Varje maskininstruktion är två bytes lång. De första 4 bitarna är op-koden (instruktionen); de sista 12 bitarna är operanden (argumentet). Tabellen nedanför innehåller op-koder (instruktioner) i hexadecimal notation, tillsammans med en kortfattad beskrivning av varje instruktion. Bokstäverna R, S och T används istället för hexadecimala siffror i de fält som representerar en registeridentifierare som varierar beroende på den specifika tillämpningen av instruktionen. Bokstäverna X och Y används istället för hexadecimala siffror i variabla fält som inte representerar ett register.

Each machine instruction is two bytes long. The first 4 bits provide the op-code; the last 12 bits make up the operand field. The table that follows lists the instructions in hexadecimal notation together with a short description of each. The letters R, S, and T are used in place of hexadecimal digits in those fields representing a register identifier that varies depending on the particular application of the instruction. The letters X and Y are used in lieu of hexadecimal digits in variable fields not representing a register.

Op-kod Op-code	Operand Operand	Beskrivning Description
1	RXY	LADDA (LOAD) registret R med bitmönstret som finns i minnescellen vars adress är XY. Exempel: 14A3 skulle göra att innehållet i minnescellen på adress A3 placeras i register 4. *LOAD the register R with the bit pattern found in the memory cell whose address is XY.* *Example: 14A3 would cause the contents of the memory cell located at address A3 to be placed in register 4.*
2	RXY	LADDA (LOAD) registret R med bitmönstret XY. Exempel: 20A3 skulle göra att värdet A3 placeras i register 0. *LOAD the register R with the bit pattern XY.* *Example: 20A3 would cause the value A3 to be placed in register 0.*
3	RXY	LAGRA (STORE) bitmönstret som finns i register R i minnescellen vars adress är XY. Exempel: 35B1 skulle göra att innehållet i register 5 placeras i minnescellen vars adress är B1. *STORE the bit pattern found in register R in the memory cell whose address is XY.* *Example: 35B1 would cause the contents of register 5 to be placed in the memory cell whose address is B1.*
4	0RS	FLYTTA (MOVE) bitmönstret som finns i register R till register S. Exempel: 40A4 skulle göra att innehållet i register A kopieras till register 4. *MOVE the bit pattern found in register R to register S.* *Example: 40A4 would cause the contents of register A to be copied into register 4.*
5	RST	ADDERA (ADD) bitmönstren i register S och T som om de var två-komplementrepresentationer och spara resultatet i register R. Exempel: 5726 skulle göra att de binära värdena i register 2 och 6 läggs till och summan placeras i register 7. ADD the bit patterns in registers S and T as though they were two's complement representations and leave the result in register R. Example: 5726 would cause the binary values in registers 2 and 6 to be added and the sum placed in register 7.
6	RST	ADDERA (ADD) bitmönstren i register S och T som om de representerade värden i flyttalsnotation och spara flyttalsresultatet i register R. Exempel: 634E skulle orsaka att värdena i register 4 och E adderas som flyttalsvärden och resultatet placeras i register 3. ADD the bit patterns in registers S and T as though they represented values in floating-point notation and leave the floating-point result in register R. Example: 634E would cause the values in registers 4 and E to be added as floating-point values and the result to be placed in register 3.
7	RST	ELLER (OR) utför OR på bitmönstren i register S och T och placera resultatet i register R. Exempel: 7CB4 skulle leda till att resultatet av att OR-operationen med register B och 4 placeras i register C. OR the bit patterns in registers S and T and place the result in register R. Example: 7CB4 would cause the result of ORing the contents of registers B and 4 to be placed in register C.
8	RST	OCH (AND) utför OCH med bitmönstren i register S och T och placera resultatet i register R. Exempel: 8045 skulle göra att resultatet av OCH att innehållet i register 4 och 5 placeras i register 0. AND the bit patterns in registers S and T and place the result in register R. Example: 8045 would cause the result of ANDing the contents of registers 4 and 5 to be placed in register 0.
9	RST	EXKLUSIV ELLER (EXCLUSIVE OR / XOR) gör XOR med bitmönstren i register S och T och placera resultatet i register R. Exempel: 95F3 skulle leda till att resultatet av EXKLUSIV ELLER med innehållet i register F och 3 placeras i register 5. EXCLUSIVE OR (XOR) the bit patterns in registers S and T and place the result in register R. Example: 95F3 would cause the result of EXCLUSIVE ORing the contents of registers F and 3 to be placed in register 5.
A	R0X	ROTERA (ROTATE) bitmönstret i register R en bit till höger X gånger. Biten som roteras ut från höger placeras längst till vänster (low-order till high-order). Exempel: A403 skulle göra att innehållet i register 4 roteras 3 bitar till höger på ett cirkulärt sätt. ROTATE the bit pattern in register R one bit to the right X times. Each time place the bit that started at the low-order end at the high-order end. Example: A403 would cause the contents of register 4 to be rotated 3 bits to the right in a circular fashion.
B	RXY	HOPPA (JUMP) till instruktionen som finns i minnescellen vid adress XY om bitmönstret i register R är lika med bitmönstret i register nummer 0. Fortsätt annars med den normala exekveringssekvensen. (Hoppet implementeras genom att kopiera XY till programräknaren under körningsfasen.) Exempel: B43C skulle först jämföra innehållet i register 4 med innehållet i registret 0. Om de två var lika skulle mönstret 3C placeras i programräknaren så att nästa utförda instruktion skulle vara den som finns på den minnesadressen. Annars skulle ingenting göras och programkörning skulle fortsätta i sin normala sekvens. JUMP to the instruction located in the memory cell at address XY if the bit pattern in register R is equal to the bit pattern in register number 0. Otherwise, continue with the normal sequence of execution. (The jump is implemented by copying XY into the program counter during the execute phase.) Example: B43C would first compare the contents of register 4 with the contents of register 0. If the two were equal, the pattern 3C would be placed in the program counter so that the next instruction executed would be the one located at that memory address. Otherwise, nothing would be done and program execution would continue in its normal sequence.
C	0	STOPPA (HALT) utförande. Exempel: C000 skulle göra att programkörningen stoppas HALT execution. Example: C000 would cause program execution to stop

A

Adress/Address	Innehåll/Content	Adress/Address	Innehåll/Content	Adress/Address	Innehåll/Content	Adress/Address	Innehåll/Content
00	10	04	82	08	B1	0C	00
01	0F	05	10	09	0C	0D	20
02	40	06	B3	0A	C0	0E	21
03	21	06	D0	0B	00	0F	FE

PC	IR	R0	R1	R2	COMMENT
0	??	??	??	05	Initial values
2	100F	??	??	05	Fetch cycle 1
2	100F	??	??	05	Decode cycle 1: LOAD R0 with the content of 0F
2	100F	FE	??	05	Execute cycle 1
4	4021	FE	??	05	Fetch cycle 2
4	4021	FE	??	05	Decode cycle 2: MOVE the content of R2 to R1
4	4021	FE	05	05	Execute cycle 2
6	8210	FE	05	05	Fetch cycle 3
6	8210	FE	05	05	Decode cycle 3: AND with R1 and R0, store result in R2
6	8210	FE	05	04	Execute cycle 3

B

Adress/Address	Innehåll/Content	Adress/Address	Innehåll/Content	Adress/Address	Innehåll/Content	Adress/Address	Innehåll/Content
00	12	04	72	08	B1	0C	00
01	0E	05	12	09	0C	0D	20
02	21	06	B3	0A	C0	0E	03
03	FE	06	D0	0B	00	0F	FF

PC	IR	R0	R1	R2	COMMENT
00	??	??	??	05	Initial values
02	120E	??	??	05	Fetch cycle 1
02	120E	??	??	05	Decode cycle 1: LOAD R2 with the content of 0E
02	120E	??	??	03	Execute cycle 1
04	21FE	??	??	03	Fetch cycle 2
04	21FE	??	FE	03	Decode cycle 2: LOAD the bit pattern FE into R1
04	21FE	??	FE	03	Execute cycle 2
06	7212	??	FE	03	Fetch cycle 3
06	7212	??	FE	03	Decode cycle 3: OR with R1 and R2, store result in R2
06	7212	??	FE	FF	Execute cycle 3