Table 2 - DEUNIAC Instruction Set

Symbol		Description
Operation	*Opcode*	
HLT	0111	Halt the computer
Arithmetic and Logic Operations		
Q(1 bit)	Opcode (4 bits)	Rd (2 bits) S1 (2 bits) S2 (2 bits)
DBL	0000	Double content of S1 and store the result in Rd
DBT	0001	Divide content of S1 by 2 and store result to Rd
ADD	0010	Add content of S1 and S2 and store result in Rd
INC	0011	Increase content of S1 and store result in Rd
AND	0100	AND contents of S1 and S2 and store result in Rd
NOT	0101	Complement content of S1 and store the result in Rd
XOR	0110	XOR contents of S1 and S2 and store result in Rd
Data Transfer		Q(1 bit) Opcode (4 bits) Rd (2 bits) S1 (2 bits) S2 (2 bits)
ST	1000	Write the content of Rd into the memory of address S1S2 if Q=0 Write the data *S1S2* into the memory of address indicated by *the content* of *register Rd* if Q=1
LD	1001	Read the data *S1S2* and load it into Rd, if Q=0 Read the memory content of address *S1S2* and load it into Rd, if Q=1
IO	1010	Transfer data from register that is indicated by S1 into OUTR, if Q=0 Registers: 00→ R0, 01→ R1, 10→ R2 Transfer data from INPR into register that is indicated b Rd, if Q=1 Registers: 00→ R0, 01→ R1, 10→ R2
TSF	1011	Transfer data from register that is indicated by S1 into Rd.
		Registers: 00→ R0, 01→ R1, 10→ R2
Program Control		- (1 bit) Opcode (4 bits) -(1 bit) Address (5 bits)
JMP	1100	if Q=0 then jumps to address (5-bits) if Q=1 and if V=1 then jumps to address (5-bits) (V is overflow flag)
CAL	1101	go to the address of the instruction memory (PUSH operation of stack memory)
RET	1110	load the previous PC content from the stack into PC (POP operation of stack memory)
JMR	1111	X (1 bit) Opcode (4 bits) XX (2 bits) Address (4 bits - signed) Use *Address* as offset and jump to address relatively