



ELTD13

Laboratório de

Microcontroladores/Microprocessadores

Prática_04a4

Prof. Enio R. Ribeiro

Universidade Federal de Itajubá - UNIFEI

Exercícios (endereç. relativo (não sinalizado/sinalizado), flag V, adição e subtração múltipla precisão, comparação)

Ex.:1) As variáveis p e q possuem, cada uma, 3 bytes. Eles são binários sinalizados com os seguintes valores: ($\$81 \leq p \leq \ff) ($0 \leq q \leq \$7f$). Faça a operação (q-p) para cada byte das variáveis p e q, isto é, ($1^\circ q - 1^\circ p$; $2^\circ q - 2^\circ p$; ... etc). Guarde, de forma adequada, os resultados nas variáveis sb1 e sb2. A variável sb1 tem elementos de 1 byte e os elementos de sb2 são de 2 bytes. O programa é cíclico. Faça as designações e alocações necessárias. Não usar variáveis extras. (Uso obrigatório de instruções aritméticas e de teste do flag V (bvc ou bvs)).

Ex.: 2) Faça um programa para somar duas variáveis: g e h. As variáveis g e h possuem, cada uma, 4 elementos. Os elementos de g e h podem representar valores entre: $0 \leq g \leq 255$ e $0 \leq h \leq 65535$. Guarde o resultado na variável sgh. O programa é cíclico. Faça as designações e alocações necessárias. Não usar variáveis extras. (Uso de adição de múltipla precisão).

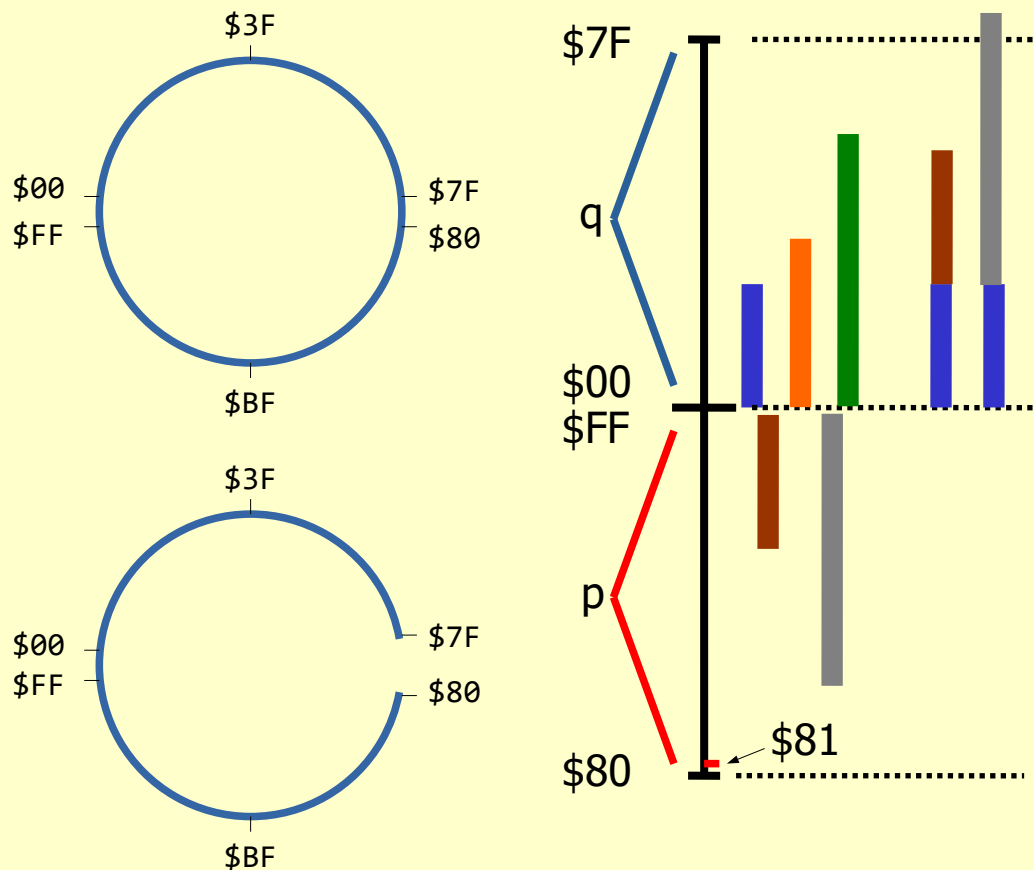
Ex.: 3) Seja m uma variável com 3 bytes, os quais são binários não sinalizados. Coloque em ordem crescente os bytes da variável m. O programa é cíclico. Faça as designações e alocações necessárias. Não usar variáveis extras. (Uso obrigatório de instruções comparações e de branch unsigned (bhi, bhs, blo, bls)).

Ex.: 4) Faça um programa para subtrair duas variáveis: m e n. As variáveis m e n possuem, cada uma, 4 elementos. Cada elemento de m e de n pode representar valores entre: $0 \leq m, n \leq 65535$. Guarde os resultados na variável smn. O programa é cíclico. Faça as designações e alocações necessárias. Não usar variáveis extras. (Uso de subtração de múltipla precisão).

Ex.: 5) Seja r uma variável com 3 bytes, os quais são binários sinalizados. Coloque em ordem decrescente os bytes da variável r. O programa é cíclico. Faça as designações e alocações necessárias. Não usar variáveis extras. (Uso obrigatório de instruções comparações e de branch signed (bge, bgt, ble, blt)).

Ex.:1) As variáveis p e q possuem, cada uma, 3 bytes. Eles são binários sinalizados com os seguintes valores: ($\$81 \leq p \leq \ff) ($0 \leq q \leq \$7f$). Faça a operação ($q-p$) para cada byte das variáveis p e q, isto é, ($1^o q - 1^o p$; $2^o q - 2^o p$; ... etc). Guarde, de forma adequada, os resultados nas variáveis sb1 e sb2. A variável sb1 tem elementos de 1 byte e os elementos de sb2 são de 2 bytes. O programa é cíclico. Faça as designações e alocações necessárias. Não usar variáveis extras. (Uso obrigatório de instruções aritméticas e de teste do flag V (bvc ou bvs)).

Ex.: 2) Faça um programa para somar duas variáveis: g e h. As variáveis g e h possuem, cada uma, 4 elementos. Os elementos de g e h podem representar valores entre: $0 \leq g \leq 255$ e $0 \leq h \leq 65535$. Guarde o resultado na variável sgh. O programa é cíclico. Faça as designações e alocações necessárias. Não usar variáveis extras. (Uso de adição de múltipla precisão).



ender. memória elementos

