

# **Miniprojekt: Minimax-Maschine Pflichtenheft**

Maximilian Lumpe, Niklas Blume, Jan Feuchter, Phu Bac Duong

Hardware Projekt 2017  
03. Januar 2017

# Contents

1	Einleitung	2
2	Aufgabe: Paketanalyse	2
3	Ist-Analyse der Basismaschine	3
4	Beschreibung des Implementierungskonzeptes	4
5	Angestrebte Projektergebnisse	4
6	Arbeitsplanung und -verteilung	5
7	Anhang: Flussdiagramme geplanter Mikroprogramme	5
8	Anhang: Verwendete Hilfsmittel	5

# 1 Einleitung

In diesem Miniprojekt im Rahmen des Hardware-Praktikums beschäftigen wir uns mit der Minimax-Maschine, welche uns grundlegend aus der Vorlesung "Grundlagen der Rechnerarchitektur" bekannt ist. Zur Lösung der Aufgaben ist es hierbei notwendig, die vorgegebene Grundstruktur der Maschine geeignet zu erweitern, um die Algorithmen zu realisieren.

Die Vorbereitung auf unser Projekt wird dokumentiert und strukturiert durch das von uns erstellte Pflichtenheft. Das Pflichtenheft wird nur unsere Vorbereitung beinhalten. Die Ergebnisse werden in einer weiteren Dokumentation enthalten sein.

## 2 Aufgabe: Paketanalyse

Nach unserem Verständniss ist das Ziel der Aufgabenstellung das Implementieren des Algorithmus "Paketanalyse" auf der Minimax-Maschine. Dieser Algorithmus wertet die Länge des Nutzdatenteils der Datenpakete aus dem Speicher der Maschine aus.

Jedes Paket besteht aus einem Header mit 80 Bits, gefolgt von dem Datenteil mit variabler Länge. Ein Paket beginnt mit dem festgelegten Bitmuster 1110. Der Header enthält eine 2 Bytes lange Kanalnummer, die bei der Bitstelle 32. beginnt. Zu einem Kanal gehören mehrere Datenpakete mit einer eindeutigen Kanalnummer. Die Anzahl der Bits, die in den Speicher geladen werden, wird als bekannt vorausgesetzt und wird in ein entsprechendes Register vorgeladen.

Nun soll der "Paketanalyse"-Algorithmus eine Tabelle, die Kanalnummern und zugehörige Datenlängen (in Bits) enthält, anlegen. Haben mehrere Paket dieselbe Kanalnummer, so werden die Längen des Nutzdatenteils zusammengefügt. Die Tabelle soll ab einer beliebigen Speicheradresse außerhalb des Paketfeldbereichs im Hauptspeicher der Maschine abgelegt werden.

Diese Aufgabenstellung soll mit dem gegebenen Minimax-Simulator simuliert und getestet werden. Die Maschine kann durch vorgegebene Bauteile erweitert werden, was sich jedoch auf die Bewertung auswirkt. Der Algorithmus wird in Form der Steuertabelle implementiert und soll außerdem als Flussdiagramm abgegeben werden.

### 3 Ist-Analyse der Basismaschine

Die Minimax-Maschine ist ein minimales Rechensystem welches im Wesentlichen aus einigen Registern (Basis ACCU, PC, IR, MDR, MAR) und einer arithmetischen Einheit (ALU), welche den Datenpfad bilden, und einem Hauptspeicher (HS) besteht. Jeder Register hat als Eingang mindesten die ALU und zusätzlich einen Eingang, der den Schreibzugriff regelt. Der Befehlsablauf des Systems wird über ein Mikroprogramm festgelegt.

Für das Projekt kann die Architektur der Minimax-Maschine um zusätzliche Register bzw. Sign Extension Units erweitert werden.

- Die Basismaschine:
  1. ACCU: Abkürzung für Accumulator ein Zwischenspeicher, um mit MDR Operationen durchführen zu können.
  2. PC: Abkürzung für program counter, enthält den Programmzähler, welcher den nächsten Befehl beeinflusst.
  3. IR: Abkürzung für instruction register, enthält Opcode(8 Bit) und Adressteil(24 bit).
  4. MDR: Abkürzung für memory data register, enthält je nach Einstellung des Multiplexers `textttMDR.Sel`, verschiedene Daten entweder aus Hauptspeicher oder aus der ALU.
  5. MAR: Abkürzung für memory adress register, enthält die Speicheradresse an der ausdem Hauptspeicher Daten geladen oder geschrieben werden sollen.
  6. HS: Abkürzung für Hauptspeicher, er wird mit 24-Bit durch MAR adressiert und gibt eine 32-Bit Zahl zurück. Auf die selbe Art funktioniert schreiben einer 32-Bit Zahl natürlich auch.
- ALU-Operationen der Basismaschine:
  1. ADD: Addiert ALU-Eingang A und ALU-Eingang B
  2. SUB.B: Subtrahiert ALU-Eingang A von ALU-Eingang B
  3. TRANS.A: Schaltet den ALU-Eingang A durch
  4. TRANS.B: Schaltet den ALU-Eingang B durch

Die möglichen Operationen auf die in der ALU implementieren Operationen sind beschränkt ( Basis : ADD, SUB, TRANS.A, TRANS.B ) . Die ALU kann aber mit zusätzliche Operationen , wie z.B dem DIV-Befehl oder dem AND-Befehl , ergänzt werden . Um eine Operation auszuführen müssen über die Multiplexer ALUSel.A und AluSel.B zwei Operanden ausgewählt werden und der ALU muss über die ALU Ctrl-Leitung der Code für die Operation übergeben werden. An den Multiplexern liegen sowohl Konstanten als auch die Register an, welche zur ALU durchgeschaltet werden können. Das Ergebnis der Operation kann entweder in einem Register (über MDR) oder im HS (Adresse im Register MAR) gespeichert werden. Zusätzlich können Flags (Basis nur ein Flag ALU RESULT == 0) gesetzt werden, welche zurück zur Control Unit (CU) geleitet werden um z. B. bedingte Sprünge auszuführen. Die uns vorliegende Minimax-Maschine arbeitet mit 32-Bit und speichert Werte mit 32-Bit in den Registern und im HS. Alle ALU-Operationen werden folglich mit 32-Bit ausgeführt. Dies stellt sich jedoch für unser Aufgabe als Hindernis, da wir die Daten bitweise untersuchen müssen, Daten aus dem HS und den Registern jedoch nur als 32-Bit Zahlen auslesen können und nicht als einzelne Bits.

Folglich wird die Basismaschine um einige Konstanten, Operationen und Register erweitert werden müssen, welche im Implementierungskonzept näher aufgeführt sind.

## **4 Beschreibung des Implementierungskonzeptes**

## **5 Angestrebte Projektergebnisse**

1. erster Stichpunkt
2. zweiter Stichpunkt
3. dritter Stichpunkt

- 6    **Arbeitsplanung und -verteilung**
- 7    **Anhang: Flussdiagramme geplanter Mikroprogramme**
- 8    **Anhang: Verwendete Hilfsmittel**