

Übungszettel 9

35. Zum Vergleich der Effizienz von Speicherzugriffen bei vier verschiedenen Architekturen betrachten wir:

- Akku-Maschine (Ein-Adreßmaschine).
- Speicher-Maschine (3-Adreßmaschine): Alle drei Operanden befinden sich im Speicher.
- Stack-Maschine (Null-Adreßmaschine): Alle Operationen laufen über den Stack ab. Nur PUSH und POP greifen auf den Speicher zu, alle anderen Befehle holen ihre Operanden vom Stack und legen die Ergebnisse wieder dort ab.
- Load-Store-Maschine (Register-Maschine): Alle Operationen finden in Registern statt, Register-Register-Befehle haben 3 Operanden. Es gibt 16 allgemeine Register, die Registerkennung ist 4 Bit lang.

Zum Vergleich der Speicherzugriffe treffen wir für alle Architekturen folgende Annahmen:

- Der Operationscode ist immer 1 Byte lang.
- Alle Speicheradressen haben eine Länge von 2 Bytes.
- Alle Operanden sind 4 Bytes lang.
- Die Länge aller Instruktionen ist ein Vielfaches von 1 Byte.
- Die Variablen A, B, und C stehen vom Beginn an im Speicher.
- Mit einem Speicherzugriff können maximal 4 Bytes übertragen werden.

Schreiben Sie für jede der 4 obigen Architekturen Assembler-Code für die Anweisung $A = B + C$. Verwenden Sie dazu die (syntaktisch der entsprechenden Architektur angepaßten) Assemblerbefehle `load`, `store`, `add`, `push`, `pop`, `move` und ermitteln Sie die Anzahl der Befehlsbytes und die Anzahl der Datenbytes für jede Code-Sequenz. Welche Architektur hat den effizientesten Code und welche die wenigsten Speicherzugriffe (= Befehle und Daten)?

36. Gegeben sei folgendes Programm für den Universalrechenautomaten (URA) sowie die Speicherbelegung am Programmbeginn:

Adresse	Label	Operation	Operanden	Kommentar
0000		mov	r4, #0	; r4 = 0
0001		mov	r1, 1000	; r1 = Inhalt von Adresse 1000
0002		mov	r2, 1004	; r2 = Inhalt von Adresse 1004
0003	loop:	cmp	r1, r2	; vergleiche r1 mit r2
0004		blt	mark1	; Sprung wenn r1 < r2
0005		sub	r3, r1, r2	; subtrahiere r2 von r1
0006		mov	r1, r3	; Ergebnis in r1 zurück
0007		inc	r4	; erhöhe r4
0008		jmp	loop	; verzweige nach loop
0009	mark1:	mov	1008, r4	; Inhalt von r4 nach Adresse 1008
000a		mov	100c, r1	; Inhalt von r1 nach Adresse 1012
000b		ret		; Programmende
...				
1000	A:	44		
1004	B:	7		
1008	C:	0		
1012	D:	0		
...				

Zeichnen Sie ein Flußdiagramm für das Programm. Was berechnet das Programm?

37. In einer DLX-Maschine sei das Multiplizierwerk fehlerhaft. Beheben Sie den Fehler, indem Sie ein DLX-Unterprogramm für die Multiplikation nach dem Verfahren der Ägyptischen Multiplikation schreiben.
- Geben Sie das Unterprogramm mit folgenden Übergabeparametern an: R1, R2 enthalten die beiden zu multiplizierenden Zahlen, das Ergebnis wird in R3 übergeben. Nehmen Sie an, dass Multiplikand und Multiplikator positive Integerwerte sind, die keinen Überlauf produzieren. Das Unterprogramm soll nach dem *Callee-Save*-Prinzip gestaltet sein.
 - Geben Sie ein Codefragment an, das es auch erlaubt negative Zahlen unter den Voraussetzungen in a. zu multiplizieren und kennzeichnen Sie die Stellen in a., wo dieses eingefügt werden soll.
 - Testen Sie das Unterprogramm in WinDLX.
38. Schreiben Sie ein DLX-Unterprogramm, das die Korrektheit eines Zeichens im 7-Bit ASCII Code mit *even parity* überprüft.
- Geben Sie das Unterprogramm mit folgenden Übergabeparametern an: R1 enthält das zu überprüfende Zeichen. Das Ergebnis wird in R3 übergeben. Das Unterprogramm soll nach dem *Callee-Save*-Prinzip gestaltet sein.
 - Geben Sie ein Codefragment an, das es dem Unterprogramm auch erlaubt Zeichen im 7-Bit ASCII Code mit *odd parity* zu prüfen. Dazu wird in R2 0 für ein Zeichen mit even parity und 1 für odd parity übergeben. Kennzeichnen Sie die Stelle in a., wo dieses eingefügt werden soll.
 - Testen Sie das Unterprogramm in WinDLX.
39. Ein A/D-Wandler liefert einen digitalisierten Spannungswert in Form eines 1-aus-16 Codes, der in das Register R1 einer DLX-Maschine geladen wurde.
- Schreiben Sie ein DLX-Unterprogramm (mit *Callee-Save*), das die Korrektheit des Digitalwerts prüft und geben Sie das Resultat der Überprüfung in R2 zurück.
 - Was müsste in ihrem Programm a. geändert werden, wenn der A/D-Wandler einen 1-aus-32 Code liefert?
 - Testen Sie das Unterprogramm in WinDLX.