

Johannes Baumann

6502 simuliert 14500

Der Chip 14500 von Motorola ist ein 1-Bit-Mikroprozessor, der sich wegen seines geringen Preises und seiner einfachen Befehlsstruktur besonders für kleine Steuerungsaufgaben eignet. Ein Problem ist normalerweise aber die Programmentwicklung für ein 14500-System. Dieses Problem wurde hier gelöst, indem 14500-Software auf einem 6502-Rechner simuliert und getestet werden kann – in diesem Fall mit einem KIM-1.

Die Struktur des 14500-Befehlssatzes entspricht derjenigen einer einfachen programmierbaren Steuerung (Tabelle 1). Auch der innere Aufbau ist sehr einfach; dementsprechend müssen viele Aufgaben von externer Logik übernommen werden. Bild 1 zeigt die Pinbelegung des 1-Bit-Prozessors und Bild 2 eine typische Beschaltung.

Bei den Befehlen werden jeweils vier Bit für den Operationscode und vier Bit für die Adressierung der Ein- und Ausgänge verwendet:

BBBBSEEE mit BBBB = Operationscode, S = Selektierung, ob Ein- oder Ausgang (1/0), EEE = Nummer des Ein- oder Ausgangs. Zwei Flags stehen zur Verfügung: Flag F (zurück zur Zeile 0) und Flag RTN (im Simulator: zurück zum Monitorprogramm). Damit man das Ergebnisregister RR lesen kann, ist der Ausgang RR der ICU mit dem Eingang 0 verbunden. Das wird benötigt, wenn man ein Bit auf 0 oder 1 setzen will:
 1 = RR + \bar{RR} ORC Eingang 0
 0 = RR · \bar{RR} ANDC Eingang 0

Ein Beispiel

Auf einer Bahnlinie stehen zwei Silos. In einem ist Sand, im anderen ist Kies. Die Transportwagen sollen automatisch gefüllt werden. Die Materialvorwahl erfolgt mit einem Handschalter. Eingänge = E und Ausgänge = A:
 Ergebnisregister E0
 Materialvorwahl E1 (Kies = 1, Sand = 0)
 Wagen vor Kiessilo E2
 Wagen vor Sandsilo E3
 Wagen voll Klappe des Kiessilos A1 (Klappe offen = 1)
 Klappe des Sandsilos A2

Das Programm in Tabelle 2 will nicht die beste Lösung sein, sondern die Möglichkeiten des 14500 zeigen.

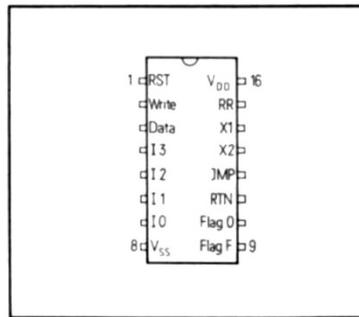


Bild 1. Pinbelegung des CMOS-Bausteins 14500 von Motorola. Er stellt einen einfachen Mikroprozessor für Steuerungsaufgaben mit 1 Bit Datenbreite dar

Tabelle 1: Befehlssatz des 14500

0	NOPO	Setze Flag 0. Keine Änderung in den Registern
1	LD	Lade Ergebnisregister
2	LDC	Lade Ergebnisregister mit Komplement
3	AND	Logisches UND
4	ANDC	Logisches UND mit dem Komplement
5	OR	Logisches ODER
6	ORC	Logisches ODER mit dem Komplement
7	XNOR	Äquivalenz (EXOR)
8	STO	Speichere Ergebnisregister
9	STOC	Speichere das Komplement vom Ergebnisregister
A	IEN	Lade Data in IEN-Register
B	OEN	Lade Data in OEN-Register
C	JMP	Setze Flag JMP. Keine Änderung in den Registern.
D	RTN	Setze Flag RTN. Keine Änderung in den Registern.
E	SKZ	Überspringe nächsten Befehl, wenn RR = 0
F	NOFF	Setze Flag F. Keine Änderung in den Registern.

Die Befehle 0, C, D, F machen das jeweilige Flag für eine Taktperiode aktiv.

Das Simulatorprogramm

Um 14500-Programme entwickeln und testen zu können, wurde ein Simulatorprogramm für 6502-Systeme geschrieben; ursprünglich zwar für den KIM-1, es läßt sich aber sehr leicht an andere Systeme adaptieren (Bild 3).

Die Register des 14500 werden in der Zero-Page des 6502-Systems simuliert:

00D0 RREG Ergebnisregister (Akku)
 00D1 OENR OEN-Register
 00D2 IENR IEN-Register

Der Programmzähler findet sich beim KIM-1 in den Zellen 00D3; 00D4 enthält die Page, auf der das zu interpretierende Programm steht (z. B. 03, wenn es ab 0300 steht).

Ferner benützt der Interpreter noch folgende Speicherzellen:

00D5 OPC Operationscode (aktueller OPC)
 00D6 ASUB Adresse des aktuellen Unterprogrammes
 00D7 ASUB+1 In dieser Adresse steht die Seite, in der die Unterprogramme stehen
 00D8 ZW11 Zwischenresultate 1
 00D9 ZW12 Zwischenresultate 2

Die Adressen 00D4 und 00D7 werden bei der Initialisierung beschrieben und nachher nicht mehr verändert.
Ein- und Ausgänge

1700 PAD Ausgänge des simulierten Systems
1701 PADD Datenrichtungsregister, definiert die Ausgänge
1702 PBD Eingänge des Systems
1703 PBDD Definiert die Eingänge, außer Bit 0, das als Ausgang definiert ist.

Monitoradressen:

1C4F START Eingang in den Monitor
Anfangsadressen:

0200 Startadresse des Interpreters
0300 Startadresse des zu interpretierenden Programmes in der Maschinensprache des 14500.

Anpassung des Programms an andere 6502-Systeme

Folgendes muß angepaßt werden, wenn man nicht den KIM-1 verwendet:

1. Die Adressen der Datenrichtungsregister (PADD in 0203,0204, PBDD in 0208,0209)

2. Die entsprechende Datenregisteradresse (PAD in 02E3,02E4 02E8,02E9 und in 013C,013D)

Wenn PBD nicht gleich PAD+2 ist, muß im Unterprogramm „LESE“ eine Änderung vorgenommen werden. Indizierte Adressierung von PBD in den Adressen 0127 bis 012C und 013B.

3. Das Unterprogramm „RTN“ muß neu geschrieben werden oder durch BRK ersetzt werden (den entsprechenden Interpunktvektor auf den Monitoreingang setzen).

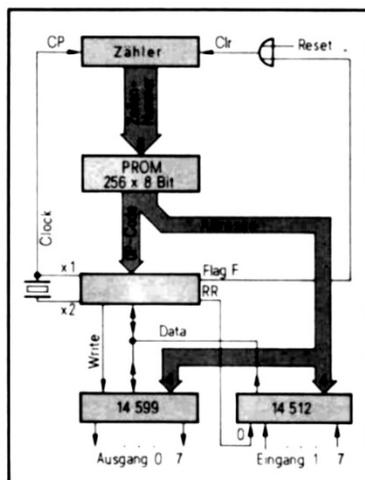


Bild 2. Struktur eines typischen 14500-Anwendersystems, wie es vom 6502-System simuliert werden kann

Tabelle 2: 14500-Beispielprogramm

Zeile	Befehl	Hex-Code	Kommentar
00	ORC E0	68	Das Ergebnisregister wird definiert
01	IEN E0	A8	Das IEN-Register wird auf 1 gesetzt
02	OEN E0	B8	Das OEN-Register wird auf 1 gesetzt
			Diese zwei Register werden gebraucht, um die Ein- oder Ausgänge zu blockieren
03	LD E1	19	Materialvorwahl?
04	OEN E0	B8	Bei Vorwahl „Sand“ blockiere die Ausgänge (entspricht etwa einem Sprung nach 09)
05	STOC A2	92	Schließe Klappe des Sandsilos
06	LD E2	1A	Wagen vor Kiessilo
07	ANDC E4	4C	und nicht voll
08	STO A1	81	dann öffne Siloklappe
09	LDC E1	29	Materialvorwahl?
0A	OEN E0	B8	Bei Vorwahl Kies blockiere die Ausgänge
0B	STOC A1	91	Schließe Kiessilo (Sicherheit)
0C	LD E3	1B	Wagen vor Sandsilo
0D	ANDC E4	4C	und nicht voll
0E	STO A2	82	dann öffne Siloklappe
0F	LD E5	1D	Der Eingang 5 wird hier gebraucht, um aus dem Interpreter wieder in den Monitor zu kommen (Retour = 1)
10	SKZ -	EF	Wenn 0, überspringe nächsten Befehl
11	RTN -	DF	Zurück zum Monitor des Simulators
12	NOFF -	FF	Beginne wieder auf Zeile 00

Bild 3. Listing des 14500-Interpreters für ein KIM-1-System

Hauptprogramm			
Ø200	A9 FF	INIT	LDA =FF
Ø202	8D Ø1 17		STA PADD
Ø205	A9 Ø1		LDA =Ø1
Ø207	8D Ø3 17		STA PBDD
Ø20A	85 D1		STA OENR
Ø20c	85 D2		STA IENR
Ø20E	A9 ØØ		LDA =ØØ
Ø210	85 DØ		STA RREG
Ø212	85 D3		STA PZAE
Ø214	A9 Ø2		LDA =Ø2
Ø216	85 D7		STA ASUB+1
Ø218	A9 Ø3		LDA =Ø3
Ø21A	85 D4		STA PZAE+1
Ø21C	EA		NOP
Ø21D	EA		NOP
Ø21E	EA		NOP
Ø21F	A2 ØØ	SIMU	LDX =ØØ
Ø221	A1 D3		LDA (PZAE,X)
Ø223	85 D5		STA OPC
Ø225	4A		LSR A
Ø226	4A		LSR A
Ø227	4A		LSR A
Ø228	4A		LSR A
Ø229	A8		TAY
Ø22A	B9 4Ø Ø2		LDA TAB,Y
Ø22D	85 D6		STA ASUB
Ø22F	E6 D3		INC PZAE
Ø231	2Ø 37 Ø2		JSR JIND
Ø234	4C 1F Ø2		JMP SIMU
Ø237	6C D6 ØØ		JIND JMP (ASUB)
Tabelle			
Ø240	5Ø 53 59	TAB	
Ø243	61 69 73		
Ø246	7B 85 8F		
Ø249	95 9D A3		
Ø24C	A9 AC B7		
Ø24F	BE		
Unterprogramme (Tiefe 1)			
Ø250	EA	NOPO	NOP
Ø251	EA		NOP
Ø252	6Ø		RTS
Ø253	2Ø ØØ Ø1 LD		JSR LESE
Ø256	85 DØ		STA RREG
Ø258	6Ø		RTS
Ø259	2Ø ØØ Ø1 LDC		JSR LESE
Ø25C	49 Ø1		EOR =Ø1

025E	85 D0		STA RREG	<u>Unterprogramme (Tiefe 2)</u>	
0260	60		RTS	02C3	85 D9 AUSG STA ZWI2
0261	20 00 01	ANDY	JSR LESE	02C5	A5 D1 LDA OENR
0264	25 D0		AND RREG	02C7	F0 21 BEQ AUSG3
0266	85 D0		STA RREG	02C9	A9 01 LDA =01
0268	60		RTS	02CB	85 D8 STA ZWI2
0269	20 00 01	ANDC	JSR LESE	02CD	A5 D5 LDA OPC
026C	49 01		EOR =01	02CF	29 0F AND =0F
026E	25 D0		AND RREG	02D1	EA NOP
0270	85 D0		STA RREG	02D2	EA NOP
0272	60		RTS	02D3	AA TAX
0273	20 00 01	OR	JSR LESE	02D4	F0 08 AUSG1 BEQ AUSG2
0276	05 D0		ORA RREG	02D6	06 D8 ASL ZWI1
0278	85 D0		STA RREG	02D8	06 D9 ASL ZWI2
027A	60		RTS	02DA	CA DEX
027B	20 00 01	ORC	JSR LESE	02DB	4C D4 02 JMP AUSG1
027E	49 01		EOR =01	02DE	A5 D8 AUSG2 LDA ZWI1
0280	05 D0		ORA RREG	02E0	49 FF EOR =FF
0282	85 D0		STA RREG	02E2	2D 00 17 AND PAD
0284	60		RTS	02E5	05 D9 ORA ZWI2
0285	20 00 01	XNOR	JSR LESE	02E7	8D 00 17 STA PAD
0288	49 01		EOR =01	02EA	60 AUSG3 RTS3
028A	45 D0		EOR RREG	0100	A5 D2 LESE LDA IENR
028C	85 D0		STA RREG	0102	F0 2F BEQ LESE4
028E	60		RTS	0104	A9 01 LDA =01
028F	A5 D0	STO	LDA RREG	0106	85 D8 STA ZWI1
0291	20 C3 02		JSR AUSG	0108	A5 D5 WENN0 LDA OPC
0294	60		RTS	010A	29 0F AND =0F
0295	A5 D0	STOC	LDA RREG	010C	C9 08 CMP =08
0297	49 01		EOR =01	010E	F0 26 BEQ ZERO
0299	20 C3 02		JSR AUSG	0110	EA EA NOP
029C	60		RTS	0112	EA EA NOP
029D	20 08 01	IEN	JSR WENN0	0115	A5 D5 WENN1 LDA OPC
02A0	85 D2		STA IENR	0117	29 08 AND =08
02A2	60		RTS	0119	4A LSR
02A3	20 00 01	OEN	JSR LESE	011A	4A LSR
02A6	85 D1		STA OENR	011B	AA TAX
02A8	60		RTS	011C	A5 D5 LDA OPC
02A9	EA	JMR	NOP	011E	29 07 AND =07
02AA	EA		NOP	0120	A8 TAY
02AB	60		RTN	0121	F0 06 LESE1 BEQ LESE2
02AC	A5 D3	RTN	LDA PZAE	0123	06 D8 ASL ZWI1
02AE	85 FA		STA 00FA	0125	88 DEY
02B0	A9 03		LDA =03	0126	4C 21 01 JMP LESE1
02B2	85 FB		STA 00FB	0129	A5 D8 LESE2 LDA ZWI1
02B4	4C 4F 1C		JMP START	012B	3D 00 17 AND PAD, X
02B7	A5 D0	SKZ	LDA RREG	012E	F0 02 BEQ LESE3
02B9	D0 02		BNE SKZ1	0130	A9 01 LDA =01
02BB	E6 D3		INC PZAE	0132	60 LESE3 RTS
02BD	60	SKZ1	RTS	0133	A9 00 LESE4 LDA =00
02BE	A9 00	NOPF	LDA =00	0135	60 RTS
02C0	85 D3		STA PZAE	0136	A5 D0 ZERO LDA RREG
02C2	60		RTS	0138	60 RTS