Aufgaben für die Konsultationen – Rechnermodell

Hendrik Fehr

23. April 2024

1 Aufgaben zu den Instruktionen

Aufgabe 1: Dokumentation der Instruktionen

Betrachtet wird das Rechnermodell dessen Blockschaltbild in Abbildung 1 gegeben ist. Die wichtigsten Komponenten sind die ALU[1], der Speicher MEM, sowie die Register AC[2], DR[3], PC[4], IR[5] und AR[6]. Die Tabelle 1 gibt die möglichen Instruktionen an, es ist ein minimaler Befehlssatz.

Ergänzen Sie die fehlende Beschreibung der Instruktionen in der freien Spalte. Nutzen Sie das Zeichen := für Zuweisungen, wobei die Zuweisung auf der Seite mit dem Doppelpunkt erfolgt. Beispielsweise bedeutet die Notation $\mathrm{AC} := \mathrm{AC} + \mathrm{DR}$, die Zuweisung der Summe aus AC und DR an das Register AC. Zugriff auf den Speicher an Adresse <adr> wird mit der Notation MEM(<adr>) dargestellt, wobei die Schreibzugriffe auf der Seite mit dem Doppelpunkt stehen und Lesezugriffe auf der anderen Seite. Nutzen Sie die Notation &<var>, um die Adresse von <var> zu ermitteln.

Aufgabe 2: Darstellung der beteiligten Einheiten

Jede Instruktion aus Tabelle 1 benötigt nur einen Teil der Einheiten des Rechnermodells. Markieren sie die Datenflüsse der beteiligten Einheiten in das Blockdiagramm in Abbildung 1. Unterscheiden Sie dabei die Fetch-Phase und die Execute-Phase. Verwenden Sie für jede Instruktion eine andere Farbe oder erstellen sie ein neues Diagramm, damit das Ergebnis übersichtlich bleibt.

[1] *arithmetic logic unit* (arithmetisch-logische Einheit)
[2] *accumulator* (Akkumulator)
[3] *data register* (Datenregister)
[4] *program counter* (Programmzähler)
[5] *instruction register* (Befehlsregister)
[6] *adress register* (Adressregister)

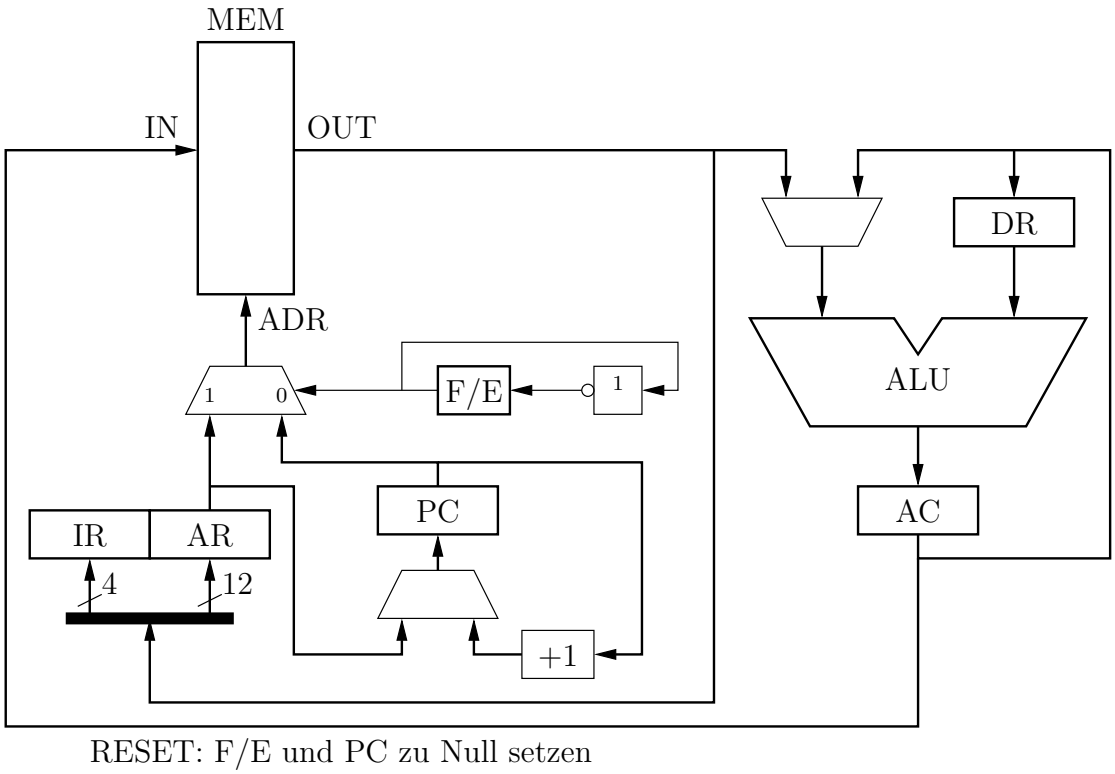

Abbildung 1: Blockschaltbild der Hauptkomponenten des Rechners. Die Steuerung des Adressmultiplexers in Abhängigkeit der Fecht- und Execute-Phasen mit dem F/E-Register ist ausführlich dargestellt. Die restlichen Steuerleitungen, z. B. für den Multiplexer vor dem PC-Register oder für die ALU sind nicht dargestellt. Dünne Signallinien stellen die Übertragung eines Bits dar. Breite Signallinien symbolisieren die parallele Übertragung mehrere Bits. Der Speicher arbeitet mit 16-Bit-Worten und 12-Bit-Adressen, wie die angegebenen Bitbreiten von IR und AR andeuten.

Tabelle 1: Binärformat der Instruktionen des Rechners (minimaler Befehlssatz).

Instruktion	Bit 15	OP	ADR	Kurzbeschreibung (Aufgabe 1)
LD x	1	0b000	$\&x$	
ST x	1	0b001	$\&x$	
BRA *label*	1	0b010	$\&label$	
BZ *label*	1	0b011	$\&label$	
MOV DR, AC	0	0b000	0	
MOV AC, DR	0	0b001	0	
ADD	0	0b010	0	
SUB	0	0b011	0	
AND	0	0b100	0	
NOT	0	0b101	0	

2 Programmieraufgaben

Erstellen Sie ein Git-Archiv und sammeln sie darin die Lösungen oder Teillösungen der folgenden Programmieraufgaben. Dieses Archiv werden Sie später noch benötigen, deshalb sollten Sie es auf Ihren persönlichen Gitlab-Bereich ablegen.

Aufgabe 1: Schleife (minimaler Befehlssatz)

Gegeben sind die zwei folgenden Programmteile:

Ausschnitt `.DATA`-Section

```
; Beispiel einer Schleife
.DATA
cnt: 3; Zählvariable 3, 2, 1 führt zum Durchlaufen, 0 zum Verlassen
```

Ausschnitt `.CODE`-Section

```
.CODE
; Initialisierung ergänzen
loop: ; Start des Schleifenkörpers
; Schleifenköper ergänzen
loop_ende: ; Ende des Schleifenkörpers
.END
```

In der `.DATA`-Section ist schon die Zählvariable für die Schleifendurchläufe vorgesehen und die Sprungmarken `loop` und `loop_ende` in der `.CODE`-Section stellen den Schleifenkörper dar.

Bitte ergänzen Sie die fehlenden Einträge in der `.DATA`-Section und vervollständigen Sie die Instruktionen in der `.CODE`-Section, sodass der Schleifenkörper so oft durchlaufen wird, wie in der Variablen `cnt` angegeben wird.

Suchen Sie die Stellen des Programms die besonders ineffizient sind. Schlagen Sie eine Instruktion zur Erweiterung der vorhandenen Instruktionen vor, die einen effizienteren Ablauf ermöglicht. Ergänzen Sie die Einträge in 1.

Tabelle 2: Ergänzung der Instruktionen des Rechners (erweiterter Befehlssatz).

Instruktion	Bit 15	OP	ADR/EXT	Kurzbeschreibung
DEC	0	0b011	1	AC := AC − 1

Aufgabe 2: Schleife (erweiterter Befehlssatz)

Der Befehlssatz aus Tabelle 1 wird um die Dekrement-Instruktion erweitert. Diese ist in Tabelle 2 dargestellt. Wiederholen Sie die Aufgabe 1 mit dem erweiterten Befehlssatz und schätzen Sie den Zeitgewinn.

Aufgabe 3: Emulierte Multiplikation

Der minimale Befehlssatz des Rechnermodells hat keine Instruktion für die Multiplikation von zwei Zahlen, weil diese durch wiederholte Addition emuliert werden kann. Ergänzen Sie die unten gegebene `.DATA`-Section zu einem vollständigen Programm.

`.DATA`-Section für Aufgabe 3

```
; Berechnet das Produkt von operand_a und operand_b und legt
; das Ergebnis in result bereit.
.DATA
zero: 0
one: 1 ; (nur erforderlich bei minimalem Befehlssatz)
operand_b: 5 ; erster Operand
operand_a: 7 ; zweiter Operand
result: ? ; Ergebnis
```

Aufgabe 4: Wiederverwendung

Das Programm zur Multiplikation aus Aufgabe 3 soll wiederverwendet werden, um mehrere Multiplikationen auszuführen. Die Wiederverwendung soll ohne Kopieren der Multiplikations-Routine auskommen. Wenn Sie einen Tipp benötigen, sehen Sie die Fußnote an.[7] Ergänzen Sie den unten gegebenen Ausschnitt um die fehlenden Variablen und Instruktionen. `.DATA`-Section zu einem vollständigen Programm.

Ausschnitt der `.DATA`-Section für Aufgabe 4

```
; Für i=1,2,3 wird das Produkt von ai und bi Berechnet und
; das Ergebnis in ri bereit gelegt.
.DATA
; Operanden
a1: 5
a2: 7
a3: 3
b1: 9
b2: 7
b3: 8
```

[7] Die Lösung besteht darin, eine veränderbare Sprunganweisung nach der Multiplikations-Routine vorzusehen, die eine Fortsetzung des Programm an der jeweils erforderlichen Stelle erlaubt.

```
; Ergebnisse
r1: ?
r2: ?
r3: ?
; fehlende Variablen ergänzen
```

Welche Hardware-Erweiterung des Rechnermodells und welche zugehörigen Instruktionen ermöglichen eine bequemere Implementierung und einen effizienteren Ablauf von Programmen mit Wiederverwendung? Rekursion soll für diese Überlegung ausgeschlossen werden, d. h. eine Routine soll sich nicht selbst wiederverwenden. Ergänzen Sie das Blockdiagramm um die Hardware-Erweiterung und tragen Sie die zugehörigen Instruktionen in die Tabelle 2 ein.