

Institut für Angewandte Mikroelektronik und Datentechnik

Optimierung der Verbindungsstrukturen in Digitalen Neuronalen Netzwerken

**Workshop on Biologically Inspired Methods on
Modelling and Design of Circuits and Systems**

5.10.2001 in Ilmenau, Germany

M.Haase, A.Wassatsch, D.Timmermann



Institut für Angewandte Mikroelektronik und Datentechnik

Fachbereich Elektrotechnik und Informationstechnik

Universität Rostock

Gliederung

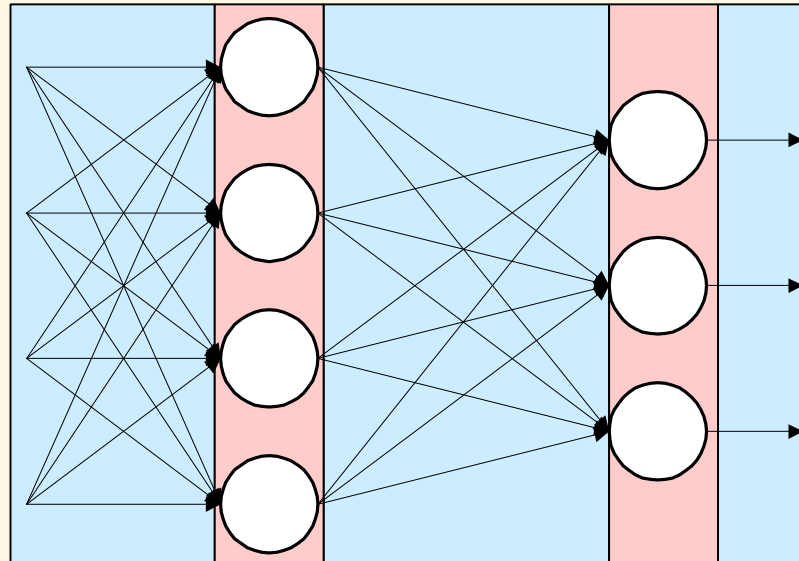
- Motivation
- Bestehende Implementierungen
- Serielle Algorithmen / Digit-Online Arithmetik
- Implementierung eines Neuronalen Netzes
- Zusammenfassung



Motivation

Bitbreite n

Fläche = $f(n)$!!!



Ansatz: serielle Algorithmen

Bestehende Implementierungen

Analog

Vorteil:

- am dichtesten am biologischen Vorbild
- Geschwindigkeit

Nachteil:

- Probleme bei der Stabilität der Gewichte (Drift), Speicherung
- nicht-lineare Multiplizierer
- Toleranzen der Bauelemente
- A/D, D/A Wandler für μ C-Steuerung

Digital-Parallel

Vorteil:

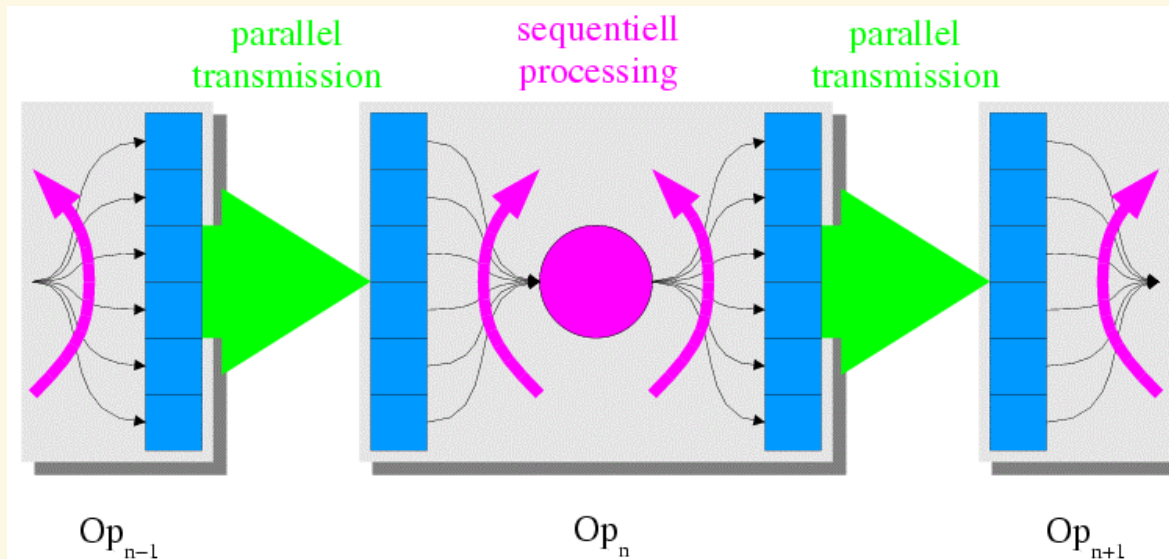
- einfache Rechnersteuerung
- Gewichtsspeicherung

Nachteil:

- Verbindungen \sim Bitbreite
- parallele Algorithmen
 - iterative Ausführung
- hoher Flächenbedarf

Parallele Datenverarbeitung

- Klassische mathematische Algorithmen der Datentechnik verarbeiten Daten parallel
- Vorteil: relativ geringe Latenzzeit
- Voraussetzung: Unabhängigkeit der Teilaufgaben



Serielle Datenverarbeitung

- zeitlich versetzte Abarbeitung von nahezu identischen Teilaufgaben führt zu einer gemeinsamen Nutzung von Schaltungsstrukturen
- kürzere Latenzzeit durch effiziente Verkettung von Grundoperationen zu einer komplexen Funktion
- geringere Chipfläche als parallele Operationen
- Voraussetzung: Redundante Zahlen



Serielle Darstellung

Last Significant Digit first

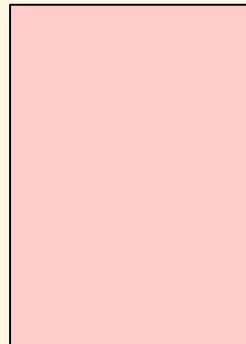
- Addition
- Multiplikation

Most Significant Digit first

- Wurzel
- Division

LSD [] [] MSD

LSD [] [] MSD



Addition

Signed Digit Darstellung

Wert	SD
0	00
1	01
-1	11
X	10

Serielle Darstellung

Last Significant Digit first

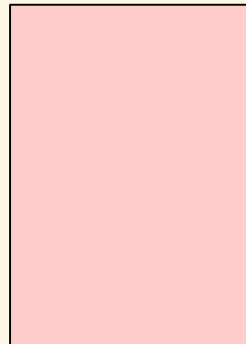
- Addition
- Multiplikation

Most Significant Digit first

- Wurzel
- Division

LSD [] [] MSD

LSD [] [] MSD



Addition

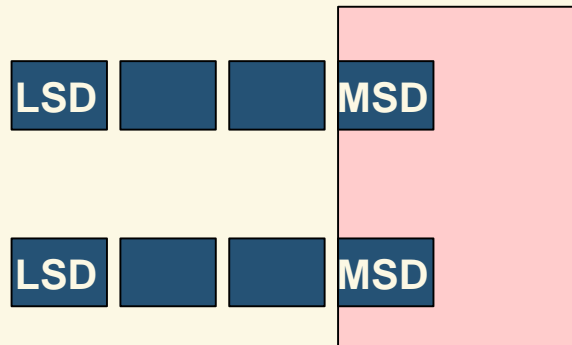
Serielle Darstellung

Last Significant Digit first

- Addition
- Multiplikation

Most Significant Digit first

- Wurzel
- Division



Addition

Serielle Darstellung

Last Significant Digit first

- Addition
- Multiplikation

Most Significant Digit first

- Wurzel
- Division



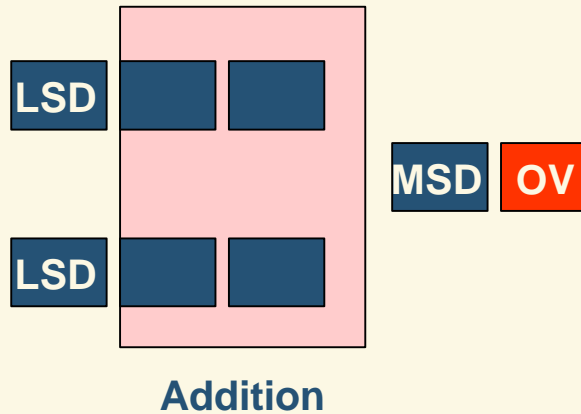
Serielle Darstellung

Last Significant Digit first

- Addition
- Multiplikation

Most Significant Digit first

- Wurzel
- Division



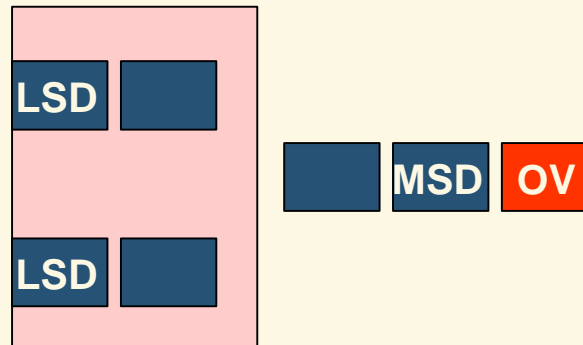
Serielle Darstellung

Last Significant Digit first

- Addition
- Multiplikation

Most Significant Digit first

- Wurzel
- Division



Addition

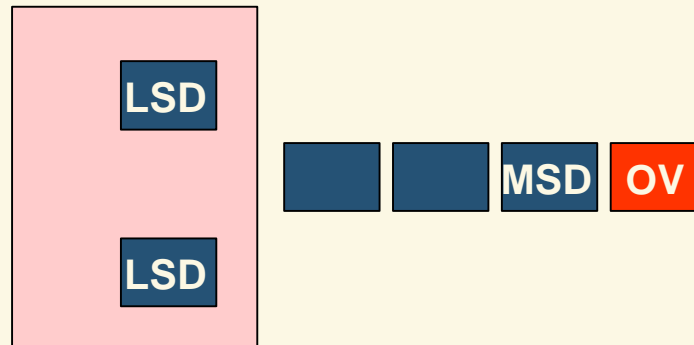
Serielle Darstellung

Last Significant Digit first

- Addition
- Multiplikation

Most Significant Digit first

- Wurzel
- Division



Addition

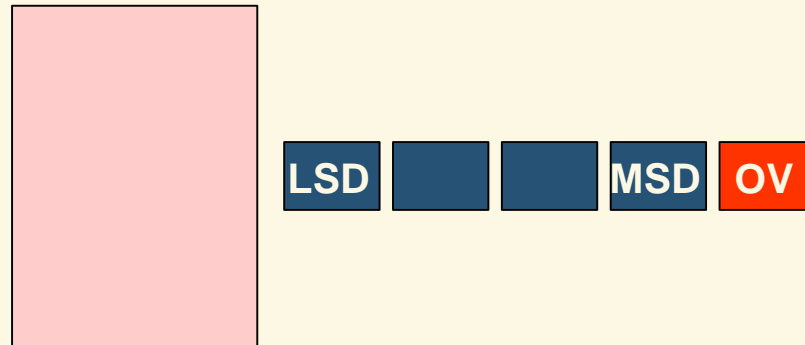
Serielle Darstellung

Last Significant Digit first

- Addition
- Multiplikation

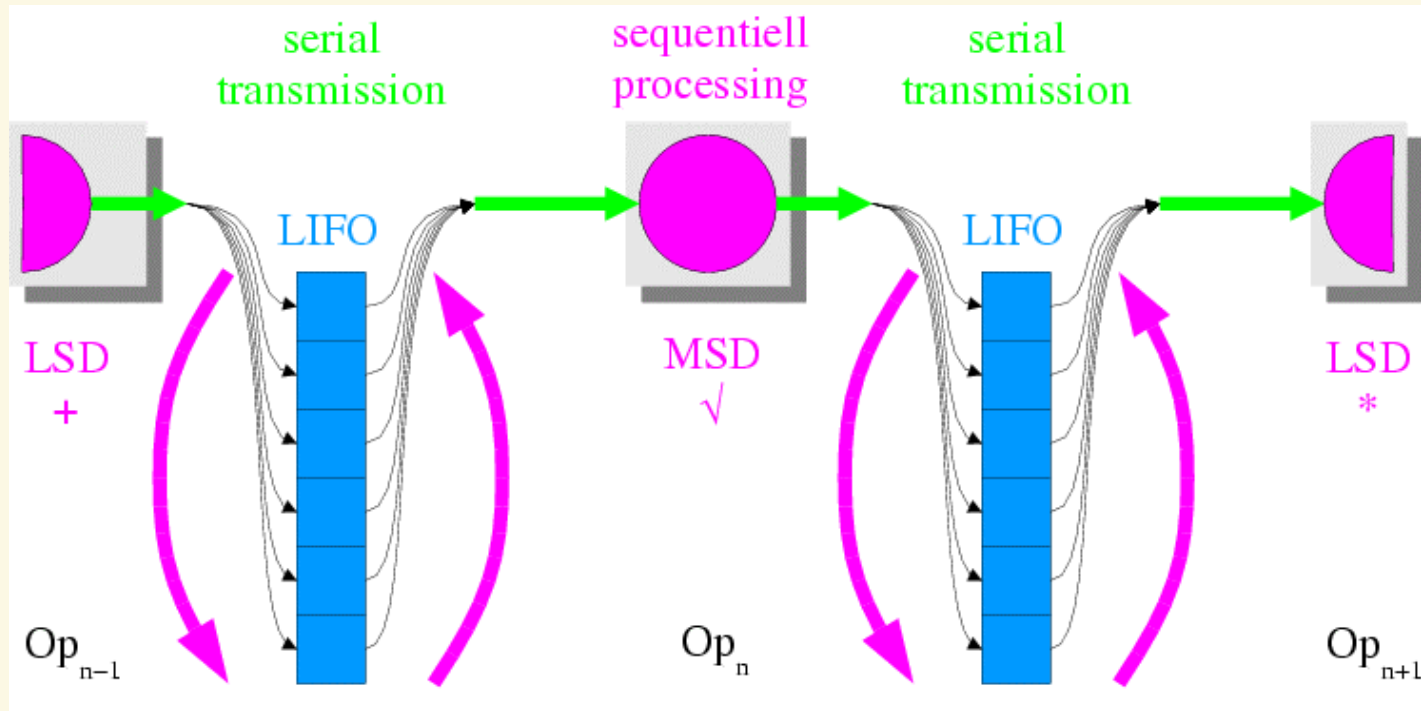
Most Significant Digit first

- Wurzel
- Division

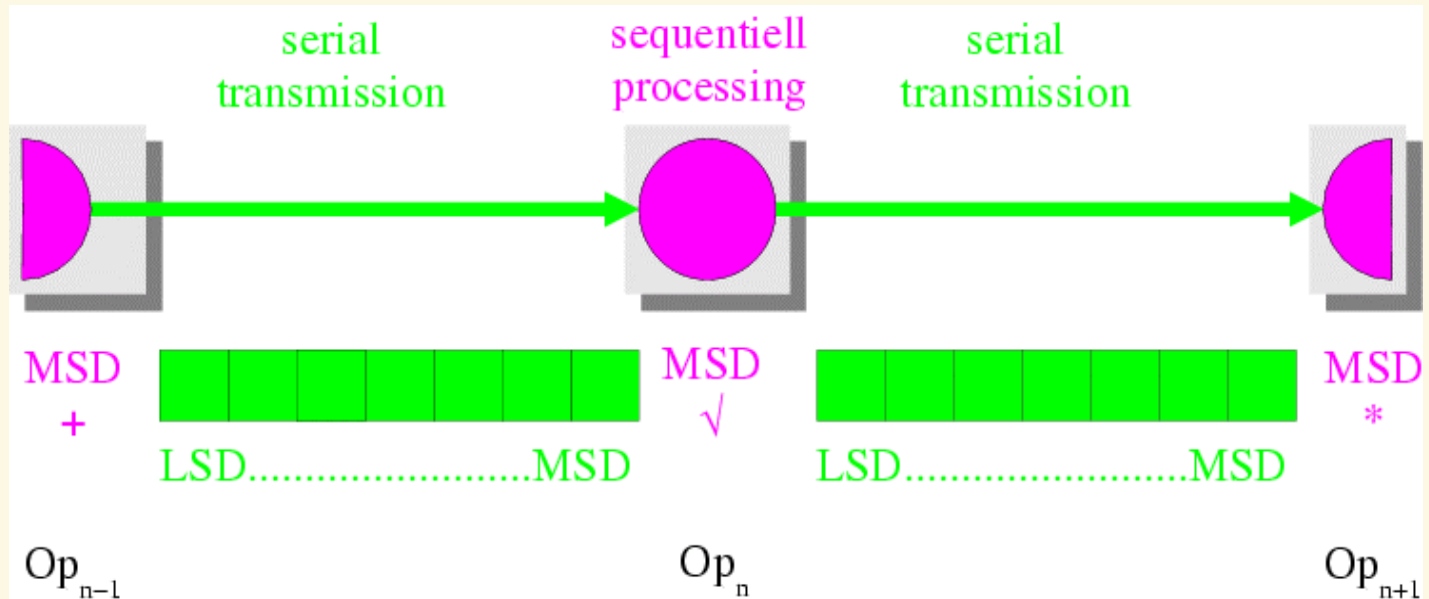


Addition

Kaskadierung von LSD/MSD Operationen



Kaskadierung von MSD Operationen



Bewertung

Vorteil

- minimaler Kommunikationsaufwand
- unabhängig von der Bitbreite der Daten
- geringer Schaltungsaufwand
- durch Kaskadierung kürzere Latenzzeit möglich

Nachteil

- Synchronisierung der seriellen Datenströme



Implementierung

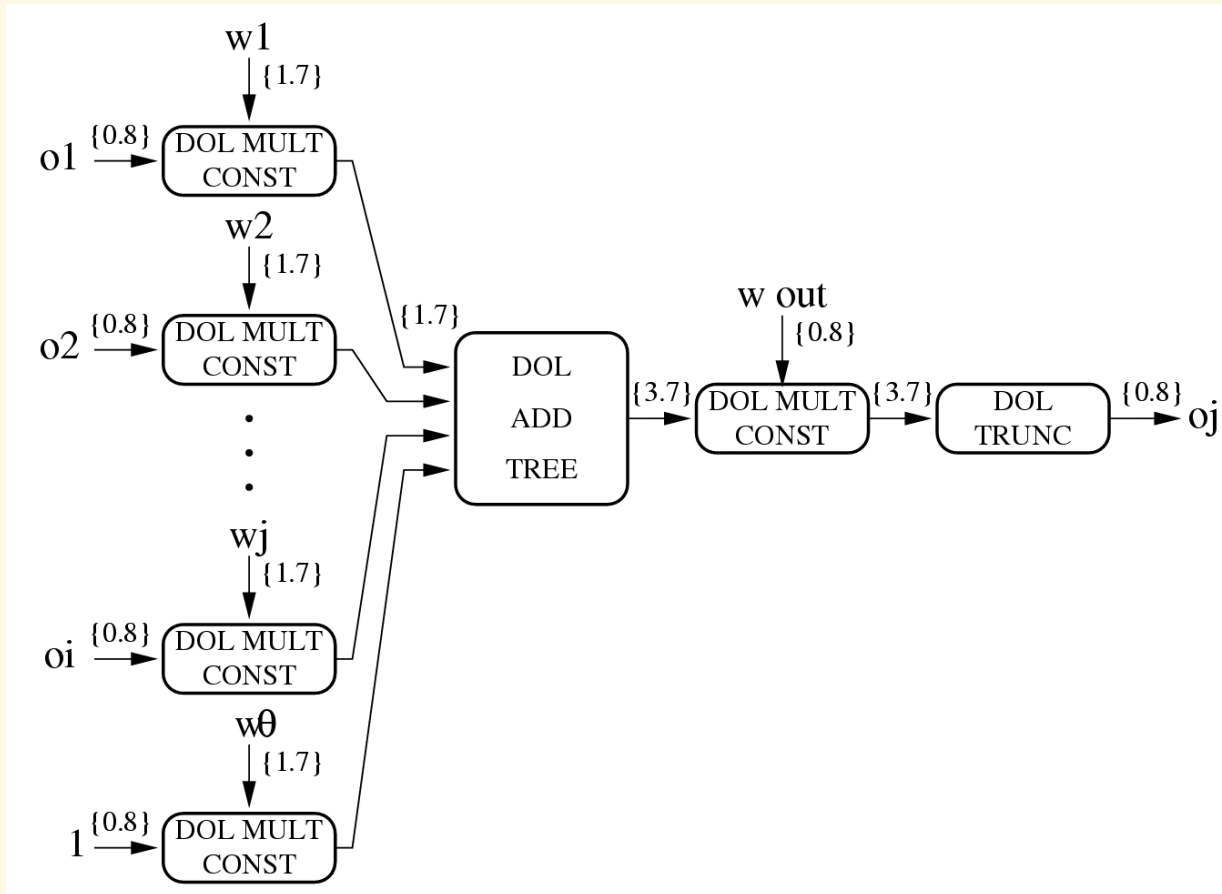
Lösung des klassischen XOR Problems

- Feed-Forward Architektur
- 2 Eingangs-, 2 Hidden-, 1 Ausgangsneuron
- Backpropagation Lernverfahren

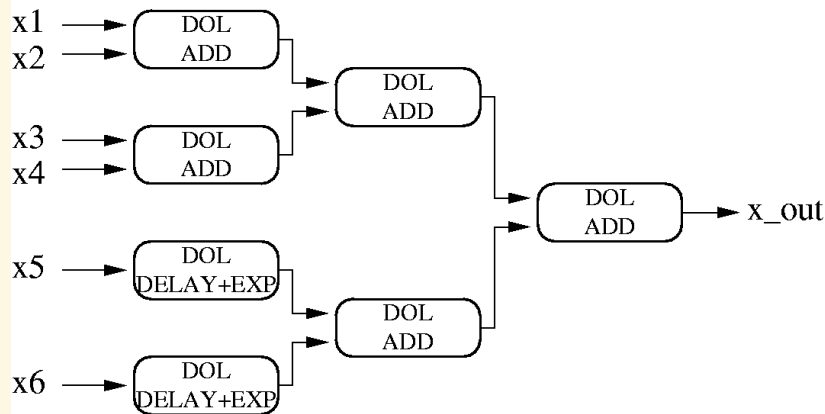
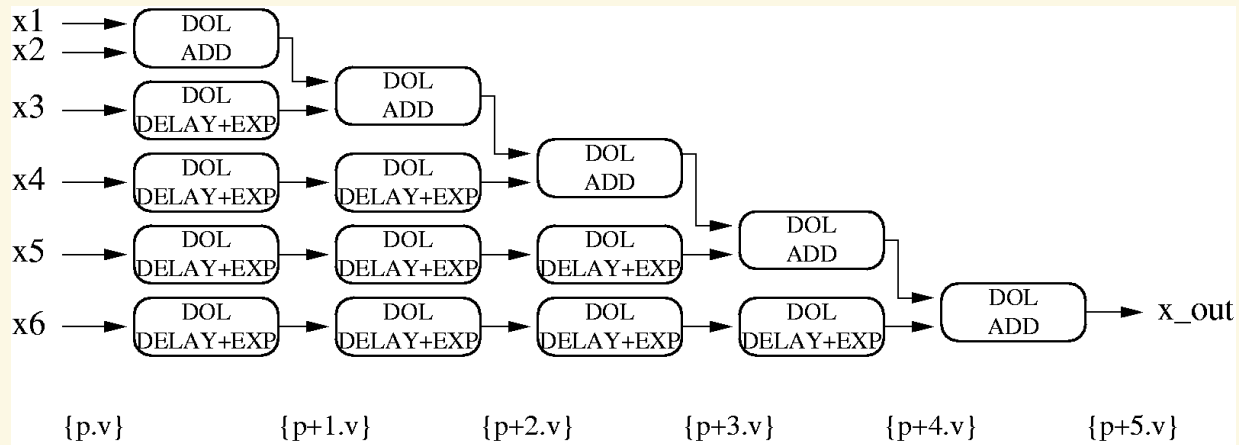
mittels serieller Algorithmen



Neuron – Digit-Online Realisierung

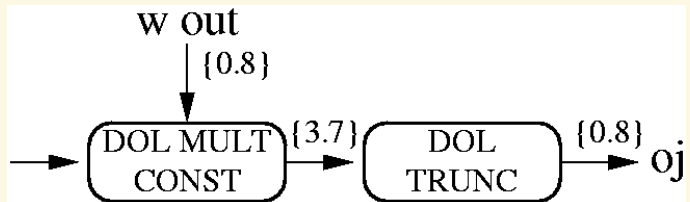
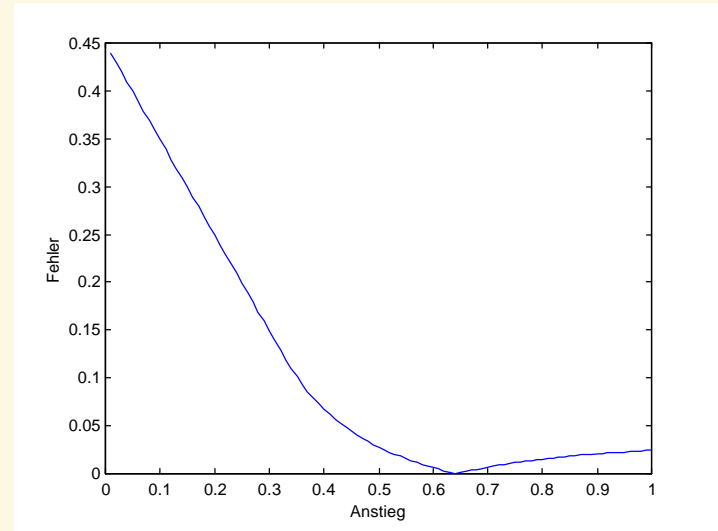
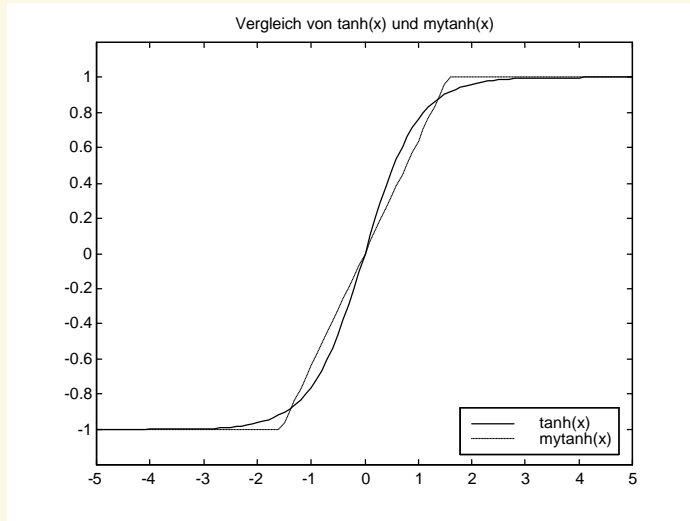


Neuron – Addierer Baum Struktur



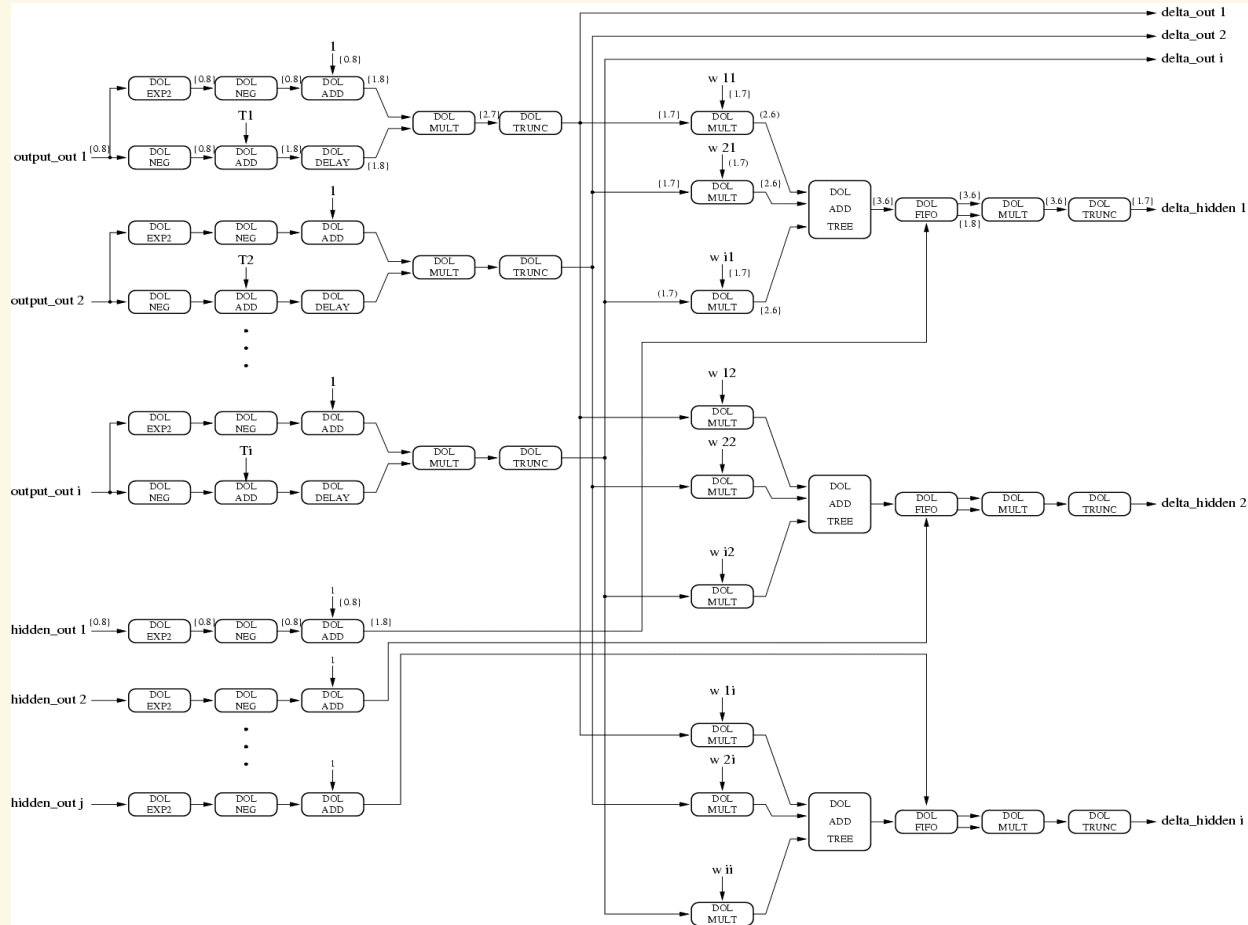
Neuron – Ausgangsfunktion

Bestimmung des optimalen Anstieges der Näherungsfunktion mytanh(x)

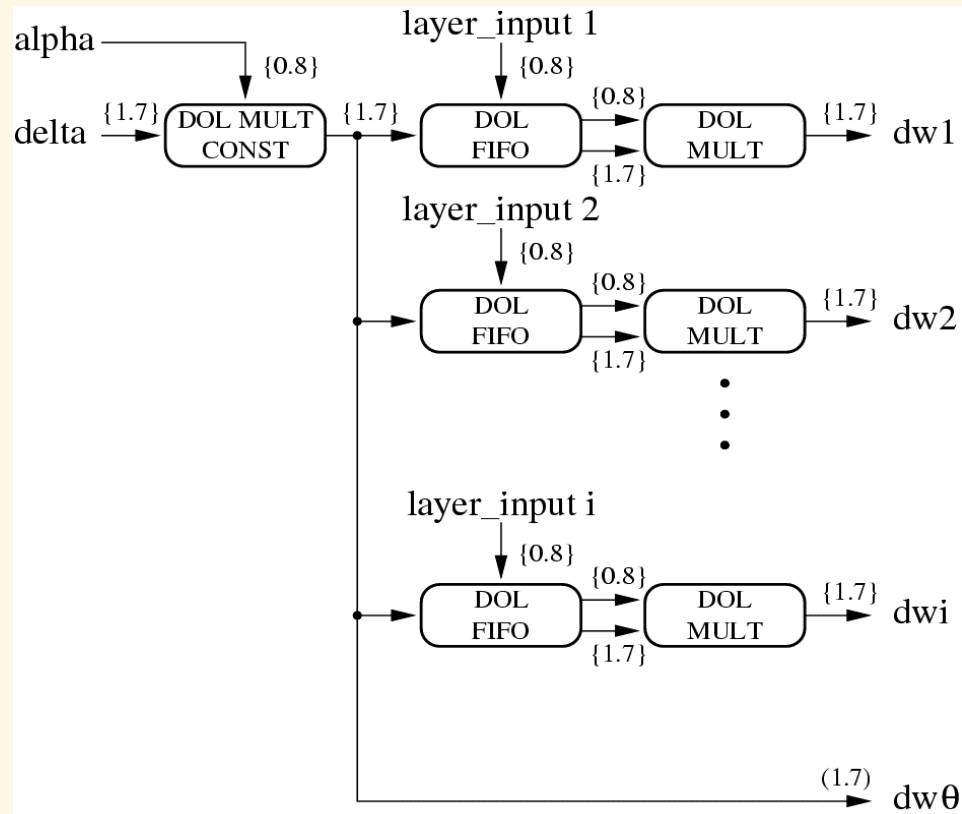


$$f'(\tanh(x)) = 1 - \tanh^2(x)$$

Backpropagation (1)

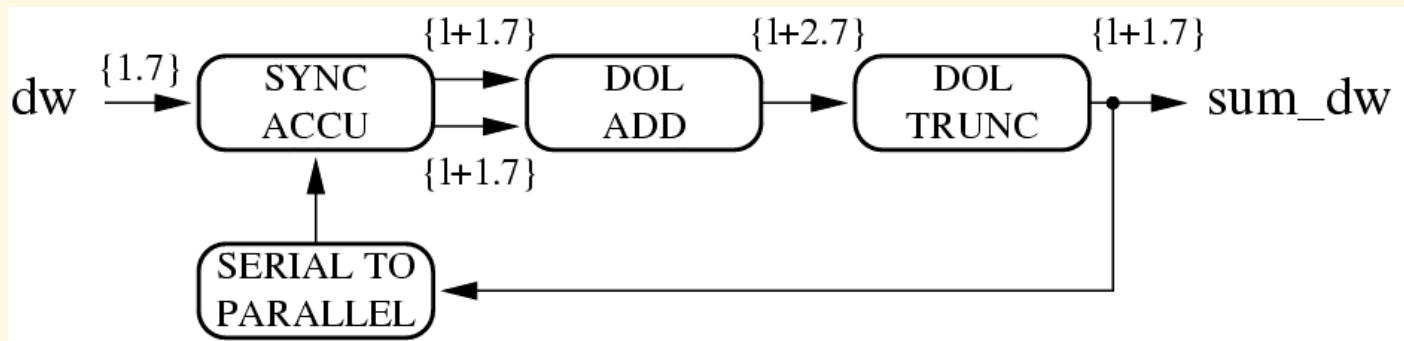


Backpropagation (2)



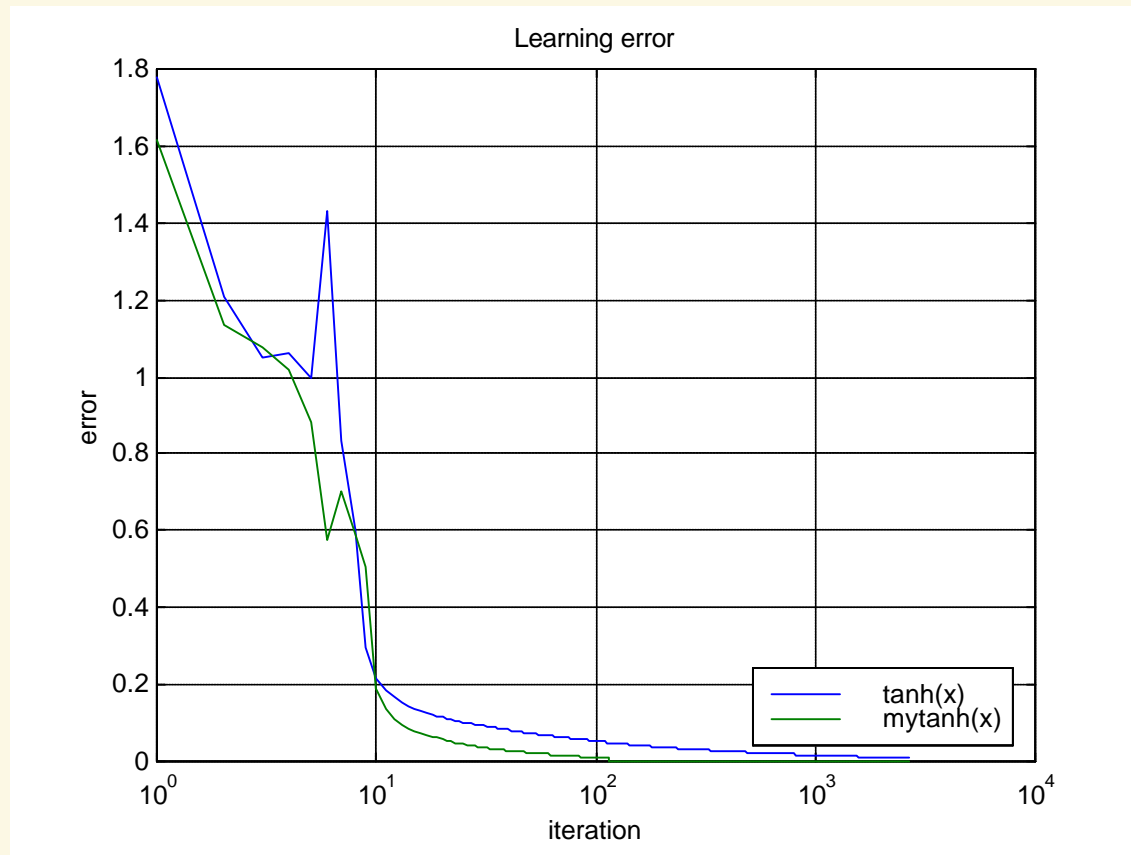
Berechnung der Gewichtsupdates

Backpropagation (3)



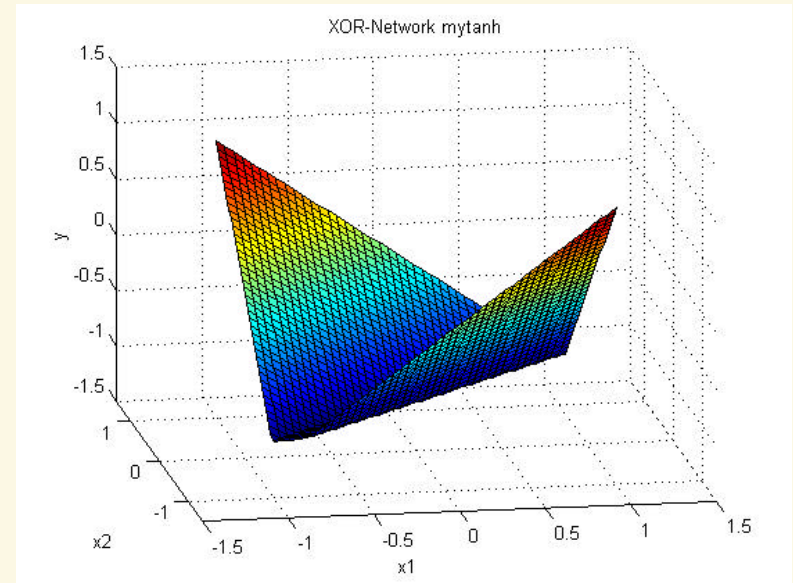
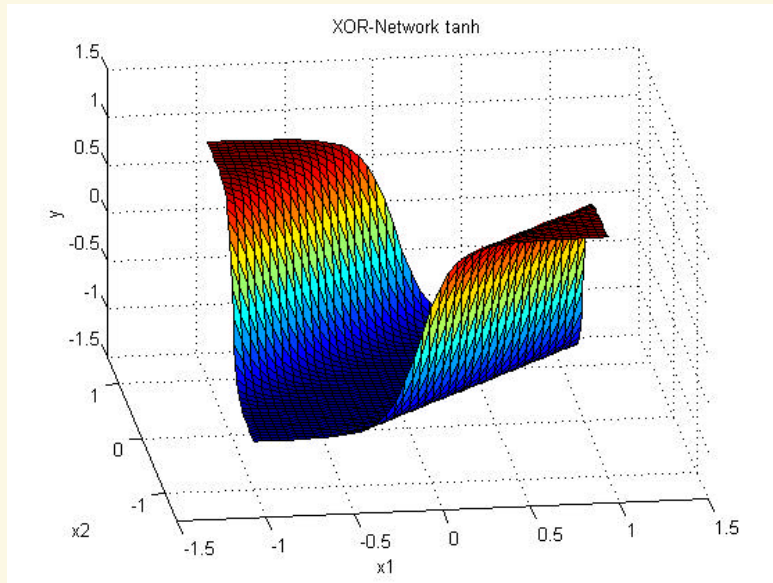
Akkumulation der Gewichtsupdates

Lernverhalten – Simulation (1)



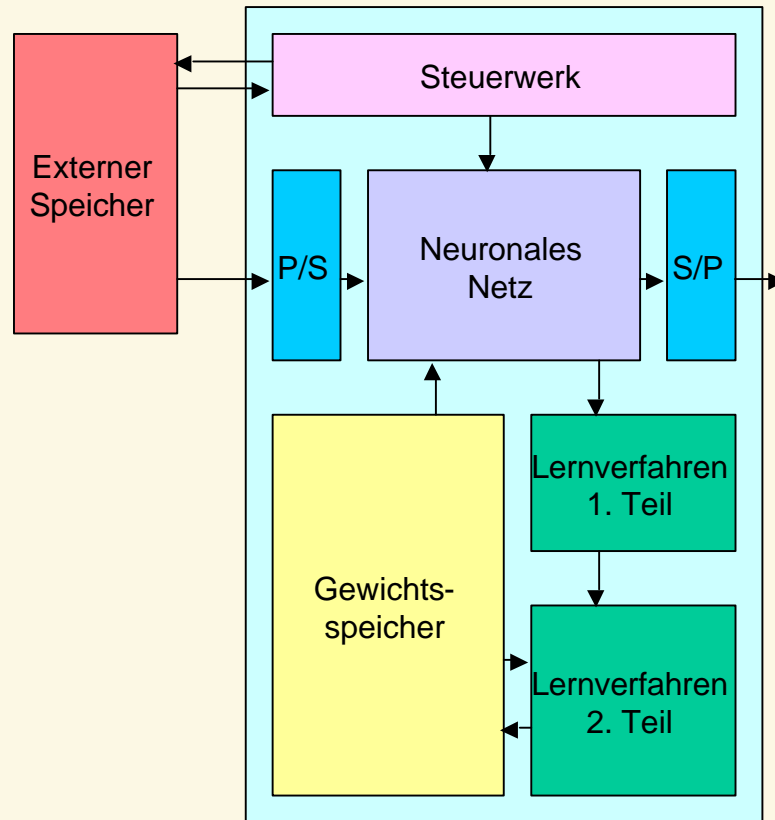
Vergleich zwischen $\tanh(x)$ und $\text{mytanh}(x)$ XOR-Problem

Lernverhalten – Simulation (2)



Ergebnis nach erfolgreichem Lernzyklus (XOR)

Aufbau des Neuro-Prozessors



Leistungsdaten des XOR-Beispiels

- 8 Bit Genauigkeit für Daten und Gewichte
- 3 Neuronen (2 hidden, 1 output)
- XILINX Virtex 1000: 64MHz Taktfrequenz

- 16 Takte je Muster in der Arbeitsphase, 38 Takte Latenzzeit
 - 4 Mill. Muster je Sek.
 - 0.25 μ s je Muster
 - 24MCPS

- 80 Takte je Muster in der Lernphase
 - 0.8 Mill. Lernmuster je Sek.
 - 1.25 μ s je Lernmuster
 - 7.2MCUPS



Entwicklungsumgebung

- MATLAB Simulationsmodell
- generische VHDL Beschreibung des Prozessors
- Software-Simulationsumgebung SYNOPSIS
- Beschleunigung durch Aptix Emulationssystem

APTIX MVP3

- Hardware-Emulations-system
- 4 XILINX Virtex 1000
- Logic-Analyser integriert



Zusammenfassung

- Minimierung der Kommunikationsstrukturen in Neuronalen Netzen
- Kaskadierung von MSD Operationen ermöglicht eine Verkürzung der Latenzzeit
- frei konfigurierbarer VHDL Core des Neuro-Prozessors mit On-Chip Lernverfahren

