# Zilog Z8671

Der Zilog Z8671 ist ein Einchip-Mikrorechner von Zilog. Er gehört zu Z8-Reihe. Im internen ROM ist ein BASIC-Interpreter enthalten.




Von Zilog gibt es neben den Z8-Datenbüchern auch ein zusätzliches Dokument „Z8671 Single-chip Interpreter, BASIC/DEBUG Software, Reference Manual, Zilog, Incorporated, 1988 (BASIC/Debug Software Reference Manual #03-3149-02)“.

Der Z8671 wurde auch von SGS-Thomson und von ST Microelectronics produziert.

Der in der DDR produzierte Z8-Clone U883 mit integriertem BASIC ist **nicht** zum Z8671 kompatibel; er enthält ein völlig anderes (eigenständig entwickeltes) BASIC.

## Literatur



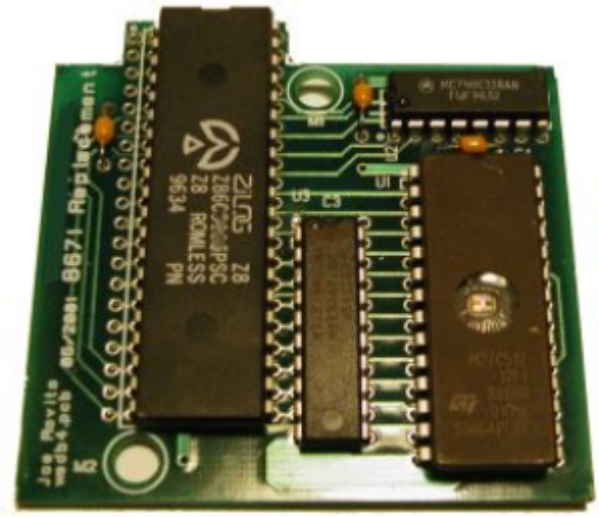





- Z8671 Single-chip BASIC Interpreter: BASIC/DEBUG Software Reference Manual, Zilog, 1988
- „New Life for the Z8671 BASIC Interpreter“ by Chris Howie (http://www.armory.com/~rstevew/Public/Micros/Z8/Z8671-BASIC/Z8671-BASIC_ROM_Main.htm) Ersatz des Z8671 durch einen Z86C03 mit externem EPROM, leicht geänderte Software. Offenbar hatte der Autor das komplette Quelllisting des BASIC/DEBUG von ZILOG
- Z8_Family_Design_Handbook_Jun88.pdf (Zilog, 660 Seiten, 38 MB) (http://www.bitsavers.org/components/zilog/z8/)
- Z8 microcomputer family May86.pdf (SGS, 3th edition, 302 Seiten, 19 MB) enthält Beispielapp. „Single Board Computer Using Z8671“ und „A Programmer's Guide to the Z8 Microcomputer“
- Z8671-Emulator (http://z8671sim.web-log.nl/)
- „Einführung in die Mikroprozessor-Anwendung“, Roland Schule; Axel Gruppe., Unter Mitarb. von Michael Zillgitt. Jean Pütz (Hrsg.)., - 1. Aufl. - Köln: vgs, 1987. (Experimente), ISBN 3-8025-1239-1
  Ein deutsches Buch über einen SBC "BERT" mit Z8671, enthält deutsche Beschreibung des Z8671 BASIC/DEBUG.

- ZiLOG Z8671 BASIC/Debug chip Replacement (http://www.ntrd.com/joe/z8671.htm) (leider offline)
- FORTH for the Zilog Z8 Microcontroller http://maptools.home.comcast.net/~maptools/Z8/ hier gibt es den Z8asm/link 32bit Version 3.02 von Zilog.
- https://web.archive.org/web/20091024152542/http://maptools.home.comcast.net/~maptools/Z8/

- http://www.bitsavers.org/components/zilog/z8/1983_Z8_Microcomputer_Technical_Manual.pdf
- https://cdn.hackaday.io/files/260591114908256/Z8671%20App%20Note.pdf



Z8671 Replacement Board circa 2001, Bild von http://www.ntrd.com/joe/z8671.htm

## Downloads

- z6871_rom.zip ROM-Inhalt
- z8671.zip z8671.bin, z8671_basic_debug.pdf, Z8_crd.txt, Zilog Z9671 datasheet pdf
- z8671_app_note.pdf

## BASIC/DEBUG

## Initialisierung

Wenn der Reset-Eingang des Z8671 auf L-Pegel geht, so stoppt er alle Aktivitäten. Beim Wiederanstieg dieses Signals beginnt der Z8, das Maschinenprogramm im internen Festwertspeicher ab Adresse %000C auszuführen; dieses bewirkt einen Betriebsprogramm-Neustart. In dessen Verlauf werden alle Programm-Unterbrechungen abgeschaltet und einige Register vorbesetzt. Dabei wird auch die Benutzung der Ports 0 bis 3, also der Prozessor-Ports festgelegt.

Danach wird die Übertragungsrate zum PC festgelegt, indem die Wert von Adresse %FFFD eingelesen wird. Dabei spielen nur die niedrigstwertigen drei Bits eine Rolle. Die höheren Bits sind beliebig und können für andere Zwecke verwendet werden.

| Wert auf %FFFD | Baudrate |
|---|---|
| x x x x x 0 0 0 | 150 |

| Wert auf %FFFD | Baudrate |
|---|---|
| x x x x x 0 0 1 | 19200 |
| x x x x x 0 1 0 | 9600 |
| x x x x x 0 1 1 | 4800 |
| x x x x x 1 0 0 | 2400 |
| x x x x x 1 0 1 | 1200 |
| x x x x x 1 1 0 | 110 |
| x x x x x 1 1 1 | 300 |

In der Reset-Routine folgt dann das Überprüfen der Speicher-Bestückung. Je nach RAM-Ausbau wird den Variablen ihr Platz zugewiesen und der Stapelzeiger für Rücksprungadressen der Unterprogramme gesetzt; außerdem werden Zeiger auf Anfang und Ende des nutzbaren BASIC-Speichers gesetzt.

Wichtig ist bei der Reset-Routine jedoch auch das Prüfen des Speichers bei niedrigen Adressen. Insbesondere interessiert sich das Betriebssystem für den Inhalt des Speicherzellen-Paares %1020 und %1021. Sollte sich hier RAM befinden, dann nimmt der Z8671 über BASIC/DEBUG den Dialog mit dem Benutzer auf, indem er den Doppelpunkt sendet. Ist an %1020 Festwertspeicher, kommt es darauf an, ob an dieser Stelle eine Zahl steht, deren höherwertiges Byte gelöscht ist. Mit anderen Worten: die Zahl muß positiv und kleiner als 256 sein. Ist dies nicht der Fall, so wird ebenfalls der Dialog aufgenommen.

Wenn die Bedingung jedoch erfüllt ist, nimmt BASIC/DEBUG die in %1020 und %1021 stehende Zahl als die erste Zeilennummer eines BASIC/DEBUG-Programms und beginnt sofort mit der Ausführung desselben.

# Register

```
%FF Stapelzeiger low
%FE Stapelzeiger high
%FD Registerblock-Zeiger
%FC Flaggenregister
%FB Programmunterbrechung Maskenregister
%FA Programmunterbrechung Anforderungsregister
%F9 Programmunterbrechung Prioritätsregister
%F8 Betriebsartenregister P0 und P1 (Daten- und Adreßbus)
%F7 Betriebsartenregister P3
%F6 Datenrichtungsregister P2
%F5 Vorteilerregister 0
%F4 Zähler-/Zeitgeberregister 0
%F3 Vorteilerregister 1
%F2 Zähler-/Zeitgeberregister 1
%F1 Betriebsartenregister Zähler/Zeitgeber
%F0 Terminal Datenregister

%EF-%80 unbelegt

%7F Arithmetik-Stapelspeicher
%6B Arithmetik-Stapelspeicher
```

```
...
%48 Programmunterbrechung Quelle
%47 Zählerüberlauf
%46 Kopie des Registers %F6
%45 Abeitsregister bei Programmunterbrechungen
%40 Arbeitsregister bei Programmunterbrechungen
%3F Zeiger Arithmetik-Stapelspeicher (Kommandos)
%3E Kopie der letzten Ausgabe an Port D
%3D Kopie der letzten Ausgabe an Port B
%3C Kopie der letzten Ausgabe an Port A
%3B Kommandointerpreter Textzeiger low
%3A -"- high
%39 Kommandointerpreter Tabellenzeiger low
%38 -"- high
%37 Kommandointerpreter Kommandozeiger low
%36 -"- high
%35 Kommandointerpreter Arbeitsregister
%30 Kommandointerpreter Arbeitsregister
%2F Kommandointerpreter Eingabespeicher
%20 Kommandointerpreter Eingabespeicher
%IF Zeiger Arithmetik-Stapelspeicher (BASIC/DEBUG)
%1E BASIC/DEBUG-Arbeitsregister
%16 BASIC/DEBUG-Arbeitsregister
%15 USR-Funktion 2. Argument low
%14 -"- high
%13 USR-Funktion 3. Argument low
%12 -"- high
%11 BASIC/DEBUG-Arbeitsregister
%10 -"-
%0F BASIC/DEBUG-Entnahmezeiger in Eingabespeicher low
%0E -"- high
%0D BASIC/DEBUG Aufnahmezeiger in Eingabespeicher low
%0C -"- high
%0B BASIC/DEBUG Obergrenze nutzbares RAM low
%0A -"- RAM high
%09 BASIC/DEBUG-Zeiger auf Programmanfang low
%08 -"- high
%07 BASIC/DEBUG-Zeiger Unterprogramm Stapelspeicher low
%06 -"- high
%05 BASIC/DEBUG-Zeiger auf Programmende low
%04 -"- high
%03 P3
%02 P2
%01 P1 = Daten-/Adreßbus (DAO bis DA7)
%00 P0 = Adreßbus (A8 bis A15)
```

# Aufbau

🔧 Fix Me!

Der orig. Quellcode ist nicht aufzufinden. Chris Howie („New Life for the Z8671 BASIC Interperter“) scheint diesen zu besitzen. Die auskommentierte Codezeile

```
251 !       LD    rACX,#2     ! USE MUD DEFAULT !(commented out)
```

deutet darauf hin.

Der Basic-Interpreter wird intern durch eine eigene IL (interpreter language) abgearbeitet, nur so passt alles in 2K ROM 🙂

Der IL-Code erinnert sehr an die Implementierung von Tom Pittman. In seiner Bibliographie schreibt Pittman auch von einer Implementierung für den Z8 („...1976 Implemented Tiny Basic (6800/6502/1802/Z8)...“). Ob er der Autor dieses BASICs ist, ist allerdings nicht bekannt.

s.a. http://www.ittybittycomputers.com/IttyBitty/TinyBasic/

From:
https://hc-ddr.hucki.net/wiki/ - **Homecomputer DDR**

Permanent link:
**https://hc-ddr.hucki.net/wiki/doku.php/elektronik/z8671?rev=1602423722**

Last update: **2020/10/11 13:42**