6502 Programmieren - Teil 2#

von R. Wilde#

Hallo Leute ! Diesmal also der zweite Teil des Maschinensprache ? Lehrganges.

Zuerst einmal eine Klarstellung: Dieser Lehrgang an sich ist kein Maschinensprache -Lehrgang, sondern vielmehr ein Assembler - Lehrgang. Denn die Maschinensprache besteht nur aus Zahlen, die der Prozessor als Befehle versteht.

Da der Programmierer mit diesen Zahlen nicht viel anfangen kann, werden diese durch einen Assembler mit jeweils drei Buchstaben dargestellt, die wiederum die Abkürzungen für die englischen Beschreibungen sind.

Ich beginne mit der Zusammenfassung aller Maschinenbefehle. Am Anfang eines Absatzes steht die Prozessor-Anweisung, dahinter die englische Bezeichnung. In dem Absatz darunter steht die Beschreibung der Funktion. In Klammern darunter sind die Flags des Prozessor-Status Registers und die Register angegeben, die durch den jeweiligen Befehl verändert werden. Dabei stehen die Buchstaben für folgende Flag's bzw. Register :

N = Negativ-Flag
V = Overflow-(Überlauf) Flag
B = Break-Flag
D = Dezimal-Flag
I = Interrupt-Flag
Z = Zero- (Null) Flag
C = Carry-Flag
P = Prozessor-Status Register
PC = Programmzähler
S = Status-Register

ADC - Add Memory to ACCU with Carry#

Zu dem Wert im ACCU soll ein fester Wert oder der Wert in einer Speicherstelle addiert werden. War das Carry-Flag gesetzt, wird noch eine 1 hinzugezählt. Ist das Gesamtergebnis größer als 255 wird das Carry-Flag gesetzt, sonst ist es = 0. Das Ergebnis steht im ACCU.
(NVZC)

AND - AND Memory with ACCU#

Der Wert im ACCU wird mit einem festen Wert oder mit dem Wert in einer Speicherstelle logisch UND verknüpft. Das Ergebnis steht im ACCU.
(NZ)

ASL ? Arithmentic Shift Left 1 Bit ACCU or Memory#

ACCU oder Speicherstelle bitweise eine Stelle nach links schieben. Bit 7 wird dabei ins Carry-Flag übernommen. Bit 0 wird = 0 gesetzt.
(NZC)

BCC - Branch on Carry Clear#

Verzweige, wenn das Carry-Flag nicht gesetzt ist.

BCS - Branch on Carry Set#

Verzweige, wenn das Carry-Flag gesetzt ist.

BEQ - Branch on Equal#

Verzweige, wenn das Zero-Flag gesetzt ist.

BIT - Test Bits in Memory with ACCU#

Bit 6 und 7 eines angegebenen Wertes oder des Wertes in einer Speicherstelle werden in das Status-Register der CPU übernommen. Gleichzeitig wird eine UND-Verknüpfung vorgenommen. Ist das Ergebnis = 0 wird das Zero-Flag der CPU gesetzt, ansonsten gelöscht. Es wird nur der Wert im Status-Register der CPU verändert.
(NVZ)

BMI - Branch on Minus#

Verzweige, wenn das Negativ-Flag im Status-Register der CPU gesetzt ist.

BNE - Branch if not Equal#

Verzweige, wenn das Zero-Flag nicht gesetzt ist.

BPL - Branch on Plus#

Verzweige, wenn das Negativ-Flag nicht gesetzt ist.

BRK - Break, Force Interrupt#

Stößt der Prozessor auf eine Break-Anweisung wird das Interrupt-Flag der CPU gesetzt und der Prozessor führt eine Interrupt-Routine aus, deren Anfangsadresse in den Speicherstellen $FFFE/ $FFFF steht.
(BI)

BVC - Branch on Overflow Clear#

Verzweige, wenn das Overflow (Überlauf) Flag nicht gesetzt ist.

BVS - Branch on Overflow Set#

Verzweige, wenn das Overflow (Überlauf) Flag gesetzt ist.

CLC - Clear Carry-Flag#

Setze das Carry-Flag = 0.
(C)

CLD - Clear Dezimal-Flag#

Setze das Dezimal-Flag = 0.
(D)

CLI - Clear Interrupt-Flag#

Setze das Interrupt-Flag = 0.
(I)

CLV - Clear Overflow-Flag#

Setze das Overflow (Überlauf) Flag = 0.
(V)

CMP - Compare ACCU with Memory#

Vergleiche den Wert im ACCU mit einem festen Wert oder dem Wert in einer Speicherstelle. Das Status-Register der CPU wird entsprechend gesetzt.
(NZC)

CPX - Compare X-Register with Memory#

Vergleiche den Wert im X-Register mit einem angegebenen Wert oder dem Wert in einer Speicherstelle. Das Status-Register der CPU wird entsprechend gesetzt.
(NZC)

CPY - Compare Y-Register with Memory#

Vergleiche den Wert im Y-Register mit einem festen Wert oder dem Wert in einer Speicherstelle. Das Status-Register der CPU wird entsprechend gesetzt.
(NZC)

DEC - Decrement Memory by One#

Zähle den Wert einer Speicher- stelle um 1 herab.
(NZ)

DEX - Decrement X-Register by One#

Zähle den Wert im X-Register um 1 herab.
(NZ)

DEY - Decrement Y-Register by One#

Zähle den Wert im Y-Register um 1 herab.
(NZ)

EOR - Exclusive OR ACCU with Memory#

Führe ein exklusives OR mit dem Wert im ACCU und einem angegebenen Wert oder dem Wert in einer Speicherstelle aus. Das Ergebnis steht im ACCU.
(NZ)

INC - Increment Memory by One#

Zähle den Wert einer Speicherstelle um 1 herauf.
(NZ)

INX - Increment X-Register by One#

Zähle den Wert im X-Register um 1 herauf.
(NZ)

INY - Increment Y-Register by One#

Zähle den Wert im Y-Register um 1 herauf.
(NZ)

JMP - Jump to new Location#

Führe das Programm bei der angegebenen Adresse fort. Das Status-Register der CPU wird nicht verändert.
(PC)

JSR - Jump to Subroutine#

Die Adresse dieser Anweisung (Wert im Programmzähler -PC-) wird auf den Stack geschoben und ein Unterprogramm, dessen Adresse angegeben ist, ausgeführt. Das Status-Register der CPU wird nicht verändert.
(PC S)

LDA - Load ACCU with Memory#

Lade den Accu mit einem angegebenen Wert oder dem Wert in einer Speicherstelle.
(NZ)

LDX - Load X-Register with Memory#

Lade das X-Register mit einem angegebenen Wert oder dem Wert in einer Speicherstelle.
(NZ)

LDY - Load Y-Register with Memory#

Lade das Y-Register mit einem angegebenen Wert oder dem Wert in einer Speicherstelle.
(NZ)

LSR ? Logical Shift Right 1 Bit ACCU or Memory#

ACCU oder Speicherstelle bitweise eine Stelle nach rechts schieben. Bit 0 wird dabei ins Carry-Flag übernommen. Bit 7 wird = 0 gesetzt.
(NZC)

NOP - No Operation#

Anweisung ohne Funktion. Dieser Befehl dient z.B. dem Freihalten von Speicherstellen oder einer kurzen Verzögerung (2 Zyklen) der nachfolgenden Anweisungen.

ORA - OR ACCU with Memory#

Logische ODER-Verknüpfung des ACCU mit einem angegebenen Wert oder dem Wert in einer Speicherstelle. Das Ergebnis steht im ACCU.
(NZ)

PHA - Push ACCU on Stack#

Der Wert des ACCU's wird in der Speicherstelle des Stack abgelegt, auf die der Stapelzeiger zeigt. Anschließend wird der Stapelzeiger um 1 abgezählt.
(S)

PHP - Push Processor-Status on Stack#

Der Wert im Status-Register der CPU wird in der Speicherstelle des Stack abgelegt, auf die der Stapelzeiger zeigt. Danach wird der Stapelzeiger um 1 abgezählt.
(S)

PLA - Pull ACCU from Stack#

Der Stapelzeiger wird um 1 erhöht und der Wert der Speicherstelle des Stack's, auf die der Stapelzeiger zeigt, wird in den ACCU übernommen.
(NZ S)

PLP - Pull Processor-Status from Stack#

Der Stapelzeiger wird um 1 erhöht und der Wert der Speicherstelle des Stack's, auf die der Stapelzeiger zeigt, wird in das Prozessor-Status Register übernommen.
(NVBDIZC S)

ROL - Rotate Left 1 Bit ACCU or Memory#

ACCU oder Speicherstelle bitweise eine Stelle nach links schieben. Das Carry-Flag wird dabei in Bit 0 und Bit 7 in das Carry-Flag geschoben.
(NZC)

ROR - Rotate Right 1 Bit ACCU or Memory#

ACCU oder Speicherstelle bitweise eine Stelle nach rechts schieben. Das Carry-Flag wird dabei in Bit 7 und Bit 0 ins Carry-Flag geschoben.
(NZC)

RTI - Return from Interrupt#

Kehre aus einer Interrupt-Routine zu der Stelle zurück, an der ein Programm durch den Interrupt unterbrochen wurde. Zu diesem Zweck wird zuerst der Wert des Prozessor-Status Registers, dann die Werte für den Programmzähler vom Stack zurückgeholt.
(NVBDIZC PC S)

RTS - Return from Subroutine#

Kehre aus einem Unterprogramm zu der Stelle zurück, an der das Unterprogramm aufgerufen wurde. Dazu werden die Werte für den Programmzähler vom Stack geholt und um 1 erhöht.
(PC S)

SBC - Subtract Memory from ACCU with Carry#

Zähle einen festen Wert oder den Wert einer Speicherstelle vom Inhalt des ACCU's unter Berücksichtigung des Carry-Flag ab. Das Ergebnis befindet sich im ACCU.
(NVZC)

SEC - Set Carry-Flag#

Setze das Carry-Flag innerhalb des Prozessor-Status Registers auf 1.
(C)

SED - Set Decimal-Flag#

Setze das Dezimal-Flag innerhalb des Prozessor-Status Registers auf 1.
(D)

SEI - Set Interrupt-Flag#

Setze das Interrupt-Flag im Prozessor-Status Register auf 1. Eine mögliche Interrupt Anforderung (IRQ) wird gesperrt.
(I)

STA - Store ACCU in Memory#

Lege den Inhalt des ACCU's in der angegebenen Speicherstelle ab.

STX - Store X-Register in Memory#

Lege den Inhalt des X-Registers in der angegebenen Speicherstelle ab.

STY - Store Y-Register in Memory#

Lege den Inhalt des Y-Registers in der angegebenen Speicherstelle ab.

TAX ? Transfer ACCU to X-Register#

Kopiere den Inhalt des ACCU´s in das X-Register. Der Wert vom ACCU bleibt unverändert.
(NZ)

TAY ? Transfer ACCU to X-Register#

Kopiere den Inhalt des ACCU´s in das Y-Register. Der Wert vom ACCU bleibt unverändert.
(NZ)

TSX ? Transfer S-Register in X-Register#

Kopiere den Inhalt des Stapelzeigers in das X-Register. Der Wert im Stapelzeiger bleibt unverändert.
(NZ)

TXA - Transfer X-Register to ACCU#

Kopiere den Inhalt des X-Registers in den ACCU. Der Wert des X-Registers bleibt unverändert.
(NZ)

TXS - Transfer X-Register to Stack-Pointer#

Kopiere den Inhalt des X-Registers in den Stapelzeiger. Der Wert des X-Registers bleibt unverändert.
(S)

TYA - Transfer Y-Register to ACCU#

Kopiere den Inhalt des Y-Registers in den ACCU. Der Wert des Y-Registers bleibt unverändert. (NZ)

Ein ziemlich trockenes Thema, das muss ich zugeben. Aber das ist NUR die Zusammenfassung der Maschinenbefehle (Assemblerbefehle) an sich. Dazu kommen noch die Adressierungsarten, die ich Euch aber erst (Danke, danke, danke !?) in der nächsten Ausgabe des Magazins zumuten möchte.

Unter dem Titel "Opcode-Printer" findet Ihr noch ein Basic-Programm, das auf EPSON-Kompatiblen Druckern eine übersichtliche Liste der Maschinenbefehle und den Adressierungsarten ausdruckt, die wohl jeder ernsthafte Maschinensprache-Programmierer gut gebrauchen kann.

Bis zum nächsten Mal, Euer R. Wilde.
CSM / 7.1988

Der Artikel entstammt der Kursreihe ?6502 Programmieren? des Compy Shop Diskettenmagazins. Die Kursreihe besteht aus 14 Kursen, die im Laufe des Jahres 2011 in unregelmäßigen Abständen einzeln veröffentlicht werden, bzw. anschließend als Zusammenzug als ABBUC-Buch ?6502 Programmieren? erscheinen.
Koordination: Volkert Barr (volkert@nivoba.de)
Version 1.1 / 2011-01-15