MPC-1 (MicroProCessor -1)

PC (Program Counter)
- Counts from 0000 to 1111, so hexadecimally from 0 tot F.
- Is reset to 0000 before every computer run
- Contains the fetch address of the current instruction at the start of every fetch cycle.

MAR (Memory Address Register)
- Receives binary addresses from the Program Counter
- Reads instructions (ROM words) from these addresses

PROM (Programmable Read-Only Memory)
- Contains ROM words of 8 bits

IR (Instruction Register)
- Splits the ROM words:
  - Most Significant Bits: opcode of 4 bits
  - Least Significant Bits: address of 4 bits

CON (Control Unit) **(!!!)**
- Resets IR and PC at the start of a computer run using the CLR-signal
- Sends clock pulses to all registers so everything runs synchronized
- Determines the behaviour of all systems based on the opcode
- The CON exit signal is a 12 bit control word with instructions for all registers: CON = $C_P E_P L_M E_R \ L_I E_I L_A E_A \ S_U E_U L_B L_O$
- This determines how the registers will react to the next positive clock flank

For example, C_P high means the program counter will be increased by one, E_P and L_M high means the contents of the PC are loaded into the MAR, etc. CON directs all of the parts of the MPC-1 computer

A (Accumulator)
- Stores temporary values during the computer run

ALU (Arithmetic-Logic unit)
- Contains a 2-complement adder / subtractor:
  - When the signal on the subtraction input (Su) is low, the ALU will make the sum with B and store the result in A
  - When Su is high, A will become the algebraic subtraction of A and B.

B (B-register)
- Temporary storage for the number to be added to or subtracted from A

O (Output register)
- Output of the result to the outside world (D)

D (Binary display)
- Shows the result in 8 LEDs, meaning, in binary

Instruction set:

Mnemonics (assembly)	op-code (binary)	Function
LDA	0000	Loads accumulator with contents of R_X (= ROM address #X)
ADD	0001	Add contents of R_X to contents of the accumulator
SUB	0010	Subtract contents of R_X to contents of the accumulator
OUT	1110	Load contents of the accumulator in the output register
HLT	1111	Stops the data processing (end program)

Example:

R_0	0000 1111	LDA R_F
R_1	0001 1110	ADD R_E
R_2	0010 1101	SUB R_D
R_3	1110 xxxx	OUT
R_4	1111 xxxx	HLT
R_5	xxxx xxxx	
R_6	xxxx xxxx	
R_7	xxxx xxxx	
R_8	xxxx xxxx	
R_9	xxxx xxxx	
R_A	xxxx xxxx	
R_B	xxxx xxxx	
R_C	xxxx xxxx	
R_D	0010 0101	37
R_E	0011 1101	61
R_F	0001 0100	20