

CST Análise e Desenvolvimento de Sistemas

AOC786201 - Fundamentos de Arquitetura e Organização de Computadores


Aritmética Binária

Soma aritmética de decimais

- Para realizar a operação soma, fazemos cada "casa" por vez, primeiro das unidades, depois dezenas, depois centenas, ...
- Como no sistema decimal temos 10 símbolos (de 0 a 9), se ao somar uma casa o valor for maior do que 9, então dizemos que "vai 1" para a casa da esquerda, deixando na casa em que está ocorrendo a operação apenas o algarismo menos significativo

Soma aritmética de decimais

- Exemplo 1:

$$\begin{array}{r} + \quad 48 \\ \quad 95 \\ \hline \quad 3 \end{array}$$


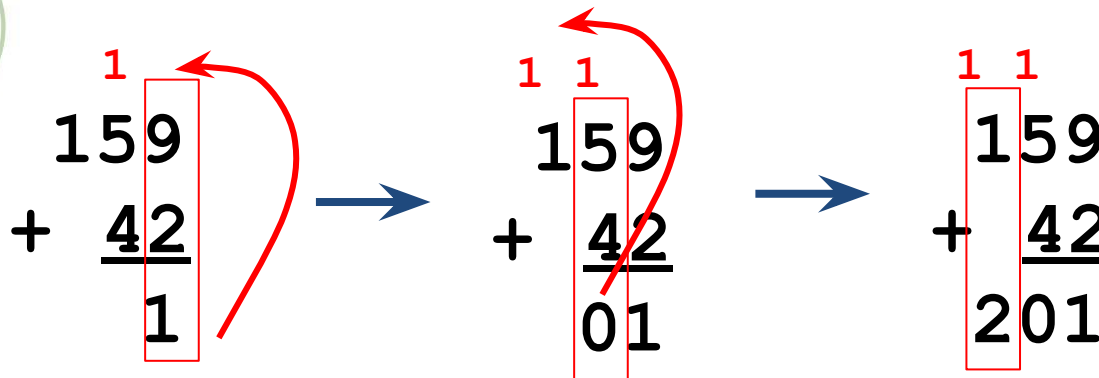


Resultado final:
 $48 + 95 = 143$

$$\begin{array}{r} + \quad 48 \\ \quad 95 \\ \hline 143 \end{array}$$

Soma aritmética de decimais

- Exemplo 2:


$$\begin{array}{r} 1 \\ 159 \\ + 42 \\ \hline 1 \end{array} \rightarrow \begin{array}{r} 1 \quad 1 \\ 159 \\ + 42 \\ \hline 01 \end{array} \rightarrow \begin{array}{r} 1 \quad 1 \\ 159 \\ + 42 \\ \hline 201 \end{array}$$

Resultado final:
 $159 + 42 = 201$

Soma aritmética de binários

- Segue a mesma lógica dos decimais, porém temos apenas 2 símbolos (0 e 1)
- Ou seja, se ao somar uma casa o valor for maior do que 1, então dizemos que "vai 1" para a casa da esquerda

Soma aritmética de binários

- Exemplo 1:

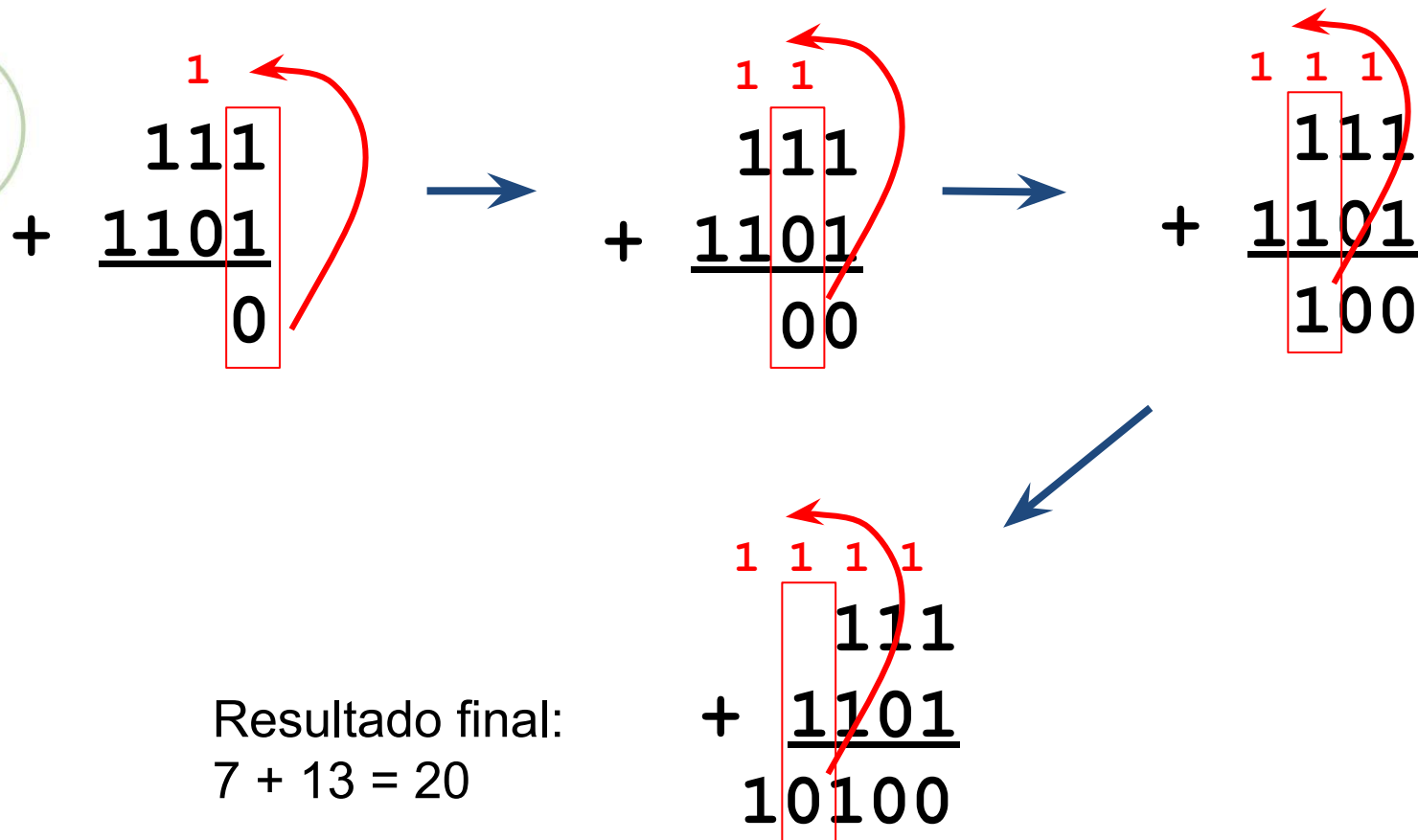
$$\begin{array}{r}
 \overset{1}{1001} \\
 + \quad 101 \\
 \hline
 0
 \end{array}
 \rightarrow
 \begin{array}{r}
 \overset{1}{1001} \\
 + \quad 101 \\
 \hline
 10
 \end{array}
 \rightarrow
 \begin{array}{r}
 \overset{1}{1001} \\
 + \quad 101 \\
 \hline
 110
 \end{array}$$

Resultado final:
 $9 + 5 = 14$

$$\begin{array}{r}
 \overset{1}{1001} \\
 + \quad 0101 \\
 \hline
 1110
 \end{array}$$

Soma aritmética de binários

- Exemplo 2:



$$\begin{array}{r}
 1 \\
 111 \\
 + 1101 \\
 \hline
 0
 \end{array}
 \rightarrow
 \begin{array}{r}
 1 \ 1 \\
 111 \\
 + 1101 \\
 \hline
 00
 \end{array}
 \rightarrow
 \begin{array}{r}
 1 \ 1 \ 1 \\
 111 \\
 + 1101 \\
 \hline
 100
 \end{array}$$

Resultado final:
7 + 13 = 20

$$\begin{array}{r}
 1 \ 1 \ 1 \ 1 \\
 111 \\
 + 1101 \\
 \hline
 10100
 \end{array}$$

Soma aritmética de binários

- Mais detalhadamente, vamos ver apenas a operação que fazemos com o bit da unidade, utilizando o exemplo dado ($7 + 13 = 20$):

$$\begin{array}{r} 111 \\ + 1101 \\ \hline 0 \end{array}$$

A red arrow points from the '0' result to a red '1' above it, indicating a carry.

- Observe que em binário “1” + “1” = “10”, ou seja, “0” e “vai 1”
- Se fosse $6 + 13 = 19$, teríamos “0” + “1” = “1” (não tem o “vai 1”)

$$\begin{array}{r} 110 \\ + 1101 \\ \hline 1 \end{array}$$

Soma aritmética de binários

- Ou seja, analisando apenas a soma da casa das unidades temos as seguintes possibilidades e resultados esperados (estamos chamando o “vai 1” de Ts):

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

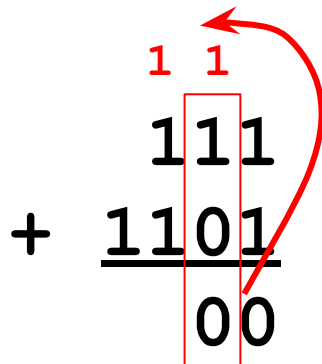
$$1 + 1 = 0 \text{ e } T_s = 1$$

Soma aritmética de binários

- O método para os demais algoritmos precisa levar em consideração que a soma anterior pode ter gerado um “vai 1”
- Como o meio somador não considera esta possibilidade, precisaremos de um novo circuito

Soma aritmética de binários

- Mais detalhadamente, vamos analisar como fazemos a operação dos demais bits (além da casa da unidade), utilizando o exemplo dado ($7 + 13 = 20$):


$$\begin{array}{r} \text{1 1} \\ 111 \\ + 1101 \\ \hline 00 \end{array}$$

- Observe que teve um “vai 1” da casa anterior, temos “1” + “1” + “0” = “10”, ou seja, “0” e “vai 1” (novamente)

Soma aritmética de binários

- Ou seja, para as casas exceto a unidade podemos ter mais possibilidades, pois além do algarismo da primeira parcela, o algarismo da segunda parcela e as saídas S e Ts, temos ainda uma terceira entrada que foi o “vai 1” da operação anterior que vamos chamar de Te:

$$A + B + T_e = S$$

0	+	0	+	0	=	0		
0	+	0	+	1	=	1		
0	+	1	+	0	=	1		
0	+	1	+	1	=	0	e Ts =	1
1	+	0	+	0	=	1		
1	+	0	+	1	=	0	e Ts =	1
1	+	1	+	0	=	0	e Ts =	1
1	+	1	+	1	=	1	e Ts =	1

Circuitos Aritméticos: Meio somador

- Vamos chamar o resultado da casa da unidade de “S” (saída) e o “vai 1” de “Ts” (transporte de saída)
- Então, temos:

$$A + B = S$$

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0 \text{ e } T_s = 1$$

A	B	Ts	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Circuitos Aritméticos: Meio somador

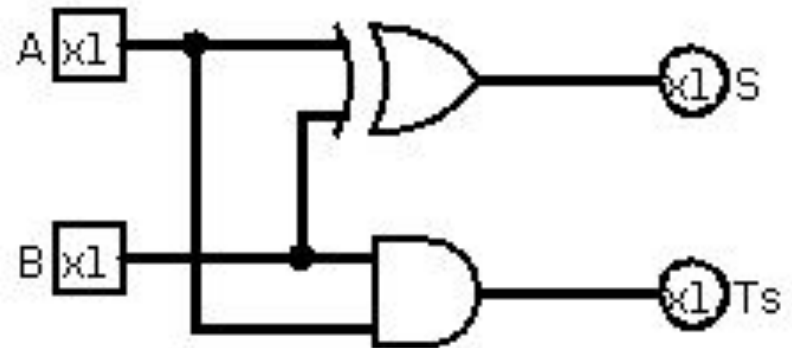
- Considerando esta tabela verdade, podemos obter a expressão de S e Ts:

$$S = A \oplus B$$

$$Ts = AB$$

A	B	Ts	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

- A saída Ts também é conhecida como Carry out



Circuitos Aritméticos: Somador completo

- Com a entrada T_e (transporte de entrada - Carry in) temos esta tabela verdade:

$$A + B + T_e = S$$

$$\begin{array}{l} 0 + 0 + 0 = 0 \\ 0 + 0 + 1 = 1 \\ 0 + 1 + 0 = 1 \\ 0 + 1 + 1 = 0 \text{ e } T_s = 1 \\ 1 + 0 + 0 = 1 \\ 1 + 0 + 1 = 0 \text{ e } T_s = 1 \\ 1 + 1 + 0 = 0 \text{ e } T_s = 1 \\ 1 + 1 + 1 = 1 \text{ e } T_s = 1 \end{array}$$

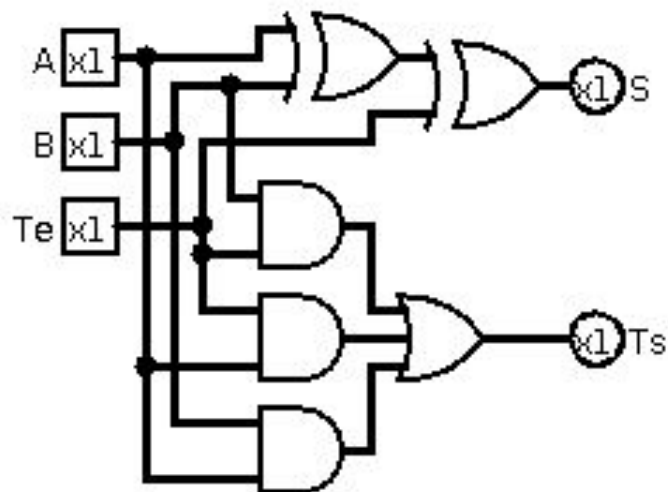
A	B	T_e	T_s	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Circuitos Aritméticos: Somador completo

- Considerando esta tabela verdade, podemos obter a expressão de S e Ts:

$$S = A \oplus B \oplus T_E$$

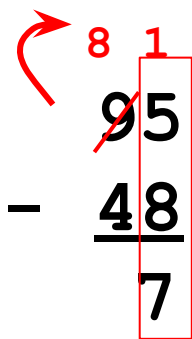
$$T_S = BT_E + AT_E + AB$$



A	B	Te	Ts	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

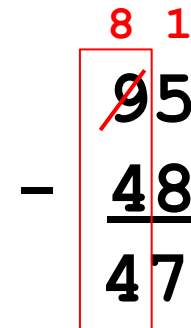
Subtração aritmética de decimais

- Para realizar a operação subtração, fazemos cada "casa" por vez, primeiro das unidades, depois dezenas, depois centenas, ...
- Ao subtrair uma casa, se o valor da casa do minuendo é maior do que da casa do subtraendo, então dizemos que "pega 1" (emprestado) da casa da esquerda


$$\begin{array}{r} 8 \quad 1 \\ \cancel{9}5 \\ - \quad \underline{48} \\ 7 \end{array}$$



Resultado final:
 $95 - 48 = 47$


$$\begin{array}{r} 8 \quad 1 \\ \cancel{9}5 \\ - \quad \underline{48} \\ 47 \end{array}$$

Subtração aritmética de binários

- Segue a mesma lógica dos decimais, porém temos apenas 2 símbolos (0 e 1), ou seja, "pega 1" para a casa da esquerda se o valor da casa do minuendo for menor do que do subtraendo

$$\begin{array}{r}
 1001 \\
 - 101 \\
 \hline
 0
 \end{array}
 \quad \rightarrow \quad
 \begin{array}{r}
 1001 \\
 - 101 \\
 \hline
 00
 \end{array}
 \quad \rightarrow \quad
 \begin{array}{r}
 \overset{0}{\curvearrowright} \overset{1}{1}001 \\
 - 101 \\
 \hline
 100
 \end{array}$$

↓

Resultado final:

$9 - 5 = 4$

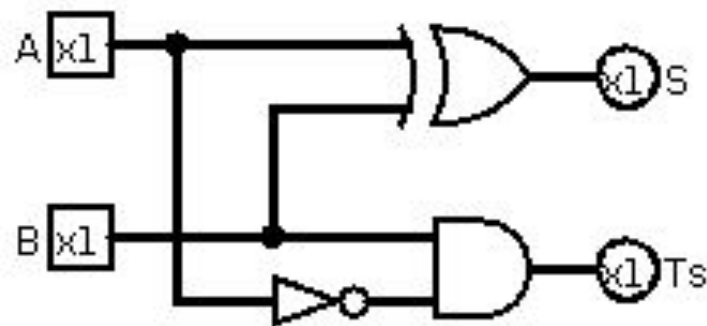
$$\begin{array}{r}
 \overset{0}{\overset{1}{\cancel{1}}}001 \\
 - 0101 \\
 \hline
 0100
 \end{array}$$

Circuitos Aritméticos: Meio subtrator

- Considerando um subtrator de um bit, sabemos que:
 - $0 - 0 = 0$
 - $0 - 1 = 1$ e transporta 1
 - $1 - 0 = 1$
 - $1 - 1 = 0$
- Considerando esta tabela verdade, podemos obter a expressão de S e Ts:

$$S = A \oplus B$$
$$Ts = \overline{A}B$$

A	B	S	Ts
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



Circuitos Aritméticos: Subtrator completo

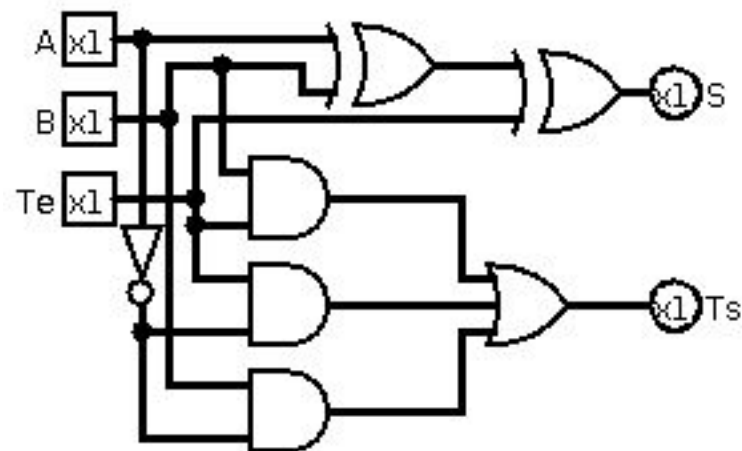
- Novamente, o meio subtrator não considera o bit de transporte de outra operação
- É necessário considerar se emprestou um para que esse bit seja subtraído
- Para entender a lógica, vamos calcular $12 - 3 = 9$, em binário

$$\begin{array}{r}
 1100 \\
 - 0011 \\
 \hline
 1001
 \end{array}$$

$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$
 $T_s=0 \quad T_s=0 \quad T_s=1 \quad T_s=1$

Col. 4 Col. 3 Col. 2 Col. 1

A	B	T_E	S	T_S
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



Regras adicionais para Soma e Subtração de decimais

- Na subtração de numerais positivos, quando o minuendo é menor que o subtraendo é necessário trocar estes numerais de posição e determinar que o resultado será negativo.
- Exemplo: $45 - 98$. O 45 é menor que 98, então faremos $98 - 45$.

$$\begin{array}{r} 45 \\ - 98 \\ \hline ? \end{array} \quad \begin{array}{c} \nearrow \\ \searrow \end{array} \quad \begin{array}{r} 98 \\ - 45 \\ \hline - ? \end{array}$$



$$\begin{array}{r} 98 \\ - 45 \\ \hline 3 \end{array}$$



$$\begin{array}{r} 98 \\ - 45 \\ \hline -53 \end{array}$$

Resultado final:

$$98 - 45 = -53$$

Regras adicionais para Soma e Subtração de decimais

- Regra dos sinais (considerando primeira parcela de maior magnitude que a segunda parcela):

Sinais iguais soma-se e colocamos o mesmo sinal

$$\bigcirc \quad + \quad + \quad = \quad +$$

$$\bigcirc \quad - \quad - \quad = \quad -$$

$$\begin{array}{r} \overset{1}{-}59 \\ + \quad \overset{1}{-}42 \\ \hline -101 \end{array}$$

Sinais diferentes subtrai-se e coloca o sinal da maior magnitude

$$\bigcirc \quad + \quad - \quad = \quad +$$

$$\bigcirc \quad - \quad + \quad = \quad -$$

$$\begin{array}{r} \overset{4}{-}3 \quad \overset{1}{-} \\ -42 \\ + \quad \underline{16} \\ -26 \end{array}$$

$$\begin{array}{r} -16 \\ + \quad \underline{42} \end{array} \quad \begin{array}{r} \overset{4}{-}3 \quad \overset{1}{-} \\ 42 \\ + \quad \underline{-16} \\ 26 \end{array}$$

Representação de sinais em sistemas computacionais

- Em sistemas computacionais, como temos apenas “0”s e “1”s precisamos representar os sinais também desta forma.
- Para isso utilizamos notações como sinal-magnitude, complemento de 1 e complemento de 2. Em geral, este último.
- Em operações com numerais em complemento de 2 não precisamos nos preocupar com a ordem das parcelas

Representação de sinais em sistemas computacionais

- Exemplo: $3 + (-5) = -2$
 - $3 = 11$ binário
 - $-5 = 1011$ binário (complemento de 2 de 4 bits)

$$\begin{array}{r} 11 \\ + \underline{1011} \\ 1110 \end{array}$$

- Resultado: 1110 binário (complemento de 2 de 4 bits) = -2

Representação de sinais em sistemas computacionais

- Exemplo: $-3 + 5 = 2$
 - $-3 = 1101$ binário (complemento de 2 de 4 bits)
 - $5 = 101$ binário

$$\begin{array}{r} 1101 \\ + \quad 0101 \\ \hline 10010 \end{array}$$

- Resultado: 0010 binário = 2
- O bit de transbordo (overflow) é descartado

Representação de sinais em sistemas computacionais

- Cuidado para não extrapolar o limite da representação!
- Exemplo: $5 + 7 = 12$
 - $5 = 101$ binário
 - $7 = 111$ binário

$$\begin{array}{r} 0101 \\ + \underline{0111} \\ 1100 \end{array}$$

- Resultado: 1100 binário que em complemento de 2 de 4 bits não é 12, mas sim -4

Representação de sinais em sistemas computacionais

- Cuidado para não extrapolar o limite da representação!
- Exemplo 2: $-5 + (-7) = -12$
 - 5 = 1011 binário (complemento de 2 de 4 bits)
 - 7 = 1001 binário (complemento de 2 de 4 bits)

$$\begin{array}{r} 1011 \\ + \quad 1001 \\ \hline 10100 \end{array}$$

- Resultado: 0100 binário que representa 4 e não -12