



HPSN

Lehrstuhl für Hochparallele VLSI-Systeme und Neuromikroelektronik

Schaltkreis- und Systementwurf

Einführung Belegarbeit

Dresden

Ziel der Lehrveranstaltung

- Implementierung eines Algorithmus als integrierte Digitalschaltung (AS/C) bestehend aus Zustandsautomat (FSM), Steuerlogik und Datenpfad.
- Verifikation der Schaltung.
- Zieltechnologie: $0.35\ \mu\text{m}$ CMOS.

Schaltung – Agenda

Schaltung

Struktur

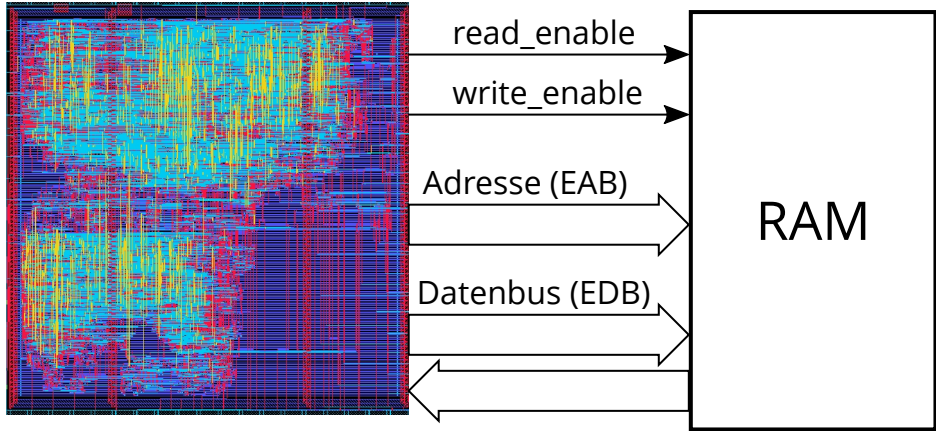
Eingangs- und Ausgangsdaten

Algorithmus

Belegarbeit

Zusammenfassung

Struktur



ASIC (Standalone Chip oder Hardwarebeschleuniger) und (externer) RAM.
Hier: Numerischer Coprozessor, Speicher mit Prozessor geteilt.

Eingangs- und Ausgangsdaten

- Datenspeicherung im RAM
- Eingangsdaten:
 - Konfigurationsdaten, z. B. Parameter des Algorithmus, Abbruchkriterien, Maximale Anzahl Iterationen
 - Argumente, z. B. Skalare Werte, Listen, Vektoren
- Ausgangsdaten: Ergebnisse, z. B. Skalare Werte, Listen, Vektoren
- Für die Schaltungssimulation ist es möglich, den RAM-Inhalt vor Ausführung aus einer Textdatei zu laden und nach Ausführung in eine Textdatei zu schreiben.

Adresse	Datum
0	parameter0
1	parameter1
2	argument0
$\vdots$	$\vdots$
n	ergebnis0
n + 1	ergebnis1

Algorithmus – Agenda

Schaltung

Algorithmus

Auswahl

Beispiele

Anmelden

Referenzimplementierung

Belegarbeit

Zusammenfassung

Algorithmusauswahl

- Geeignet sind numerische Algorithmen mit Integer, Festkomma- oder Fließkomma-Datenformaten
- Ungeeignet sind Algorithmen mit überwiegend Bit Manipulationen (Ansatz mit FSM und Datenpfad hier unpassend)
- Schlecht geeignet sind Algorithmen, die vorwiegend Speicherzugriffe ausführen (wenig Optimierungsspielraum)
- Komplexität:
 - Zwischen 16 und 64 Zustände der FSM
 - Es müssen Verzweigungen enthalten sein!
- Fokus der Lehrveranstaltung ist die Umsetzung als Digitalschaltung (Einarbeitung in Algorithmus muss selbstständig erfolgen)

Algorithmus

Beispiele 1

Lösungen von Gleichungen oder Gleichungssystemen:

- Newton-Verfahren
- Bisektion
- Fixpunkt-Verfahren
- Gauß-Algorithmus

Näherungsverfahren für Funktionen:

- CORDIC
- Reihenentwicklung von Funktionen (z. B. $\cosh(2x^2 + 3)$, $\sinh(3x^2 - 2)$, $e^x - x^2$, ...)
- Newton-Verfahren

Matraxeigenschaften

- Determinante
- Eigenwert, Eigenvektoren

Algebra

- größter gemeinsamer Teiler
- kleinstes gemeinsames Vielfaches
- erweiterter ggT
- Polynomdivision
- Langzahlarithmetik (Addition/Subtraktion, Multiplikation/Division)
- Potenzen modulo n (Square-and-Multiply)
- Primzahltests

Algorithmus

Beispiele 2

Numerische Lösung von Differentialgleichungen und wissenschaftliches Rechnen:

- Wärmeleitungsgleichung (z. B. Temperaturverteilung in einem festen Medium)
- Bewegungsgleichung (z. B. Wurf von Bällen)
- Technische Mechanik, Statik Probleme
- Elektrische Felder (z. B. Potentialverläufe in Medien, Kraft zwischen geladenen Körpern)
- Lösung der Schwingungsgleichung

Mögliche Verfahren:

- Euler vorwärts/rückwärts (explizit/implizit),
- Runge-Kutta (verschiedene)

Algorithmus

Beispiele 3

Signalverarbeitung

- Digitale Filter (FIR, IIR, Kalman)
- Diskrete Sinus/Cosinus Transformation
- Faltung zweier Funktionen

Geometrie

- Umrechnung Polarkoordinaten in kartesische Koordinaten (Näherung sin/cos)
- Polygoneigenschaften:
 - Fläche
 - Mittelpunkt
 - Schwerpunkt
 - Test: Punkt in Polygon
- Test: Ähnlichkeit zweier Dreiecke
- Volumen und Oberfläche von Rotationskörpern

Statistik, Daten

- Varianzberechnung, Mittelwert, Kovarianz, Autokorrelation
- Interpolation (z. B. Spline)
- Regression (Polynom, nichtlinear)

Sonstiges

- Kalenderalgorithmus (Eingabe: Datum, Ausgabe: Wochentag)
- Berechnung von fraktalen Graphiken (z. B. Apfelmännchen, Julia Menge)
- Einfache Bildverarbeitungsalgorithmen (z. B. Filter)
- Kürzester Weg (z. B. Dijkstra)
- Rasterung (Computergraphik, z. B. Linie, Kreis)

Eigene kreative Vorschläge willkommen!

Algorithmus

Anmelden

- Sobald Sie sich für einen Algorithmus entschieden haben: Mail an lab-icdesign@groups.tu-dresden.de mit
 - Name + Cluster-Login
 - Algorithmus
 - Relevante Details (z. B. „FIR Filter mit n Stufen, Festkommaarithmetik“).
 - Warten auf Antwort, ob Thema noch verfügbar ist(Bitte TU-Mailadresse als Absender verwenden).
- Innerhalb von 14 Tagen nach unserer Antwort: Senden Sie uns den Quelltext (Sprache nach Wahl) einer ersten eigenen Referenzimplementierung (*nur den wesentlichen Teil des Algorithmus, nur ASIC-taugliche Rechenoperationen*) an die Mailingliste
 - OK: Thema reserviert
 - Nachbesserungsbedarf: nachbessern
 - keine Implementierung: Thema wird wieder freigegeben
- Thema ist dann verbindlich für den Beleg, Referenzimplementierung kann noch weiter bearbeitet werden.
- *Studenten älterer Semester dürfen ihr eingetragenes Thema bis zur Belegabgabe weiter bearbeiten.*

Referenzimplementierung

Potentielle Probleme

- Komplexe Datenstrukturen
 - In Software leicht,
 - in Hardware gibt es nur einen linear adressierten Speicher, vergleichbar mit einem Array.
 - Kompliziertere Datenstrukturen müssen darauf abgebildet werden (können).
- Funktionsaufrufe
 - In Software normal und sinnvoll,
 - in Zustandsautomat als eingebetteter Unterablauf möglich,
 - Rücksprung zu mehreren Aufrufstellen bereits aufwendiger (nach möglichkeit vermeiden);
 - insbesondere Rekursion ist aufwendig (vermeiden, verringert im Allgemeinen den Performancevorteil der Hardwareimplementierung).
- Abstraktion, Bibliotheken
 - In Software sinnvoll und erwünscht,
 - in Hardware stehen nur Grundoperationen (Addition/Subtraktion, Multiplikation, Division) und einfache logische Operationen zur Verfügung.
 - Was darüber hinausgeht, muss selbst algorithmisch umgesetzt werden (z. B. Wurzel, Sinus, ...).

Referenzimplementierung

Negativbeispiel (Namenssortierung)

- Komplexe Datenstrukturen
 - Person als Objekt
 - 2 Variabel lange Strings als Attribute
- Funktionsaufrufe
 - Vergleichsfunktion (wird von sort aufgerufen)
 - Indirekt: Stringvergleichsfunktion (in less than Funktion)
- Essentielle Teile sind Bibliotheksfunktionen
 - Stringvergleich
 - Sortierfunktion

Referenzimplementierung (ungeeignet):

```
1 class Person:
2     def __init__(self, first_name, last_name):
3         self.first_name = first_name
4         self.last_name = last_name
5
6     # custom less than comparison
7     def __lt__(self, other):
8         if (self.last_name < other.last_name):
9             return True
10        if (self.last_name == other.last_name):
11            if (self.first_name < other.first_name):
12                return True
13        return False
14
15
16 def sortPersons(person_list):
17     person_list.sort()
```

Referenzimplementierung

Beispiel (Bubblesort)

Referenzimplementierung (Anmeldung):

```
1 void bubblesort (unsigned list[], unsigned n)
2 {
3     bool changed = false;
4
5     do {
6         changed = false;
7
8         for (unsigned i = 0; i < n-1; i++) {
9             if (list[i + 1] > list[i]) {
10                 unsigned temp = list[i + 1];
11                 list[i + 1] = list[i];
12                 list[i] = temp;
13
14                 changed = true;
15             }
16         }
17     } while (changed);
18 }
```

Optimierte Version (Beleg):

```
1 void bubblesort (unsigned list[], unsigned n)
2 {
3     int fc = 0; // flag change
4     do {
5         fc = 0;
6
7         unsigned dc = 0;
8         while (dc < n-1) {
9             unsigned dc1 = dc + 1;
10             unsigned r1 = list[dc];
11             unsigned r2 = list[dc1];
12
13             if (r2 > r1) {
14                 list[dc] = r2;
15                 list[dc1] = r1;
16                 fc = 1;
17             }
18             dc = dc1;
19         }
20     } while (fc == 1);
21 }
```

Belegarbeit – Agenda

Schaltung

Algorithmus

Belegarbeit

 Allgemein

 Inhalt

 Upload

Zusammenfassung

Belegarbeit

- Erstellung eines schriftlichen Beleges (Belegarbeit)
- Sprache: Deutsch oder Englisch
- Abgabe üblicherweise zum 15.03. (WS) bzw. 15.09. (SS)
- Im Semester der Abgabe: *Prüfungseinschreibung!*
- Abgabe elektronisch als .pdf Datei.
- Deckblatt und Selbstständigkeitserklärung zusätzliche *ausgedruckt und unterschrieben*
- Details auf der Webseite <https://tu-dresden.de/ing/elektrotechnik/iee/hpsn/studium/pruefungen>

Inhalt der Belegarbeit

- Beschreibung des Algorithmus, Referenzimplementierung in Software, Speicherbelegung
- Zahlenformat, Wertebereichs-/Genauigkeitsbetrachtungen
- Datenflussgraph (DFG), verschiedene Schedulingvarianten (ASAP, ALAP, Pipelining)
- Auswahl einer Realisierungsmöglichkeit
- Architektur des Datenpfades (Erstellung mittels vorgefertigter Baublöcke)
- Register-Transfer-Folgen (RT-Folgen)
- FSM Zustandsgraph und Zustandskodierung
- Verhaltensbeschreibung der FSM und der Steuerlogik in Verilog
- Simulation und Verifikation der FSM und der Steuerlogik zusammen mit dem Datenpfad
 - Abdeckung bezüglich Wertebereichs, Verzweigungen, ...
 - Verhaltensbeschreibung von Datenpfad und dessen Steuerlogik und FSM
 - P & R-Netzliste der Gesamtschaltung
- RTL- und Layoutsynthese der Gesamtschaltung, Analyse und Auswertung zu Constraints, kritischen Timingpfaden, Fläche, Utilization

Inhalt der Belegarbeit

Weitere Bewertungskriterien

- Definition des *Entwurfsziels* (z. B. wenig Hardware, schnell, ...) als Kriterium für DFG Variante und Optimierung
- Umsetzung und Optimierung im Kontext des gewählten *Entwurfsziels*
- *Verifikation*: Wie wurde sicher gestellt, dass die Implementierung korrekt funktioniert?
- Qualität der schriftlichen Darstellung (verständlich, nachvollziehbar, ...)
- Belegform: Deckblatt, Inhaltsverzeichnis, Quellenangaben, Literaturverzeichnis, Einleitung, Zusammenfassung und Auswertung, ...

Inhalt der Belegarbeit

Zusätzliche Hinweise

- Screenshots der Designtools (z. B. relevante Waveforms, finales Layout)
- Immer mit Erläuterungen (Was ist zu sehen?).
- „Interessante“ Aspekte erläutern (z. B. kritisches Timing, Verzweigungen der FSM, ...)
- Unkommentierter Quelltext als Referenz (z. B. FSM, Steuerlogik) in den Anhang der Arbeit

Upload des Belegs Über die Webseite

<https://tu-dresden.de/ing/elektrotechnik/iee/hpsn/studium/pruefungen>, Praktikumsanmeldung.

- Upload nur in der Prüfungszeit bis zum Abgabetermin.
- Nur aus TU-Netzwerk (zu Hause: VPN).
- Nur 1 Dokument kann hochgeladen werden,
- erneutes Hochladen (= Überschreiben) bis zur Abgabefrist möglich.
- Für alte Jahrgänge:
 - SSE muss registriert sein,
 - der *Login* muss nicht dem genutzten Login entsprechen oder freigeschaltet sein.

Einbettungsbox

Hinweis: Der nachfolgende Inhalt stammt von <https://hpsn.et.tu-dresden.de>.

Praktikumsregistrierung

Name: Schlußler
Vorname: Jens-Uwe
ZIH-E-Mail: Jens-Uwe.Schluessler@tu-dresden.de

Studiengang (zB. ET12): ET15

Matrikel: 12345678

Login: seje20

Veranstaltungen:

☐ PRZ (VLSI-Prozessorentwurf)
☒ SSE (Rechnergestützter Schaltkreisentwurf)
☐ OSM (Oberseminar Informationstechnik (VHDL-Entwurf, Schreiber))
☐ OSS (Oberseminar Schaltungstechnik (CCN, Prof. Ellinger))
☐ ICBC (Integrated Circuits for Broadband Communications)

SSE Beleg hochladen Keine Datei ausgewählt.

Abgabetermin 15. September 2020

Zusammenfassung – Agenda

Schaltung

Algorithmus

Belegarbeit

Zusammenfassung

Zusammenfassung und Checkliste

Jetzt bzw. innerhalb erster Semesterwochen:

- zum Praktikum anmelden
- Praktikumseinführung
- Algorithmus auswählen → anmelden

Während des Semesters:

- Algorithmus implementieren (ggf. Konsultation nutzen)
- Belegarbeit schreiben

Ende bzw. Prüfungszeit des Abgabesemesters:

- Anmelden zur Prüfung
- Belegarbeit abgeben