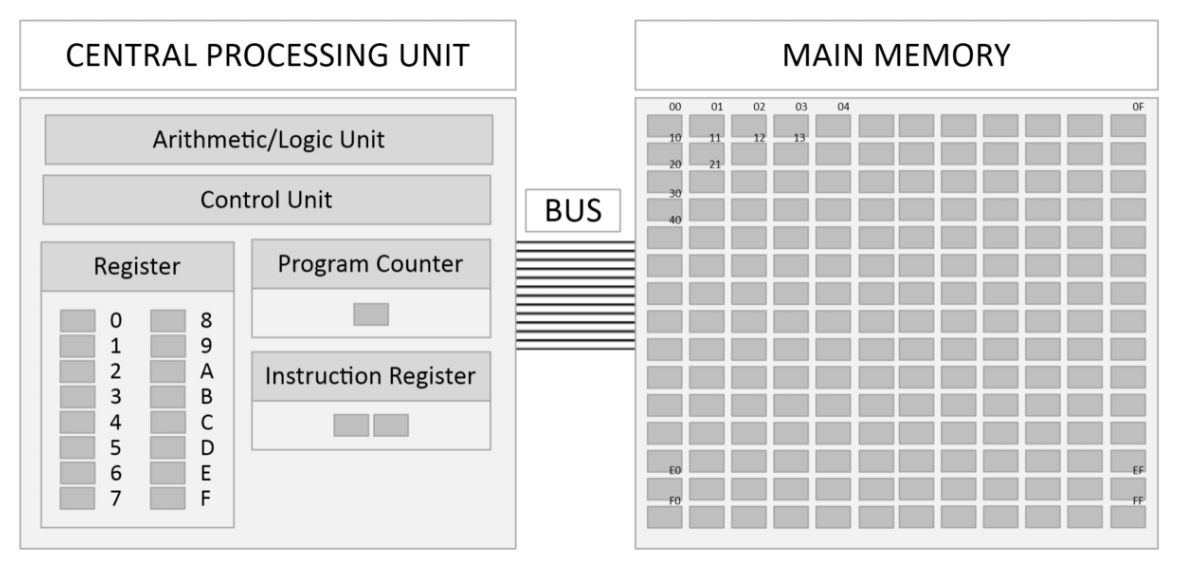
# Architektur der Beispielmaschine

[Aka VizMachine]

* Die Maschine besitzt 16 general-purpose Register die von 0 bis F nummeriert sind.
* Jedes Register ist ein Byte breit.
* Das jeweilige Register wird innerhalb der Instruktionen durch 4 Bit angegeben (0000 entspricht Register 0, 0100 Register 4, 1111 Register F, ...)
* Die Maschine hat einen Hauptspeicher von 256 Byte.
* Jede Speicherzelle (ein Byte) wird durch eine Adresse von 0 bis 255 (00 bis FF hexadezimal) angesprochen.
* Floating-point Werte werden wie folgt dargestellt (vom höchstwertigen Bit abwärts): 1 Bit Vorzeichen, 3 Bit Exponent, 4 Bit Mantisse.
* Jede Maschineninstruktion ist 2 Byte lang und besteht aus einem op-code mit 4 Bit und einem Operanden-Feld von 12 Bit.
* In der folgenden Beschreibung stehen die Buchstaben R,S,T im Operanden für eine Hexadezimalziffer, die die Nummer eines Registers darstellt.
* Die Buchstaben X und Y stehen im Operanden für Hexadezimalziffern, die kein Register angeben. Sie repräsentieren einen Hexadezimalen Wert oder eine Speicheradresse zwischen 0 und FF.



|  |  |  |
| --- | --- | --- |
| **Op-code** | **Operand** | **Beschreibung** |
| 1 | RXY | LOAD Register R mit dem Inhalt der Speicherzelle mit Adresse XY.  ( Register/Memory Direct Addressing) |
| 2 | RXY | LOAD Register R mit dem Wert (Bitmuster) XY.  (Immediate Value) |
| 3 | RXY | STORE Inhalt aus Register R in Speicherzelle mit Adresse XY. |
| 4 | 0RS | MOVE Inhalt aus Register R ins Register S. |
| 5 | RST | ADD Inhalt aus Register S und Register T (2-er Komplement Interpretation), Ergebnis wird in Register R gespeichert. |
| 6 | RST | ADD Inhalt aus Register S und Register T (Floating-Point Interpretation), Ergebnis wird in Register R gespeichert. |
| 7 | RST | OR der Bitmuster der Registern S und T, speichere Ergebnis in Register R. |
| 8 | RST | AND der Bitmuster der Registern S und T, speichere Ergebnis in Register R. |
| 9 | RST | XOR der Bitmuster der Registern S und T, speichere Ergebnis in Register R. |
| A | R0X | ROTATE das Bitmuster in Register R ein Bit nach rechts, X-mal. |
| B | RXY | JUMP zur Instruktion in der Speicherzelle mit Adresse XY, falls Inhalt von Register R gleich dem Inhalt von Register 0 ist. |
| C | 000 | HALT. |

**Zusätzliche Operationen**

|  |  |  |
| --- | --- | --- |
| **Op-code** | **Operand** | **Description** |
| D | XYZ | WAIT in Millisekunden angegeben durch XYZhex Wert. |
| E | RST | WRITE Daten von Register R in die Speicherzelle, deren Adresse in Register T angegeben ist.  (Register Indirect) |