Como zerar o Acumulador do Neander

ou

The Art of (superfluous) Programming

Atribuição Direta

- Alto nível:
  AC = 0

- Assembler do Neander

Endereço	Instrução	Comentário
0	LDA y	Mem(y) = 0; usar para y um endereço da área de dados (p.ex. 128)

Subtração

Usar o fato que, para um valor x inteiro qualquer,

$x - x = 0$

Alto nível:

$AC = x - x$

ou $AC = x + (-x)$

ou $AC = x + not(x) + 1$

ou $AC = not(x) + 1 + x$

Subtração

- Assembler do Neander

Endereço	Instrução	Comentário
0	LDA x	x é um endereço qualquer (por exemplo, 0 ou 128)
2	NOT	
3	ADD y	Mem(y) = 1; usar para y um endereço da área de dados (p.ex. 128)
4	ADD x	

Operação Lógica E

Usar o fato que, para um valor x binário qualquer,
x AND not(x) = F (ou zero)

Alto nível:

AC = x AND not(x)

ou AC = not(x) AND x

Operação Lógica E

- Assembler do Neander

Endereço	Instrução	Comentário
0	LDA x	x é um endereço qualquer (por exemplo, 0 ou 128)
2	NOT	
3	AND x	

Operação Lógica OU

Usar o fato que, para um valor x binário qualquer,

x OR not(x) = V

e, por conseqüência, not(x OR not(x)) = F (ou zero)

Alto nível:

AC = not(x OR not(x))

ou AC = not(not(x) OR x)

Operação Lógica OU

- Assembler do Neander

Endereço	Instrução	Comentário
0	LDA x	x é um endereço qualquer (por exemplo, 0 ou 128)
2	NOT	
3	OR x	
4	NOT	

Operações OU e soma

Uma variação do método anterior, baseado no fato de x OR not(x) gerar uma palavra com 1's binários (11…1111) e que esta representação é do número –1 em complemento de dois

Alto nível:

AC = (x OR not(x)) + 1

ou AC = not(x) OR x + 1

Operação OU e soma

- Assembler do Neander

Endereço	Instrução	Comentário
0	LDA x	x é um endereço qualquer (por exemplo, 0 ou 128)
2	NOT	
3	OR x	
4	ADD y	Mem(y) = 1; usar para y um endereço da área de dados (p.ex. 128)

Operações soma e NOT

Uma variação do método anterior, baseado no fato de x + not(x) gerar uma palavra com 1's binários (11…1111)

Alto nível:

AC = not (x + not(x))

ou AC = not(not(x) + x)

Operação soma e NOT

- Assembler do Neander

Endereço	Instrução	Comentário
0	LDA x	x é um endereço qualquer (por exemplo, 0 ou 128)
2	NOT	
3	ADD x	
4	NOT	

Incrementos sucessivos

Somar um a um número até atingir zero (lembrar que a representação dos números com um comprimento fixo de dígitos é circular, ou seja, incrementando-se o maior positivo obtém-se o menor negativo).

Alto nível:

```
while (AC != 0) do {AC = AC +1}
```

ou

```
DENOVO:     if (AC = 0) then goto FIM
            else {AC = AC + 1; goto DENOVO}
```

Incrementos sucessivos

- Assembler do Neander

Endereço	Instrução	Comentário
0	JZ 6	Endereço 6 corresponde ao label FIM
2	ADD y	Mem(y) = 1; usar para y um endereço da área de dados (p.ex. 128)
4	JMP 0	Endereço 0 corresponde ao label DENOVO
6		Continuação do programa

Incrementos ou Decrementos sucessivos

Se o número é negativo, somar um a um número até atingir zero. Se o número é positivo, subtrair um até atingir zero (com isto o programa é, na média, mais eficiente que a solução anterior).

Alto nível:

```
if (AC <0)
then while (AC != 0) do {AC = AC +1}
else while (AC != 0) do {AC = AC – 1}
```

ou

```
if (AC < 0) then goto DENOVO_N;
DENOVO_P:      if AC = 0 then goto FIM
                  else AC = AC – 1; goto DENOVO_P;
DENOVO_N:      if AC = 0 then goto FIM
                  else AC = AC + 1; goto DENOVO_P;
```

Incrementos ou Decrementos sucessivos

- Assembler do Neander

Endereço	Instrução	Comentário
0	JN 8	Endereço 8 corresponde ao label DENOVO_N
2	JZ 14	Endereço 14 corresponde ao label FIM
4	ADD y	Mem(y) = –1; usar para y um endereço da área de dados (p.ex. 128)
6	JMP 2	Endereço 2 corresponde ao label DENOVO_P
8	JZ 14	Endereço 14 corresponde ao label FIM
10	ADD z	Mem(z) = 1; usar para z um endereço da área de dados (p.ex. 129)
12	JMP 8	Endereço 8 corresponde ao label DENOVO_N
14		Continuação do programa

Deslocamentos sucessivos para esquerda (multiplicação por 2)

A multiplicação de um número pela base (2) insere um zero no dígito menos significativo, desloca todos os dígitos uma casa para a esquerda e elimina o dígito mais significativo. Realizando esta operação 8 vezes, garante-se que todos os 8 bits serão zero.

Alto nível:

```
for i=1 to 8
do AC = AC * 2
```

```
ou          i = 1;
DENOVO:     AC = AC * 2;
            i = i + 1;
            IF i < 9 then goto DENOVO;
```

```
ou          I = 1;
DENOVO:     AC = AC + AC;
            i = i + 1;
            IF i – 9 < 0 then goto DENOVO;
```

Deslocamentos sucessivos para esquerda (multiplicação por 2)

Endereço	Instrução	Comentário
0	STA x	Salva o valor atual do AC em um endereço temporário (p.ex.128)
2	LDA UM	Mem(UM) = 1; usar um endereço da área de dados (p.ex. 129)
4	STA I	Usar para I um endereço da área de dados (p.ex. 130)
6	LDA x	;
8	ADD x	; AC = AC + AC
10	STA x	;
12	LDA I	;
14	ADD UM	; I = I + 1
16	STA I	;
18	ADD M9	Mem(M9) = –9; usar um endereço da área de dados (p.ex. 131)
20	JN 6	
22	LDA x	Continuação do programa

Deslocamentos para esquerda com contagem regressiva

Mesmo método que o anterior, mas agora o controle do laço vai de 8 a zero.

Alto nível:

```
for i=8 downto 1
do AC = AC * 2
```

```
ou      i = 8;
DENOVO:     AC = AC * 2;
        i = i – 1;
        if (i =0) then goto FIM;
        goto DENOVO
```

Deslocamentos para esquerda com contagem regressiva

Endereço	Instrução	Comentário
0	STA x	Salva o valor atual do AC em um endereço temporário (p.ex. 128)
2	LDA Oito	Mem(Oito) = 8; usar um endereço da área de dados (p.ex.129)
4	STA I	Usar para I um endereço da área de dados (p.ex. 130)
6	LDA x	;
8	ADD x	; AC = AC + AC
10	STA x	;
12	LDA I	;
14	ADD M1	; I = I – 1; Mem(M1) = –1 (p.ex. endereço 131)
16	STA I	;
18	JZ 22	Endereço 22 corresponde ao label FIM
20	JMP 6	Endereço 6 corresponde ao label DENOVO
22	LDA x	Continuação do programa

Deslocamentos para esquerda até obter zero

Mesmo método que os dois anteriores, mas agora o laço executa até obter zero.

Alto nível:

```
while (AC != 0)
do AC = AC * 2
```

Ou

```
DENOVO:     if AC = 0 then goto FIM;
            AC = AC + 2;
            goto DENOVO
```

Deslocamentos para esquerda até obter zero

Endereço	Instrução	Comentário
0	JZ 8	Endereço 8 corresponde ao label FIM
2	STA x	Usar para x um endereço da área de dados (p.ex. 128)
4	ADD x	AC = AC + AC
6	JMP 0	Endereço 0 corresponde ao label DENOVO
8		Continuação do programa

Mais idéias?

Novos métodos são bem-vindos.....