

# Protokoll: Microcontroller Z8 - IT 2

Gruppe 5: Byron Worms, Raik Dankworth, Martin Knoll

15. Mai 2012

## 1 Was ist der Z8?

Bei dem Z8 handelt es sich um eine Reihe von älteren Mikrokontrollern, die sich dadurch auszeichnen, dass sie einen recht simplen Aufbau besitzen aber dennoch einen recht großen Funktionsumfang aufweisen.

Was ist ein Mikrokontroller? Bei einem Mikrokontroller handelt es sich um ein Ein-Chip-Computersystem, was neben der logischen und arithmetischen Einheit zusätzliche Bausteine/-Baueinheiten aufweisen kann, wie zum Beispiel: Ports, Timer, Programmspeicher, Datenspeicher und Komparatoren.

## 2 Was macht den Z8 aus?

Die Registeranzahl variiert in der Z8-Reihe zwischen 128 und 4096 Registern mit der Bandbreite von 8-Bit, wobei jedes dieser Register als „General-Purpose-Register“ (kurz: GPR) angesehen wird. Es gibt folglich keine Akkumulatoren oder ähnlich speziellen Registern, dennoch sind einige für gewisse Funktionen vorreserviert (0-3: Port-Register, die letzten 16: Steuerregister).

Für die Kommunikation nach außen bietet die Z8-Reihe vier Ports, wobei nicht festgelegt ist welcher Port als Ausgangs- bzw. Eingangsport fungiert.

Für Aufgaben, die zeitabhängig laufen müssen oder eine gewisse Anzahl von Iterationen benötigen, gibt es zwei Funktionseinheiten, die entweder als Timer oder als Zähler konfiguriert werden können.

Wie in jedem Mikroprozessor auch gibt es die ALU und eine Interrupt-Einheit.

Einer der größten Unterschiede zu herkömmlichen CPUs bzw. Mikroprozessoren ist die Abspeicherung von Daten und dem Programm. In der Z8-Reihe teilen sich Daten und Programm gemeinsam einen Speicher, der 64kB groß sein kann. Der Speicher kann nur einmal beschrieben werden (ROM-Speicher).

## 3 Programmierung des Z8

Es gibt folgende Möglichkeiten den Z8 zu programmieren:

- Mit Maschinencode (binäre Programmierung)
- Assembler
- Mit Hilfe von Compilern auch durch Hochsprachen (z.B. C)

## 4 Ziel des Versuchs

Es ging hauptsächlich darum die Hardwarestruktur und den Befehlssatz der Z8 Reihen kennen zu lernen. Die schließt das Sammeln von Erfahrung bzgl. der Programmierung auf der Maschinebene ein, sowie das Testen unter Echtzeitbedingungen.

## 5 Aufbau des Versuchs

Da der Programmspeicher im Z8 nur einmal beschrieben werden kann (ROM), wurde in dem Versuch ein Emulator genutzt (Z86CCP00ZEM). Anderenfalls müsste man nach jedem Be-

schreiben des Z8s einen neuen Mikrokontroller-Chip nutzen, da der Alte nicht wiederbeschreibbar ist. Der Grund für die Entscheidung den Z8 zu emulieren und nicht zu simulieren ist Folgender: Bei der Simulation wird versucht ein Teil des originalen Problems wiederzugeben, wobei die Simulation nicht zu 100% mit dem eigentlichen Hardwareverhalten übereinstimmen muss (so tun als ob). Der Emulator kann durch einen Z8 ersetzt werden, ohne das Unterschiede bemerkbar sind (nachahmen).

## 6 Durchführung

### 3.2a Portspiegelung

Die Aufgabenstellung war eine simple Portspiegelung. Dabei sollte Port 0 als Ausgabeport deklariert werden und Port 2 als Eingabeport. Der Ausgabeport sollte die auf dem Port 2 liegende Eingabe übernehmen. Visualisiert wurde dies durch LEDs (Eine LED pro Bit in jedem Port).

Der Programmablauf ist hierbei recht simple. Anfangs initialisiert man die beiden Ports (Ein-/Ausgabe) und kopiert dann in einer Endlosschleife die Daten von dem Eingabeport in den Ausgabeport.

	TITLE	"3a"	hat keinen Effekt
	ORG	0Ch	Startadresse fuer Linker
	LD	P2M, #0FFh	Port 2 Managementregister auf Input
	LD	P01M, #00h	Port 0/1 Managementregister auf Output
LOOP	LD	P0, P2	P2 nach P0 laden
JP	LOOP		Endlosschleife

### 3.2b Laufflicht

In der Aufgabe ging um eine Realisierung eines Laufflichts. Dafür mussten beiden Ports (0, 2) als Ausgabeport deklariert werden. Die erste Hürde, die es zu überwinden galt war die Funktionsweise der beiden 8-Bit Ports. Dies haben wir über einen Rotationbefehl realisiert, wobei der Überhang in dem Carry-Flag gespeichert wurde. Dass das Laufflicht überhaupt wahrnehmbar für das menschliche Auge war, musste eine Verzögerungsschleife eingebaut werden, die eine Laufdauer von  $2^{16}$  Durchgängen hatte.

	TITLE	"3b"	hat keinen Effekt
	ORG	0Ch	Startadresse fuer Linker
	LD	P2M, #00h	Port 2 Managementregister auf Output
	LD	P01M, #00h	Port 0/1 Managementregister auf Output
	LD	P2, #80h	Hoechstes Bit in P2 auf high, Rest low
	LD	P0, #00h	P0 auf low
	RCF		Carry auf 0
LOOP	RRC	P2	P2 nach rechts rotieren, mit Uebertrag
	RLC	P0	P0 nach links rotieren, mit Uebertrag
	JP	C, LOOP	wenn Carry, noch ein Durchlauf ohne warten
WAIT	DECW	RR4	Dekrementiere R4 und R3 (auf Word-Breite)

JP	NZ, WAIT	wenn $RR4 \neq 0$ , Warteschleife
JP	LOOP	Endloschleife