Lógica e Sistemas Digitais

Adição e subtração no domínio dos números naturais e relativos com representação limitada de operandos e resultados

Indicadores de erro e indicadores relacionais

João Pedro Patriarca (jpatri@cc.isel.ipl.pt)

Slides inspirados nos slides do prof. Mário Véstias

ISEL

Erro na representação de resultados na adição e subtração

- Por norma, num sistema computacional, o número de bits para codificar operandos é igual ao número de bits para codificar resultados
- Qualquer resultado tem uma interpretação incorreta sempre que não seja possível codificar o valor do resultado com o número de bits definido
- A afirmação anterior é válida quer no domínio dos naturais quer no domínio dos relativos como também quer na operação adição quer na operação subtração
- Exemplos de resultados não representáveis no domínio $\mathbb{N}$ com codificação a 4 bits (0..15)

$$10 + 9 = 19 \qquad 4 + 12 = 16 \qquad 0 - 1 = -1 \qquad 11 - 15 = -4$$

- Exemplo de resultados não representáveis no domínio $\mathbb{Z}$ com codificação a 4 bits (-8..+7)

$$7 + 7 = 14 \qquad -5 + (-4) = -9 \qquad -3 - 7 = -10 \qquad 6 - (-2) = 8$$

Flag de erro na adição no domínio $\mathbb{N}$

- Exemplo em binário com operandos e resultados a 4 bits

$2 + 7 = 9$

		0010	(2)
	+	0111	(7)
Arrasto ⇨	0	1001	(9)

$10 + 9 = 19$

		1010	(10)
	+	1001	(9)
Arrasto ⇨	1	0011	(16+3)

$4 + 12 = 16$

		0100	(4)
	+	1100	(12)
Arrasto ⇨	1	0000	(16+0)

- A interpretação errada do resultado na adição de números naturais acontece sempre que a operação produz arrasto (*carry*)

Flag de erro na subtração no domínio $\mathbb{N}$

- Exemplo em binário com operandos e resultados a 4 bits

$13 - 2 = 13 + (-2) = 11$					$0 - 1 = 0 + (-1) = -1$					$11 - 15 = 11 + (-15) = -4$			
			1	(1)				1	(1)			1	(1)
			1101	(13)				0000	(0)			1011	(11)
		+	1101	(13)			+	1110	(14)		+	0000	(0)
Arrasto	⇨	1	1011	(16+11)	Arrasto	⇨	0	1111	(15)	Arrasto ⇨	0	1100	(12)

- A interpretação errada do resultado na subtração de números naturais acontece sempre que a operação adição não produz arrasto (*carry*)
- O arrasto na subtração (*borrow*) é o inverso do arrasto na adição (*carry)*

Flag de erro na adição no domínio $\mathbb{Z}$

- Exemplo em binário com operandos e resultados a 4 bits

$$7 + (-5) = +2$$
$$\begin{array}{r|ll} & 0111 & (+7) \\ + & 1011 & (-5) \\ \hline 1 & 0010 & (+2) \end{array}$$

$$-8 + 6 = -2$$
$$\begin{array}{r|ll} & 1000 & (-8) \\ + & 0110 & (+6) \\ \hline 0 & 1110 & (-2) \end{array}$$

$$5 + 2 = +7$$
$$\begin{array}{r|ll} & 0101 & (+5) \\ + & 0010 & (+2) \\ \hline 0 & 0111 & (+7) \end{array}$$

$$-4 + (-3) = -7$$
$$\begin{array}{r|ll} & 1100 & (-4) \\ + & 1101 & (-3) \\ \hline 1 & 1001 & (-7) \end{array}$$

$$4 + 6 = +10$$
$$\begin{array}{r|ll} & 0100 & (+4) \\ + & 0110 & (+6) \\ \hline 0 & 1010 & (-6) \end{array}$$

$$-1 + (-8) = -9$$
$$\begin{array}{r|ll} & 1111 & (-1) \\ + & 1000 & (-8) \\ \hline 1 & 0111 & (+7) \end{array}$$

- Impossível exceder o domínio na adição de operandos com sinais diferentes (- retângulos verdes)
- Operandos com o mesmo sinal (- retângulos vermelhos), existe erro se o resultado tiver um sinal diferente do sinal dos operandos

Flag de erro na subtração no domínio $\mathbb{Z}$

- Exemplo em binário com operandos e resultados a 4 bits

$7 - 2 = 7 + (-2) = +5$

$$\begin{array}{r|l l}
 & 1 & (1) \\
 & 0111 & (7) \\
+ & 1101 & (-3) \\ \hline
1 & 0101 & (+5)
\end{array}$$

$-8 - (-4) = -8 + 4 = -4$

$$\begin{array}{r|l l}
 & 1 & (1) \\
 & 1000 & (-8) \\
+ & 0011 & (+3) \\ \hline
0 & 1100 & (-4)
\end{array}$$

$1 - (-3) = 1 + 3 = +4$

$$\begin{array}{r|l l}
 & 1 & (1) \\
 & 0001 & (+1) \\
+ & 0010 & (+2) \\ \hline
0 & 0100 & (+4)
\end{array}$$

$-2 - 5 = -2 + (-5) = -7$

$$\begin{array}{r|l l}
 & 1 & (1) \\
 & 1110 & (-2) \\
+ & 1010 & (-6) \\ \hline
1 & 1001 & (-7)
\end{array}$$

$0 - (-8) = 0 + 8 = +8$

$$\begin{array}{r|l l}
 & 1 & (1) \\
 & 0000 & (+0) \\
+ & 0111 & (+7) \\ \hline
0 & 1000 & (-8)
\end{array}$$

$-6 - 3 = -6 + (-3) = -9$

$$\begin{array}{r|l l}
 & 1 & (1) \\
 & 1010 & (-6) \\
+ & 1100 & (-4) \\ \hline
1 & 0111 & (+7)
\end{array}$$

- Mesmas regras que na adição

- O sinal observado no complemento do valor a subtrair corresponde ao sinal do complemento restringido que é relevante para o complemento do valor -8 (5º. exemplo)

Overflow (exceder o domínio nos números relativos)

- Definição de *Overflow*:

  Há *Overflow* quando a soma de dois números positivos produz um número negativo ou quando a soma de dois números negativos produz um número positivo

- Expressão algébrica

$$
\begin{array}{ccccc}
 & C_n & ... & C_1 & \\
 & A_n & ... & A_1 & A_0 \\
+ & B_n & ... & B_1 & B_0 \\
\hline
C_{n+1} & S_n & ... & S_1 & S_0
\end{array}
$$

$$Ov = \overline{A_n}.\overline{B_n}.S_n + A_n.B_n.\overline{S_n}$$

- Expressão baseada nos arrastos C_n e C_{n+1}

$$Ov = C_n \oplus C_{n+1}$$

- Expressão baseada nos arrastos C_n e C_{n+1} mas sem acesso a C_n

$$S_n = A_n \oplus B_n \oplus C_n \Leftrightarrow C_n = A_n \oplus B_n \oplus S_n \Longrightarrow Ov = A_n \oplus B_n \oplus S_n \oplus C_{n+1}$$

Indicadores relacionais

Considerações gerais

- A comparação entre dois números é realizada <u>apenas e somente</u> por análise do resultado da subtração
- Siglas típicas para os dois domínios (considerando a subtração $A - B$)
  - A (*Above*): A é maior que B nos $\mathbb{N}$
  - AE (*Above or Equal*): A é maior ou igual que B nos $\mathbb{N}$
  - B (*Below*): A é menor que B nos $\mathbb{N}$
  - BE (*Below or Equal*): A é menor ou igual a B nos $\mathbb{N}$
  - G (*Greater*): A é maior que B nos $\mathbb{Z}$
  - GE (*Greater or Equal*): A é maior ou igual a B nos $\mathbb{Z}$
  - L (*Less*): A é menor que B nos $\mathbb{Z}$
  - LE (*Less or Equal*): A é menor ou igual que B nos $\mathbb{Z}$
  - Z (*Zero*) ou E (*Equal*): A é igual a B quer nos $\mathbb{N}$, quer nos $\mathbb{Z}$
- Um resultado negativo indica que o operando A é inferior a B
- Um resultado positivo indica que o operando A é superior ou igual a B

Flags relacionais

- A relação de igualdade entre dois operandos é obtida por análise de todos os bits do resultado; o OR negado de todos os bits do resultado implementa a *flag Zero* ou *flag Equal*, independentemente do domínio ($Equal = \overline{S_n + \cdots + S_1 + S_0}$)
- A relação entre dois operandos, para o domínio $\mathbb{N}$, é obtida por análise da *flag borrow*

$$
\begin{array}{cccccc}
 & & & & & 1 \\
 & & A_n & \ldots & A_1 & A_0 \\
 & + & \overline{B_n} & \ldots & \overline{B_1} & \overline{B_0} \\
\hline
C_{n+1} & & S_n & \ldots & S_1 & S_0
\end{array}
$$

- O valor direto de *borrow* implementa a relação *Below*, nos $\mathbb{N}$
- A relação entre dois operandos, para o domínio $\mathbb{Z}$, é obtida por análise do sinal do resultado (S_n) e pela *flag overflow*
- O XOR entre o sinal do resultado e a *flag overflow* implementa a relação *Less*, nos $\mathbb{Z}$ ($Less = S_n \oplus Ov$)

Exemplos

Completar considerando a operação $R = A - B - C_{in}$ (a dimensão da codificação dos números é indicado em cada tabela)

<table>
<tr><td colspan="2">Ex1 – 4 bits</td><td>R</td><td>A</td><td>B</td><td>Cin/Bin</td><td rowspan="2">Cout/Bout</td><td rowspan="2">OV</td><td rowspan="2">BL</td><td rowspan="2">GE</td></tr>
<tr><td colspan="2">Base 2</td><td>0001</td><td></td><td>1000</td><td rowspan="3">0</td></tr>
<tr><td rowspan="2">Base 10</td><td>$\mathbb{N}$</td><td></td><td></td><td></td><td></td><td>-</td><td>0</td><td>-</td></tr>
<tr><td>$\mathbb{Z}$</td><td></td><td></td><td></td><td>-</td><td></td><td>-</td><td></td></tr>
</table>

<table>
<tr><td colspan="2">Ex2 – 6 bits</td><td>R</td><td>A</td><td>B</td><td>Cin/Bin</td><td rowspan="2">Cout/Bout</td><td rowspan="2">OV</td><td rowspan="2">BL</td><td rowspan="2">GE</td></tr>
<tr><td colspan="2">Base 2</td><td></td><td></td><td>110101</td><td rowspan="3">1</td></tr>
<tr><td rowspan="2">Base 10</td><td>$\mathbb{N}$</td><td></td><td></td><td></td><td></td><td>-</td><td></td><td>-</td></tr>
<tr><td>$\mathbb{Z}$</td><td>-23</td><td></td><td></td><td>-</td><td></td><td>-</td><td></td></tr>
</table>

<table>
<tr><td colspan="2">Ex3 – 3 bits</td><td>R</td><td>A</td><td>B</td><td>Cin/Bin</td><td rowspan="2">Cout/Bout</td><td rowspan="2">OV</td><td rowspan="2">BL</td><td rowspan="2">GE</td></tr>
<tr><td colspan="2">Base 2</td><td></td><td></td><td></td><td rowspan="3"></td></tr>
<tr><td rowspan="2">Base 10</td><td>$\mathbb{N}$</td><td></td><td></td><td></td><td></td><td>-</td><td>1</td><td>-</td></tr>
<tr><td>$\mathbb{Z}$</td><td>-4</td><td></td><td></td><td>-</td><td></td><td>-</td><td>0</td></tr>
</table>

Solução Ex1

<table>
<tr><td colspan="2">Ex1 – 4 bits</td><td>R</td><td>A</td><td>B</td><td>Cin/Bin</td><td rowspan="2">Cout/Bout</td><td rowspan="2">OV</td><td rowspan="2">BL</td><td rowspan="2">GE</td></tr>
<tr><td colspan="2">Base 2</td><td>0001</td><td>1001</td><td>1000</td><td rowspan="3">0</td></tr>
<tr><td rowspan="2">Base 10</td><td>ℕ</td><td>1</td><td>9</td><td>8</td><td>0</td><td>-</td><td>0</td><td>-</td></tr>
<tr><td>ℤ</td><td>1</td><td>-7</td><td>-8</td><td>-</td><td>0</td><td>-</td><td>1</td></tr>
</table>

- Sendo R=1, o resultado poderá ser 1 (sem erro) ou -15 (com erro)
- Nos naturais, para R=-15, A=-7 o que torna impossível esta hipótese (o valor -7 não é representável nos naturais)
- Nos naturais, para R=1, A=9, logo BL=0 e Bout=0
- Nos complementos, A=-7, B=-8, logo R=1 e, portanto, OV=0 e GE=1

Solução Ex2

<table>
<tr><td colspan="2">Ex2 – 6 bits</td><td>R</td><td>A</td><td>B</td><td>Cin/Bin</td><td rowspan="2">Cout/Bout</td><td rowspan="2">OV</td><td rowspan="2">BL</td><td rowspan="2">GE</td></tr>
<tr><td colspan="2">Base 2</td><td>101001</td><td>011111</td><td>110101</td><td rowspan="3">1</td></tr>
<tr><td rowspan="2">Base 10</td><td>$\mathbb{N}$</td><td>41</td><td>31</td><td>53</td><td>1</td><td>-</td><td>1</td><td>-</td></tr>
<tr><td>$\mathbb{Z}$</td><td>-23</td><td>+31</td><td>-11</td><td>-</td><td>1</td><td>-</td><td>1</td></tr>
</table>

- Nos complementos B=-11
- Nos complementos R=-23 (com OV=0) ou R=+41 (com OV=1)
- Nos complementos, para R=-23, implica A=-33 tornando impossível esta hipótese porque com 6 bits o menor valor representável é -32
- Nos complementos, para R=+41, A=+31. Satisfazendo as condições, OV=1 e GE=1
- Nos naturais, R=41, A=31, B=53, logo o resultado excede o domínio e, por isso, Bout=1; como A<B, BL=1

Solução Ex3

Ex3 – 3 bits		R	A	B	Cin/Bin	Cout/Bout	OV	BL	GE
Base 2		100	000	011					
Base 10	$\mathbb{N}$	4	0	3	1	1	-	1	-
	$\mathbb{Z}$	-4	0	+3		-	0	-	0

- Nos relativos, R=-4 (com OV=0) ou R=+4 (com OV=1)
- Para GE=0, A<B nos complementos, logo o resultado tem de ser negativo, ou seja, R=-4. Esta condição inviabiliza A>0 e B<0
- Para BL=1, A<B nos naturais inviabilizando que nos complementos A<0 e B>=0 e também que A<0 e B<0, logo apenas será possível com A>=0 e B>=0
- Para satisfazer R=-4, B>A nos naturais e nos relativos, os únicos valores de A e de B que satisfazem as condições são 0 e 3, respetivamente, com Bin=1

Exercícios

Completar considerando a operação $R = A - B - C_{in}$ (a dimensão da codificação dos números é indicado em cada tabela)

Ex1 – 4 bits		R	A	B	Cin/Bin	Cout/Bout	OV	BL	GE
Base 2		0000							
Base 10	$\mathbb{N}$		0				-	1	-
	$\mathbb{Z}$					-		-	1

Ex2 – 6 bits		R	A	B	Cin/Bin	Cout/Bout	OV	BL	GE
Base 2				110101					
Base 10	$\mathbb{N}$				1		-		-
	$\mathbb{Z}$	-22				-		-	

Ex3 – 3 bits		R	A	B	Cin/Bin	Cout/Bout	OV	BL	GE
Base 2									
Base 10	$\mathbb{N}$	7		0			-	0	-
	$\mathbb{Z}$					-		-	1