# TRABALHO 1 DE MICROPROCESSADORES

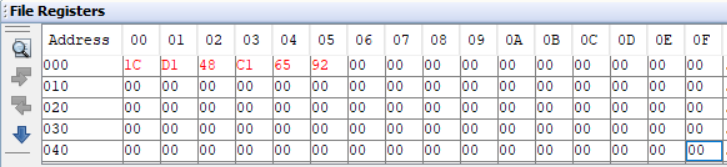
ALUNO: Francilândio Lima Serafim (472644)

## EX01 – Resolução

Para armazenar números de 16 bits na memória, foi utilizada a estratégia de colocar a parte mais significativa em uma variável e a parte menos significativa em outra, já que no PIC18F4550 cada posição de memória vale 8 bits. Desse modo, utilizando as variáveis A1 e A0, foi possível armazenar o primeiro número de 16 bits, sendo A1 a parte mais significativa e A0 a menos, e o mesmo foi feito para as variáveis B1 e B0. Foram declaradas também duas variáveis para armazenar o resultado da soma dos dois números de tal forma que SOMA1 armazena a parte mais significativa e SOMA0 armazena a menos. As declarações de variáveis foram de tal forma que A1, A0, B1, B0, SOMA1 e SOMA0, nessa ordem, ocupam os endereços de memória de 0 a 5.

Iniciando o programa, foi usada a instrução MOVLW para colocar o valor 1C no W, e depois usada a instrução MOVWF para colocar o valor de W na variável A1, e o mesmo foi feito para atribuir os valores D1, 48 e C1 nas variáveis A0, B1 e B0. Ou seja, no caso de teste foram utilizados os números A1D1 (A) e 48C1 (B) em hexadecimal como mostra a Figura 1.

Figura 1 – Valores do teste nos registradores.



Na sequência utilizou-se a função MOVF para mover o valor de A0 para W, e a função ADDWF para somar o valor de W com o valor de B0, armazenando o resultado no W e depois com a função MOVWF foi movido o valor do acumulador para a variável SOMA0, ou seja, foi feita a soma das partes menos significativas. O mesmo processo foi feito para as partes mais significativas com a diferença de que a função para a soma usada foi ADDWFC, pois da soma das partes menos significativas havia possibilidade de ter carry, e a função citada considera esse caso, ademais foi armazenado o valor dessa soma em SOMA1 usando a função MOVWF.

O código utilizado para a resolução está representado na Figura 2.

Figura 2 – Código de resolução.

