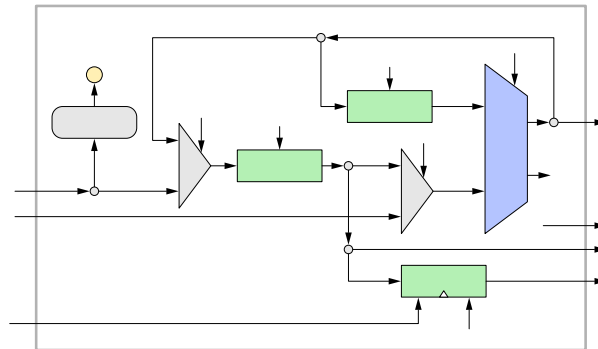


10. Übungsblatt - zu bearbeiten bis 15.01.2024

## Aufgabe 1 Hack-Prozessor: Steuerlogik

- Von welchen Größen hängt das Steuersignal `load` des Adreßregisters **A** (*address register*) des Hack-Prozessors ab? Hinweis: Beachten Sie sowohl die Bits der auszuführenden Maschinenbefehle als auch Steuerleitungen innerhalb der CPU.
- Geben Sie einen logischen Ausdruck (Boolesche Formel) an, mit der das Steuersignal `load` des Adressregisters **A** (*address register*) des Hack-Prozessors aus den in Teilaufgabe a) identifizierten Größen berechnet werden kann!
- Wo in der unten gezeigten schematischen Darstellung des Hack-Prozessors ist die Implementierung der in b) entwickelten logischen Funktion angesiedelt?



## Aufgabe 2 Hack-Prozessor: Steuerlogik

- Von welchen Größen hängt das Steuersignal `load` des Programmzählers `PC` (*program counter*) des Hack-Prozessors ab? Hinweis: Beachten Sie sowohl die Bits der auszuführenden Maschinenbefehle als auch Steuerleitungen innerhalb der CPU.
- Geben Sie einen logischen Ausdruck (Boolesche Formel) an, mit der das Steuersignal `load` des Programmzählers `PC` (*program counter*) des Hack-Prozessors aus den in Teilaufgabe a) identifizierten Größen berechnet werden kann! Versuchen Sie, diesen Ausdruck so einfach wie möglich zu halten! (Es ist aber keine Minimierung mit den früher in der Vorlesung betrachteten Verfahren gefragt.)

**Aufgabe 3** Hack-Assembler- und -Maschinensprache

- a) Übersetzen Sie den rechts stehenden Quelltext aus der Hack-Assemblersprache in die Hack-Maschinensprache!
- b) Wie kann man den rechts stehenden Quelltext auf vier Anweisungen kürzen, die das gleiche Ergebnis liefern?
- c) Für welche Assemblerbefehle stehen die folgenden Maschinenspracheanweisungen?
- (i) 1111000000010000, (ii) 0000000000000111, (iii) 1111000010011000.

```
@i
M=M-1
@i
D=M
@k
M=D+M
```

**Aufgabe 4** Hack-Assemblersprache

- a) Was berechnet das rechts gezeigte Programm 1? Wie sieht die Symboltabelle für dieses Programm aus? Hinweis: Hack-CPU-Emulator
- b) Was ändert sich am Programmablauf und an der Symboltabelle, wenn die ersten vier Anweisungen geändert werden zu:

```
@sum
M=0
@i
M=0
```

Wie verändert sich hierdurch das Ergebnis?

- c) Betrachten Sie das rechts gezeigte Programm 2. Was berechnet es? Wie unterscheidet es sich in seinen Berechnungen von Programm 1?
- d) Wie könnte man das Ergebnis von Programm 1 mit weniger Anweisungen berechnen? Schaffen Sie es, mit weniger als 14 Anweisungen (Markierungsdefinitionen wie (LOOP) nicht gerechnet) auszukommen? Hinweis: Berechnungsreihenfolge! Testen Sie Ihr Programm im Hack-CPU-Emulator!

## Programm 1

```
@i
M=1
@sum
M=0
(LLOOP)
@i
D=M
@R0
D=D-M
@WRITE
D; JGT
@i
D=M
@sum
M=D+M
@i
M=M+1
@LOOP
0; JMP
(WRITE)
@sum
D=M
@R1
M=D
(END)
@END
0; JMP
```

## Programm 2

```
@sum
M=0
@i
M=0
@TEST
0; JMP
(LLOOP)
@i
D=M
@sum
M=D+M
@i
M=M+1
(TEST)
@i
D=M
@R0
D=D-M
@LOOP
@LOOP
D; JLE
@sum
D=M
@R1
M=D
(END)
@END
0; JMP
```