Operações Aritméticas

Operações Aritméticas em Complemento de 2

Alex Dias Gonsales

Operações Aritméticas em Complemento de 2

- Representação em complemento de 2
- Operações Aritméticas
  - Troca de Sinal
  - Soma
  - Subtração
- *Overflow*

Representação complemento de 2

- Observação: será utilizada representação em complemento de 2 com n=4 bits

Número	Positivo		Número	Negativo
0000	0		----	--
0001	1		1111	-1
0010	2		1110	-2
0011	3		1101	-3
0100	4		1100	-4
0101	5		1011	-5
0110	6		1010	-6
0111	7		1001	-7
----	-		1000	-8

Troca de Sinal

- Trocar o sinal de um número é o mesmo que obter o complemento de 2 do número
  - Calcular “Bn - a” ou
  - Inverter todos os bits e somar 1 ou
  - Usar método da potência negativa
- Exemplo: complemento de 2 do número 3

$B^n - 3 =$ $16 - 3 = 13 = 1101_2$	Inverter e somar 1
10000 = 16 - 0011 = 3 ---- 1101 = -3	0011 = 3 1100 = 3 invertido + 1 ---- 1101 = -3

Soma

- Somar os dois operandos e ignorar o “vai um”, se ocorrer.

-3 + 2 = -1	-3 + (-2) = -5
1101 = -3 + 0010 = 2 ---- 1111 = -1	1101 = -3 + 1110 = -2 ---- **1**1011 = -5 (ignorar o "vai um")

Subtração

- Em qualquer sistema de numeração a subtração pode ser efetuada através de uma soma:
  - x - y = x + (-y)

6 - 2 = 4 6 + (-2) = 4	-3 – 2 = -3 + (-2) = -5
0110 = 6 + 1110 = -2 ---- 0100 = 4	1101 = -3 + 1110 = -2 ---- 1011 = -5

Overflow

- O *overflow* (estouro ou transbordamento) ocorre quando ao realizar uma operação o resultado ficar fora da faixa representável.
  - Overflow durante a troca de sinal
  - Overflow durante a soma
  - Overflow durante a subtração

Overflow

- Troca de Sinal
  - Atenção: a troca de sinal do menor número negativo gera *overflow*, pois não existe representação positiva para esse número. Exemplo:
    - Menor número negativo = -8
    - -8 = 1000_2 → não existe +8

$B^n - 8 =$ $16 - 8 = 8 = 1000_2$	Inverter e somar 1
10000 = 16 - 1000 = 8 ---- 1000 = -8	1000 = -8 0111 = -8 invertido + 1 ---- 1000 = -8

Overflow

- *Overflow* ao efetuar uma soma ou subtração

4 + 5 = 9	- 4 + (-7) = -11
0100 = 4 + 0101 = 5 ---- 1001 = -7	1100 = -4 + 1001 = -7 ---- 0101 = 5

-4 – 5 = -4 + (-5) = -9
1100 = -4 + 1011 = -5 ---- 0111 = 7

Overflow

- O *overflow* ocorre independentemente do *carry*. Veja 5 exemplos abaixo:

decimal	complemento de 2	carry	overflow	resultado
3 + 4 = 7	0011 + 0100 = **0**0111 = 7	não	não	correto
7 + -1 = 6	0111 + 1111 = **1**0110 = 6	sim	não	correto
7 + 3 = 10	0111 + 0011 = **0**1010 = -6	não	sim	deveria ser 10
-8 + -1 = -9	1000 + 1111 = **1**0111 = 7	sim	sim	deveria ser -9

Overflow

- Como detectar *overflow*
  - Regra 1: se o "vai um" do bit mais significativo for diferente do "vem um" do bit mais significativo então ocorreu *overflow*.
  - Regra 2: comparar os sinais dos operandos e do resultado segundo a tabela a seguir

operando1	operando 2	esperado	obtido	Estouro
+	+	+	+	não
+	+	+	-	sim
-	-	-	-	não
-	-	-	+	sim
+	-	+ ou -	+ ou -	nunca ocorre
-	+	+ ou -	+ ou -	nunca ocorre