

Atividade 01: arquitetura do conjunto de instruções

1) Considerando a equação a seguir faça o que se pede:

- a) Defina a sintaxe e a semântica de um conjunto de instruções que possibilite a implementação da equação. O conjunto de instruções deve usar armazenamento interno baseado em Pilha.
- b) O programa assembly que utiliza o conjunto de instruções definido no item (a) que implementa a seguinte equação:

$$S = \frac{(C - A^B)}{(B - C) \times A}$$

2) Considerando a equação a seguir faça o que se pede:

- a) Defina a sintaxe e a semântica de um conjunto de instruções que possibilite a implementação da equação. O conjunto de instruções deve usar armazenamento interno baseado em Acumulador.
- b) O programa assembly que utiliza o conjunto de instruções definido no item (a) que implementa a seguinte equação:

$$S = \frac{(C - A^3)}{(B - 5) \times (A + 2)}$$

Trabalho realizado por: **Guilherme Rafael Graeff**

RESPOSTAS:

1.a)

Usarei (A) como exemplo ((A) = espaço de memória):

push (A) : Coloca o conteúdo de (A)
no topo da pilha, se a pilha estiver
não estiver vazia todos os elementos
se deslocam dando espaço ao A.

pull (A) : (A) recebe o conteúdo topo da pilha.

adcn : topo <-- topo+topo⁻¹

subt : topo <-- topo-topo⁻¹

divi : topo <-- topo/topo⁻¹

mult : topo <-- topo*topo⁻¹

elev : topo <-- topo^topo⁻¹

b)

push C ;
push B ;
subt ;
push A ;
mult ;
push B ;
push A ;
elev ;
push C ;
subt ;
divi ;
pull S ;

2.a)

Usarei (A) como exemplo ((A) = espaço de memória, A como valor constante e 'w' é o acumulador) :

movmw (A) ; = move da memória para o acumulador.

movmw A ; = move do banco de constantes para o acumulador.

movwm (A) ; = move da do acumulador para a memória.

add (A) ; w <-- w + (A)

add A ; w <-- w + A

sub (A) ; w <-- w - (A)

subb A ; w <-- w - A

mul (A) ; w <-- w * (A)

mull A ; w <-- w * A

div (A) ; w <-- w / (A)

divv A ; w <-- w / A

elev (A) ; w <-- w ^ (A)

elevv A ; w <-- w + ^ A

b)

movmw A ;
addd 2 ;
movwm aux ;
movmw B ;
subb 5 ;
mul aux ;
movwm S ;
movmw A ;
elevv 3 ;
movwm aux ;
movmw C ;
sub aux ;
div S ;
movwm S ;

