

**Teste do simulador da Maquina de Von Neumann**

**Santiago Quintero Hincapié**

**11726111**

**PCS 3216**

**USP - POLI**

O teste foi feito com um programa que utiliza todas as instruções, com o objetivo de testar a correta funcionalidade de todas elas.

MNEMÔNICOS	CÓDIGO	INSTRUÇÃO / PSEUDO-INSTRUÇÃO
JP J	/0xxx	JUMP INCONDICIONAL
JZ Z	/1xxx	JUMP IF ZERO
JN N	/2xxx	JUMP IF NEGATIVE
LV V	/3xxx	LOAD VALUE
+ +	/4xxx	ADD
- -	/5xxx	SUBTRACT
* *	/6xxx	MULTIPLY
/ /	/7xxx	DIVIDE
LD L	/8xxx	LOAD FROM MEMORY
MM M	/9xxx	MOVE TO MEMORY
SC S	/Axxx	SUBROUTINE CALL
RS R	/Bxxx	RETURN FROM SUBROUTINE
HM H	/Cxxx	HALT MACHINE
GD G	/Dxxx	GET DATA
PD P	/Exxx	PUT DATA
OS O	/Fxxx	OPERATING SYSTEM CALL

O programa criado:

CI	CO	OP
000	3	013
002	4	051
004	9	01a
006	a	020
008	4	01a
00a	9	01c
00c	e	000
00e	8	01a
010	c	012
012	8	01c
014	6	053
016	7	054
018	0	02a
...		
022	d	000
024	f	000
026	6	052
028	b	020
02a	e	000
02c	5	055
02e	1	034
...		
034	4	050
036	2	03c
...		
03c	e	000

os dados:

Endereco	Dado
050	-2
051	17
052	-1
053	9
054	-7
055	18

A entrada e o endereço:

Entrada:	44
Endereço:	000

#### Descrição de cada linha:

1. Salva no acumulador 13
2. Faz o acumulador =  $13 + 17$
3. Salva na memória o que tem no acumulador (30)
4. Faz uma chamada na subroutine que esta no endereço 022
5. Detém a execução e espera a entrada, neste caso 44
6. Não faz nada (instrução f, operating system call)
7. Faz o acumulador =  $44 * -1$
8. Retorna ao endereço (+2) onde foi chamada a subroutine (008)
9. Faz o acumulador =  $-44 + 30$
10. Salva na memória o que tem no acumulador (-14)
11. Mostra na saída o que tem no acumulador (-14)
12. Carrega no acumulador o que tem no endereço 01a (30)
13. Detém a execução do programa e começa no endereço 012
14. Carrega no acumulador o que tem no endereço 01c (-14)
15. Faz o acumulador =  $-14 * 9$
16. Faz o acumulador =  $-126 / (-7)$
17. Vai ao endereço 02a
18. Mostra na saída o que tem no acumulador (18)
19. Faz o acumulador =  $18 - 18$
20. Vai ao endereço 034 se o acumulador é 0 (sim é)
21. Faz o acumulador =  $0 + (-2)$
22. Vai ao endereço 03c se o acumulador é negativo (é -2)
23. Mostra na saída o que tem no acumulador (-2)

**Resultado da memória e da saída:**

CI	CO	OP
000	3	013
002	4	051
004	9	01a
006	a	020
008	4	01a
00a	9	01c
00c	e	000
00e	8	01a
010	c	012
012	8	01c
014	6	053
016	7	054
018	0	02a
01a	X	XXX
01c	X	XXX
01e	X	XXX
020	X	XXX
022	d	000

  

Endereco	Dado
050	-2
051	17
052	-1
053	9
054	-7
055	18

  

Entrada:  LOAD

Endereço:  RUN

Saida:  Next Step

Acomulador:  Reset