

EA869 – Relatório: Atividade Computacional I

Bryan Wolff

RA: 214095

Questão 1

No r16 armazenei um valor exemplar para n, no caso, 5. No r17 eu carreguei o valor 1 onde iniciará a soma e no r18 eu carreguei o valor 0 e nele estará a resposta final. Criei um loop para que sempre adicione o valor do r17 no r18, acumulando, e a cada vez que for feito isso, incremento o r17 e comparo com o r16. Se r17 atingir o valor de r16, quer dizer que já chegou no valor de n, ou seja, deve parar de acumular as somas no r18 e parar o programa.

Questão 2

a) *ANDI Rx, M*; zera todos os bits que possuem zero em relação a máscara, já os bits referentes as posições dos bits 1 da máscara, permanecem com o mesmo valor. Como o registrador possui 8 bits, e temos interesse em manter o bit 3 e 4 iguais e zerar os outros, a máscara será 00011000=24. Portanto a instrução deve ser:

andi r16,24

b) *SBR Rx,M* ; Seta todos os bits no registrador Rx para um, que são um na máscara constante M. Como o registrador tem bits da forma 00000000 e pede para setar os bits 2 e 7, então o valor da máscara deve ser correspondente a cadeia binária 10000100 = 132. Logo a instrução deve ser:

sbr r16,132

c) *CBR Rx,M* ; zera todos os bits no registrador Rx que estão em um dentro do valor de máscara constante M. Utilizando o mesmo raciocínio do item anterior, para zerar os bits 6, 5 e 1, a cadeia binária da máscara deve ser 01100010 = 98. Logo a instrução comando deve ser:

cbr r16,98

Questão 3

Vamos partir de um número palíndromo para fazer o exercício e armazená-lo no r16. O exercício forneceu um exemplo de número (abdc efgh = 1011 1101 = 189) então coloquei ele no r16. Vou definir também r20 com o valor 0, e assumirá 1 se, e somente se, eu provar que o valor contido em r16 é palíndromo. Para isso, vou utilizar mais dois registradores, no r17 usando a função *andi* com o número 15, vou isolar os bits 0000efgh para usar como comparação posteriormente. No r18, com as funções *bst* e *bld*, através da flag T, irei transferir os bits abcd do r16 e colocarei na forma 0000dcba no r18. Se o valor de r18 for igual ao valor de r17, o número é palíndromo, e irei atribuir o valor 1 ao r20, caso contrário, o programa irá encerrar com o valor 0 no r20.

Questão 4

Armazenei na memória de programa o valor de “a” e “b” utilizando a diretiva *ORG* e *DB* inserindo-os no primeiro local de memória (0x0000). A partir disso, iniciei o código do programa em outro lugar da memória (0x002C) com a diretiva *ORG*, e com

ajuda do ponteiro *Z* e a instrução *LPM*, consigo transferir da memória do programa, os valores de “a” e “b” para os registradores r16 e r17, respectivamente.

Antes de iniciar, defini que o **quociente estará no r18 e o resto no r19**, e eu considere 3 casos possíveis. No caso $a = b$, sabemos que o quociente será 1 e o resto 0, portanto defini r18 contendo 1 e r19 contendo 0 para este caso. No caso $a < b$, sabemos que não conseguiríamos dividir, portanto r19 (resto) assumira o mesmo valor do dividendo (a), e o quociente (r18) terá valor 0.

Agora no caso $a > b$, o método que será abordado para fazer a divisão desses dois números inteiros, será a partir do método de subtrações sucessivas, onde irei subtrair o divisor do dividendo quantas vezes eu conseguir (em um loop) até atingir zero ou um valor negativo. A quantidade de vezes que eu conseguir fazer essa subtração será o quociente. Assim que a subtração assumir um valor zero ou negativo, o programa deverá encerrar, contendo o quociente e o resto, respectivamente em r18 e r19.