

Praktikum Rechnerstrukturen– Aufgabenblatt 1

Die folgenden Aufgaben sollen in 2er Gruppen gemeinsam an einem PC bearbeitet werden. Das Kopieren von fertigen Lösungen, die von Personen außerhalb der Arbeitsgruppe erstellt wurden wird als Täuschungsversuch gewertet und kann zum Ausschluss vom Praktikum führen.

Die Lösungen sind im Praktikum zu präsentieren und direkt danach per moodle abzugeben. Die Bearbeitungsdauer beträgt ca. 3 Praktikumstermine.

Einleitung

In diesem Aufgabenblatt sollen Sie den im weiteren Praktikum verwendeten Simulator *Hades* kennenlernen. Anhand eines einfachen Beispiels sollen Sie sich mit den Grundfunktionen von *Hades* vertraut machen und lernen, ein existierendes Schaltnetz zu laden und zu simulieren, zu verändern, sowie anschließend ein Schaltnetz selber aufbauen zu können.

Aufgaben

1-1. Hades kennenlernen

- Bitte laden Sie zunächst aus moodle den Simulator *Hades* in ein Verzeichnis ihrer Wahl innerhalb ihres Account herunter. Laden Sie ebenfalls die in moodle gegebene Beispieldatei herunter.
- Starten Sie Hades und öffnen Sie das Beispiel.
- Verschaffen Sie sich einen Überblick über das Schaltnetz (SN), seine Funktionsweise und seine funktionellen Komponenten.
- Simulieren Sie das Schaltnetz in Hades. Verändern Sie die Belegung des Eingangsvektors über die Schalter links.
- Was bedeutet die Ausgabe des SN? (Tipp: Das SN funktioniert nicht 100% richtig) Bedenken Sie, dass die Komponenten des Eingangsvektors, z.B. in Wertetabellen nach fallenden Indizes sortiert angegeben werden, d.h. $X=(x_3, x_2, x_1, x_0)$.

1-2. Gegebenes Schaltnetz verändern

- Die Eingabe der Eingangsvektoren als Binärzahl mit einzelnen Schaltern ist unkomfortabel und fehleranfällig. In Hades gibt es daher ein Hex-Tastenfeld (hex switch), dass zur Eingabe von 4-Bit Binärzahlen verwendet werden kann. Bitte ersetzen Sie die einzelnen Schalter, die bisher die Belegung des Eingangsvektors bestimmen, durch dieses Hex-Tastenfeld. Testen Sie Ihre Schaltung auf unveränderte Funktion.
- Analog zur Eingabe gibt es eine Komponente, die 4-Bit Zahlen Hexadezimal darstellen kann (hex display). Bitte ergänzen Sie die Schaltung um eine Hexadezimalanzeige, die die aktuelle Eingangsbelegung anzeigt. Testen Sie Ihre Schaltung auf unveränderte Funktion.
- Die Beispielschaltung verwendet nur Gatter mit zwei Eingängen, was den erlaubten Operatoren der Schaltalgebra entspricht. Es ist allerdings üblich auch ODER- und UND-Gatter zu verwendenden, die drei oder vier Eingänge haben. Ein ODER-Gatter mit 3 Eingängen kann z.B. als die Funktion $\text{or}: \mathbb{B}^3 \rightarrow \mathbb{B}$ mit der Funktionsgleichung $\text{or}(x_2, x_1, x_0) = (x_2 \vee x_1 \vee x_0)$ geschrieben werden und in einem erweiterten Booleschen Ausdruck direkt verwendet werden.

- i. Zeigen Sie durch Anwenden der Rechenregeln der Schaltalgebra, dass diese Gatter mit drei oder vier Gattern sich analog zu Gattern mit nur zwei Eingängen verhalten und damit ihre Verwendung problemlos möglich ist.
 - ii. Wie beeinflussen diese Gatter mit mehr als zwei Eingängen die Tiefe eines Ausdrucks?
 - iii. Ändern Sie das SN in Hades so, dass nun Gatter mit zwei bis vier Eingängen sinnvoll verwendet werden.
- d. Wie sie in der vorherigen Aufgabe analysiert haben, implementiert das Schaltnetz die gewünschte Funktionalität nicht korrekt.
Geben sie die vom Schaltnetz implementierte Booleschen Funktion an. Lesen Sie hierzu den implementierten Booleschen Ausdruck aus dem gegebenen Schaltnetz ab.
- e. Korrigieren Sie das SN, so dass es die korrekte Boolesche Funktion berechnet. Geben Sie den korrekten Booleschen Ausdruck an. Testen Sie Ihre Schaltung auf Korrektheit.

1-3. Gegebenes Schaltnetz selber aufbauen

- a. Bauen Sie einen 1-Bit Volladdierer aus einzelnen Gattern auf. Realisieren Sie das SN wie in der VL behandelt durch strukturellen Entwurf. Verwenden die dabei den in der VL besprochenen Halbaddierer als Komponente.
- b. Vervollständigen Sie dieses SN um sinnvolle Komponenten für Ein- und Ausgabe. Nutzen Sie die Funktion, LEDs und Schaltern Namen geben zu können.
- c. Testen Sie ihre Schaltung auf Korrektheit.

1-4. Entwurf eines Schaltnetz aus einer textuellen Beschreibung

Es soll ein zweistufiges Schaltnetz entwickelt und in Hades aufgebaut werden, das eine totale Boolesche Funktion implementiert, deren Erfüllbarkeitsmenge alle Eingangsvektoren der Breite vier enthält, deren Hamminggewicht zwei ist (d.h. die genau zwei Einsen enthalten).

- a. Geben Sie die Boolesche Funktion formal an, als
 - i. Tabelle
 - ii. Boolescher Ausdruck in DNF. Versuchen Sie den Ausdruck so weit wie möglich zu vereinfachen.
 - iii. Operatorbaum
- a. Geben Sie die Erfüllbarkeitsmenge formal an.
- b. Implementieren Sie die Funktion als Schaltnetz. Versuchen Sie die Anzahl der für den Aufbau nötigen Gatter so gering wie möglich zu halten ohne dabei Übersicht und Nachvollziehbarkeit des Entwurfes zu verschlechtern. Bedenken Sie, dass Gatter mit mehr als 2 Eingängen einen intern erhöhten Aufwand bedeuten und daher ein 3-wertiges Gatter mit Gewicht 2 zählt und ein 4-wertiges mit Gewicht 3.
- c. Ergänzen Sie Ihre Schaltung um geeignete Ein- und Ausgabemöglichkeiten und testen Sie sie auf Korrektheit. (Dieses Testen wird in Zukunft nicht mehr als Teilaufgabe explizit genannt werden, sondern ist immer implizit als Teil der Implementierung einer Schaltung in Hades gefordert.)
- d. Auf welche Weise haben Sie den Gatterbedarf reduziert? Dokumentieren Sie ihren Lösungsweg.

Abgabe / Präsentation im Praktikum

Alle Aufgaben sind im Praktikum (i.a. an einem der Laborrechner) zu präsentieren. Die Aufgaben eines Aufgabenzettels sollen in der angegebenen Reihenfolge bearbeitet und einzeln präsentiert werden. Gelöste Aufgaben *sollen* erst präsentiert werden und dann die nächste Aufgabe bearbeitet werden. Die Bearbeitung eines neuen Aufgabenzettels *darf* erst begonnen werden, wenn der vorhergehende abgegeben und präsentiert ist.

Die Lösungen der Aufgaben sind zu dokumentieren. D.h. alle erstellten Simulationsschaltungen sind mit sprechenden Namen zu versehen. Wenn vorbereitende Arbeiten (z.B. das Ermitteln einer Schaltfunktion oder ähnlicher Entwurfsschritte) zur Aufgabenstellung gehören, sind diese schriftlich knapp festzuhalten und elektronisch abzugeben. Dabei sind ggf. die erstellten Hades-Simulation mit dem Dateinamen zu referenzieren. Handschriftliche eingescannte Lösungen sind erlaubt.

Die erstellten Lösungen müssen auf den Laborrechnern fehlerfrei laufen. Falls Sie Lösungen auf eigenen Laptops/Notebooks erstellt haben, müssen diese trotzdem auf den Laborrechnern fehlerfrei laufen. Präsentationen auf dem eigenen Laptop/Notebook erfolgen auf eigene Gefahr.

Hochladen in moodle

Die präsentierten Lösungen sind unverändert gruppenweise in moodle hochzuladen. Je Gruppe bitte nur eine Abgabe gemäß der unten angegebenen Form.

- Die Hades-Schaltungen sind umgehend nach der Präsentation im Praktikum, d.h. in jedem Fall *am selben Kalendertag* hochzuladen.
- Der komplette Praktikumsbericht (d.h. inklusive der zuvor bereits abgegeben Hades-Dateien) zu einem abgeschlossenen Aufgabenblatt ist *spätestens am darauffolgenden Kalendertag* abzugeben. Erfolgt dies am selben Kalendertag, darf die erste Abgabe entfallen.

Die Abgaben sollen in folgender Form geschehen:

- Je Abgabe eine Archivdatei (.zip oder .rar) mit folgendem Inhalt:
 - Alle Dateien, die Teil der Lösung sind mit selbsterklärenden Namen. Ggf. bitte Verzeichnisse verwenden, die die Lösungen von Teilaufgaben voneinander trennen.
 - Bei umfangreicheren Abgaben muss eine Datei `readme.txt` im Wurzelverzeichnis des Archivs vorhanden sein, die ggf. notwendige Hinweise zur Verwendung der Lösung enthält.
 - Als Dokumentformate für Berichte sind nur .txt, .doc, .docx, .pdf und .jpg zulässig.
 - Eine Textdatei mit Namen `gruppe.txt` im Wurzelverzeichnis des Archivs, die die Namen der Gruppenmitglieder mit Matrikelnummer zeilenweise auflistet.
 - Die Archivdatei muss einen Dateinamen haben, der eindeutig das Fach, die Aufgabe und die Gruppe enthält. Folgendes Format ist zu verwenden:
SS2015-RS_Aufg<x>_<Name>_<Vorname>.zip (bzw. .rar). Ein Beispiel:
SS2015-RS_Aufg1_Blaurock_Ole.zip.
Dabei ist es egal von welchem Gruppenmitglied der Name verwendet wird.
- Pro Gruppe ist nur eine Lösung in moodle hochzuladen.
 - Sollte eine Gruppe mehrere Lösungen abgeben, wird eine nach dem Zufallsprinzip ausgewählt und die anderen Lösungen ignoriert.
- Lösungen, die nicht den formalen Vorgaben entsprechen, werden nicht berücksichtigt.
- Nicht vergessen, die Lösungen in moodle als endgültig abgegeben zu markieren.