

MPC-1 (MicroProCessor -1)

PC (Program Counter)

- Telt van 0000 tot 1111, dus hexadecimaal van 0 tot F.
- Wordt vóór iedere computerrun op 0000 gereset.
- Bevat bij het begin van iedere ophaalcyclus het adres van de huidige instructie

MAR (Memory Address Register)

- Ontvangt binaire adressen van Program Counter
- Leest instructies (ROM woorden) van die adressen

PROM (Programmable Read-Only Memory)

- Bevat ROM woorden van 8 bytes

IR (Instruction Register)

- Splitst de ROM woorden:
 - Most Significant Bits: opcode van 4 bytes
 - Least Significant Bits: adres van 4 bytes

CON (Control Unit) (!!!)

- Reset IR en PC aan het begin van een computerrun met CLR-sigitaal
- Stuurt klokimpulsen naar alle registers zodat alles synchroon loopt
- Bepaalt uit de opcode hoe alle apparaten zich moeten gedragen
- Het CON uitgangssigitaal is een 12 bit stuurwoord met instructies voor alle registers: $CON = C_P E_P L_M E_R L_I E_I L_A E_A S_U E_U L_B L_O$
- Dit bepaalt hoe de registers op de volgende positieve klokflank zullen reageren

Zo betekent bijvoorbeeld C_P hoog dat de programmateller met één wordt vermeerderd, E_P en L_M hoog wil zeggen dat de inhoud van de PC in het MAR wordt geladen, enz.

CON dirigeert al de diverse onderdelen van de MPC-1 computer

A (Accumulator)

- Slaats tussenliggende waarde op gedurende de computerrun

ALU (Arithmetic-Logic unit)

- Bevat een 2-complement opteller/ aftrekker:
 - Als het sigitaal op de aftrekingang (S_u) laag is, maakt de ALU de som met B
 - Wanneer S_u hoog is, krijgt u het algebraïsch verschil met B.

B (B-register)

- Tijdelijke opslagplaats voor het getal dat bij A moet opgeteld of afgetrokken worden

O (Output register)

- Uitvoer van de uitkomstwaarde naar de buitenwereld (D)

D (Binary display)

- Toont de uitkomstwaarde in 8 LEDs, dus binair

Instructieset:

Mnemonics (assembly)	op-code (binair)	Functie
LDA	0000	Laad accumulator met inhoud van R_X (= ROM adres #X)
ADD	0001	Tel inhoud van R_X bij inhoud van de accumulator op
SUB	0010	Trek inhoud van R_X van inhoud van de accumulator af
OUT	1110	Laad accumulatorinhoud in het uitvoerregister
HLT	1111	Zet de dataverwerking stop (einde programma)

Voorbeeld:

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