

Atividade 2 - Arquitetura do Conjunto de Instruções

Andrew Gabriel - 2011100015

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 Load/Store.

b) O programa assembly que utiliza o conjunto de instruções definido no item (a) que implementa a seguinte equação:

$$S = \frac{(C - 4^B)}{(B - C) \times A}$$

a)

```
Load mem, reg ; reg <- (mem)
Store mem, reg ; (mem) <- reg
Sub regDestino, regX, regY ; regDestino <- (regX - regY)
Mult regDestino, regX, regY ; regDestino <- (regX * regY)
Divi regDestino, regX, regY ; regDestino <- (regX / regY)
Powi regDestino, valuei, regX ; regDestino <- (valuei ^ regX)
```

b)

Programa Assembly:	Descrição:
Load A, R1;	(R1 <- A)
Load B, R2;	(R2 <- B)
Load C, R3;	(R3 <- C)
Sub RR4, R2, R3;	(RR4 <- (B-C))
Mult RR5, RR4, R1;	(RR5 <- (B-C) *A)
Powi RR6, 4, R2;	(RR6 <- (4^B))
Sub RR7, R3, RR6;	(RR7 <- (C-4^B))
Divi RS, RR7, RR5;	(RS <- (C-4^B) / ((B-C) *A))
Store S, RS;	(S <- (C-4^B) / ((B-C) *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:

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

a)

```
moveMreg mem; reg <- (mem)
moveRegM mem; (mem) <- reg
SumI valuei; reg <- reg + valuei
Sub mem; reg <- reg - (mem)
SubI valuei; reg <- reg - valuei
Mult mem; reg <- reg * (mem)
Div mem; reg <- reg / (mem)
PowI valuei; reg <- reg^valuei
```

b)

Programa Assembly:

Descrição:

moveMreg A;	(reg <- A)
SumI 2;	(reg <- (A+2))
moveRegM S;	(S <- (A+2))
moveMreg B;	(reg <- B)
SubI 5;	(reg <- (B-5))
Mult S;	(reg <- (B-5) * (A+2))
moveRegM S;	(S <- (B-5) * (A+2))
moveMreg A;	(reg <- A)
PowI 3;	(reg <- (A^3))
moveRegM T;	(T <- (A^3))
moveMreg C;	(reg <- C)
Sub T;	(reg <- (C-A^3))
Div S;	(reg <- (C-A^3) / (B-5) * (A+2))
moveRegM S;	(S <- (C-A^3) / (B-5) * (A+2))