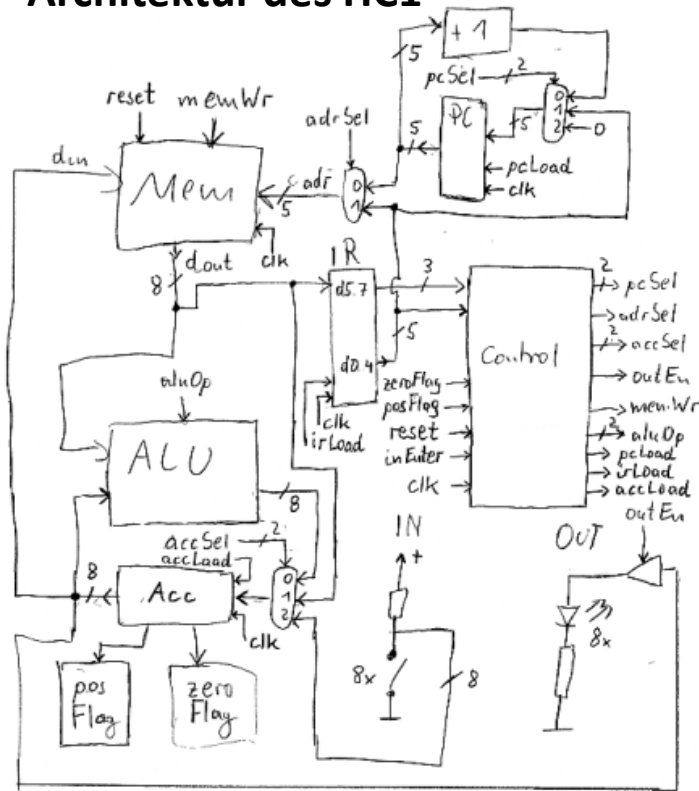


Assembler / Linker für den HC1-Mikrorechner

- **Ausgangssituation**
- Anforderungen
- Implementierung
 - Assembler
 - Linker
- Live-Vorführung

Ausgangssituation – HC1

Skizze der Architektur des HC1



Instructionset des HC1

Instruction	Encoding	Operation	Comment
LOAD A, address	000 aaaaa	$A \leftarrow M[aaaaa]$	Load A with content of memory location aaaaa
STORE A, address	001 aaaaa	$M[aaaaa] \leftarrow A$	Store A into memory location aaaaa
ADD A, address	010 aaaaa	$A \leftarrow A + M[aaaaa]$	Add A with M[aaaaa] and store the result back into A
SUB A, address	011 aaaaa	$A \leftarrow A - M[aaaaa]$	Subtract A with M[A] and store the result back into A
NAND A, address	100 eeeee	$A \leftarrow \text{not} (A \text{ and } M[bbbbbb])$	Perform logical NAND operation of A and M[bbbbbb] and store the result back into A
IN A	100 00000	$A \leftarrow \text{Input}$	Input to A
OUT A	100 00001	$\text{Output} \leftarrow A$	Output from A
JZ address	101 aaaaa	IF $A == 0$ THEN $PC \leftarrow aaaaa$	Jump to address if A is zero
JPOS address	110 aaaaa	IF $A > 0$ THEN $PC \leftarrow aaaaa$	Jump to address if A is a positive number
J address	111 aaaaa	$PC \leftarrow aaaaa$	Jump always to address

Notations:
 A accumulator
 M memory
 PC program counter
 aaaaa five bits for specifying a memory address
 eeeee five bits for specifying a memory address > 1



- Ausgangssituation
- **Anforderungen**
- Implementierung
 - Assembler
 - Linker
- Live-Vorführung

Kurzübersicht

- **Assembler**
 - Assembler-Code in Maschinencode übersetzen
 - Syntaktische Fehler im Assembler-Code erkennen
 - (Brauchbare) Fehlermeldungen generieren
- **Linker**
 - Zusammenführen mehrerer Quellcode-Dateien
 - Speicheradressen anpassen

- Ausgangssituation
- Anforderungen
- **Implementierung**
 - **Assembler**
 - Linker
- Live-Vorführung

- Quelldateien einlesen
 - Feld
 - zeichenweise vergleichen (Mnemonic, label)
- Argument (Adresse) prüfen
 - Opcode + Adresse zusammenfügen
- Symboltabelle aktualisieren
 - Index für Zeile und Datei

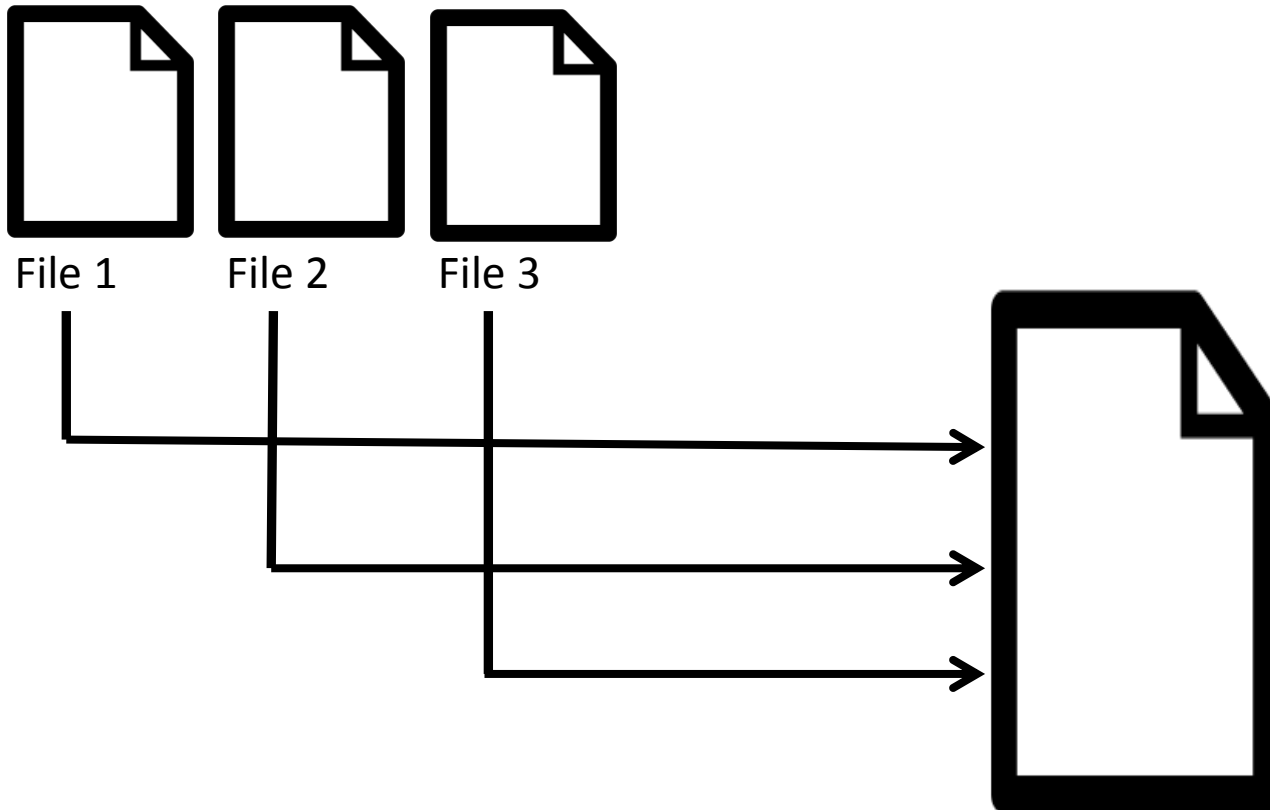
- Mnemonic
 - Opcode
 - Bezeichnung (LOAD)
 - Flag, für korrekte Erkennung

- Ausgangssituation
- Anforderungen
- **Implementierung**
 - Assembler
 - **Linker**
- Live-Vorführung

Funktionsweise (1)

Quelldateien übersetzt in
Maschinencode (von Assembler
erzeugt)

Zusammenführen zu einer „großen“
Datei mit Maschinencode



Funktionsweise (2)

```

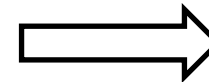
      IN A
      STORE A, zahl
      J lab1
weiter: OUT A
ende:   J ende
zahl:   20
  
```

File 1

```

lab1: ADD A, zahl
      J weiter
  
```

File 2



```

      IN A
      STORE A, zahl
      J lab1
weiter: OUT A
ende:   J ende
zahl:   20
lab1:   ADD A, zahl
      J weiter
  
```

Output

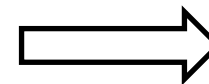
```

10000000
001aaaaa
111aaaaa
10000001
111aaaaa
  
```



```

010aaaaa
111aaaaa
  
```



```

10000000
001aaaaa
111aaaaa
10000001
111aaaaa
010aaaaa
111aaaaa
  
```

Funktionsweise (3)

```

                IN A
                STORE A, zahl
                J lab1
weiter:         OUT A
ende:          J ende
zahl:          20
lab1:          ADD A, zahl
                J weiter
  
```

```

10000000
001aaaaa
111aaaaa
10000001
111aaaaa
010aaaaa
111aaaaa
  
```

Symboltabelle (angelegt von Assembler)

Label	Wo wird Referenz genommen?	Wo ist Label definiert?
zahl	File 1, Zeile 2	File 1, Zeile 6
weiter	File 2, Zeile 2	File 1, Zeile 4

Sonstige Daten

Datei	Anzahl Instruktionen
File 1	6
File 2	2

Funktionsweise (4)

Daten für Linker

```

      IN A
      STORE A, zahl
      J lab1
weiter: OUT A
ende:   J ende
zahl:   20
lab1:   ADD A, zahl
        J weiter
  
```

```

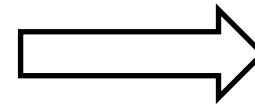
10000000
001aaaaa
111aaaaa
10000001
111aaaaa
010aaaaa
111aaaaa
  
```

Symboltabelle (angelegt von Assembler)

Label	Wo wird Referenz genommen?	Wo ist Label definiert?
zahl	File 1, Zeile 2	File 1, Zeile 6
weiter	File 2, Zeile 2	File 1, Zeile 4

Sonstige Daten

Datei	Anzahl Instruktionen
File 1	5
File 2	2



Final

```

10000000
00100101
11100101
10000001
11100100
01000101
11100011
  
```

- Ausgangssituation
- Anforderungen
- Implementierung
 - Assembler
 - Linker
- **Live-Vorführung**

Vielen Dank für Ihre Aufmerksamkeit