

# Aritmética Binária

## Caminho de Dados

Ivanildo Miranda

Octávio Augusto Deiroz

### **Aritmética Binária e Caminho de Dados**

- Representação Binárias
- Representação Hexadecimal
- Números sem Sinal
- Números com Sinal
- Operações Aritméticas (soma e subtração) com inteiros
- Operações Lógicas
- Instruções do MIPS
- Construção de uma ULA
- Multiplicação
- Divisão
- Representação em ponto flutuante

## Aritmética Binária e Caminho de Dados Representação Binária

- Notação Posicional Decimais

$$547 = 5 \times 10^2 + 4 \times 10^1 + 7 \times 10^0$$

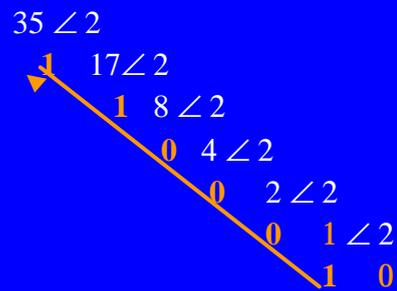
- Notação Posicional Binária

$$1000100011 =$$

$$1 \times 2^9 + 0 \times 2^8 + 0 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

## Aritmética Binária e Caminho de Dados Representação Binária

- Algoritmo de conversão de Decimal Para Binário



Assim :

$$35_{10} = 100011_2$$

### Aritmética Binária e Caminho de Dados Representação Hexadecimal

|      |   |
|------|---|
| 0000 | 0 |
| 0001 | 1 |
| 0010 | 2 |
| 0011 | 3 |
| 0100 | 4 |
| 0101 | 5 |
| 0110 | 6 |
| 0111 | 7 |
| 1000 | 8 |
| 1001 | 9 |
| 1010 | A |
| 1011 | B |
| 1100 | C |
| 1101 | D |
| 1110 | E |
| 1111 | F |

Assim :

$$35_{10} = 100011_2 = 0010\ 0011_2 = 2\ 3_{16}$$

Observe que foram acrescentados 0's não significativos à representação binária, para facilitar a conversão

### Aritmética Binária e Caminho de Dados Números sem Sinal

- Os registradores do MIPS são de 32 bits e portanto os números inteiros são representados como padrões de 32 bits.
- Podemos ter duas situações : números sem e com sinal
- No primeiro caso, podemos ter  $2^{32}$  combinações, o que nos permite representar de 0 a  $2^{32} - 1$  (0 a 4.294.967.295)

Como representar números com sinal ?

## Aritmética Binária e Caminho de Dados Números com Sinal

Como representar números com sinal ?

Primeira idéia : Reservar um dos bits para o sinal -

- 0 representa números positivos
- 1 representa números negativos

|   |    |    |    |    |    |    |    |     |     |     |     |   |   |   |   |   |   |
|---|----|----|----|----|----|----|----|-----|-----|-----|-----|---|---|---|---|---|---|
| S | 30 | 29 | 28 | 27 | 26 | 25 | 24 | ... | ... | ... | ... | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 1  | 0  | 1  | 1  | 1  | 0  | 0  | ... | ... | ... | ... | 0 | 0 | 0 | 1 | 1 | 0 |

## Aritmética Binária e Caminho de Dados Números com Sinal

Problemas com o uso do bit mais significativo para representar o sinal :

- Estabelecer o valor do bit de sinal, após as operações (necessita de uma tabela de comparações)
- Existência de dois zeros (um positivo e outro negativo)

## Aritmética Binária e Caminho de Dados Números com Sinal

Alternativa :

Representar os números como complemento de 2, ou seja, todos os números iniciados com 0 são positivos e todos os números iniciados com 1 são negativos. :

```

0000 0000 0000 0000 0000 0000 0000 = 0
0000 0000 0000 0000 0000 0000 0001 = 1
0000 0000 0000 0000 0000 0000 0010 = 2
...
0111 1111 1111 1111 1111 1111 1111 = 2.147.483.647
1000 0000 0000 0000 0000 0000 0000 = -2.147.483.648
1000 0000 0000 0000 0000 0000 0001 = -2.147.483.647
1000 0000 0000 0000 0000 0000 0010 = -2.147.483.646
...
1111 1111 1111 1111 1111 1111 1110 = -2
1111 1111 1111 1111 1111 1111 1111 = -1
    
```

## Aritmética Binária e Caminho de Dados Soma de Inteiros sem Sinal

Soma – Segue o mesmo procedimento da soma de decimais :

Exemplo : 107+61 = 168

|          |   |   |   |   |   |   |   |   |
|----------|---|---|---|---|---|---|---|---|
| Vai Um-> | 1 | 1 | 1 | 1 | 1 | 1 | 1 |   |
| ...0000  | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| ...0000  | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| Total    | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |

## Aritmética Binária e Caminho de Dados Soma e Subtração de Inteiros com Sinal

Subtração :

Exemplo :  $168-61=107 \Rightarrow 168+(-61)=107$

Basta somar o complemento :

|          |   |   |   |   |   |   |   |   |
|----------|---|---|---|---|---|---|---|---|
| Vai Um-> |   |   |   |   |   |   |   |   |
| ...0000  | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| ...1111  | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| Total    | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |

## Aritmética Binária e Caminho de Dados Operações Lógicas - AND

AND (E)  $A \cdot B = C$

| A | B | C |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Exemplo :

|     |   |   |   |   |   |   |   |   |
|-----|---|---|---|---|---|---|---|---|
| A   | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| B   | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| A.B | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

## Aritmética Binária e Caminho de Dados

### Operações Lógicas - OR

OR (OU)  $A+B=C$

| A | B | C |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Exemplo :

|     |   |   |   |   |   |   |   |   |
|-----|---|---|---|---|---|---|---|---|
| A   | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| B   | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| A+B | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |

## Aritmética Binária e Caminho de Dados

### Operações Lógicas - NOT

NOT (Negação)  $\bar{A}=B$

| A | B |
|---|---|
| 0 | 1 |
| 1 | 0 |

Exemplo :

|         |   |   |   |   |   |   |   |   |
|---------|---|---|---|---|---|---|---|---|
| A       | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| NOT (A) | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |

## Aritmética Binária e Caminho de Dados

### Achar o complemento de um número

Basta inverter o número e somar 1

Exemplo : Achar o complemento de 107

|           |           |   |   |   |   |   |   |   |   |
|-----------|-----------|---|---|---|---|---|---|---|---|
| 107       | ...00000  | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| NOT (107) | ...111111 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| (+1)      | ...000000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| = -107    | ...111111 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |

## Aritmética Binária e Caminho de Dados

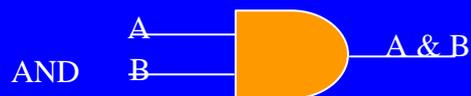
### Instruções MIPS – Tabela Resumo

| Categoria  | Instrução | Exemplo             | Resultado             |
|------------|-----------|---------------------|-----------------------|
| Aritmética | add       | add \$s1,\$s2,\$s3  | $\$s1 = \$s2 + \$s3$  |
|            | addi      | addi \$s1,\$s2,k    | $\$s1 = \$s2 + k$     |
|            | addu      | addu \$s1,\$s2,\$s3 | $\$s1 = \$s2 + \$s3$  |
|            | addiu     | addiu \$s1,\$s2,k   | $\$s1 = \$s2 + k$     |
|            | sub       | sub \$s1,\$s2,\$s3  | $\$s1 = \$s2 - \$s3$  |
|            | subu      | subu \$s1,\$s2,\$s3 | $\$s1 = \$s2 - \$s3$  |
| Lógica     | and       | and \$s1,\$s2,\$s3  | $\$s1 = \$s2 \& \$s3$ |
|            | andi      | andi \$s1,\$s2,k    | $\$s1 = \$s2 \& k$    |
|            | or        | or \$s1,\$s2,\$s3   | $\$s1 = \$s2 ! \$s3$  |
|            | ori       | ori \$s1,\$s2,k     | $\$s1 = \$s2 ! k$     |

## Aritmética Binária e Caminho de Dados Instruções MIPS

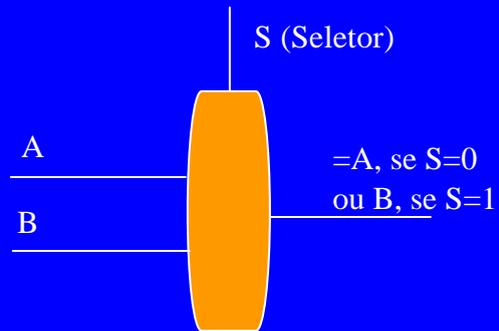
| Categoria                | Instrução | Exemplo         | Resultado           |
|--------------------------|-----------|-----------------|---------------------|
| Lógica<br>(Deslocamento) | sll       | sll \$s1,\$s2,k | $\$s1 = \$s2 \ll k$ |
|                          | srl       | srl \$s1,\$s2,k | $\$s1 = \$s2 \gg k$ |

## Aritmética Binária e Caminho de Dados Construção de uma ULA – Blocos Básicos



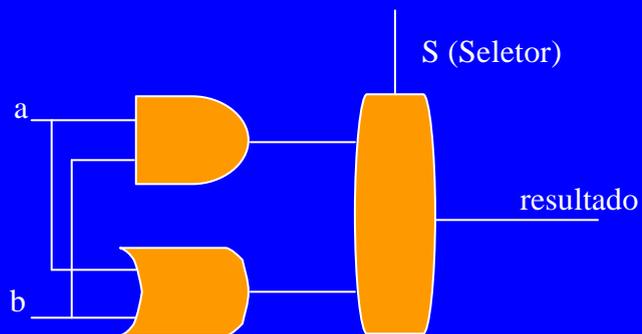
**Aritmética Binária e Caminho de Dados**  
**Construção de uma ULA – Blocos Básicos**

Multiplexador



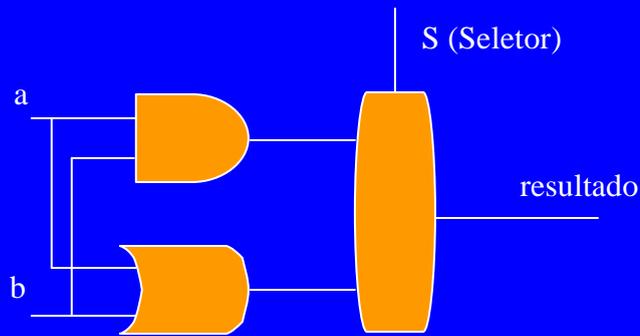
**Aritmética Binária e Caminho de Dados**  
**Construção de uma ULA – Blocos Básicos**

Unidade Lógica de 1 bit para realizar AND e OR



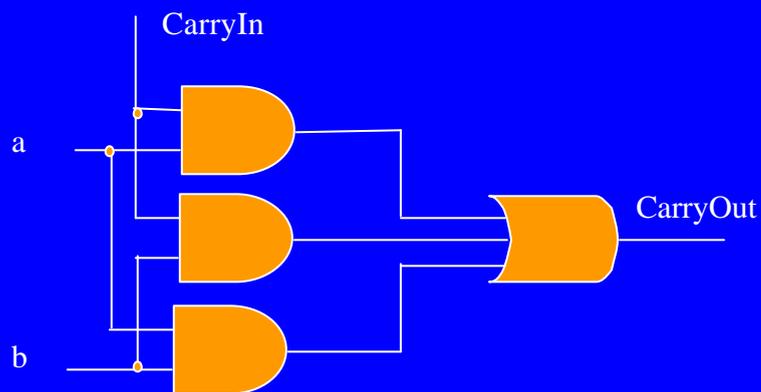
**Aritmética Binária e Caminho de Dados**  
**Construção de uma ULA – Blocos Básicos**

Unidade Lógica de 1 bit para realizar AND e OR



**Aritmética Binária e Caminho de Dados**  
**Construção de uma ULA – Blocos Básicos**

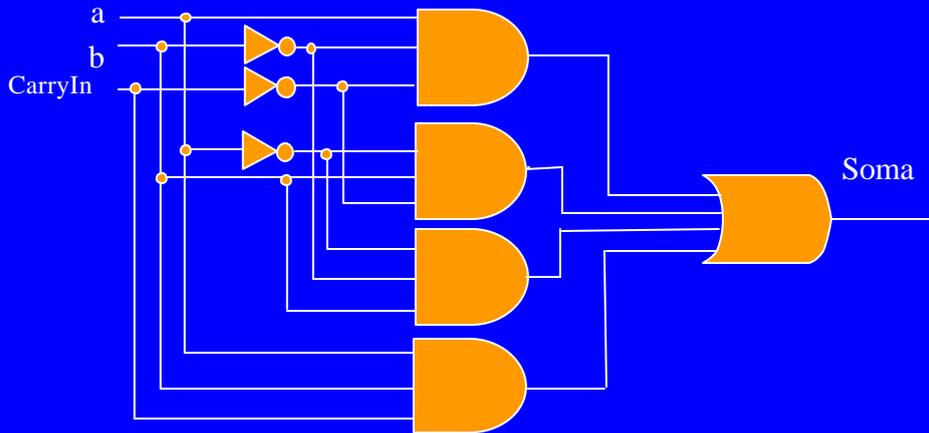
Circuito para Geração do “vai-um” (Carry-out)



$$\text{CarryOut} = (b \cdot \text{CarryIn}) + (a \cdot \text{CarryIn}) + (a \cdot b) + (a \cdot b \cdot \text{CarryIn})$$

**Aritmética Binária e Caminho de Dados**  
**Construção de uma ULA – Blocos Básicos**

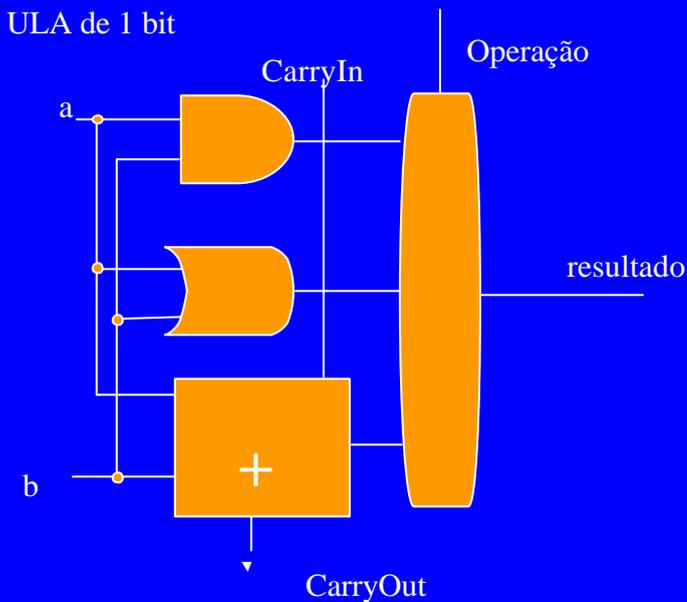
Circuito para Geração da Soma



$$\text{Soma} = (\overline{a} \cdot \overline{b} \cdot \text{CarryIn}) + (\overline{a} \cdot b \cdot \text{CarryIn}) + (a \cdot \overline{b} \cdot \text{CarryIn}) + (a \cdot b \cdot \text{CarryIn})$$

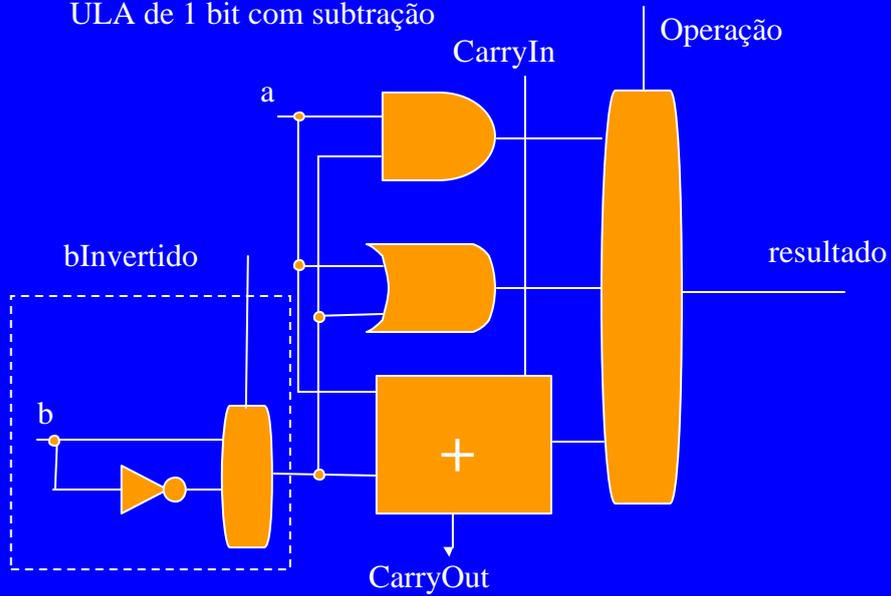
**Aritmética Binária e Caminho de Dados**  
**Construção de uma ULA – Blocos Básicos**

ULA de 1 bit



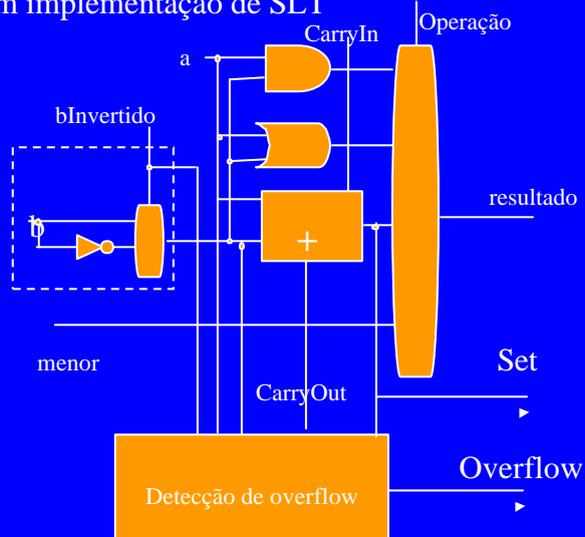
**Aritmética Binária e Caminho de Dados**  
**Construção de uma ULA – Blocos Básicos**

ULA de 1 bit com subtração

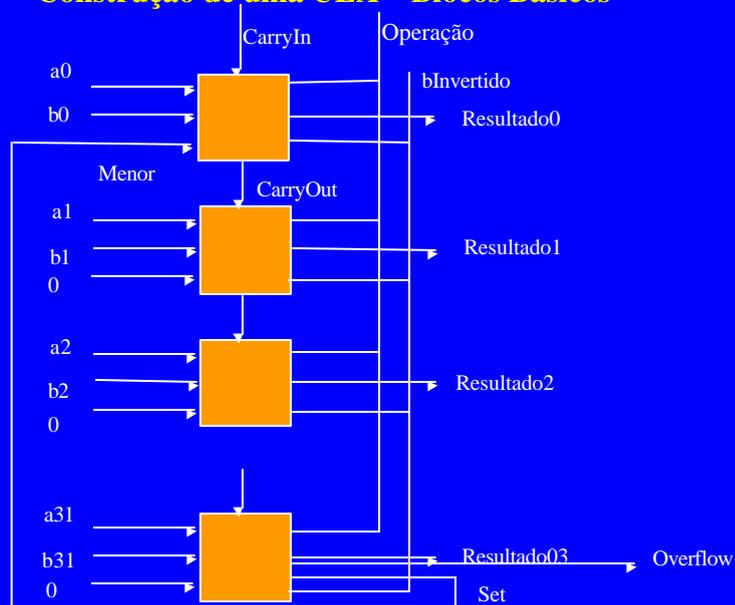


**Aritmética Binária e Caminho de Dados**  
**Construção de uma ULA – Blocos Básicos**

ULA com implementação de SLT

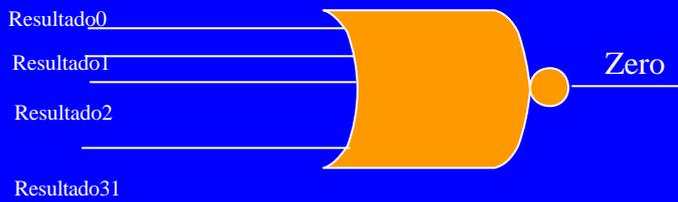


## Aritmética Binária e Caminho de Dados Construção de uma ULA – Blocos Básicos

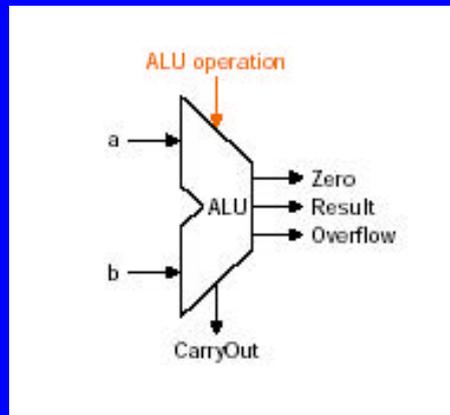


## Aritmética Binária e Caminho de Dados Construção de uma ULA – Blocos Básicos

Detecção de 0



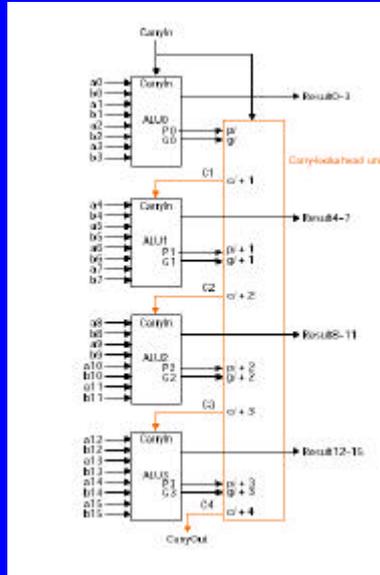
## Aritmética Binária e Caminho de Dados Construção de uma ULA – Blocos Básicos



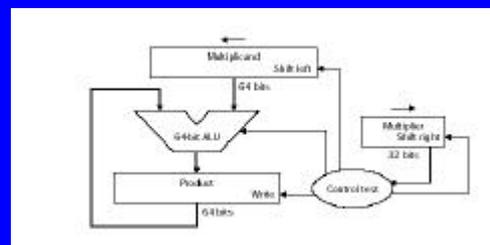
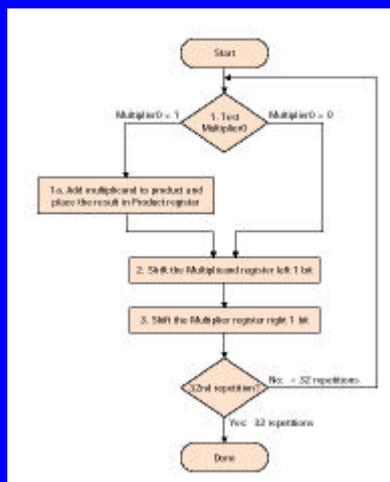
## Aritmética Binária e Caminho de Dados Construção de uma ULA

- Sistema de Carry em cascata apresenta problemas de desempenho
- Solução : Implementar um sistema Carry Lookahead

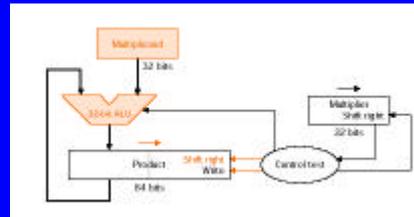
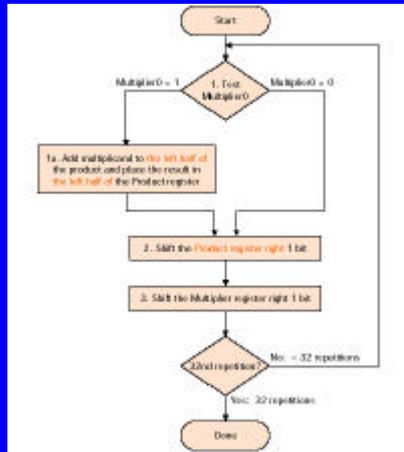
## Aritmética Binária e Caminho de Dados Construção de uma ULA



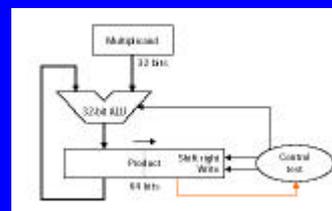
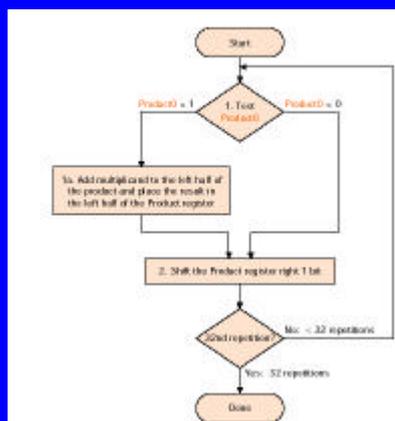
## Aritmética Binária e Caminho de Dados Usando Uma ULA para Multiplicar



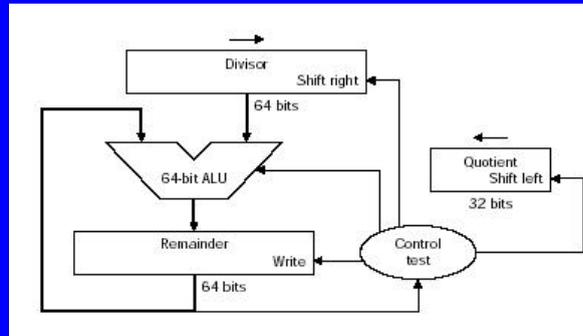
## Aritmética Binária e Caminho de Dados Usando Uma ULA para Multiplicar – Melhorado 1



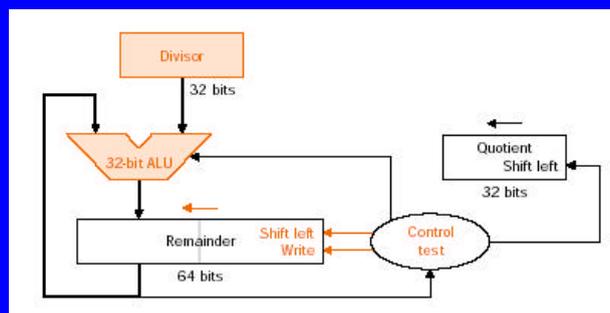
## Aritmética Binária e Caminho de Dados Usando Uma ULA para Multiplicar – Melhorado 2



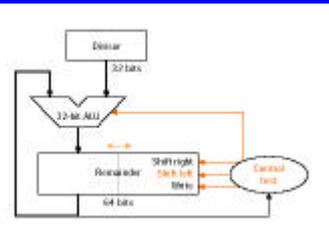
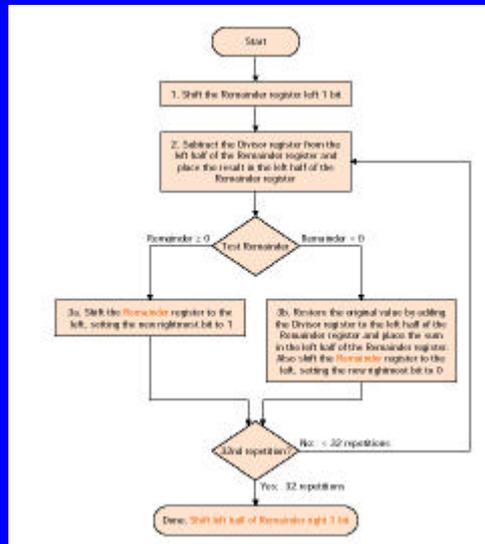
## Aritmética Binária e Caminho de Dados Usando Uma ULA para Dividir –



## Aritmética Binária e Caminho de Dados Usando Uma ULA para Dividir – Melhorado 1



## Aritmética Binária e Caminho de Dados Usando Uma ULA para Dividir – Melhorado 2



## Aritmética Binária e Caminho de Dados Operações em Ponto Flutuante

### Representação em Ponto Flutuante

#### Precisão Simples

|       |    |          |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-------|----|----------|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 31    | 30 | 29       | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18       | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| s     |    | expoente |    |    |    |    |    |    |    |    |    |    | mantissa |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 1 bit |    | 8 bits   |    |    |    |    |    |    |    |    |    |    | 23 bits  |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

#### Precisão Dupla

|       |    |                                  |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |
|-------|----|----------------------------------|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|
| 31    | 30 | 29                               | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18       | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |  |  |  |  |  |
| s     |    | expoente                         |    |    |    |    |    |    |    |    |    |    | mantissa |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |
| 1 bit |    | 11 bits                          |    |    |    |    |    |    |    |    |    |    | 23 bits  |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |
|       |    | Mantissa - continuação - 32 bits |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |

## Aritmética Binária e Caminho de Dados Operações em Ponto Flutuante

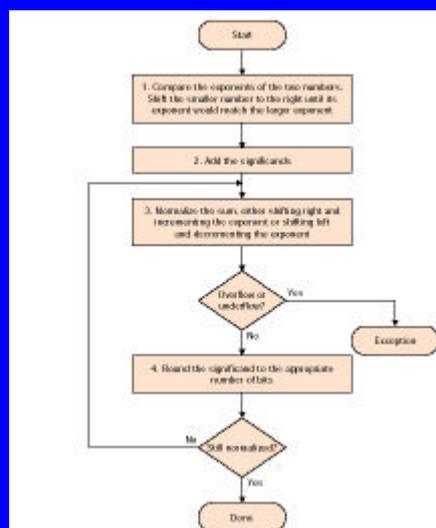
### Representação em Ponto Flutuante - Detalhes

- O número 1 da mantissa não é representado
- A conversão para decimal se dá pela fórmula  $(-1)^s \times (1 + \text{mantissa})^{\text{Expoente-peso}}$
- Notação normalizada pelo padrão IEEE754
- Propriedades de números reais nem sempre são verdadeiras usando notação de ponto flutuante. Exemplo :

$$1 \times 10^{38} + 1 - 1 \times 10^{38} \neq 1 \times 10^{38} - 1 \times 10^{38} + 1$$

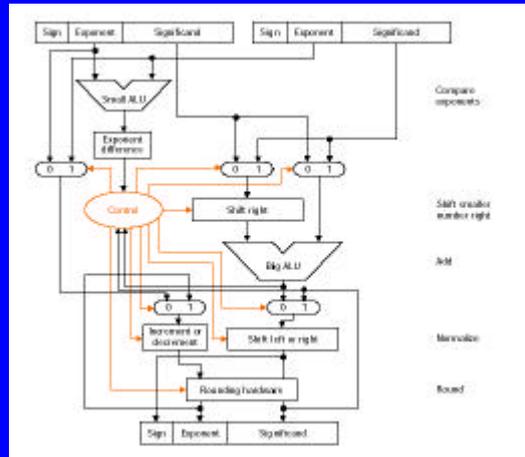
## Aritmética Binária e Caminho de Dados Operações em Ponto Flutuante

### Algoritmo de soma em ponto flutuante



## Aritmética Binária e Caminho de Dados Operações em Ponto Flutuante

Somador em ponto flutuante



## Aritmética Binária e Caminho de Dados Operações em Ponto Flutuante

Somador em ponto flutuante

