

# Protokoll: Mikroprozessor Z80 - Grundlagenversuch - IT 7

Gruppe 5: Byron Worms, Raik Dankworth, Martin Knoll

12. Juni 2012

## 1 Was ist ein Z80?

Der Z80 ist ein Mikroprozessor, der sich durch folgende Eigenschaften von vorherigen Mikroprozessoren hervorhebt.

- Einphasentakt
- 158 Basisbefehle
- vorteilhafte Registerorganisation (zwei Blöcke mit 6 allgemeinen Registern)
- einfaches Umschalten zwischen den Registern - Blöcken zur Bearbeitung von Unterprogrammen ohne Registersicherungen
- 3 schnelle Interrupt - Behandlungsarten
- ein nicht maskierbarer Interrupt (NMI)

## 2 Versuchsziel

Das Praktikum diene dazu, die Arbeitsweise eines 8 - Bit Mikroprozessors kennen zu lernen. Dabei lag der Schwerpunkt auf der Verdeutlichung der Ein - und Ausgabesignale des Prozessors.

## 3 Versuchsvorbereitung

Dabei gilt es mehrere Fragen zu beantworten.

### 3.1 Was ist ein Maschinenzyklus?

Ein Maschinenzyklus besteht bei dem Z80 aus 3-6 Takten. Die Verarbeitung eines Maschinenbefehls erfolgt in einem bestimmten Schema, das sich aus mehreren Maschinenzyklen zusammensetzt. Der erste Schritt ist das Befehlsladen, danach erfolgt die Befehlsdekodierung und dessen Ausführung. Danach wird das Ergebnis nur noch gespeichert.

### 3.2 Wie ist die Registerstruktur des Z80 aufgebaut?

Der Z80 hat einen 16 Bit Stackpointer, zwei 16 Bit Indexregister, zwei Akkumulatoren und zwei Flagregister. Dazu sind noch 2 Blöcke mit 6 8 - Bit Registern. Die Besonderheit besteht darin, dass Operationen nur auf einen Block ausgeführt werden, so dass Programmwechsel leicht umgesetzt werden können, da man nur den Block wechseln braucht.

### 3.3 Wodurch werden Daten - und Befehlsbytes unterschieden?

Die werden den Zustand des Maschinenzuklus unterschieden. Beim Befehlsladen wird das anliegende Byte als Befehlsbyte aufgefasst und sonst als Datenbyte.

### 3.4 Was ist ein Interrupt und sind die möglichen Interruptmodi?

Ein Interrupt ist ein Signal, das den aktuellen Prozess unterbrechen kann. Dabei wird zwischen maskierbare und nicht maskierbare Interrupts unterschieden. Die möglichen Modi sind Mode 0, welcher durch ein Reset ausgelöst wird, wobei die CPU auf eine Eingabe eines Peripheriegeräts wartet. Im Mode 1 wird an die Adresse 38h gesprungen und die ISR (Interrupt - Service - Routine), welche dort beginnt, wird aufgerufen. In Mode 2 wird an eine beliebige Adresse gesprungen und deren ISR gestartet.

### 3.5 Was ist ein nicht maskierbarer Interrupt?

Maskierbare Interrupts können von der CPU unterbunden werden, während nicht maskierbare Interrupts immer ausgeführt werden.

### 3.6 Wodurch ist der HALT - Zustand des Z80 gekennzeichnet?

Der Z80 pausiert im HALT Zustand die Befehlsabarbeitung und wartet auf einen Interrupt.

### 3.7 Wie groß ist der Adressbereich für anschließbare Speicher und Ein-/Ausgabe- Port der Z80 CPU?

Der Z80 besitzt einen 16 - Bit Adressbus, womit  $2^{16}$  Adressen zu Verfügung stehen, was ungefähr 65 kByte entspricht. Bei der Adressierung von E/A-Ports wird der Adressbus gleichzeitig zur Beschreibung des Ports verwendet. Dadurch entfallen 8 Bit zur Beschreibung, so dass noch 8 Bit zur Adressierung nutzbar sind, wodurch 256 Ports angesteuert werden können.

### 3.8 Was bedeutet bidirektionaler Datentransfer?

Beim Z80 wird der Adress - und Datenbus zur Ein- und Ausgabe verwendet.

## 4 Versuchsdurchführung

Es folgt das Protokoll der durchgeführten Protokolle. Tabellen der einzelnen Zustände sind als Anlage angefügt.

### 4.1

In der ersten Aufgabe sollten wir die WAIT -, RESET - und BUSRQ - Signale setzen und dessen Wirkung überprüfen. WAIT - Signal zeigt der CPU, dass die angesprochenen Speicher oder E/A - Bausteine noch nicht zur Datenübertragung bereit sind. RESET - Signal sorgt dafür, dass die CPU alle Daten auf den Adress- und Datenbus löscht und in den M1 Zustand zurückgeht. Genauso wird der Befehlszähler auf 1 zurückgesetzt. Der BUSRQ - Signal ist ähnlich zu dem RESET - Signal, nur dass der Befehlszähler nicht auf 1 zurückgesetzt wird.

Tabelle

## 4.2

Bei dieser Aufgabe sollten wir in den Speicher etwas Schreiben und danach aus dem Speicher laden und beobachten was auf dem Adress - und Datenbus übertragen wird. Dafür haben wir als erstes in den Akkumulator eine Zahl geschrieben, diese danach in den Speicher geschrieben und zum Schluss aus der gleichen Speicheradresse die Zahl zurück in den Akkumulator geladen.

## 4.3

Diese Aufgabe ist ähnlich zur Aufgabe 2, wobei zwei spezielle Befehle zum Beschreiben und Laden eines Ports verwendet werden sollten.

## 4.4

Hierbei ging es darum den Interruptbehandlungszyklus kennen zulernen. Dafür musste man als erstes die Interruptbehandlung aktivieren und danach konnte man den Interrupt auslösen. Der Unterschied zu einer normalen M1 - Zyklus bestand daran, dass er nicht den Befehl vom Datenbus gelesen hat, sondern gleich zur ISR gesprungen ist.

## 4.5

Aufgabe 5 war ähnlich zur Aufgabe 4, wobei wir die Aktivierung der Interruptbehandlung nicht ausgeführt werden musste.

## 4.6

Dabei sollten wir die CPU in einen HALT - Zustand bringen und ihn mit einem Interrupt aus diesem Zustand rausholen. Die Bearbeitung dieser Aufgabe haben wir in Aufgabe 4 und 5 schon mit erfüllt.

## 4.7

In dieser Aufgabe sollte der Reset - Befehl 38h ausgeführt werden, welcher dafür sorgt, dass die CPU neugestartet und das Programm an der Stelle 38 weitergeführt wird.

## 4.8

Für die letzten Aufgabe sollten wir eine Operation selbst wählen und diese dann auf den Registern des Z80 anwenden. Wir haben uns dabei auf das Inkrementieren des A Register beschränkt. Dafür mussten wir als erstes eine Zahl in das Register schreiben und dieses dann mit einem weiteren Befehl inkrementieren.

[illegible]