

Virtual Machine

Formato dei comandi

I comandi sono codificati in un microcodice che fa uso di interi a 32 bit. Il byte più significativo contiene il codice dell'istruzione e il byte meno significativo contiene l'eventuale operando

Comandi accettati

<i>N°</i>	<i>Istruzione</i>	<i>Opcode(hex)</i>
0	LDNA Load Numeric in Accumulator	0x000000**
1	LDAR Load Accumulator in Register	0x010000**
2	LDRA Load Register in Accumulator	0x020000**
3	ADDN Add Numeric to Accumulator	0x030000**
4	ADDR Add Register to Accumulator	0x040000**
5	SUBN Subtract Numeric to Accumulator	0x050000**
6	SUBR Subtract Register to Accumulator	0x060000**
7	MULN Multiply Numeric to Accumulator	0x070000**
8	MULR Multiply Register to Accumulator	0x080000**
9	DIVN Divide Accumulator by Numeric	0x090000**
10	DIVR Divide Accumulator by Register	0x0A0000**
11	ANDN Binary bit-to-bit AND between Accumulator and Numeric	0x0B0000**
12	ANDR Binary bit-to-bit AND between Accumulator and Register	0x0C0000**
13	ORNA Binary bit-to-bit OR between Accumulator and Numeric	0x0D0000**
14	ORRA Binary bit-to-bit OR between Accumulator and Register	0x0E0000**

<i>N°</i>	<i>Istruzione</i>	<i>Opcode(hex)</i>
15	JMPA Jump if Accumulator is 0	0x0F000000
16	JNZA Jump if Accumulator is NOT 0	0x10000000
17	JMPR Jump if Register is 0	0x11000000
18	JNZR Jump if Register is NOT 0	0x12000000
19	GOTO	0x130000**
20	HALT Break Execution	0x14000000
21	GRAB Grab a number and put it in Accumulator	0x15000000
22	PRNT Print Accumulator	0x16000000

Esempi

Somma di due numeri:

<i>Assembly</i>	<i>Linguaggio Macchina</i>
GRAB	352321536
LDAR100	16777316
GRAB	352321536
ADDR100	67108964
PRNT	369098752
HALT	335544320

Ritorna uno se un numero è pari, zero se dispari:

<i>Assembly</i>	<i>Linguaggio Macchina</i>
GRAB	352321536
ANDN001	184549377
JNZA	268435456
GOTO7	318767111
LDNA0	0
PRNT	369098752
HALT	335544320
LDNA1	1
PRNT	369098752
HALT	335544320