

Rechnerarchitektur

SS 2017

Übungszettel 1

1. In einer 32-Bit-Stackmaschine laufen alle Operationen über den Stack ab. Nur **push** (Datum auf den Stack, 3 Taktzyklen) und **pop** (Datum vom Stack, 3 Taktzyklen) greifen auf den Speicher zu. Weitere Kenndaten der Rechnerarchitektur sind:
 - Der Operationscode ist 1 Byte lang.
 - Alle Speicheradressen haben eine Länge von 2 Bytes.
 - Alle Operanden sind 4 Bytes lang.
 - Die Variablen A und B stehen vom Beginn an im Speicher.
 - (a) Schreiben Sie ein Programmstück für die Stackmaschine, das $A^2 - B^2$ berechnet und das Ergebnis in die Variable C speichert. Es stehen die Befehlscodes **add** (2 Taktzyklen), **sub** (2 Taktzyklen) und **mult** (8 Taktzyklen) zur Verfügung. Sie sollten dabei mit *einer* Multiplikation auskommen.
 - (b) Wieviele Befehle und Taktzyklen benötigt Ihr Programm? Wieviele Bytes (Instruktionen und Daten) werden beim Programmablauf vom Speicher in die Maschine transferiert?
 - (c) Es stehen nun die zusätzlichen Befehle **dup** (dupliziert das oberste Stackelement, 1 Taktzyklus) und **exc** (vertauscht die beiden obersten Stackelemente, 1 Taktzyklus) zur Verfügung.
 - (d) Ermitteln Sie die Parameter Ihres neuen Programms (analog zu b).
2. Implementieren Sie die Anweisung $e = (a + b)(c + (a + b)d)$ auf einer Stackmaschine.
 - (a) Sie haben die Befehle **push** (3 Taktzyklen), **pop** (3), **add** (1), **mult** (5) zur Verfügung.
 - (b) Sie haben zusätzlich den Befehl **dup** (1 Taktzyklus) zur Verfügung, der das oberste Stackelement dupliziert. Die Addition ist kommutativ ($a + b = b + a$).

Zählen Sie für beide Fälle die Anzahl der Befehle, die Anzahl der Speicherzugriffe (mit und ohne dem Einlesen der Befehlscodes) und die Anzahl der Taktzyklen, die das Programm benötigt. Ermitteln Sie auch, wie viele Bytes im Programmablauf zwischen Speicher und CPU transferiert werden, wenn ein Operationscode 1 Byte lang ist, Speicheradressen 2 Bytes und Operanden 4 Bytes. a , b , c und d stehen von Beginn an im Speicher.
3. Für zwei vorzeichenlose Zahlen $Z1 = A \cdot 2^{16} + B$, $Z2 = C \cdot 2^{16} + D$ soll der mathematische Ausdruck $E = A \cdot C \cdot 2^{32} + (B \cdot C + A \cdot D) \cdot 2^{16} + B \cdot D$
 - (a) mit einer Akku-Maschine: **LOAD** x , **STORE** x , **ADD** x , **MULT** x , **SHL** i
 - (b) mit einer Speicher-Maschine: **ADD** x, y, z , **MULT** x, y, z , **SHL** x, y, i

unter ausschließlicher Verwendung der jeweils angegebenen Opcodes berechnet werden. i ist ein Immediate-Wert (z.B. 16). Wir nehmen an, dass die Speicherstellen ($A - E$) und der Akku alle Ergebnisse aufnehmen können. Zwischenergebnisse können in zusätzlichen Speicherstellen ab F gespeichert werden. Sie sollten die Anzahl der Stores jedoch minimal halten. Geben Sie jeweils den Assembler Code zur Berechnung obigen Ausdrucks an. Vergleichen Sie die Anzahl der Instruktionen und der notwendigen Speicherzugriffe.