Technische Universität Dresden

Fakultät Informatik Institut für Technische Informatik, Professur für Adaptive Dynamische Systeme

Grundlagen der Technischen Informatik

2. Laborübung

Prof. Dr.-Ing Diana Göhringer

6. November 2026

section

1 Ausführung von Programmen und Befehlen

1.1.) Analyse eines Programmcodes

Geben Sie für das nachstehende Programmstück die Speicher-Adressen und die Mnemonik für den gegebenen Maschinencode für die einzelnen Befehle an (Beginn bei Addresse 0x02).

Adresse	HEX	Mnemonik	Kommentar
0x02, 0x03	0x31 00	LXI SP, 0x00	Initialisiere Stackpointer auf 0
		MVI A, 0x76	
		@P: PUSH A	
		JMP @P	

1.2.) SINGLE-STEP-Betrieb

Geben Sie das Programm in den Von-Neumann-Rechner ein und testen Sie das Programm auf die gewünschte Funkionalität. Befolgen Sie dazu die nachfolgenden Punkte.

1. Überprüfen Sie die Einstellungen des Boards. Befinden sich alle Kippschalter in der richtigen Positoin (z.B. MREQ muss auf 1 gesetzt werden)?
2. Geben Sie an der Adresse 0x00 einen Sprungbefehl ein, der zum Start des Programms springt.
3. Geben Sie das Programm beginnend an der Startadresse 0x02 in den Von-Neumann-Rechner ein.
4. Setzen Sie den Programmzähler zurück.
5. Arbeiten Sie das Programm zunächst im Einzelschritt-Betrieb (SINGLE-STEP, INT.-CYCLE) ab. Beobachten Sie dabei die Belegung der Register und Steuerleitungen. Tragen Sie die Belegungsänderung in die untenstehende Tabelle ein.
6. Wenn Sie mit der Funktionsweise des Rechners ausreichend vertraut geworden sind, schalten Sie in den kontinuierlichen Betrieb um.
7. Lassen Sie das Programm fertig ablaufen und von einer betreuenden Person überprüfen, bevor Sie fortfahren.

Befehl / Speicherplatz	Takt	PC	Addr. Buffer	Speicher					Data Buffer	L	SP	Akku	S Z C
				SEL	RD	WR	Adr.R.	D.-R.					
	0	0x00	**	0	0	0	0x00	0x00	**	**	**	**	* * *
Befehlsanfang	1		0x00										
	2	0x01											
JMP 0x02	3			1	1		0x00	0xC3					
(0xC3 0x02)	4			0	0				0xC3				
Adresse: 0x00, 0x01	5		0x01										
	6	0x02											
	7			1	1		0x01	0x02					
	8			0	0				0x02				
Befehlsende	9	0x02											
	10												
	11												
	12												
	13												
	14												
	15												
	16												
	17												
	18												
	19												
	20												
	21												
	22												
	23												
	24												
	25												
	26												
	27												
	28												
	29												
	30												

Befehl / Speicherplatz	Takt	PC	Addr. Buffer	Speicher					Data Buffer	L	SP	Akku	S Z C
				SEL	RD	WR	Adr.R.	D.-R.					
	31												
	32												
	33												
	34												
	35												
	36												
	37												
	38												
	39												
	40												
	41												
	42												
	43												
	44												
	45												
	46												
	47												
	48												
	49												
	50												
	51												
	52												
	53												
	54												

1.3.) Untersuchung des Programmablaufs

Vervollständigen Sie in folgender Tabelle den Ablauf des Programmes. Als Referenz sind die internen Taktzyklen gegeben, zu denen die Ausführung eines Befehls vollendet wird. Ergänzen Sie zu jedem Befehl die resultierende Belegung des Programmzählers, des Akkumulators und des Stapelregisters. Nicht initialisierte Register sind mit '*' markiert.

Interner Takt	Befehl	PC	Akkumulator	Stackpointer
0	-	0x00	*	*
9	JMP 0x02	0x02	*	*
18	LXI SP, 0x00	0x04		
27				
36				
45				
54				
63				
72				
81				
4464				
4473				
4482				
4491				
4500				

1.4.) Bus-Cycle-Betrieb

Lassen Sie das Programm erneut im BUS-CYCLE-Betrieb durchlaufen und geben Sie diejenigen Takte an, nach denen ein BUS-CYCLE jeweils beginnt oder endet.

Hinweis: Der BUS-CYCLE ist zum Debuggen von Programmen geeignet.

Takte: 3, 7,

1.5.) Schleifen

Die Schleife des Programms verfügt über keine Abbruchbedingung, trotzdem hält das Programm irgendwann an. Wie heißt das Phänomen, welches uns hier hilft Befehle zu sparen, in der Praxis aber vorwiegend ungeplant auftritt und dabei eine häufige Sicherheitslücke darstellt?
Antwort:

1.6.) Funktionalität des Programms

Welche Funktion wird durch das Programm realisiert? Warum ist die im Rahmen der Aufgaben sinnvoll