**LA MACCHINA UNIVERSALE DI VON NEUMANN.**

**DISPOSITIVO OPERATIVO**

Le parole sono di 40 bit

La memoria M contiene 1024 parole M(1), M(2), … M(1024)

La CPU aveva due [registri](https://it.wikipedia.org/wiki/Registro_(informatica)), l'accumulatore (AC) e il moltiplicatore/quoziente (MQ)

Le istruzioni, da 20 bit, erano rappresentate a coppie in una singola parola di 40 bit.

Dei venti bit di ciascuna istruzione, otto erano usati per il codice, mentre i restanti 12

ospitavano un indirizzo a una parola della memoria

codice indirizzo operando codice indirizzo operando

**CCCCCCCC** ZZZZZZZZZZZZ **CCCCCCCC** ZZZZZZZZZZZZ

**DISPOSITIVO LINGUISTICO**

**SET ISTRUZIONI DELLA MACCHINA IAS**

**CODICE BINARIO E**

**CODICE MNEMONICO DESCRIZIONE**

**TRASFERIMENTO DATI**

00001010 LOAD MQ Transfer contents of register MQ to the

accumulator AC

00001001 LOAD MQ,M(X) Transfer contents of memory location X to MQ

00100001 STOR M(X) Transfer contents of accumulator to memory

location X

00000001 LOAD M(X) Transfer M(X) to the accumulator

00000010 LOAD –M(X) Transfer - M(X) to the accumulator

00000011 LOAD |M(X)| Transfer |M(X)| to the accumulator

00000100 LOAD –|M(X)| Transfer -|M(X)| to the accumulator

**TRASFERIMENTO DEL CONTROLLO ASSOLUTO E CONDIZIONATO**

00001101 JUMP M(X,0:19) Take next instruction from left half of M(X)

00001110 JUMP M(X,20:39) Take next instruction from right half of M(X)

00001111 JUMP+ M(X,0:19) If number in the accumulator is nonnegative,

take next instruction from left half of M(X)

00010000 JUMP-M(X,20:39) If number in the accumulator is no positive,

take next instruction from right half of M(X)

**ARITMETICA**

00000101 ADD M(X) Add M(X) to AC; put the result in AC

00000111 ADD |M(X)| Add |M(X)| to AC; put the result in AC

00000110 SUB M(X) Subtract M(X) from AC; put the result in AC

00001000 SUB |M(X)| Subtract |M(X)| from AC; put the remainder in AC

00001011 MUL M(X) Multiply M(X) by MQ; put most significant bits of result in AC,

put least significant bits in MQ

00001100 DIV M(X) Divide AC by M(X); put the quotient in MQ and the remainder in AC

00010100 LSH Multiply accumulator by 2, i.e., shift left one bit position

00010101 RSH Divide accumulator by 2, i.e., shift right one position

**SOSTITUZIONE DI INDIRIZZI DI OPERANDI NELLE ISTRUZIONI**

00010010 STOR M(X,8:19) **Replace left address field at M(X) by 12 rightmost bits of AC**

00010011 STOR M(X,28:39) Replace right address field at M(X) by 12 leftmost bits of AC

--------------------------------------------------------------------------------------------------------------------

**DISPOSITIVO LINGUISTICO DELLA MACCHINA VIRTUALE DI TURING**

Le azioni da eseguire nell’intervallo i-esimo sono definite da uno stato descritto da una **quintupla** di elementi:

**Si**: sigla che identifica l’istruzione corrente;

**Xi**: il simbolo letto sul nastro;

**Yi**: il simbolo da sostituire a **Xi;**

**Vi**: spostamento (destra/sinistra) per scegliere il prossimo carattere,

**Si+1**: sigla che identifica l’istruzione successiva.

---------------------------------------------------------------------------------------------------------------------

**Von Neumann bottleneck**

A way of solving the problem is by using special type of computer known as **Reduced Instruction Set Computers** (RISC).

The main intention of the RISC is to **reduce the total number of memory references made by the CPU**; instead it uses large number of registers for the same purpose.