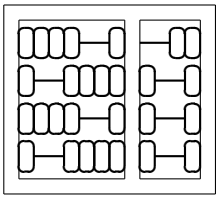

MC404 - Organização de Computadores e Linguagem de Montagem
Instituto de Computação
Universidade Estadual de Campinas

Lab. 01

1 Objetivo

Neste laboratório serão vistos a representação binária sinalizada e não-sinalizada de números, operações aritméticas e o comportamento do registrador de flags no processador ATmega88.

2 Preliminar

Com uma representação binária não-sinalizada de n bits, conseguimos representar uma faixa de valores entre 0 e $2^n - 1$. Já na representação sinalizada em complemento de 2, esse mesmo número de bits pode representar uma faixa de valores entre $-(2^{n-1})$ e $+(2^{n-1} - 1)$. Simbolicamente, um número sinalizado D em complemento de 2 é definido como:

```
+D = D
-D = M - D
```

onde $M = 2^n$ é conhecido como *módulo*.

Em complemento de 2, tem-se como regra prática para identificar se um valor é positivo ou negativo a inspeção do bit mais significativo, conhecido como *bit de sinal*. Se esse bit for 0, o número é positivo. Caso contrario (*bit de sinal* = 1), o número é negativo.

A forma como vemos um número (sinalizado ou não) depende da interpretação que damos aos bits. Desta forma, a sequência binária 10000, com $n = 5$, pode ser vista como o número $+16$ ou -16.

Ao efetuarmos operações aritméticas sobre números de n bits, é possível que o valor resultante não seja representável com esse mesmo número de bits. A esse fato damos o nome de *transbordamento* (*overflow*). Um transbordamento pode ocorrer tanto em operações com números sinalizados quanto com números não-sinalizados. No processador ATmega88, um transbordamento ocorrido durante uma operação com números não-sinalizados é indicado pela flag *Carry* (Bit 0 no registrador de status), enquanto que transbordamentos em operações com números sinalizados são indicados pela flag *Two's Complement Overflow* (Bit 3 no registrador de status).

Os bits do Registrador de Status podem ser mapeados da seguinte forma:

Bit 7 - I: Global Interrupt Enable

Habilita a ocorrência de interrupções. O controle de cada interrupção é realizado por um grupo diferente de registradores, mas se este bit estiver desabilitado, todas as interrupções estarão desabilitadas.

Bit 6 - T: Bit Copy Storage

Operações de cópia de bits (BLD, BST) utilizam este bit como fonte ou destino.

Bit 5 - H: Half Carry Flag
Indica ocorrência de *Half Carry* (*Carry* do bit 3 para o 4) em algumas operações aritméticas (Útil em aritmética BCD).

Bit 4 - S: Sign Bit
Sempre é um "ou exclusivo' entre o Bit 2 e Bit 3.

Bit 3 - V: Two's Complement Overflow Flag
Indica overflow em operações aritméticas com complemento a 2.

Bit 2 - N: Negative Flag
Indica um resultado negativo em uma operação lógica/aritmética.

Bit 1 - Z: Zero Flag
Indica um resultado igual a zero em uma operação lógica/aritmética.

Bit 0 - C: Carry Flag
Indica ocorrência de *carry* em uma operação lógica/aritmética.

3 Atividade 1

Nesta atividade vamos observar os efeitos da operação de adição sobre o registrador de status do ATmega88. Conforme descrito acima, este registrador armazena informações a respeito da execução da instrução aritmética mais recente, podendo ser utilizado para alterar o fluxo de execução por instruções condicionais.

Abra o AVR Studio e crie um projeto chamado lab01 e insira no arquivo as definições para o ATmega88:

```
.nolist
.include "m88def.inc"
.list
```

em seguida insira o seguinte código:

```
ldi r16,0x10
ldi r17,0x40
add r16,r17

rjmp PC
```

Em seguida clique no menu "Build"e em seguida na opção "Build and Run". Pressione F11 para executar cada uma das instruções do código individualmente. Execute as instruções inseridas e, para cada uma delas, observe como os registradores e as flags(**C**=Carry, **V**=Overflow, **S**=Sign Bit) se comportam.

Para observar os valores modificados em cada instrução, basta observar a tabela posicionada no lado esquerdo da tela do AVR Studio. Nesta tabela podem ser vistos o valor dos registradores alterados (r16 e r17) e também o registrador de status (SREG, com os bits devidamente sinalizados).

Como deve ter ficado claro, a primeira instrução move o valor hexa 10 para o registrador de 8 bits `r16`. A segunda instrução move 40 para o registrador `r17`. Já a terceira instrução adiciona os valores dos registradores `r16` e `r17`, guardando o resultado em `r16`.

Vamos agora variar os valores dos operandos nas duas primeiras instruções e anotar o resultado da adição. Cada linha da tabela 1 contém um par de operandos, em hexadecimal, que deve ser usado no lugar de 40 e 10 do exemplo anterior. Primeiramente converta os valores para decimal (com e sem sinal), depois modifique os valores dos operandos no programa para cada par da tabela, e torne a montar e executar as 3 instruções. Anote na tabela 1 o resultado obtido em `r16` (hexa), converta o valor para decimal com e sem sinal, e anote os estados das flags de carry, overflow e sinal após a adição (lembre-se que $n = 8$).

Tabela 1: Adição

1o. operando			2o. operando			resultado (r16)			flags
hex	dec(s)	dec(u)	hex	dec(s)	dec(u)	hex	dec(s)	dec(u)	
10			F0						c = ____ v = ____ s = ____
13			70						c = ____ v = ____ s = ____
3A			2F						c = ____ v = ____ s = ____
F0			A9						c = ____ v = ____ s = ____
99			B9						c = ____ v = ____ s = ____

Analise seus resultados e responda:

1. Os resultados obtidos conferem com os quais você esperava?

2. Em quais condições ocorre transbordamento para adição **sem** sinal?

3. Em quais condições ocorre transbordamento para adição **com** sinal?

4 Atividade 2

Esta atividade é parecida com a primeira, mas agora vamos usar uma operação de subtração ao invés da adição. Seguindo o mesmo procedimento da atividade 1, entre com as três instruções a seguir no seu programa:

```
ldi r16, <1o-operando>
ldi r17, <2o-operando>
SUB r16, r17                ; r16 = r16 - r17
```

onde `<1o-operando>` e `<2o-operando>` devem assumir cada par de valores hexa dados pela tabela 2.

Tabela 2: Subtração

1o. operando			2o. operando			resultado (r16)			flags
hex	dec(s)	dec(u)	hex	dec(s)	dec(u)	hex	dec(s)	dec(u)	
10			10						c = ____ v = ____ s = ____
90			13						c = ____ v = ____ s = ____
D1			3A						c = ____ v = ____ s = ____
57			F0						c = ____ v = ____ s = ____
47			99						c = ____ v = ____ s = ____

Novamente analise seus resultados e responda:

1. Os resultados obtidos conferem com os quais você esperava?

2. Em quais condições ocorre transbordamento para subtração **sem** sinal?

3. Em quais condições ocorre transbordamento para subtração **com** sinal?

Considere as instruções:

```
ADD  r16, r17
SUB  r16, r17
```

Para quais operandos o resultado dessas duas instruções é o mesmo? Podemos afirmar que as flags de carry, overflow e sinal se comportam de forma semelhante nos casos em que o resultado das instruções é o mesmo?