Laboratório 4

Números positivos e negativos, todos representados em binário.



Dicas

- Você não precisa entregar nenhum código como resposta. Procure entender os conceitos e explorar as variações.
- Você pode utilizar o simulador RISC-V para testar os códigos que você desenvolver. Para isso, veja as dicas fornecidas logo abaixo.
- Não deixe de colocar comentários nos seus códigos. Procure organizar o código de forma que ele fique mais fácil de entender.
- As dicas desse laboratório estão colapsadas. Para expandi-las, clique na pequena seta do lado direito da caixa de texto.

Dados são apenas bits

Conforme visto em sala de aula e nas atividades de laboratório anteriores, os dados armazenados num computador são apenas bits e eles precisam ser interpretados de acordo com o contexto que o programa/programador deseja.

Nesse sentido, a letra A é representada pelo número 65 na tabela ASCII. Isso significa que, um byte em memória com o valor 65 pode tanto representar o número 65 quanto a letra A. O significado real desse byte será dado pela forma de uso posterior do programa. Se o programa imprime como letra, aparecerá A, se ele for impresso como número, aparecerá 65.

Algo similar acontece com os números positivos e negativos. Da mesma forma que temos, em C, os tipos int e unsigned int representam números inteiros com sinal e sem sinal. Todos eles em complemento de 2. O que muda é o contexto de uso.



Atividade 1

Faça um programa que leia um número do teclado, guarde num registrador, e imprima esse número na tela. Você já fez algo muito similar anteriormente, pode reutilizar seu código sem problemas. Teste seu programa com números positivos e negativos, o que aconteceu?

Dica

Se você leu e escreveu o mesmo registrador, o simulador atual vai imprimir o número exatamente da mesma forma. Mas, se você fizer uma operação com ele, você verá que aparecerá um número bem grande na tela. Então, se quiser imprimir números negativos, você precisa tratar o caso de números bem grandes como já fez anteriormente com um programa que testava se o número era positivo ou negativo antes de imprimir, lembra?

Você pode fazer operações com todos os dados

Você já sabe que a letra A é representada pelo código ASCII 65. Já a letra a (minúscula) é representada pelo número 97. Como o alfabeto tem as letras em sequência na tabela ASCII (a letra B tem código 66 e a b tem código 98), você pode passar uma letra de maiúscula para minúscula apenas somando 32 (=97-65) ao código ASCII da letra maiúscula. Se você subtrair 32 de uma letra minúscula, você também passa essa letra para maiúscula.

Mas dá para ser mais simples ainda, como \$32 = 2^5\$, pelo intervalo dos caracteres, se você ativar o bit 5 de uma letra maiúscula, ela será transformada em minúscula. Se você desativar o bit 5 de uma letra minúscula, ela será transformada em maiúscula. Aqui falamos em ativar através de uma operação OU com a instrução ORI com o imediato 32. Para desativar, você precisa fazer uma operação E com a instrução ANDI com o imediato 223 (=255-32).

Apesar das duas abordagens parecerem iguais, elas possuem uma diferença sutil: para você passar o A para minúscula, você tanto pode utilizar a soma quanto o OU. Mas, para isso, você precisa saber se tem uma letra maiúscula pois se tiver uma letra minúscula e somar 32 (consulte a tabela ASCII para descobrir o que vai acontecer), você não terá a mesma letra minúscula. Já se fizer um OU mesmo na letra minúscula, ela se manterá meinúscula pois o bit que você está ativando já está ativado.

Atividade 2

Faça um programa que leia um caractere em letras maiúsculas do teclado, guarde num registrador e imprima a versão minúscula dele na tela.

Dica

Para ler um caracter do teclado, você deve utilizar o código 5 em t0 com a instrução addi t0, zero, 5.

Atividade 3

Faça um programa que leia um caractere em letras minúsculas do teclado, guarde num registrador e imprima a versão maiúscula dele na tela.



Faça um programa que leia um caractere do teclado, guarde num registrador e imprima uma versão dele na tela. Se o caractere for uma letra minúscula, imprima a versão maiúscula. Se for uma letra maiúscula, imprima a versão minúscula. Caso contrário, imprima o caractere sem alterações.

Dica

O código ASCII da letra Z é 90. Assim, se você tiver um caracter entre 65 e 90, você sabe que ele é maiúsculo. Você pode usar uma regra similar para os caracteres minúsculos.

Os números podem ser representados em hexadecimal

No laboratório passado, você imprimiu números em binário, o que gerou um conjunto muito grande de dígitos na tela pois nossos números são de 32 bits. Agora que você já sabe representar os números em hexadecimal, você pode encontrar uma representação menor para os números.

Atividade 5

Faça um programa que leia um número do teclado entre 0 e 15, guarde num registrador, e imprima esse número na tela em hexadecimal utilizando apenas um dígito. Ao final, não deixe de imprimir a letra h.

Dica

Na hora de imprimir um dígito, você pode separar em dois grupos, os números de 0 a 9 e as letras de A-F. Para imprimir os números, você pode usar a instrução ADDI com o imediato 48. Para imprimir as letras, você pode usar a instrução ADDI com o imediato 55 (=65-10).

Desafio final

Agora é hora de imprimir um número em hexadecimal. Para isso, você precisa separar os dígitos de hexadecimal e imprimir cada um deles separadamente.

Atividade 6

Faça um programa que leia vários números do teclado, guarde num registrador, e imprima cada número na tela em hexadecimal utilizando 8 dígitos hexadecimal. Ao final de cada número, não deixe de imprimir a letra h. Seu programa deve parar quando o usuário digitar o número 0.



Você consegue separar os dígitos de hexadecimal com uma máscara de 4 bits ativos fazendo um AND, de forma bem similar à que fez para imprimir os números em binário (só que agora com 4 bits).