OPERANDS • EXECUTE • WRITE BACK

FETCH • DECODE • FETCH

Befehlsdecoder

HLT = F4 *z* 1111 0100	**COB** = 0*zz*	**Stop**	**Ld** 01 111	stoppe alles (schreibe I/O)
INC = 40 *z* 0100 0000	**ADD** = 1*zz*	**+** *z*	**La** 01 100	addiere *z* plus **1**
DEC = 48 *z* 0100 1000	**SUB** = 2*zz*	**–** *z*	**Ls** 01 101	subtrahiere *z* minus **1**
JMP = E9 *z* 1110 1001	**BRA** = 6*zz*	**S** *z*	–	springe nach *z*
JZ/ISZ=74 *z* 0111 0100	**BRZ** = 7*zz*	**0** *z*	–	wenn *z*==0/NUL: springe *z*+2 o. *zz*
MOV = A0 *z* 1010 0000	**LDA** = 5*zz*	–	**Pr** *z* 11 *zzzzzz*	lies *z* nach R1/EAX
MOV = B0 *z* 1011 0000	**STO** = 3*zz*	–	**Ps** *z* 10 *zzzzzz*	schreibe R1/EAX nach *z*
IN = E4 *z* 1110 0100	**IN** = 901	–	**Lu** 01 110	lies I/O-Butter in *z* (o. R1/EAX)
OUT = E6 *z* 1110 0110	**OUT** = 902	–	**Ld** 01 111	schreibe I/O-Butter aus *z* (o. R1/EAX)

CU

+R2– R1/EAX

XOR

AND **ALU**

Übertrag

I/O

	•0	•1	•2	•3	•4	•5	•6	•7	•8	•9	•A	•B	•C	•D	•E	•F
0•	0000 0000 00	0000 0001 01	0000 0010 02	0000 0011 03	0000 0100 04	0000 0101 05	0000 0110 06	0000 0111 07	0000 1000 08	0000 1001 09	0000 1010 10	0000 1011 11	0000 1100 12	0000 1101 13	0000 1110 14	0000 1111 15
1•	0001 0000 16	0001 0001 17	0001 0010 18	0001 0011 19	0001 0100 20	0001 0101 21	0001 0110 22	0001 0111 23	0001 1000 24	0001 1001 25	0001 1010 26	0001 1011 27	0001 1100 28	0001 1101 29	0001 1110 30	0001 1111 31
2•	0010 0000 32	0010 0001 33	0010 0010 34	0010 0011 35	0010 0100 36	0010 0101 37	0010 0110 38	0010 0111 39	0010 1000 40	0010 1001 41	0010 1010 42	0010 1011 43	0010 1100 44	0010 1101 45	0010 1110 46	0010 1111 47
3•	0011 0000 48	0011 0001 49	0011 0010 50	0011 0011 51	0011 0100 52	0011 0101 53	0011 0110 54	0011 0111 55	0011 1000 56	0011 1001 57	0011 1010 58	0011 1011 59	0011 1100 60	0011 1101 61	0011 1110 62	0011 1111 63
4•	0100 0000 64	0100 0001 65	0100 0010 66	0100 0011 67	0100 0100 68	0100 0101 69	0100 0110 70	0100 0111 71	0100 1000 72	0100 1001 73	0100 1010 74	0100 1011 75	0100 1100 76	0100 1101 77	0100 1110 78	0100 1111 79
5•	0101 0000 80	0101 0001 81	0101 0010 82	0101 0011 83	0101 0100 84	0101 0101 85	0101 0110 86	0101 0111 87	0101 1000 88	0101 1001 89	0101 1010 90	0101 1011 91	0101 1100 92	0101 1101 93	0101 1110 94	0101 1111 95
6•	0110 0000 96	0110 0001 97	0110 0010 98	0110 0011 99	0110 0100 100	0110 0101 101	0110 0110 102	0110 0111 103	0110 1000 104	0110 1001 105	0110 1010 106	0110 1011 107	0110 1100 108	0110 1101 109	0110 1110 110	0110 1111 111
7•	0111 0000 112	0111 0001 113	0111 0010 114	0111 0011 115	0111 0100 116	0111 0101 117	0111 0110 118	0111 0111 119	0111 1000 120	0111 1001 121	0111 1010 122	0111 1011 123	0111 1100 124	0111 1101 125	0111 1110 126	0111 1111 127

SPEICHERWERK

Dr. WJC Röhricht

Papiercomputer

Die Geschichte der Papiercomputer beginnt im hohen Mittelalter, als Autoren wie Ramon LLULL[1] komplexe Heuristiken mit Kreisen und Rädern in Bücher wie die **«Ars magna»** zeichneten. Im heutigen Sinne tauchten die ersten Veranschaulichungen moderner Computer in der Mitte des 20ten Jhdts. auf: 1965 entstand der so genannte **«Little Man Computer»** von Dr. Stuart MADNICK zur Veranschaulichung der VON-NEUMANN-Architektur mit In- und Output, Speicher- und Rechenwerk[2]. Die wahrscheinlich größte Verbreitung erfuhr dann der bekannte, 1983 von Rudolf BACK (WDR) und Ulrich ROHDE (PC-Magazin) erfundene, **«Know-How-Computer»**[3]. Mehrere hunderttausend Exemplare wurden davon verteilt. Hier hat man Programmspeicher und Datenregister getrennt voneinander dargestellt, Steuer- und Rechenwerk leistete der Nutzer und statt Nullen und Einsen dienten Streichhölzer als Daten. Fünf einfache Befehle erlaubten alle Grundrechenarten auszuführen. Mitte bis Ende der 1980er Jahre wurden auch in der damaligen DDR Papiercomputer vorgestellt, und zwar in der Kinder- und Jugend-Zeitschrift „FRÖSI": 1986, 1988 und 1989 konnten die Leser und Leserinnen in drei Versionen aus diversen „Speichern" grafische Elemente z.B. zu Häusern oder Schiffen zusammensetzen – zur Veranschaulichung von CAD-CAM mit dem **«Frösi-Papiercomputer»**[4].

Lehr- und Lernmittel

Alle diese Papiercomputer – im Grunde auch die mittelalterlichen Heuristiken – dienten natürlich nicht wirklich der automatischen Informationsverarbeitung, sondern haben ihren Platz in der Lehre i.w.S.

Es lassen sich so mit einfachen Mitteln die tatsächlichen Vorgänge in einem (modernen) Rechner recht genau darstellen. Aus den bekannten Vorlagen versucht deswegen auch der **EJG-Papiercomputer** die für das Verständnis der inneren Abläufe in einem PC hilfreichen „Bausteine" zu kombinieren: In- und Output (**I/O**) und einheitlicher **Speicher** für Programm und Daten haben wir dem «Little Man Computer» entnommen, den stark vereinfachten Befehlssatz u.a. dem «Know-How-Computer». Eigenentwickelt hingegen sind Steuerwerk (**CU**) und Rechenwerk (**ALU**) entworfen, um die Arbeit dieser wesentlichen Bausteine deutlicher zu machen. Die Schülerinnen und Schüler können übrigens einfach mit dem Dezimalsystem rechnen, aber auch binäre oder hexadezimale Daten können dargestellt und verarbeitet werden, was auch den Zusammenhang der Stellenwertsysteme besser veranschaulichen hilft. Die **Reduktion** der Rechenoperationen auf die Addition (und die **Subtraktion als Addition des Zweierkomplements**) von **1** erlaubt auch eine einfache symbolische Darstellung des Addierers als **XOR-AND-Logikgatter**.

Programmierung

Für die praktische Anwendung ist das Schreiben von einfachen Programmen hilfreich. Der implementierte **Befehlsdecoder** im Steuerwerk ermöglicht dabei die Nutzung des EJG-Papiercomputers sowohl mit den fünf Befehlen des Know-How-Computers, dem Befehlssatz des LMC, mit Teilen des Opcodes von ZUSES **Z3** als auch mit einigen Assembler-Mnemonics. Der extrem vereinfachte hexadezimale und binäre Opcode dient der „maschinennahen" Programmierung. Auch die Abarbeitung des **VON-NEUMANN-Zyklus** ermöglicht unsere Darstellung des Steuerwerks. Das Speicherwerk mit einem **Adressraum von 7 Bit** wurde so aufgebaut, dass die Adressierung sowohl dezimal, als auch binär oder hexadezimal erfolgen kann.

Daten und Programmbefehle werden händisch in die Speicherzellen notiert. Als Beispiel soll hier das angepasste Standardprogramm des Know-How-Computers dienen, welches die einfache Addition der Inhalte zweier Speicherzellen implementiert:

- Zelle **64**: **`S 67`**
- Zelle **65**: **`+ 00`**
- Zelle **66**: **`– 01`**
- Zelle **67**: **`O 01`**
- Zelle **68**: **`S 65`**
- Zelle **69**: **Stop**

In die Zellen **`00`** und **`01`** sollten positive ganze Zahlen notiert werden. Dann startet die Simulation in Speicherzelle **64**.

[1] HNF (2016): Das Universum des Ramon Llull.– https://blog.hnf.de/das-universum-des-ramon-llull/

[2] Wikipedia: Little Man Computer.– https://en.wikipedia.org/wiki/Little_man_computer

[3] BACK, R. & ROHDE, U. (1983): DER KNOW HOW COMPUTER. Entwickelt von PC Magazin und WDR ComputerClub.– PC Magazin, Beilage

[4] WILKENDORF, D. & HAMBACH, R. (1986): CAD-CAM mit dem „Frösi"-Papiercomputer.– FRÖSI 1986/04, Beilage

Viel Freude beim Ausprobieren
wünscht Dr. WJC Röhricht