Anhang A
Mathematische Herleitungen

A.1 Ableitung der Sigmoidfunktion

$$g(x) = \frac{1}{1+e^{-cx}} = (1+e^{-cx})^{-1} \tag{A.1}$$

$$g'(x) = -(1+e^{-cx})^{-2} \cdot -c \cdot e^{-cx} \tag{A.2}$$

$$= c \cdot \frac{e^{-cx}}{(1+e^{-cx}) \cdot (1+e^{-cx})} \tag{A.3}$$

$$= c \cdot \frac{e^{-cx} \cdot e^{cx}}{(1+e^{-cx}) \cdot (1+e^{-cx}) \cdot e^{cx}} \tag{A.4}$$

$$= c \cdot \frac{1}{(1+e^{-cx}) \cdot (e^{cx}+1)} \tag{A.5}$$

$$= c \cdot \frac{1}{(1+e^{-cx})} \cdot \frac{1}{(1+e^{cx})} \tag{A.6}$$

$$\boxed{g'(x) = c \cdot g(x) \cdot g(-x)} \tag{A.7}$$

A.2 Optimale Geradenannäherung der Sigmoidfunktion

Die Bestimmung der optimalen Geradensteigung und des Übergangspunktes x_1 erfolgt in zwei Schritten. Als erstes wird die Steigung der Sigmoidfunktion im Nullpunkt bestimmt, um einen Anhaltspunkt für die erforderliche Grundsteigung k der Geraden zu bekommen. Hierzu wird die im vorigen Abschnitt hergeleitete Ableitung verwendet:

$$g'(x) = c \cdot g(x) \cdot g(-x) \tag{A.8}$$

$$\Rightarrow \quad g'(0) = c \cdot g(0) \cdot g(0) \tag{A.9}$$

$$\Rightarrow \quad \boxed{g'(0) = \frac{1}{4} \cdot c = \frac{1}{k}} \tag{A.10}$$

mit

$$g(0) = \frac{1}{1+e^{0}} = \frac{1}{1+1} = \frac{1}{2} \tag{A.11}$$

Aus (A.10) ist ersichtlich, daß für die Konstante c sinnvollerweise eine Zweierpotenz zu verwenden ist, da für die Grundsteigung k nur Zweierpotenzen der Form

$$k = 2^{S-1} \quad \text{mit} \quad S = 0, 1, ..., 7 \tag{A.12}$$

zur Verfügung stehen, andernfalls ist das Ergebnis aus (A.10) entsprechend auf die nächste zulässige Zweierpotenz für k auf- oder abzurunden.

Im zweiten Schritt wird der optimale Wechsel der Geradensteigungen, d. h. x_1, bestimmt. Als Maß für den optimalen Wechsel kann die Fläche zwischen beiden Kurven

$$F(x_1) = \left| G(x_m) - Y(x_1, x_m) \right| \tag{A.13}$$

herangezogen werden. Hierzu werden sowohl die Sigmoidfunktion als auch die Geraden über ein Intervall integriert. Da die Sigmoidfunktion punktsymmetrisch zur Ordinate ist, genügt es, wenn die Integration nur über den positiven Wertebereich von x durchgeführt wird, d. h. als untere Grenze des Intervalls wird $x = 0$ gewählt. Somit ergibt sich für die Fläche unter der Sigmoidfunktion im Intervall $[0...x_m]$:

$$\int_0^{x_m} g(x) = -\frac{1}{c} \ln \frac{e^{-cx}}{1+e^{-cx}} \Bigg|_0^{x_m} \quad \text{(vgl. [Bro85] Integral Nr. 453, S. 61)} \tag{A.14}$$

$$= -\frac{1}{c} \ln \frac{e^{-cx_m}}{1+e^{-cx_m}} + \frac{1}{c} \ln \frac{e^0}{1+e^0} \tag{A.15}$$

$$= -\frac{1}{c} \ln \frac{e^{-cx_m}}{1+e^{-cx_m}} + \frac{\ln 0,5}{c} \tag{A.16}$$

$$= -\frac{1}{c} \ln \frac{e^{-cx_m}}{1+e^{-cx_m}} + \frac{\ln 0,5}{c} \tag{A.17}$$

$$\boxed{\int_0^{x_m} g(x) = x_m + \frac{\ln 0,5}{c} = G(x_m)} \tag{A.18}$$

mit

$$\ln \frac{e^{-cx_m}}{1+e^{-cx_m}} \approx -cx_m \quad \text{für} \quad cx_m >> 1 \ . \tag{A.19}$$

Für die Geraden ergeben sich drei Bereiche, in denen die Gerade jeweils eine andere Steigung aufweist:

1. Bereich $[0...x_1]$: $y_1 = \frac{x}{k} + b$ (A.20)
2. Bereich $[x_1...x_2]$: $y_2 = \frac{x}{8 \cdot k} + (b + a)$ (A.21)
3. Bereich $[x_2...x_m]$: $y_3 = 1$. (A.22)

Aufgrund der Sigmoidfunktion ist die Konstante b vorgegeben zu (vgl. (A.11))

$$\boxed{b = g(0) = \frac{1}{2}}. \tag{A.23}$$

Der Übergang von Bereich 1 nach Bereich 2 findet statt bei

$$y_1(x_1) = y_2(x_1). \tag{A.24}$$

Hieraus ergibt sich der notwendige Wert für den Parameter a in Abhängigkeit von x_1 bei gegebener Grundsteigung k:

$$\Rightarrow \quad \frac{x_1}{k} + b = \frac{x_1}{8 \cdot k} + b + a \tag{A.25}$$

$$\Rightarrow \quad a = \frac{x_1}{k} - \frac{x_1}{8 \cdot k} = \frac{x_1}{k} \cdot \left(1 - \frac{1}{8}\right) \tag{A.26}$$

$$\Rightarrow \quad \boxed{a = \frac{7}{8 \cdot k} \cdot x_1}. \tag{A.27}$$

Der Übergang von Bereich 2 nach Bereich 3 findet statt bei

$$y_2(x_2) = y_3(x_2) = 1. \tag{A.28}$$

Somit ergibt sich x_2 wie folgt:

$$\Rightarrow \quad \frac{1}{8 \cdot k} \cdot x_2 + (b + a) = 1 \tag{A.29}$$

$$\Rightarrow \quad x_2 = 8 \cdot k \cdot (1 - a - b) \tag{A.30}$$

unter Verwendung von (A.27) und Einsetzen von (A.23) ergibt sich damit:

$$\Rightarrow \quad x_2 = 8 \cdot k \cdot \left(1 - \frac{7}{8 \cdot k} \cdot x_1 - b\right) \tag{A.31}$$

$$\Rightarrow \quad x_2 = 8 \cdot k \cdot (1 - b) - 7 \cdot x_1 = 8 \cdot k \cdot \left(1 - \frac{1}{2}\right) - 7 \cdot x_1 \tag{A.32}$$

$$\Rightarrow \quad \boxed{x_2 = 4 \cdot k - 7 \cdot x_1}. \tag{A.33}$$

Somit ist der Übergangspunkt x_2 nur noch von x_1 abhängig, da k eine gewählte Konstante ist.

Die Fläche unter den drei Geraden ergibt sich zu

$$Y(x_1, x_m) = Y_1(x_1) + Y_2(x_1) + Y_3(x_1, x_m) \tag{A.34}$$

$$Y(x_1, x_m) = \int_0^{x_1} y_1(x) + \int_{x_1}^{x_2} y_2(x) + \int_{x_2}^{x_m} y_3(x) \quad \text{für} \quad 0 \le x_1 < x_m \quad \text{und} \quad x_1 < x_2 < x_m \,. \tag{A.35}$$

Die Berechnung der drei Integrale erfolgt zunächst getrennt, um die Übersichtlichkeit zu wahren. Zunächst Bereich 1:

$$\int_0^{x_1} y_1(x) = \int_0^{x_1} \frac{1}{k} \cdot x + \frac{1}{2} = \left. \frac{1}{2 \cdot k} \cdot x^2 + \frac{1}{2} \cdot x \right|_0^{x_1} \tag{A.36}$$

$$= \frac{1}{2 \cdot k} \cdot x_1^2 + \frac{1}{2} \cdot x_1 - \frac{1}{2 \cdot k} \cdot 0^2 - \frac{1}{2} \cdot 0 \tag{A.37}$$

$$\boxed{\int_0^{x_1} y_1(x) = \frac{1}{2 \cdot k} \cdot x_1^2 + \frac{1}{2} \cdot x_1}. \tag{A.38}$$

Für Bereich 2 ergibt sich:

$$\int_{x_1}^{x_2} y_2(x) = \int_{x_1}^{x_2} \frac{1}{8\cdot k}\cdot x + \frac{1}{2} + a = \left.\frac{1}{16\cdot k}\cdot x^2 + \left(\frac{1}{2}+a\right)\cdot x\right|_{x_1}^{x_2} \tag{A.39}$$

$$= \frac{1}{16\cdot k}\cdot x_2^2 + \left(\frac{1}{2}+a\right)\cdot x_2 - \frac{1}{16\cdot k}\cdot x_1^2 - \left(\frac{1}{2}+a\right)\cdot x_1 \tag{A.40}$$

einsetzen von (A.33):

$$= \frac{1}{16\cdot k}\cdot (4\cdot k - 7\cdot x_1)^2 + \left(\frac{1}{2}+a\right)\cdot(4\cdot k - 7\cdot x_1)$$
$$- \frac{1}{16\cdot k}\cdot x_1^2 - \left(\frac{1}{2}+a\right)\cdot x_1 \tag{A.41}$$

$$= \frac{1}{16\cdot k}\cdot (16\cdot k^2 - 56\cdot k\cdot x_1 + 49\cdot x_1^2) + \left(2\cdot k + 4\cdot k\cdot a - \frac{7}{2}\cdot x_1 - 7\cdot a\cdot x_1\right)$$
$$- \frac{1}{16\cdot k}\cdot x_1^2 - \left(\frac{1}{2}+a\right)\cdot x_1 \tag{A.42}$$

$$= k - \frac{7}{2}\cdot x_1 + \frac{49}{16\cdot k}\cdot x_1^2 + 2\cdot k + 4\cdot k\cdot a - \frac{7}{2}\cdot x_1 - 7\cdot a\cdot x_1$$
$$- \frac{1}{16\cdot k}\cdot x_1^2 - \left(\frac{1}{2}+a\right)\cdot x_1 \tag{A.43}$$

einsetzen von (A.27):

$$= k - \frac{7}{2}\cdot x_1 + \frac{49}{16\cdot k}\cdot x_1^2 + 2\cdot k + 4\cdot k\cdot\left(\frac{7}{8\cdot k}\cdot x_1\right) - \frac{7}{2}\cdot x_1 - 7\cdot\left(\frac{7}{8\cdot k}\cdot x_1\right)\cdot x_1$$
$$- \frac{1}{16\cdot k}\cdot x_1^2 - \left(\frac{1}{2} + \left(\frac{7}{8\cdot k}\cdot x_1\right)\right)\cdot x_1 \tag{A.44}$$

$$= k - \frac{7}{2}\cdot x_1 + \frac{49}{16\cdot k}\cdot x_1^2 + 2\cdot k + \frac{7}{2}\cdot x_1 - \frac{7}{2}\cdot x_1 - \frac{49}{8\cdot k}\cdot x_1^2$$
$$- \frac{1}{16\cdot k}\cdot x_1^2 - \frac{1}{2}\cdot x_1 - \frac{7}{8\cdot k}\cdot x_1^2 \tag{A.45}$$

$$\boxed{\int_{x_1}^{x_2} y_2(x) = -\frac{4}{k}\cdot x_1^2 - 4\cdot x_1 + 3\cdot k}. \tag{A.46}$$

Als letztes ergibt sich für Bereich 3:

$$\int_{x_2}^{x_m} y_3(x) = \int_{x_2}^{x_m} 1 = \left. x\right|_{x_2}^{x_m} = x_m - x_2 \tag{A.47}$$

einsetzen von (A.33):

$$\boxed{\int_{x_2}^{x_m} y_3(x) = x_m - 4\cdot k + 7\cdot x_1}. \tag{A.48}$$

Gleichungen (A.38), A.46) und (A.48) eingesetzt in (A.34) ergibt:

$$Y(x_1, x_m) = \left(\frac{1}{2 \cdot k} \cdot x_1^2 + \frac{1}{2} \cdot x_1\right) + \left(-\frac{4}{k} \cdot x_1^2 - 4 \cdot x_1 + 3 \cdot k\right) + (x_m - 4 \cdot k + 7 \cdot x_1) \tag{A.49}$$

$$\boxed{Y(x_1, x_m) = -\frac{7}{2 \cdot k} \cdot x_1^2 + \frac{7}{2} \cdot x_1 - k + x_m}. \tag{A.50}$$

Gleichungen (A.18) und (A.50) eingesetzt in (A.13) ergibt:

$$F(x_1) = \left| G(x_m) + \frac{7}{2 \cdot k} \cdot x_1^2 - \frac{7}{2} \cdot x_1 + k - x_m \right| \tag{A.51}$$

$$= \left| x_m + \frac{\ln 0,5}{c} + \frac{7}{2 \cdot k} \cdot x_1^2 - \frac{7}{2} \cdot x_1 + k - x_m \right| \tag{A.52}$$

$$\boxed{F(x_1) = \left| \frac{7}{2 \cdot k} \cdot x_1^2 - \frac{7}{2} \cdot x_1 + k + \frac{\ln 0,5}{c} \right|}. \tag{A.53}$$

Der optimale Wechsel der Geraden von Bereich 1 nach Bereich 2 bei x_1 ergibt sich, indem in (A.53) nach einer Nullstelle bezüglich x_1 gesucht wird. Diese ergibt sich bei ermitteltem k wie folgt:

$$0 = \frac{7}{2 \cdot k} \cdot x_1^2 - \frac{7}{2} \cdot x_1 + k + \frac{\ln 0,5}{c} \tag{A.54}$$

umformen in die Form

$$0 = x^2 + p \cdot x + q \tag{A.55}$$

ergibt

$$0 = x_1^2 - k \cdot x + \frac{2 \cdot k}{7} \cdot \left(k + \frac{\ln 0,5}{c}\right) \tag{A.56}$$

mit

$$p = -k \qquad \text{und} \qquad q = \frac{2 \cdot k}{7} \cdot \left(k + \frac{\ln 0,5}{c}\right). \tag{A.57}$$

Nach der bekannten Lösungsformel für quadratische Gleichungen (z. B. [Bro85])

$$x_{1;2} = -\frac{p}{2} \pm \sqrt{\frac{p^2}{4} - q} \tag{A.58}$$

ergibt sich als gesuchte Lösung für x_1

$$\boxed{x_1 = \frac{k}{2} - \sqrt{\frac{k^2}{4} - \frac{2 \cdot k}{7} \cdot \left(k + \frac{\ln 0,5}{c}\right)}}. \tag{A.59}$$

Sofern die Grundsteigung k direkt nach (A.10) berechnet wird, ohne auf den nächsten zulässigen Zweierpotenzwert zu runden, ergibt sich für das Verhältnis von x_1 zu k unter Verwendung von (A.10) ein konstanter Wert nach

$$x_1 = \frac{k}{2} - \sqrt{\frac{k^2}{4} - \frac{2 \cdot k}{7} \cdot \left(k + \frac{k \cdot \ln 0,5}{4}\right)} \tag{A.60}$$

$$= \frac{k}{2} - \sqrt{\frac{k^2}{4} - \frac{2 \cdot k^2}{7} \cdot \left(1 + \frac{\ln 0,5}{4}\right)} \tag{A.61}$$

$$= \frac{k}{2} - k\sqrt{\frac{1}{4} - \frac{2}{7} \cdot \left(1 + \frac{\ln 0,5}{4}\right)} \tag{A.62}$$

$$\boxed{\frac{x_1}{k} = \frac{1}{2} - \sqrt{\frac{1}{4} - \frac{2}{7} \cdot \left(1 + \frac{\ln 0,5}{4}\right)} = 0,3825} \cdot \tag{A.63}$$

Mit (A.63) hat auch der Parameter a nach (A.27) den konstanten Wert

$$\boxed{a = \frac{7}{8} \cdot \frac{x_1}{k} = 0,3347} \cdot \tag{A.64}$$

A.3 Taylorreihe

Die *Taylorsche Reihe* oder kurz Taylorreihe dient zur Potenzreihenentwicklung einer Funktion um den Punkt x_0. Für den Sonderfall $x_0 = 0$ wird sie auch *MacLaurinsche Reihe* genannt. Diese hat die Form

$$f(x) = f(0) + \frac{f'(0)}{1!} \cdot x + \frac{f''(0)}{2!} \cdot x^2 + \cdots + \frac{f^{(n)}(0)}{n!} \cdot x^n + R_n \tag{A.65}$$

mit R_n = Restglied.

Um die Reihenentwicklung für die Sigmoidfunktion nach (A.1) durchführen zu können, benötigt man neben der Ableitung der Sigmoidfunktion (A.7) auch die Ableitung der Funktion $g(-x)$. Diese ergibt sich analog zu (A.1) bis (A.7) zu

$$g'(-x) = -c \cdot g(x) \cdot g(-x)\,. \tag{A.66}$$

Nun läßt sich die Reihenentwicklung der Sigmoidfunktion nach

$$t(x) = g(0) + \frac{g'(0)}{1!} \cdot x + \frac{g''(0)}{2!} \cdot x^2 + \frac{g'''(0)}{3!} \cdot x^3 \tag{A.67}$$

durchführen, wobei die nullte bis dritte Ableitung zu bestimmen sind. Für die nullte und erste Ableitung erhält man nach (A.11) und (A.10)

$$\boxed{g(0) = \frac{1}{2}} \quad \text{und} \quad \boxed{g'(0) = \frac{1}{4} \cdot c}. \tag{A.68}$$

Für die zweite Ableitung ergibt sich nach der Produktregel

$$g''(x) = (c \cdot g(x) \cdot g(-x))' = c \cdot (c \cdot g(x) \cdot g(-x) \cdot g(-x) + g(x) \cdot -c \cdot g(x) \cdot g(-x)) \tag{A.69}$$

$$g''(x) = c^2 \cdot (g(x) \cdot g^2(-x) - g^2(x) \cdot g(-x)) \tag{A.70}$$

$$g''(0) = c^2 \cdot \left(\frac{1}{2} \cdot \frac{1}{4} - \frac{1}{4} \cdot \frac{1}{2}\right) \tag{A.71}$$

$$\boxed{g''(0) = 0} \cdot \tag{A.72}$$

Nach dem gleichen Schema ergibt sich für die dritte Ableitung

$$g'''(x) = 2 \cdot c^3 \cdot (g(x) \cdot g^3(-x) - 2 \cdot g^2(x) \cdot g^2(-x)) \tag{A.73}$$

$$\boxed{g'''(0) = -\frac{2}{16} \cdot c^3}. \tag{A.74}$$

Gleichungen (A.68), (A.72) und (A.74) eingesetzt in (A.67) ergeben die Lösung

$$\boxed{t(x) = \frac{1}{2} + \frac{1}{4} \cdot c \cdot x - \frac{1}{48} \cdot c^3 \cdot x^3}. \tag{A.75}$$

Anhang B
VHDL-Modelle des NeNEB-Chips

B.1 Design-Hierarchie

Das Design des NeNEB-Chips ist über 6 Ebenen verteilt. Die oberste Ebene kennzeichnet dabei das Toplevel-Design des NeNEB-Chips. Aus der Nummer der Ebene kann erkannt werden, wieviel-stufige Teildesigns ein Design einer Ebene hat, d. h. ein Design der Ebene 1 verwendet keine weiteren Teildesigns, ein Design der Stufe 2 verwendet 1-stufige Teildesigns, ein Design der Stufe 3 verwendet bis zu zweistufige Teildesigns, ein Design der Stufe 4 verwendet bis zu dreistufige Teildesigns usw.

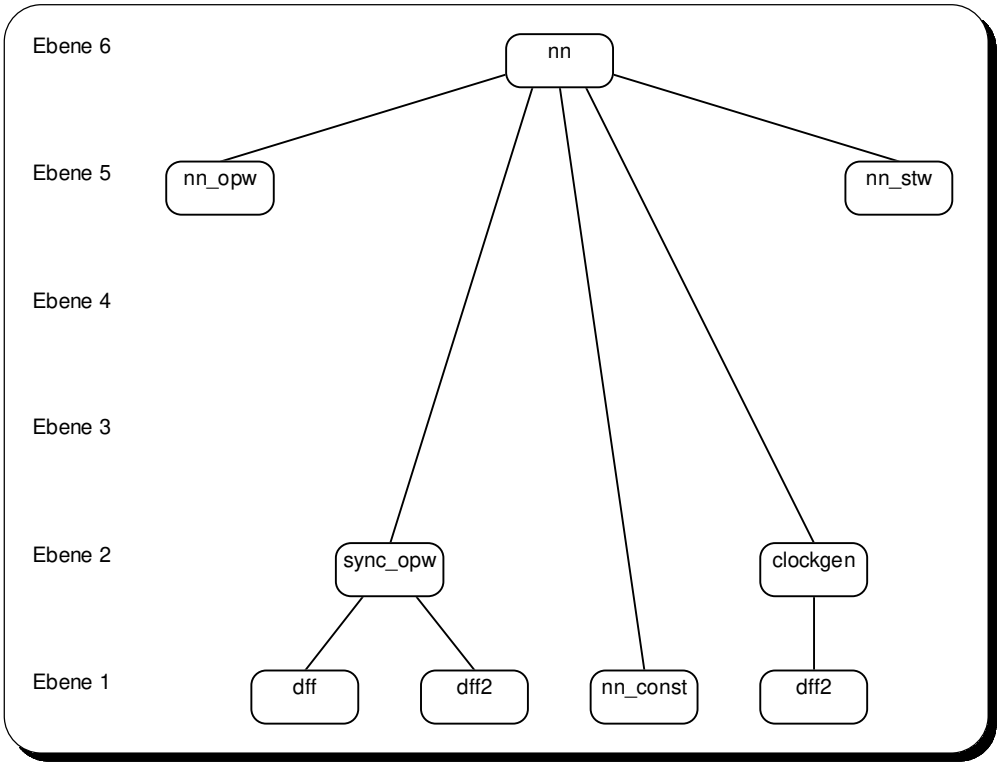

Bild B.1: Toplevel-Design des NeNEB-Chips

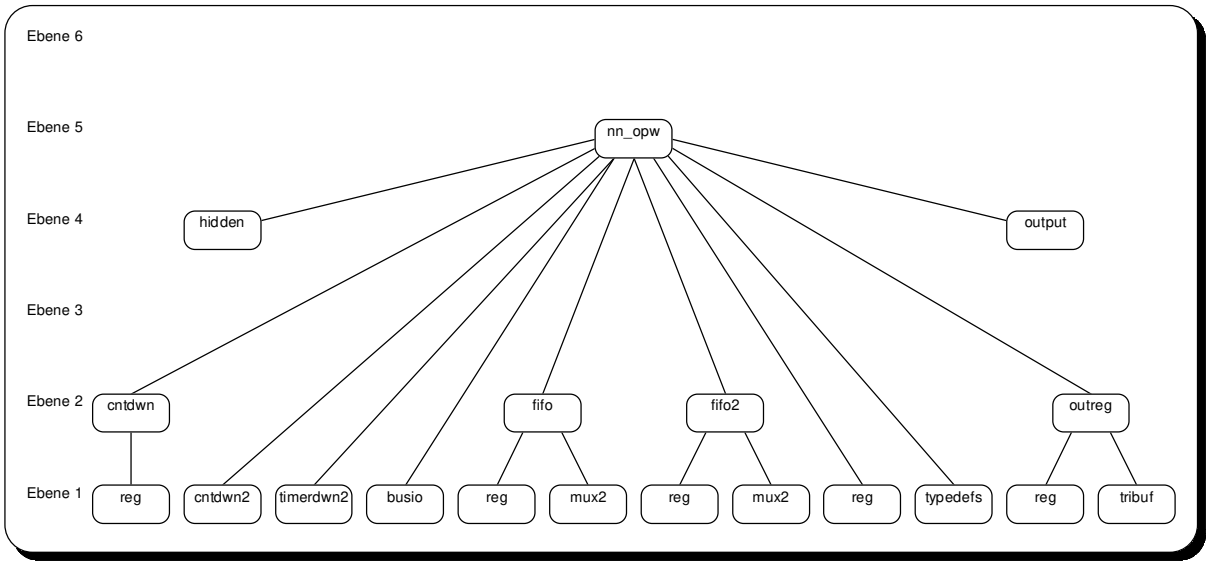

Bild B.2: Design-Hierarchie des Operationswerkes

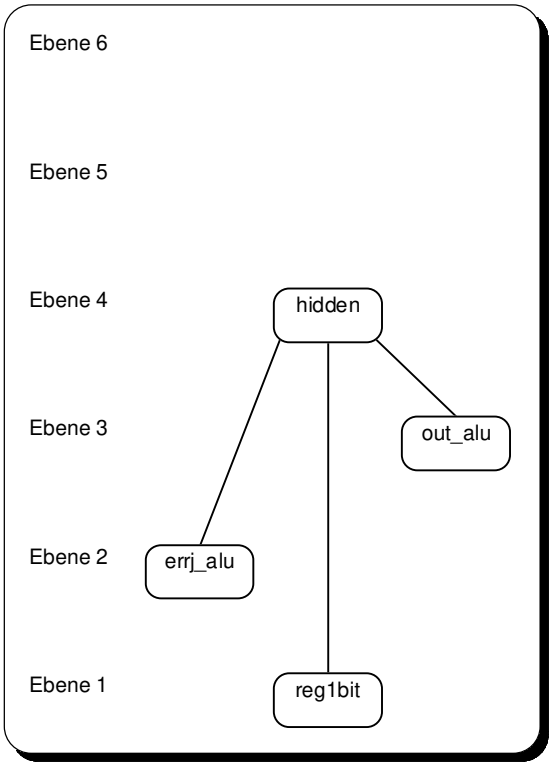

Bild B.3: Design-Hierarchie eines verborgenen Neurons

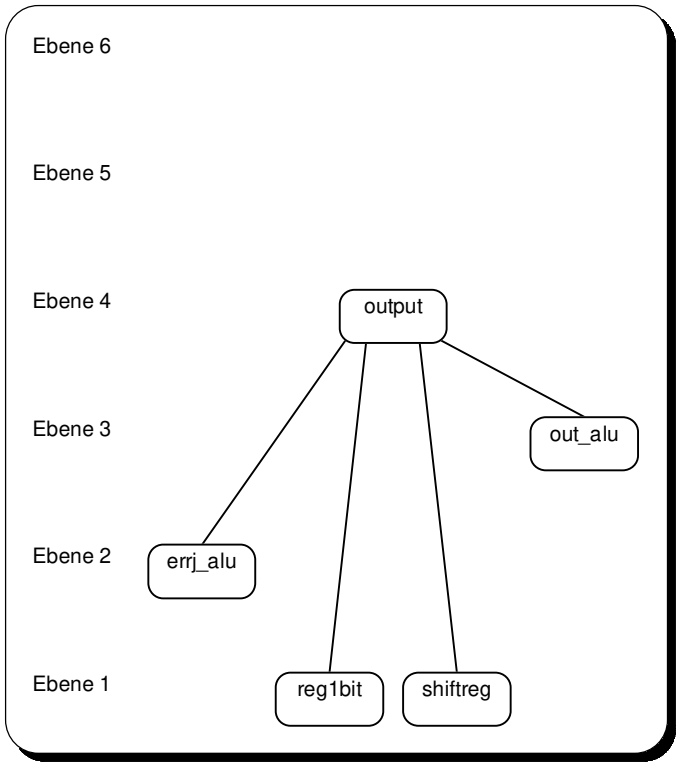

Bild B.4: Design-Hierarchie eines Ausgabeneurons

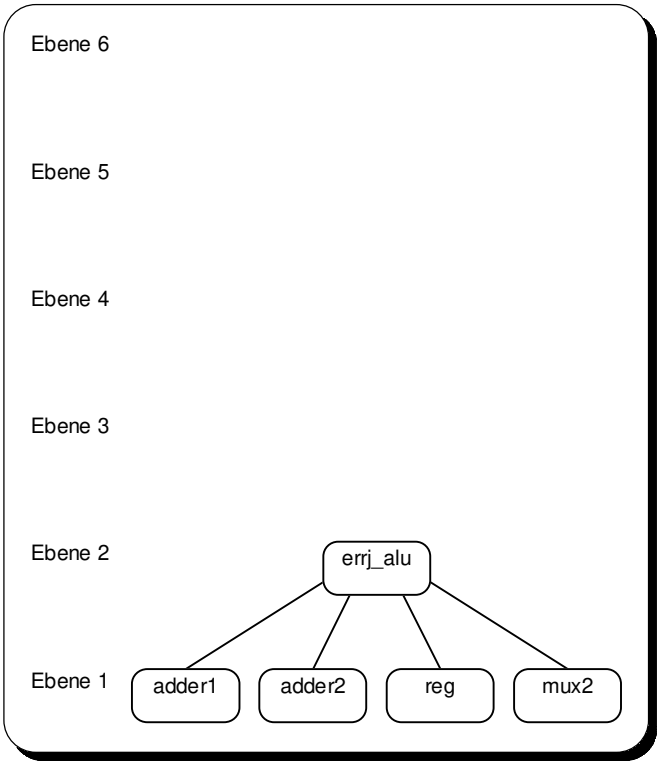

Bild B.5: Design-Hierarchie der Erregungsfunktion

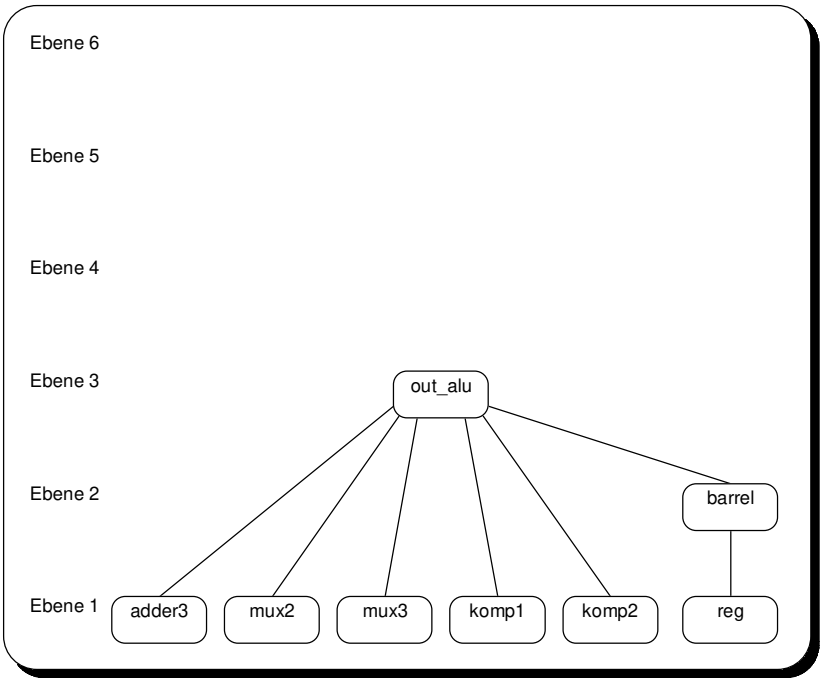

Bild B.6: Design-Hierarchie der Ausgabefunktion

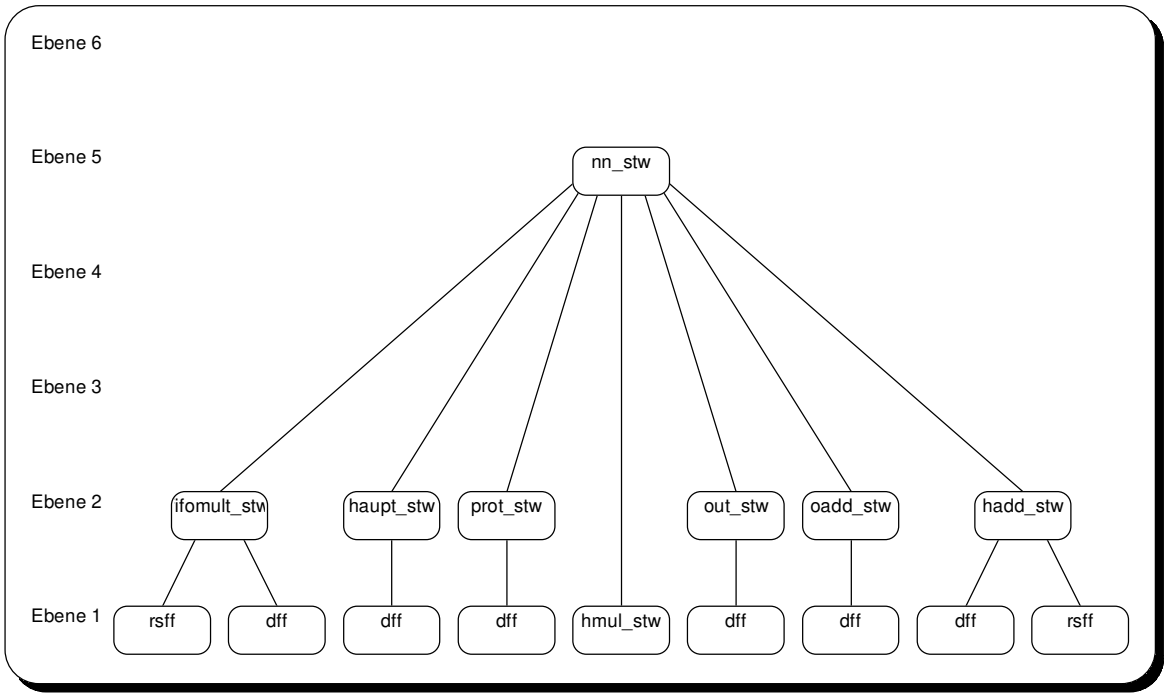

Bild B.7: Design-Hierarchie des Steuerwerkes

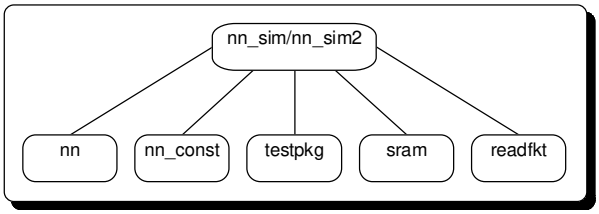

Bild B.8: Design-Hierarchie der Simulationsumgebung

B.2 Alphabetische Übersicht über die einzelnen VHDL-Modelle

- **adder1.vhd:** Addierer in Teil 1 der Erregungsfunktion
- **adder2.vhd:** Addierer in Teil 2 der Erregungsfunktion
- **adder3.vhd:** Addierer in der Aktivierungsfunktion
- **barrel.vhd:** Barrel-Schieberegister in der Aktivierungsfunktion
- **busio.vhd:** Ansteuerung für bidirektionalen Bus
- **clockgen.vhd:** Takterzeugung
- **cntdwn.vhd:** Rückwärtszähler mit Reload-Register und Zählerausgabe
- **cntdwn2.vhd:** Rückwärtszähler ohne Reload-Register mit Zählerausgabe
- **dff.vhd:** D-Flip-Flop mit synchronem Rücksetzeingang
- **dff2.vhd:** D-Flip-Flop mit zusätzlichem negiertem Ausgang
- **errj_alu.vhd:** Erregungsfunktion Teil 1 und Teil 2
- **fifo.vhd:** Serialisierer für die Ausgaben der verborgenen Neuronen
- **fifo2.vhd:** Ringspeicher für die Gewichte der Ausgabeneuronen
- **fimu_stw.vhd:** Steuerwerk Stufe 3
- **hadd_stw.vhd:** Steuerwerk Stufe 2
- **haup_stw.vhd:** Hauptsteuerwerk
- **hidden.vhd:** Berechnung eines verborgenen Neurons
- **hmul_stw.vhd:** Steuerwerk Stufe 1
- **komp1.vhd:** Komparator für positive Werte
- **komp2.vhd:** Komparator für negative Werte
- **mux2.vhd:** Multiplexer mit 2 Eingängen
- **mux3.vhd:** Multiplexer mit 3 Eingängen
- **nn.vhd:** Gesamtchip
- **nn_const.vhd:** Konstantendeklarationen für den Gesamtchip
- **nn_opw.vhd:** gesamtes Operationswerk des NeNEB-Chips
- **nn_sim.vhd:** erste Simulationsumgebung
- **nn_sim2.vhd:** zweite Simulationsumgebung
- **nn_stw.vhd:** gesamtes Steuerwerk des NeNEB-Chips
- **oadd_stw.vhd:** Steuerwerk Stufe 4
- **output.vhd:** Berechnung eines Ausgabeneurons
- **outreg.vhd:** Speicher für Ausgabewerte

- **out_alu.vhd:** Aktivierungsfunktion
- **out_stw.vhd:** Steuerwerk Stufe 5
- **pack.vhd:** Package mit Typdeklarationen
- **prot_stw.vhd:** Protokollautomat
- **readfkt.vhd:** Funktion zum Einlesen von pgm-ascii-Dateien
- **reg1bit.vhd:** 1-Bit-Register
- **register.vhd:** Register variabler Breite
- **rsff.vhd:** RS-Flip-Flop mit Setzvorrang
- **shiftreg.vhd:** Schieberegister zur Serialisierung der Gewichte der Ausgabeneuronen
- **sram.vhd:** Model eines statischen RAM
- **sync_opw.vhd:** Operationswerk zur Synchronisation der Eingangssignale
- **testpkg.vhd:** Konstantendeklarationen für die Simulationsumgebung
- **timerdn2.vhd:** Rückwärtszähler ohne Reload-Register ohne Zählerausgabe
- **timerdwn.vhd:** Rückwärtszähler mit Reload-Register ohne Zählerausgabe

B.3 Zustandsgraphen der Steuerwerksautomaten

Auf den folgenden Seiten sind die Zustandsgraphen aller Steuerwerksautomaten abgedruckt. Alle Automaten sind getaktete synchrone Moore-Automaten. Der Zustandsname steht im Symbol über dem Strich, die aktiven Ausgangssignale stehen unter dem Strich. In einem Symbol nicht aufgeführte Ausgangssignale sind inaktiv. An den Kanten zwischen den Zuständen stehen nur die jeweils ausschlaggebenden Eingangsvariablen mit ihrem relevanten Pegel. Dabei bedeutet der reine Signalname ein aktives Eingangssignal und ein Signalname mit vorangestelltem "¬" ein inaktives Eingangssignal. Der Begriff "sonst" an einer Kante bedeutet, daß diese Kante nur dann verfolgt wird, wenn keine andere mit einem Eingangssignal versehene Kante aktiv ist. Kleine Ziffern am Anfang der Kanten einiger Symbole bedeuten die Auswertungsreihenfolge der Kanten, wobei die Kante mit Nummer 1 zuerst ausgewertet wird.

B.3.1 Protokollautomat

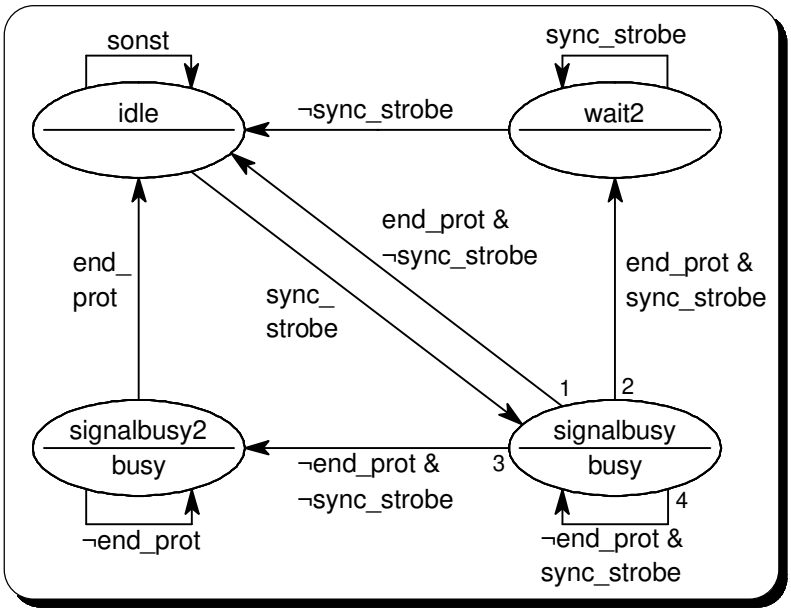

Bild B.9: Zustandsdiagramm des Protokollautomaten

B.3.2 Hauptsteuerwerk

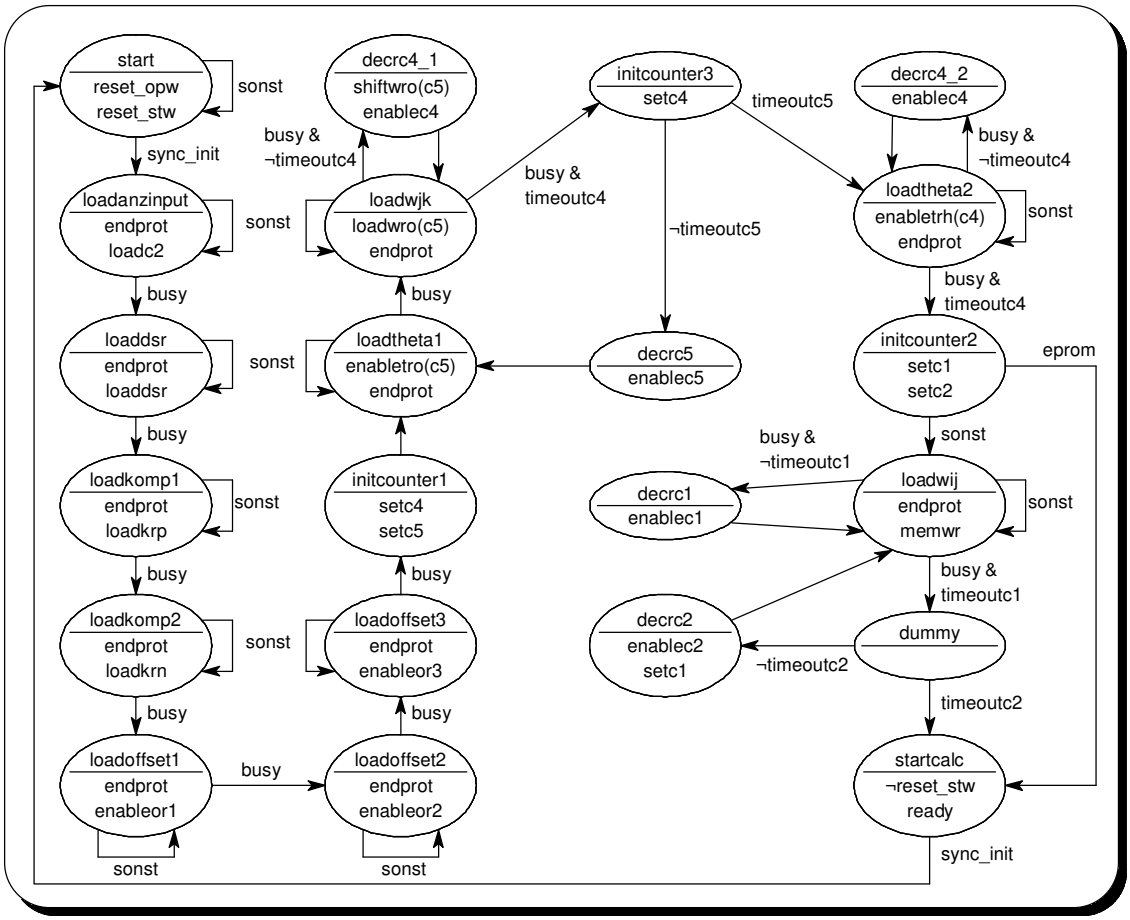

Bild B.10: Zustandsdiagramm des Hauptsteuerwerkes

B.3.3 Stufe 1

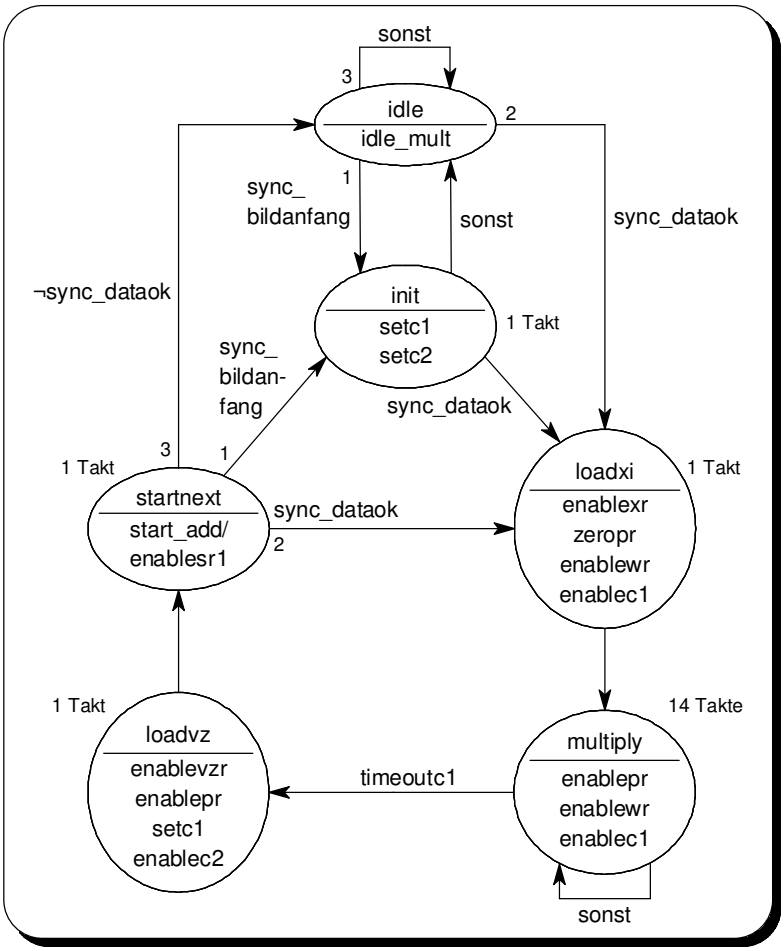

Bild B.11: Zustandsdiagramm der Stufe 1

B.3.4 Stufe 2

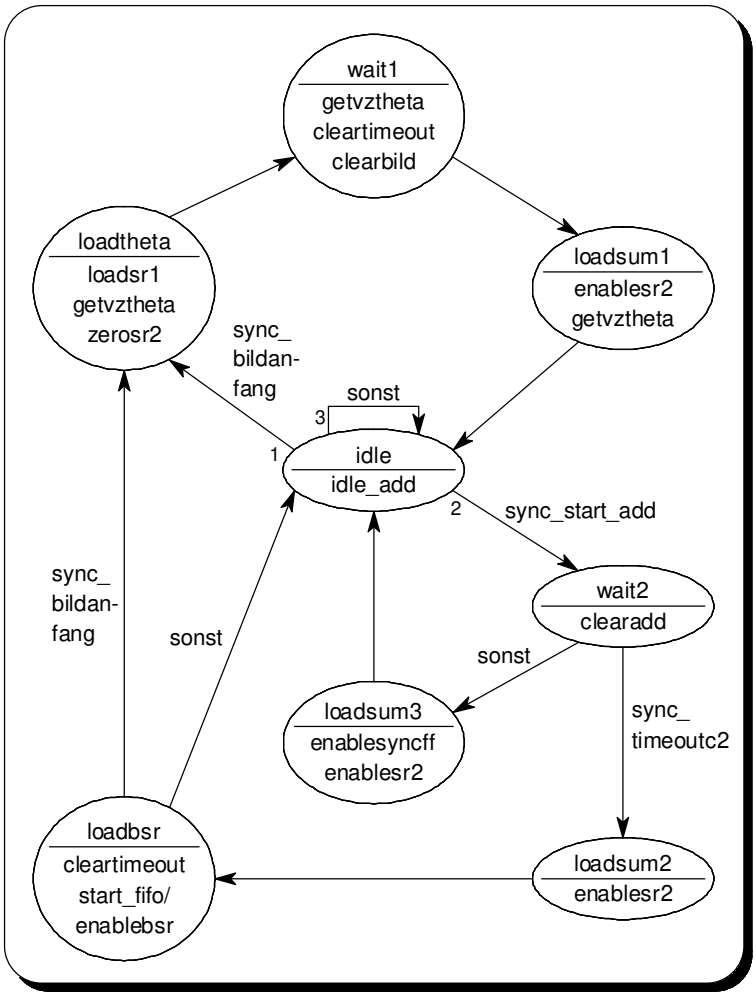

Bild B.12: Zustandsdiagramm der Stufe 2

B.3.5 Stufe 3

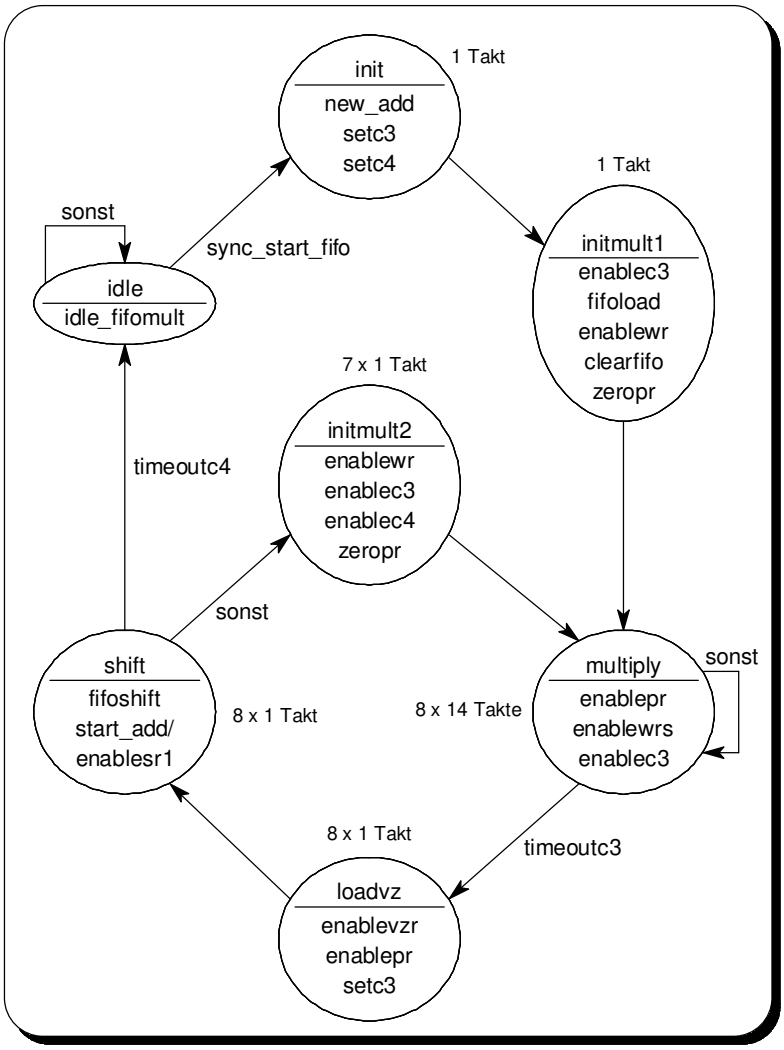

Bild B.13: Zustandsdiagramm der Stufe 3

B.3.6 Stufe 4

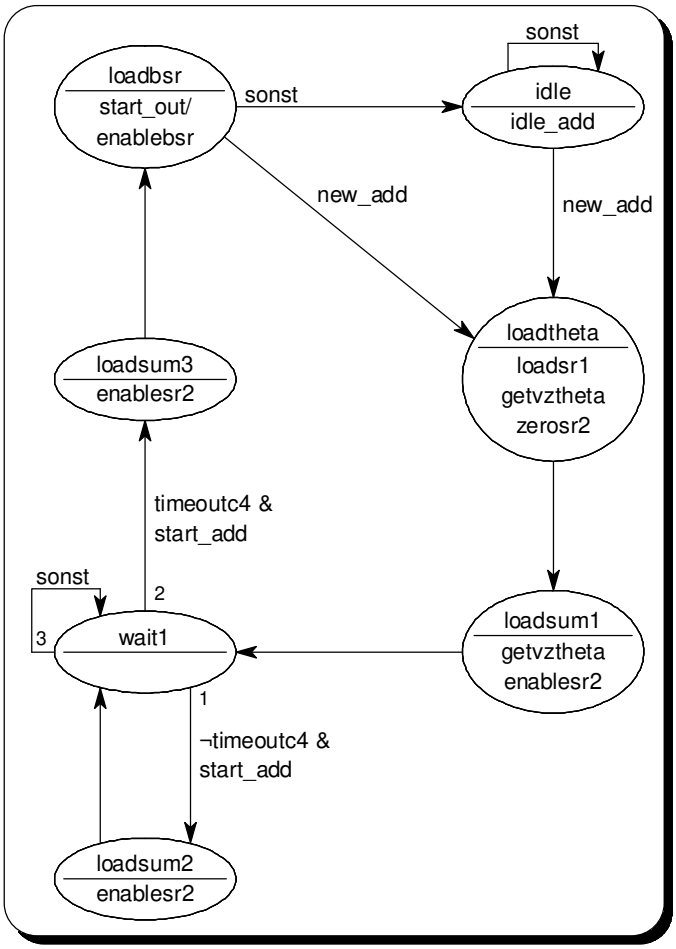

Bild B.14: Zustandsdiagramm der Stufe 4

B.3.7 Stufe 5

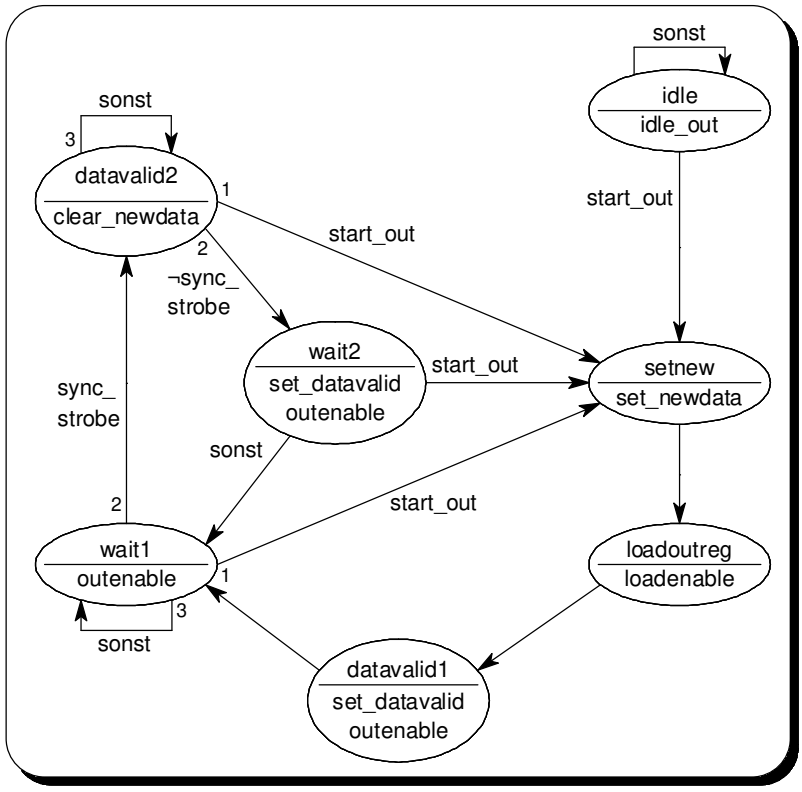

Bild B.15: Zustandsdiagramm der Stufe 5

Anhang C
AHDL-Modelle des Mosaik-FPGA

C.1 Design-Hierarchie

Das Mosaik-FPGA besteht aus zwei FPGA-Bausteinen und hat deshalb zwei Toplevel-Designs, eines für das Steuerwerk, das andere für das Operationswerk. Die weiteren von AHDL bereitgestellten Primitiven für Flip-Flops und Bustreiber sind in der Design-Hierarchie nicht enthalten.

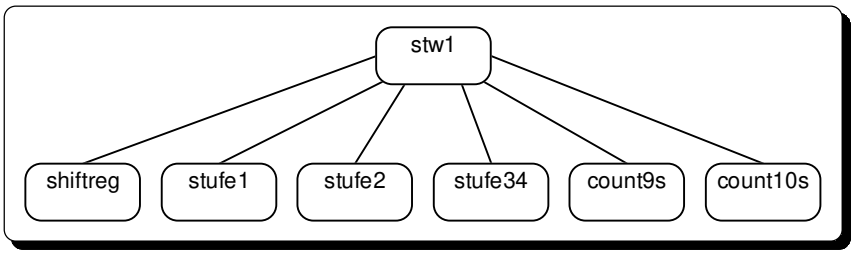

Bild C.1: Design-Hierarchie des Steuerwerkes des Mosaik-FPGAs

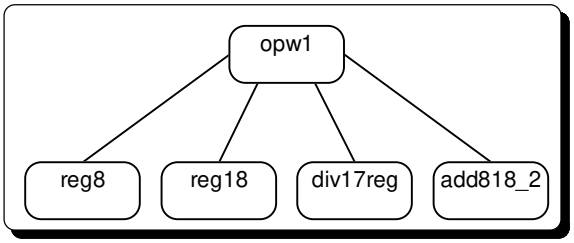

Bild C.2: Design-Hierarchie des Operationswerkes des Mosaik-FPGAs

C.2 Übersicht über die einzelnen AHDL-Modelle

- **add818_2.tdf:** Addierer mit unterschiedlich breiten Operanden
- **count10s.tdf:** 10-Bit-Rückwärtszähler ohne Zählstandsausgabe mit timeout
- **count9s.tdf:** 9-Bit-Rückwärtszähler mit Zählstandsausgabe ohne timeout
- **div17reg.tdf:** Barrel-Schieberegister mit 17-Bit-Eingang und 8-Bit-Ausgang
- **opw1.tdf:** Operationswerk Toplevel
- **reg18.tdf:** 18-Bit-Register
- **reg8.tdf:** 8-Bit-Register
- **shiftreg.tdf:** Potenzierungsregister
- **stufe1.tdf:** Stufe 1 des Steuerwerkes
- **stufe2.tdf:** Stufe 2des Steuerwerkes
- **stufe34.tdf:** Stufe 3 und 4 des Steuerwerkes
- **stw1.tdf:** Steuerwerk Toplevel

C.3 Zustandsgraphen der Steuerwerksautomaten

Für die Darstellung der Graphen gelten die Ausführungen unter Anhang B.3.

C.3.1 Stufe 1

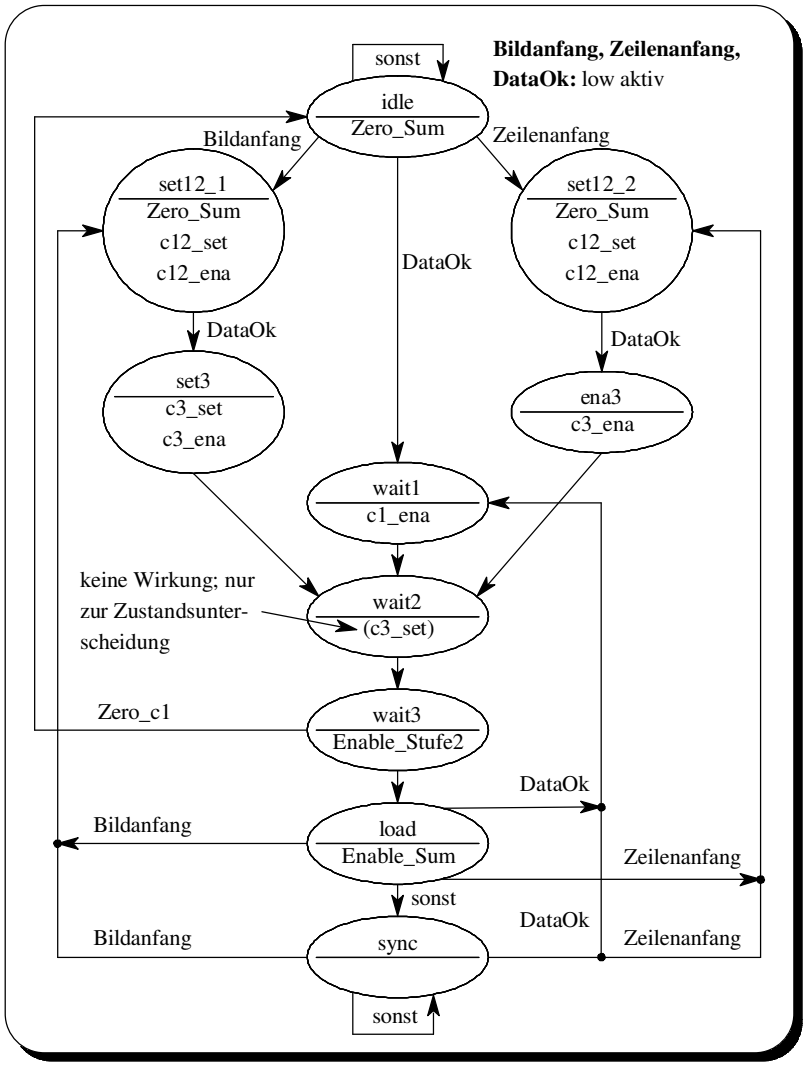

Bild C.3: Zustandsdiagramm der Stufe 1

C.3.2 Stufe 2

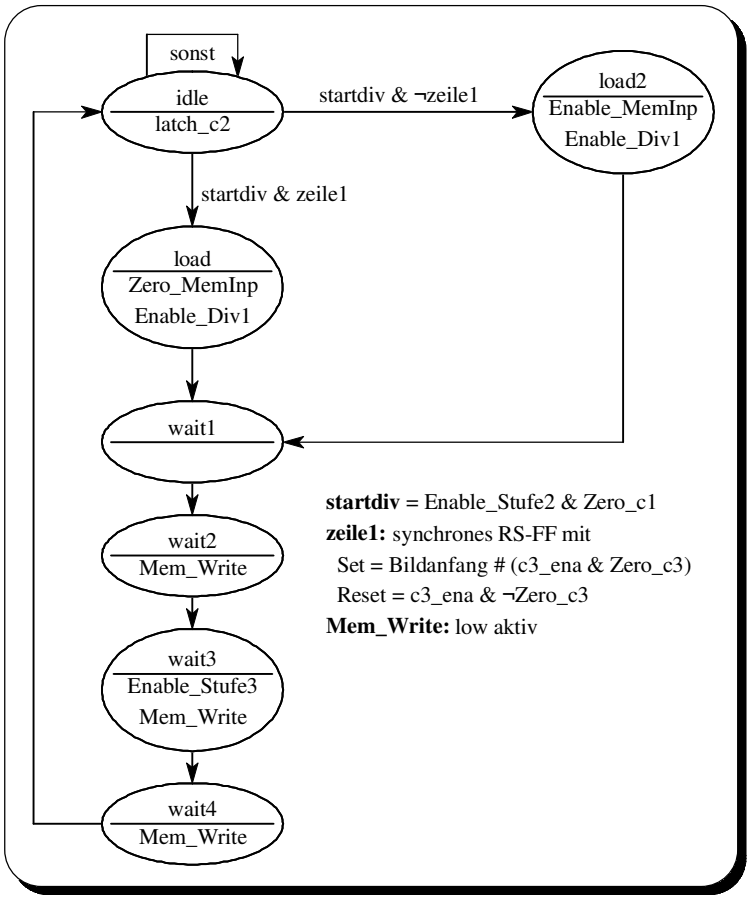

Bild C.4: Zustandsdiagramm der Stufe 2

C.3.3 Stufe 3

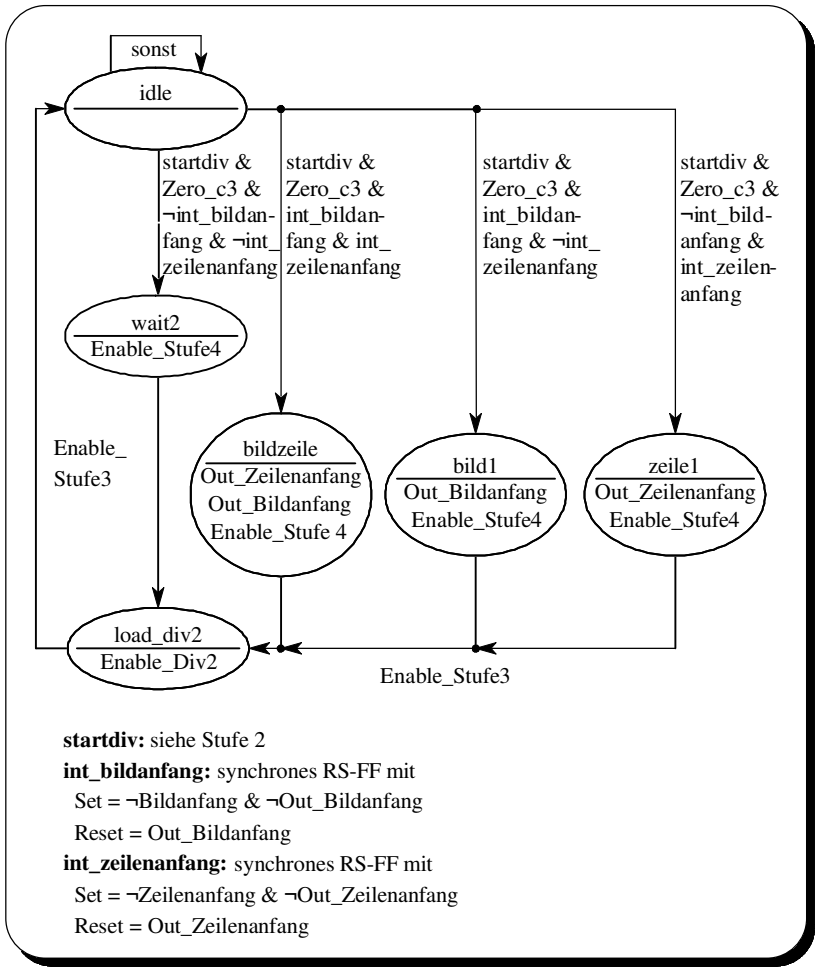

Bild C.5: Zustandsdiagramm der Stufe 3

C.3.4 Stufe 4

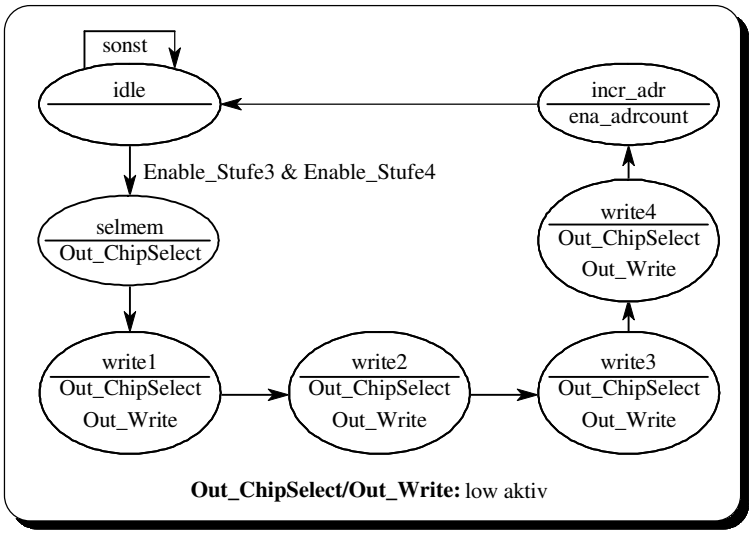

Bild C.6: Zustandsdiagramm der Stufe 4

Anhang D
Simulationsprogramme

Alle nachfolgend kurz beschriebenen Programme sind reine zweckmäßige Entwicklerversionen ohne viel Bedienungskomfort. Bei den Beschreibungen ist am Anfang stets ein Abdruck der Programmausgabe enthalten, die durch die Eingabe *"programm ?"* erfolgt.

D.1 Programme für den NeNEB-Chip

D.1.1 bpgsim

```
Simulator fuer die Recall-Phase eines BPG-Netzes.

Aufruf: bpgsim -nNetzdatei -iQuelldatei [-oAusgabedatei]
               [-wGewichtenachkommastellen] [-sSignalnachkommastellen]
               [-fFunktionsnummer]
        Quelldatei: Patterndatei (.pat)
                    Bildsteuerdatei (.rif)
                   oder Bilddatei (.raw)
        Nachkommastellen: positiv Integer (1..32, default: -w8, -s8)
                          binaere Nachkommastellen!
        Funktionsnummer: 1 - Sigmoid
                         2 - 3-Geradenannaeherung
```

Dieses Programm simuliert weitgehend den Berechnungsalgorithmus des NeNEB-Chips. Es diente bei der funktionalen Simulation des NeNEB-Chips als Referenz.

D.1.2 conv-nn

```
Konvetiert Netzlisten aus BPG in Binaerfiles fuer
NeNEB-Initialisierung

Aufruf: conv2nn -nNetzdatei -aAusgabedatei
                [-oobere_Schwelle
                 -uuntere_Schwelle
                 -mMittel_Offset
                 -kSteigung_der_Ausgabefunktion]
        Offset: positiv Real 0.00-0.99 (default -m0.5)
        Schwellen: +/- 0-0.99 (default -o0.9, -u0.1)
```

Mit diesem Programm lassen sich die Initialisierungsdateien für den NeNEB-Chip erzeugen. Alle Daten werden als 16-Bit-Werte gespeichert.

D.1.3 Aufbau der Netzdatei mit Beispielen

```
1   outputs 1
2   hiddens 2
3   inputs  2

4   TEMP= 0.2
5   CYCL= 10000
6   UPDT= 1
7   ETHA= 0.01
8   ALFA= 0.01
9   wRND= 0.5

10   -0.5   1.0 -2.0

11   -0.5   1.0  1.0
12   -1.5   1.0  1.0
```

Das vorstehende Beispiel zeigt eine vom BPG-Simulationsprogramm *bpg* erzeugte Netzdatei für das XOR-Problem. In Zeile 1 - 3 werden die Anzahl der Ausgabeneuronen, der verborgenen Neuronen und der Eingabestellen angegeben. Zeile 4 gibt die Größe der Temperaturkonstanten der Sigmoidfunktion (= 1/c) an. In Zeile 5 ist die Anzahl der Lernzyklen angegeben. Die Zeilen 7 und 8 geben die Werte für die Lernkonstante η und den Parameter α des Impulstermes an. Der Wert in Zeile 9 ist der Startwert für die synaptischen Gewichte.

Zeile 10 enthält die Gewichte für das Ausgabeneuron, wobei zuerst der Schwellwert θ steht. In den Zeilen 11 und 12 stehen die Gewichte für die verborgenen Neuronen, wobei auch hier die Schwellwerte an erster Stelle stehen. Das folgende Beispiel zeigt noch eine Netzdatei des 1-aus-8-Decoders.

```
outputs 8
hiddens 8
inputs 3

TEMP= 0.500000
CYCL= 10000
UPDT= 1
ETHA= 0.100000
ALFA= 0.200000
wRND= 0.500000

 -0.42 -2.17 -2.64  1.39 -2.02  1.72 -2.71 -0.20 -2.15
 -1.78 -2.08 -2.99 -0.32  3.00 -0.57 -1.92  0.80  1.35
 -1.13  2.34 -3.71 -0.07 -2.04  0.52  1.09 -2.52 -1.86
 -1.89  1.27 -1.78 -0.87  2.76 -1.22  0.19  1.46 -3.18
 -1.65 -1.84  1.09 -0.47 -2.10  1.22 -2.02 -3.24  2.44
 -1.00 -2.02  1.60 -0.08 -1.04 -3.81 -2.27  1.15  1.08
 -0.78 -0.84  2.12 -2.53 -2.51  0.33  1.58 -1.76 -1.94
 -1.13  0.39  1.07 -2.95  0.94 -3.82  1.97  0.34 -0.03

  0.39  1.38 -2.98 -0.50
  1.63 -3.53  0.15  0.04
 -0.81  1.62  1.74  0.13
  0.27  1.78 -0.84 -3.28
 -1.35  1.49  0.05  2.93
  1.46 -0.51 -3.06  0.37
  0.39  1.32  1.21 -2.86
  0.54 -2.25  2.83 -2.07
```

D.2 Programme für das Mosaik-FPGA

D.2.1 pcpix

```
Bildvergröberung als Pyramide mit Darstellung; default 256*256,
256 Farben

Aufruf: pcpix [Quelldatei [Zieldatei [MinHöhe [Breite Höhe]]]]
```

Mit diesem Programm wurde der im Mosaik-FPGA implementierte Auflösungsreduktions-Algorithmus überprüft. Es kann für ein angegebenes RAW-Bild interaktiv schrittweise die Auflösung reduziert und erhöht werden, wobei die Darstellung stets in gleicher Größe erfolgt.

D.2.2 mosaic

```
Bildvergröberung von default 256*256 (max. 640*480) nach 32 (min. 1)
Zeilen

Aufruf: mosaic [Quelldatei [Zieldatei [MinHöhe [Breite Höhe]]]]
```

Dieses Programm ist eine abgespeckte Version des Programms *pcpix* ohne Bildausgabe.

D.2.3 hist

```
Histogrammausgleich für 256 Farben. Mittlere 90% Pixel werden
berücksichtigt.

Aufruf: hist [Quelldatei [Zieldatei [%-Schwarz %-Weiß [Breite Höhe]]]]
```

Dieses Programm führt für ein RAW-Bild einen Histogrammausgleich nach dem bei ALVINN angewendeten Verfahren durch.

D.2.4 interla

```
  Trennt ein Bild (Default 256 Spalten) in zwei Halbbilder, wobei
  eines die geraden Zeilen, das andere die ungeraden Zeilen enthält.

Aufruf: interla [Quelldatei [Zieldatei1 Zieldatei2 [Zeilenlänge]]]
```

Dieses Programm trennt ein RAW-Bild in zwei Halbbilder, wobei das eine Halbbild die ungeraden und das andere die geraden Zeilen enthält.

D.2.5 window

```
  Schneidet ein Fenster aus einem Bild.

Aufruf:
 window [Quelldatei Zieldatei [X-Start [Y-Start [Breite Höhe [alte
Breite]]]]]
```

Mit diesem Programm kann aus einem RAW-Bild ein Teil ausgeschnitten werden.

D.2.6 split

```
Splittet eine Datei in mehrere kleine Dateien zu je n-Records.
Die Dateiextensions werden von 000 beginnend durchnummeriert.
Optional kann ein Header mit m-Records überlesen werden.
```

Mit diesem Programm kann eine Datei in mehrere neue Dateien gesplittet werden. Alle Eingaben werden interaktiv abgefragt. Dieses Programm war notwendig, um die im Arbeitsbereich KOGS digitalisierten Bilder aus einer großen Digitalisierungsdatei zu separieren.

D.2.7 strip

```
Liest die ersten n-Records einer Datei und speichert diese
unter neuem Namen ab.
```

Mit diesem Programm kann ein Teil einer Datei in einer neuen Datei gespeichert werden. Alle Eingaben werden interaktiv abgefragt.

Anhang E
Inhalt der beigelegten Diskette

A:\
ÃÄÄÄAHDL Mosaik-FPGA AHDL-Designfiles (Endung .tdf)
³ Mosaik-FPGA VHDL-Simulationsmodelle (Endung .sti)
³ Mosaik-FPGA Kommandoprozeduren (Endung .cms)
³
ÃÄÄÄBILDER Bilddateien für NeNEB-Chip (Endung .raw) für die folgenden
³ ³ Probleme
³ ÃÄÄÄBUCHSTAB Buchstabenproblem (auch als GIF-Dateien)
³ ÃÄÄÄDECODER 1-aus-8-Decoder
³ ÃÄÄÄEISENBAH Eisenbahnbilder
³ ÀÄÄÄXOR XOR-Problem
³
ÃÄÄÄBPG Netzdateien mit den Netzparametern (Endung .net),
³ ³ Initialisierungsdateien für NeNEB-Chip (Endung .bin),
³ ³ Patterndateien für Programm *bpg* (Endung .pat),
³ ³ Ausgabedateien des Programms *bpgsim* (Endung .out), für die
³ ³ folgenden Probleme
³ ÃÄÄÄBUCHSTAB Buchstabenproblem
³ ÃÄÄÄDECODER 1-aus-8-Decoder
³ ÃÄÄÄEISENBAH Eisenbahnbilder
³ ÀÄÄÄXOR XOR-Problem
³
ÃÄÄÄPROGS Simulations- und Hilfsprogramme als Turbo-Pascal 6.0 Source
³ (Endung .pas) und als ausführbare Programme (Endung .exe)
³ unter MS-DOS.
³
ÀÄÄÄVHDL NeNEB-Chip VHDL-Designfiles (Endung .vhd)
NeNEB-Chip Kommandoprozeduren (Endung .cms)

Die meisten Textdateien befinden sich auf der Diskette im MS-DOS-Format. Lediglich die Dateien im Verzeichnis VHDL und die Netzdateien liegen im UNIX-Format vor. Diese können mit dem Hilfsprogramm *lin2dos* ins MS-DOS-Format konvertiert werden. Eine Konvertierung von MS-DOS nach UNIX leistet das Hilfsprogramm *dos2lin*. Beide Programme benötigen Quell- und Zieldateiname und testen nicht das Vorhandensein der Zieldatei.

Literaturverzeichnis

[Alt93] Altera Databook, Altera Corporation, 1993

[Bor91] Handbücher zu Turbo Pascal 6.0, Borland GmbH, 2. Aufl. 1991

[Clo94] Cloutier, Jocelyn et Simard, Patrice Y.: Hardware Implementation of the Backpropagation without Multiplication, IEEE S. 46-55, 1994

[Dud86] Schüler Duden Informatik, Bibliographisches Institut, 1986

[Gun80] Gehirn und Nervensystem, Spektrum der Wissenschaft Verlagsgesellschaft, 1980

[Hah95] Hahn, Lars: Realisierung eines Echtzeit-Video-Digitalisierers unter Verwendung einer Hardware-Beschreibungssprache (AHDL, VHDL) als FPGA-Prototyp, Studienarbeit Universität Hamburg, Fachbereich Informatik, 1995

[IEE88] IEEE: Standard VHDL Language Reference Manual (IEEE Std. 1076-1988), IEEE 1988

[Jan89] Janosch, Bernd: Optische Neurocomputer, c't Magazin für Computertechnik, Ausgabe 12/89, S. 38ff, Verlag Heinz Heise

[Kli94] Klindworth, André: Einführung in AHDL, Universität Hamburg, Fachbereich Informatik, 1994

[Kol89] Kolla, Reiner: Einführung in den VLSI Entwurf, Teubner Verlag, 1989

[Lag87] Lagemann, Klaus: Rechnerstrukturen, Springer Verlag, 1987

[Lar91] Larsson, Lars: Simulation neuronaler Netzwerke und Test des Z-Kammer-Triggers am H1-Detektor, Diplomarbeit Universität Hamburg, Fachbereich Physik, 1991

[Lar93] Larsson, Lars: Hardware-Implementation Neuronaler Netzwerke (BPG), Oberseminar Vortragsfolien Universität Hamburg, Fachbereich Informatik, 1993

[Lar94] Larsson, Lars: Portierung des BackPropagation-Algorithmus auf einen massiv parallelen Rechner, Oberseminar Vortragsfolien Universität Hamburg, Fachbereich Informatik, 1993

[Mae93] Maeder, Andreas: VHDL-Kurzbeschreibung, Universität Hamburg, Fachbereich Informatik, 1993

[Mar93] Marchesi, M. et Orlando, G. et Piazza, F. et Uncini, A.: Fast Neural Networks without Multipliers, IEEE Trans. Neural Networks, Vol. 4, S. 53-62, 1993

[Pom89] Pomerleau, Dean: ALVINN: An Autonomous Land Vehicle in a Neural Network, Carnegie Mellon University, Computer Science Departement, 1989

[Pom93] Pomerleau, Dean A.: Neural Network Perception for Mobile Robot Guidance, Kluwer Academic Publishers, 1993

[Pos89] Post, H.: Entwurf und Technologie hochintegrierter Schaltungen, Teubner Verlag, 1989

[Rey91] Reyneri, L. M. et Filippi, E.: An analysis on the Performance of Silicon Implementations of the Backpropagation Algorithms for Artifical Neural Networks, IEEE Trans. Computers, Vol. 40, S. 1380-1389, 1991

[Roj93] Rojas, Raúl: Theorie der neuronalen Netze, Springer Verlag, 1993

[Ros58] Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organisation in the brain, Psychological Review 65:386-408, 1958
in: Anderson, J. A., Rosenfeld, E.: Neurocomputing: Foundations of Research, MIT Press 1988, S. 89ff

[Rum86] Rummelhart, McCelland: Parallel Distributed Processing, Volume 2, MIT Press, 1986

[Sak93] Sakaue, S. et Kohda, T. et Yamamoto, H. et Maruno, S. et Shimeki, Y.: Reduction of Required Precision Bits for Back-Propagation Applied to Pattern Recognition, IEEE Trans. Neural Networks, Vol. 4, S. 270-275, 1993

[Sam91] Referenzhandbuch, Ami Pro für Windows Version 3.0, Samna Corporation, 1991

[Sim94] Simard, Patrice Y. et Graf, Hans Peter: Backpropagation without Multiplakation, Neural Information Processing Systems, Morgan Kaufmann, Vol. 6, S. 232-239, 1994

[Spa76] Spaniol, Otto: Arithmetik in Rechnenanlagen, Teubner Verlag, 1976

[Syn93] Synopsys VHDL Simulator Reference Manual, Synopsys Inc., 1993

[Zer91] Zerbe, Klaus: Die Neuro-Chips kommen, c't Magazin für Computertechnik, Ausgabe 5/91, S. 182ff, Verlag Heinz Heise

Danksagung

Ich danke Herrn Prof. Klaus Lagemann, der mit sanftem Druck und Verständnis dafür gesorgt hat, daß diese Arbeit rechtzeitig fertiggestellt werden konnte.

Frau Prof. Bärbel Mertsching danke ich für das besondere Interesse an dieser Arbeit.

Herrn Lars Larsson danke ich für die der Arbeit zugrunde liegende interessante Themenstellung und die hervorragende Betreuung. Außerdem danke ich Ihm für das ständige thematische Korrekturlesen und die wertvollen Hinweise beim Zusammenschreiben der Arbeit.

Weiterhin möchte ich mich bei den Herren Norman Hendrich und Andreas Mäder für die Hilfen bei der Entwicklung und Simulation des VHDL-Codes bedanken.

Herrn André Klindworth danke ich für die Hilfen bei der Entwicklung des Mosaik-FPGA und für ein grandioses SRAM-VHDL-Modell, daß mir die Simulation des NeNEB-Chips enorm erleichterte.

Nicht zuletzt gilt mein Dank Herrn Georg Heßmann, der für mich in allen UNIX-Fragen erste Anlaufstelle war, sowie allen weiteren Mitarbeitern des Arbeitsbereichs TECH des Fachbereichs Informatik der Uni Hamburg, die bei Fragen stets ein offenes Ohr hatten.

Ganz besonders danke ich meiner Schwester Annegret Krol für das zeitraubende Korrekturlesen der vielen Tippfehler, sowie meiner Mutter Hannelore Krol, die mich zum Informatikstudium ermutigt hatte und mir damit bei einer schweren Entscheidung half. Ihr widme ich diese Arbeit.

Stephan Krol, Januar 1996

Erklärung

Hiermit erkläre ich, daß ich diese Diplomarbeit selbständig angefertigt und nur die angegebenen Hilfsmittel benutzt habe.

Pinneberg, den .....

.....

(Stephan Krol)