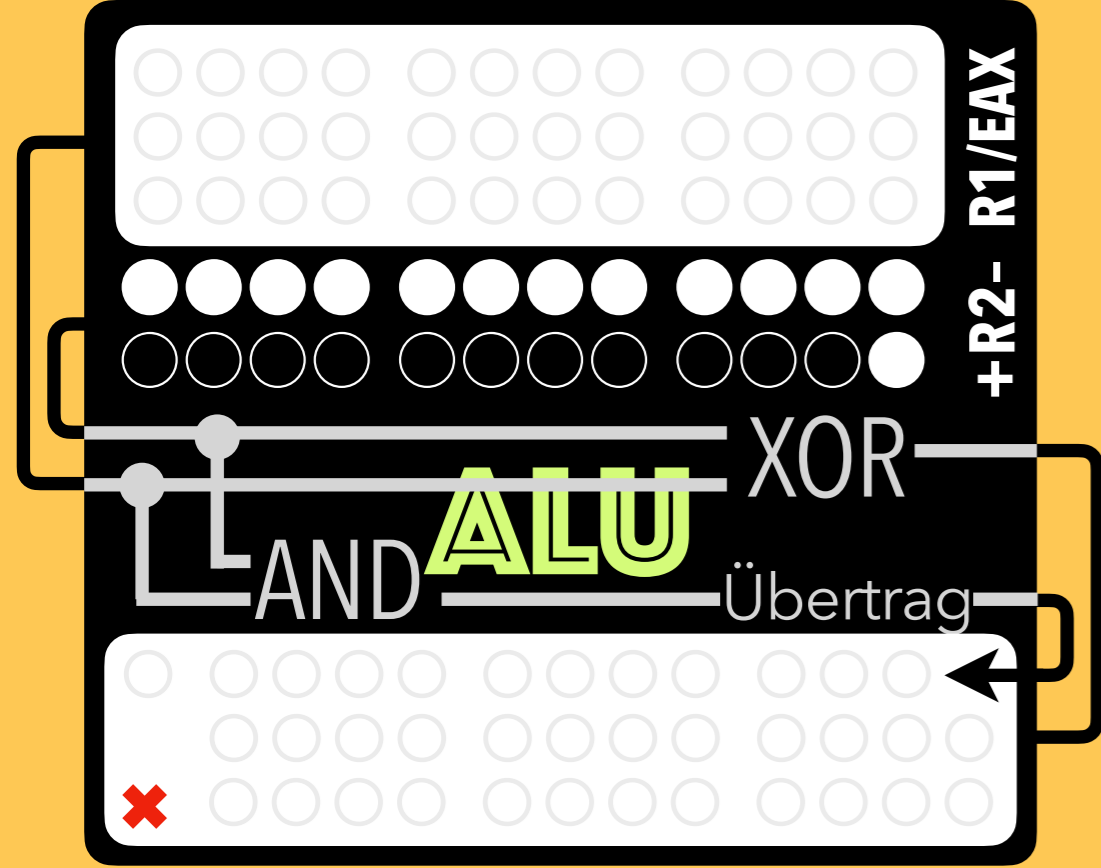
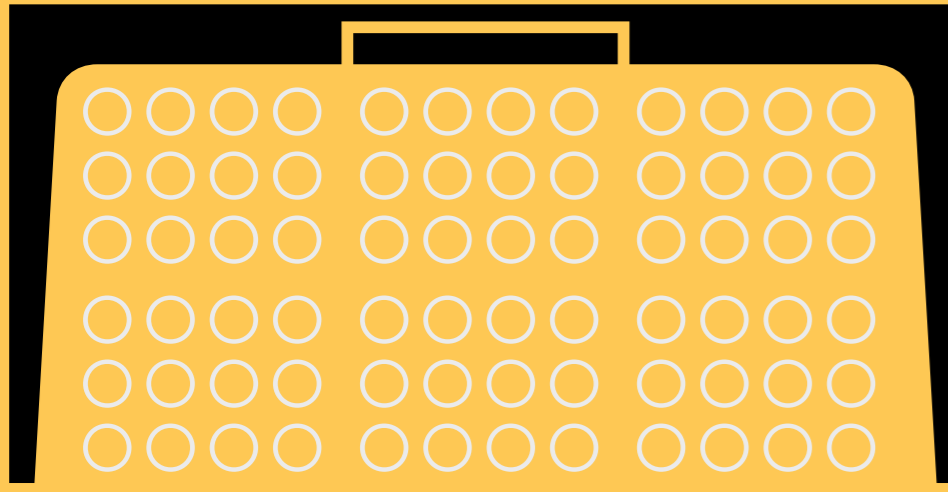
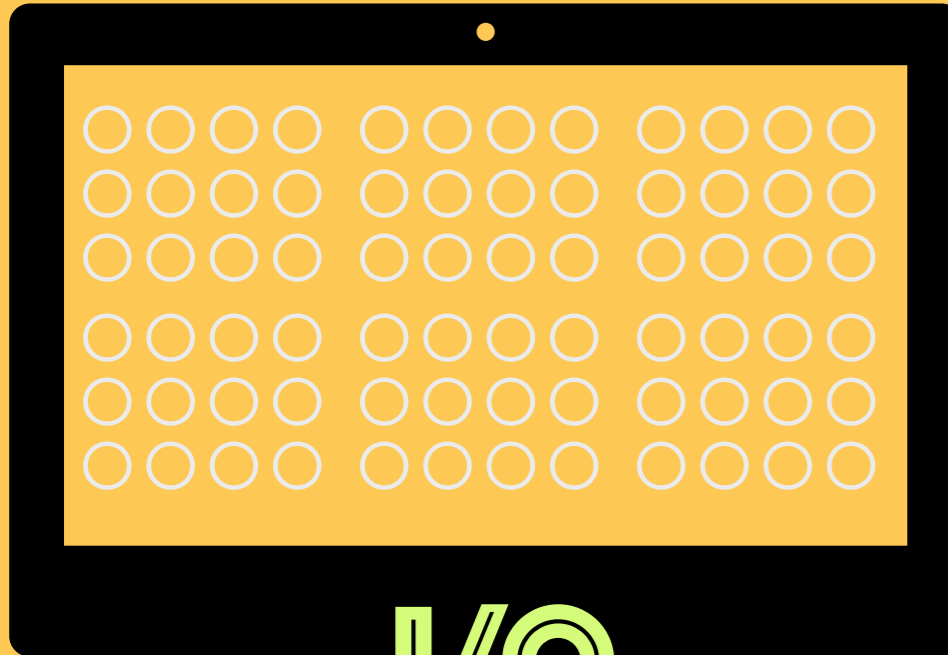


| OPERANDS | | EXECUTE | | WRITE BACK | |
|----------------------------------|----------------------|--------------------------|---------------------------------------|-----------------------------|--|
| HLT = F4 z 1111 0100 | COB = 0zz | Stop | Ld 01 111 | stoppe alles (schreibe I/O) | |
| INC = 40 z 0100 0000 | ADD = 1zz + z | La 01 100 | addiere z plus 1 | | |
| DEC = 48 z 0100 1000 | SUB = 2zz - z | Ls 01 101 | subtrahiere z minus 1 | | |
| JMP = E9 z 1110 1001 | BRA = 6zz S z | - | springe nach z | | |
| JZ/ISZ =74 z 0111 0100 | BRZ = 7zz 0 z | - | wenn z==0/NUL: springe z+2 o. zz | | |
| MOV = A0 z 1010 0000 | LDA = 5zz - | Pr z 11 zzzzzz | lies z nach R1/EAX | | |
| MOV = 80 z 1011 0000 | STO = 3zz - | Ps z 10 zzzzzz | schreibe R1/EAX nach z | | |
| IN = E4 z 1110 0100 | IN = 901 - | Lu 01 110 | lies I/O-Buffer in z (o. R1/EAX) | | |
| OUT = E6 z 1110 0110 | OUT = 902 - | Ld 01 111 | schreibe I/O-Buffer aus z (o. R1/EAX) | | |

CU



I/O



| | · 0 | · 1 | · 2 | · 3 | · 4 | · 5 | · 6 | · 7 | · 8 | · 9 | · A | · B | · C | · D | · E | · F |
|----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 0. | 0000 0000 | 0000 0001 | 0000 0010 | 0000 0011 | 0000 0100 | 0000 0101 | 0000 0110 | 0000 0111 | 0000 1000 | 0000 1001 | 0000 1010 | 0000 1011 | 0000 1100 | 0000 1101 | 0000 1110 | 0000 1111 |
| 1. | 0001 0000 | 0001 0001 | 0001 0010 | 0001 0011 | 0001 0100 | 0001 0101 | 0001 0110 | 0001 0111 | 0001 1000 | 0001 1001 | 0001 1010 | 0001 1011 | 0001 1100 | 0001 1101 | 0001 1110 | 0001 1111 |
| 2. | 0010 0000 | 0010 0001 | 0010 0010 | 0010 0011 | 0010 0100 | 0010 0101 | 0010 0110 | 0010 0111 | 0010 1000 | 0010 1001 | 0010 1010 | 0010 1011 | 0010 1100 | 0010 1101 | 0010 1110 | 0010 1111 |
| 3. | 0011 0000 | 0011 0001 | 0011 0010 | 0011 0011 | 0011 0100 | 0011 0101 | 0011 0110 | 0011 0111 | 0011 1000 | 0011 1001 | 0011 1010 | 0011 1011 | 0011 1100 | 0011 1101 | 0011 1110 | 0011 1111 |
| 4. | 0100 0000 | 0100 0001 | 0100 0010 | 0100 0011 | 0100 0100 | 0100 0101 | 0100 0110 | 0100 0111 | 0100 1000 | 0100 1001 | 0100 1010 | 0100 1011 | 0100 1100 | 0100 1101 | 0100 1110 | 0100 1111 |
| 5. | 0101 0000 | 0101 0001 | 0101 0010 | 0101 0011 | 0101 0100 | 0101 0101 | 0101 0110 | 0101 0111 | 0101 1000 | 0101 1001 | 0101 1010 | 0101 1011 | 0101 1100 | 0101 1101 | 0101 1110 | 0101 1111 |
| 6. | 0110 0000 | 0110 0001 | 0110 0010 | 0110 0011 | 0110 0100 | 0110 0101 | 0110 0110 | 0110 0111 | 0110 1000 | 0110 1001 | 0110 1010 | 0110 1011 | 0110 1100 | 0110 1101 | 0110 1110 | 0110 1111 |
| 7. | 0111 0000 | 0111 0001 | 0111 0010 | 0111 0011 | 0111 0100 | 0111 0101 | 0111 0110 | 0111 0111 | 0111 1000 | 0111 1001 | 0111 1010 | 0111 1011 | 0111 1100 | 0111 1101 | 0111 1110 | 0111 1111 |

SPEICHERWERK

Papiercomputer

Die Geschichte der Papiercomputer beginnt im hohen Mittelalter, als Autoren wie Ramon LLULL¹ 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 Rechenwerk². 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.

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

Lehr- und Lernmittel

Alle diese Papiercomputer – mit Ausnahme der 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-Logiggatter**.

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**: **0 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**.

*Viel Freude beim Ausprobieren
wünscht Dr. WJC Röhrich*