

# Laboratório 3

## Instruções Lógicas e Manipulação de Bits

Neste laboratório vamos examinar o uso de instruções lógicas no Nios II. Instruções lógicas são úteis na manipulação de sequências de bits e no manuseamento de dados no nível de bit, em que somente poucos bits são de interesse especial. Elas são essenciais quando trabalhamos com operações de entrada e saída, como veremos no próximo laboratório. Novamente, faremos uso do sistema *DE2 Basic Computer*.

### Parte I

Considere o programa em linguagem de montagem do Nios II mostrado na Figura 1. Este programa determina o número máximo de 1s consecutivos em uma palavra de 32 bits. Por exemplo, a palavra 0x937a (1001001101111010) tem um máximo de 4 1s consecutivos. O código da figura calcula o número de 1s consecutivos para a palavra 0x90abcdef. O código fonte está disponível na página da disciplina (arquivo *lab3\_part1.s*). Note que este código é baseado em um ideia não óbvia para contar o número de 1s consecutivos: o número é deslocado e uma operação lógica AND é feita com o resultado deslocado até que o número consista de apenas 0s. Tente entender como essa ideia funciona antes de continuar.

```
.equ TEST_NUM, 0x90abcdef /* The number to be tested */



.global _start
_start:
    movia r8, TEST_NUM /* Load r8 with the number to be tested */
    mov r9, r8 /* Copy the number to r9 */

STRING_COUNTER:
    mov r10, r0 /* Clear the counter to zero */
STRING_COUNTER_LOOP:
    beq r9, r0, END_STRING_COUNTER /* Loop until r9 contains no more 1s */
    srli r11, r9, 0x01 /* Calculate the number of 1s by shifting the number */
    and r9, r9, r11 /* and ANDing it with the shifted result. */
    addi r10, r10, 0x01 /* Increment the counter. */
    br STRING_COUNTER_LOOP
END_STRING_COUNTER:
    mov r12, r10 /* Store the result into r12 */

END:
    br END /* Wait here when the program has completed */
.end
```

Figura 1. Programa em linguagem de montagem para contar o número de 1s consecutivos.

Prossiga da seguinte forma:

1. Crie um diretório *lab3* e um subdiretório *lab3\_part1*. Copie o arquivo *lab3\_part1.s* para este diretório.
2. Usando o programa Altera Monitor, crie um novo projeto, *part1*, no diretório *lab3\_part1*. Especifique o sistema *DE2 Basic Computer* e que o programa em linguagem de montagem contido no arquivo *lab3\_part1.s* será usado.
3. Compile e carregue o programa na placa DE2.
4. Execute o programa. Quando o programa estiver executando, você não conseguirá ver nenhuma mudança (tal como o conteúdo dos registradores ou posições de memória) na janela do programa monitor, porque ele não pode se comunicar com o processador na placa DE2. Mas, se você parar o programa, o estado atual desses componentes será mostrado. Faça assim e observe que o programa parou sua execução na última instrução de salto, carregada na posição de memória 0x28. Também note que o número de 1s consecutivos do nosso número de teste é 4, como indicado pelo conteúdo do registrador *r12*.
5. Retorne ao início do programa ao clicar no ícone . Agora utilize a opção *single step*, para executar instrução por instrução, clicando no ícone . Observe como as instruções modificam o conteúdo dos registradores do processador.
6. Coloque o valor 0x08 no contador de programa (pc), o que fará com que as duas primeiras instruções que carregam o número de teste no registrador *r8* não sejam executadas. Ainda, carregue o registrador *r8* com o valor 0xabcdef90. Quantos 1s consecutivos existe neste número? Execute o programa e veja se você acertou.

## Parte II

O programa da Figura 1 usa uma ideia elaborada ao fazer a operação AND do valor a ser testado com sua versão deslocada para determinar o número de 1s consecutivos. Desenvolva um outro programa que implemente a mesma tarefa mas não use essa ideia. Execute e teste seu programa.

## Parte III

Considere o problema de encontrar a maior sequência de 1s e 0s alternantes. Por exemplo, o número binário 101101010001 tem uma sequência de 6 1s e 0s alternantes, como destacado aqui: 101**1010**10001. Assuma que o primeiro e último bits possam fazer parte da maior sequência. Por exemplo, 1010 possui 4 bits consecutivos com 1s e 0s alternantes.

Escreva um programa que determine a maior sequência de 1s e 0s alternantes - escreva o resultado no registrador *r16*. Dicas (você não necessariamente precisa segui-las):

- Construa subrotinas que determinam a maior sequência de 1s e de 0s.
- O que acontece quando uma operação XOR é realizada com um número de n bits e uma sequência de 0s e 1s alternantes, também de n bits?

Execute e teste seu programa.

## Relatório

Além do código fonte dos programas escritos, seu grupo também deve incluir uma discussão da solução para a Parte III.

Copyright ©2011 Altera Corporation.

Tradução para o português de Alexandro Baldassin.