

Atividade Assíncrona 4 – Aritmética de Números Inteiros no MIPS
ARQUITETURA DE COMPUTADORES
Luiza Batista Laquini – 2019107786

Atividade 1

Questão 1:

Padrão de bits	Inteiro com sinal	Inteiro sem sinal
0x0000	0	0
0x7fff	32767	32767
0x8000	-32768	32768
0xffff	-1	65535

Padrão de bits	Inteiro com sinal	Inteiro sem sinal
0x00000000	0	0
0x00007fff	32767	32767
0x00008000	32768	32768
0x0000ffff	65535	65535

Questão 2:

Oper1	Oper2	Oper1 + Oper2	Oper1 – Oper2	Exemplo
valor > 0	valor > 0	S	N	1 + 2147483647 = overflow 1 – 2147483647 = -2147483646
valor > 0	valor < 0	N	S	2147483647 + (-2147483648) = -1 2147483647 – (-2147483648) = overflow
valor < 0	valor > 0	N	S	-2147483648 + 2147483647 = -1 -2147483648 – 2147483647 = overflow
valor < 0	valor < 0	S	N	-2147483648 + (-2147483648) = overflow -2147483648 – (-2147483648) = 0
0	valor > 0	N	N	0 + 2147483647 = 2147483647 0 – 2147483647 = -2147483647
0	valor < 0	N	S	0 + (-2147483648) = -2147483648 0 – (-2147483648) = overflow
valor > 0	0	N	N	2147483647 + 0 = 2147483647 2147483647 – 0 = 2147483647
valor < 0	0	N	N	-2147483648 + 0 = -2147483648 -2147483648 – 0 = -2147483648

Atividade 2

A saída do programa foi:

Soma = -2

Subtração = 2147483647

A explicação para o “-2” é que o bit de sinal se altera com a operação.

Questão 3: Sim, ocorre *overflow* na soma ($a + b$). O resultado correto seria 4294967294, porém o *signed int* não possui a quantidade de bits suficientes para representar esse número e, portanto, retorna um resultado errado (-2).

Atividade 3

a	b	Resultado impresso	Overflow?	Justificativa
+2000000000	+2000000000	<i>arithmetic overflow</i>	S	A soma excede 32 bits em casas positivas
+2000000000	-2000000000	0	N	A soma de positivo com negativo (independente da ordem) vai sempre resultar em um valor <u>entre</u> os dois operandos, portanto, se as entradas são válidas, não há overflow
-2000000000	+1	-1999999999	N	A soma de positivo com negativo (independente da ordem) vai sempre resultar em um valor <u>entre</u> os dois operandos, portanto, se as entradas são válidas, não há overflow
-2000000000	-2000000000	<i>arithmetic overflow</i>	S	A soma excede 32 bits em casas negativas

Atividade 4

a	b	Resultado impresso	Overflow?	Justificativa
+2000000000	+2000000000	0	N	A subtração de duas entradas positivas válidas vai sempre resultar em um valor <u>entre</u> os dois operandos
+2000000000	-2000000000	<i>arithmetic overflow</i>	S	A subtração de um positivo com um negativo (nessa ordem) vai se transformar, na verdade, em uma soma de positivos, podendo gerar overflow por exceder os 32 bits em casas positivas
-2000000000	+2000000000	<i>arithmetic overflow</i>	S	A subtração de um negativo com um positivo (nessa ordem) vai se transformar, na verdade, em uma soma de negativos, podendo gerar overflow por exceder os 32 bits em casas negativas
-2000000000	-2000000000	0	N	A subtração de duas entradas positivas válidas vai sempre resultar em um valor <u>entre</u> os dois operandos

Atividade 5

a	b	Resultado impresso	Overflow?	Justificativa
Adição				
+2000000000	+2000000000	-294967296 (errado)	S	A soma excede 32 bits em casas positivas
+2000000000	-2000000000	0	N	A soma de positivo com negativo (independente da ordem) vai sempre resultar em um valor <u>entre</u> os dois operandos, portanto, se as entradas são válidas, não há overflow
-2000000000	+2000000000	0	N	A soma de positivo com negativo (independente da ordem) vai sempre resultar em um valor <u>entre</u> os dois operandos, portanto, se as entradas são válidas, não há overflow
-2000000000	-2000000000	294967296 (errado)	S	A soma excede 32 bits em casas negativas
Subtração				
+2000000000	+2000000000	0	N	A subtração de duas entradas positivas válidas vai sempre resultar em um valor <u>entre</u> os dois operandos
+2000000000	-2000000000	-294967296 (errado)	S	A subtração de um positivo com um negativo (nessa ordem) vai se transformar, na verdade, em uma soma de positivos, podendo gerar overflow por exceder os 32 bits em casas positivas
-2000000000	+2000000000	294967296 (errado)	S	A subtração de um negativo com um positivo (nessa ordem) vai se transformar, na verdade, em uma soma de negativos, podendo gerar overflow por exceder os 32 bits em casas negativas
-2000000000	-2000000000	0	N	A subtração de duas entradas positivas válidas vai sempre resultar em um valor <u>entre</u> os dois operandos

OBS: Nota-se que a diferença entre usar *signed int* e *unsigned int* é que em *unsigned int* não obtemos mensagens específicas de *overflow* e sim resultados errados. Isso acontece porque o bit do sinal é preenchido.

Atividade 6

Questão 4:

Dividendo	Divisor	Quociente	Resto
22	7	3	1
-22	7	-3	-1
22	-7	-3	1
-22	-7	3	-1

Atividade 7

Questão 5: Sim. Os dois números são de 32bits, portanto, a multiplicação entre eles excede 32bits e não pode ser representada, gerando *overflow*.

Questão 6: Não. O resultado exibido foi retirado dos 32bits menos significativos de um resultado de 64bits que não teve como ser representado:

00111111 11111111 11111111 11111111 00000000 00000000 00000000 00000001.

Resultado exibido: 1

Atividade 8 (multiplicação COM sinal)

a	b	Registrador hi (hexadecimal)	Registrador lo (hexadecimal)
2	1	0x00000000	0x00000002
2	-1	0xffffffff	0xffffffe
2 ¹⁸	2 ¹⁴	0x00000001	0x00000000
2 ¹⁸	-2 ¹⁴	0xffffffff	0x00000000
2147483647	2147483647		
2147483647	-2147483647		

a	b	Resultado esperado	Resultado obtido
2	1	2	2
2	-1	-2	-2
2 ¹⁸	2 ¹⁴	4294967296	0
2 ¹⁸	-2 ¹⁴	-4294967296	0
2147483647	2147483647	Overflow	1
2147483647	-2147483647	Overflow	-1

Questão 7: Porque ocorre *overflow*.

Atividade 9 (multiplicação SEM sinal)

a	b	Registrador hi (hexadecimal)	Registrador lo (hexadecimal)
2	1	0x00000000	0x00000002
2	-1	0x00000001	0xffffffe
2 ¹⁸	2 ¹⁴	0x00000001	0x00000000
2 ¹⁸	-2 ¹⁴	0x0003ffff	0x00000000
2147483647	2147483647	0x3ffffff	0x00000001
2147483647	-2147483647	0x3ffffff	0xffffffff

a	b	Resultado esperado	Resultado obtido
2	1	2	2
2	-1	-2	<i>break instruction</i>
2 ¹⁸	2 ¹⁴	4294967296	<i>break instruction</i>
2 ¹⁸	-2 ¹⁴	-4294967296	<i>break instruction</i>
2147483647	2147483647	Overflow	<i>Overflow</i>
2147483647	-2147483647	Overflow	<i>break instruction</i>

Atividade 10 (divisão COM sinal)

a	b	Quociente impresso	Resto impresso	Erro
11	2	5	1	-
11	-2	-5	1	-
-11	2	-5	-1	-
-11	-2	5	-1	-
-1	2	0	-1	-
11	0	0	0	<i>break instruction</i>
-2147483648	-1	-2147483648	0	<i>Overflow</i>

Atividade 11 (divisão SEM sinal)

a	b	Quociente impresso	Resto impresso	Erro
11	2	5	1	-
11	-2	0	11	<i>Overflow</i>
-11	2	2147483642	1	<i>Overflow</i>
-11	-2	0	-11	<i>Overflow</i>
-1	2	2147483647	1	<i>Overflow</i>
11	0	0	0	<i>break instruction</i>
-2147483648	-1	-2147483648	0	<i>Overflow</i>