Analoge Decodierung

Pavol Hanus

Zusammenfassung — **Zuerst wird ein kurzer Überblick gegeben und die Motivation hinter der Verwendung der analogen Technik dargestellt. Die Umsetzung der Log-likelihood-Algebra mittels analoger Grundschaltungen wird erläutert. Das Prinzip der Addition von Wahrscheinlichkeiten mittels Strömen und der Log-likelihood-Verhältnisbildung mittels Spannungsdifferenzen wird erklärt. Anschließend wird auf die Zusammensetzung der analogen Basisblöcke eingegangen. Es werden Decoder betrachtet, die auf dem Graphen bzw. Trellis eines Codes basieren. Schließlich wird die Realisierung des gesamten Turbo Decoders mittels analoger Technik als Gesamtheit besprochen. Zum Schluss werden einige Realisierungsbeispiele geboten und Vorteile als auch offene Fragen auf dem Gebiet der analogen Decodierung erörtet.**

1 Einleitung

Die Welt der „digitalen“ Übertragung ist analog, d.h. kontinuierlich in der Zeit und in den Werten. Es werden keine Bits über den Luft-, bzw. optischen Kanal übertragen, sondern eher analoge Wellenfronten und diese anschließend A/D gewandelt. Obwohl die digitalen Signalprozessoren großen Fortschritt gemacht haben, stellen hohe Bitraten im Bereich von mehreren Gbit/s, kleine Chipflächen und geringe Leistungsaufnahme immernoch große Probleme dar. Natürlich werden wir weiterhin in der Zukunft digital codierte Signale (wie z.B. die „forward error correction“ (FEC) Codes) hernehmen, um ihre vorteilhafte Eigenschaften nutzen zu können. Es wäre aber sinnvoll einige Teile des Decodierprozesses in der analogen VLSI Technik zu implementieren, weil diese 10 bis 1000 mal schneller ist, wesentlich weniger Leistung verbraucht und eine kleinere Chipfläche benötigt. Die Soft-In/Soft-Out Komponentendecoder eines Turbo-Codes tauschen auch untereinander Softwerte aus. Sie verwenden also nicht nur einen analogen Eingang, sondern auch einen analogen Ausgang. Hierfür bietet es sich geradezu an, die Decodierung in analoger VLSI Technik ganz ohne Algorithmen und ohne getaktete Prozessoren in parallelen, nichtlinearen und rückgekoppelten Schaltkreisen durchzuführen.

Der erste Vorschlag zur analogen Decodierung mittels der Log-Likelihood-Algebra stammt von Hagenauer [Hag98]. Als Grundbaustein für wahrscheinlichkeitsbasierende Decodierung am Trellis den Gilbert Multiplikator zu verwenden, schlug erstmals Loeliger [Loe99] vor. Anschließend wurden einige analoge Decoder implementiert [Lus99], [Moe00], [Win01].

2 Grundlagen

Am Eingang eines Soft-In/Soft-Out Decoders muss die Information über den aktuellen Kanal und die a priori Information zusammengeführt werden. Verwendet man bei der Darstellung die Log-Likelihood-Algebra, so entspricht diese Zusammenführung einer Gewichtung des Ausgangs des Matched Filter y durch die Kanalzustandsinformation L_C und anschließender Addition der a priori Information $L(x)$. Für das a posteriori Log-Likelihood-Verhältnis $L(x|y)$ gilt dann:

$$L(x|y) = L_C y + L(x). \tag{1}$$

Die Kanalzustandsinformation hängt nur vom Kanal ab und lässt sich vom Kanalmodell ableiten. Für einen Gauß'schen Fading-Kanal gilt z.B.:

$$L_C = 4a\frac{E_S}{N_0}. \tag{2}$$

Dabei entspricht a der Fading-Amplitude und $\frac{E_S}{N_0}$ dem Signal zu Rausch Verhältnis.

Im Falle einer binären Zufallsvariable X gilt für das Log-Likelihood-Verhältnis $L(X)$:

$$L(X) = \ln \frac{P_X(x=0)}{P_X(x=1)}. \tag{3}$$

Die Wahrscheinlichkeiten $P_X(x=0)$ und $P_X(x=1)$ sind dabei gegeben durch:

$$P_X(x=0) = \frac{1}{1+e^{-L(X)}}. \tag{4}$$

und

$$P_X(x=1) = \frac{e^{-L(X)}}{1+e^{-L(X)}}. \tag{5}$$

Für den Erwartungswert $E(X)$, auch als Softbit mit $\lambda(X)$ bezeichnet, erhält man:

$$E(X) = \lambda(X) = (+1)P_X(x=0) + (-1)P_X(x=1) = tanh(L(X)/2). \tag{6}$$

Grundbausteine eines analogen Decoders sind bipolare Transistoren im Vorwärtsbetrieb (Bild1) und bipolare Transistoren in Diodenschaltung (Bild2), die logarithmische und exponentielle Verläufe nachbilden.

In Gleichung (9) entspricht I_C dem Kollektorstrom, I_S dem Sättigungsstrom, V_{BE} der Basis-Emitterspannung und V_T der Temperaturkonstanten (≈ 26 mV bei 300 °K).

Verschaltet man die bipolar Transistoren zu einem Transistorenpaar, wie in Bild3 links gezeigt, so erhält man für I_0 und I_1 folgende Gleichungen:

$$I_0 = I\frac{1}{1+e^{-\frac{\Delta V}{V_T}}} \tag{7}$$

und

$$I_1 = I\frac{e^{-\frac{\Delta V}{V_T}}}{1+e^{-\frac{\Delta V}{V_T}}}. \tag{8}$$

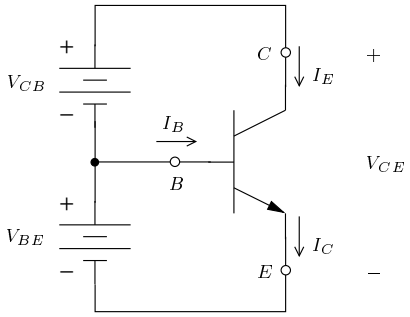

Bild 1: Bipolarer Transistor im Vorwärtsbetrieb.

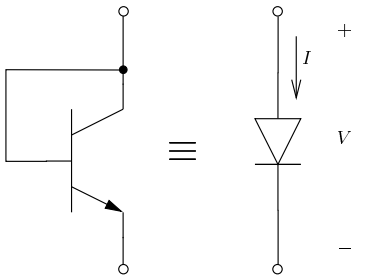

Bild 2: Bipolarer Transistor - Diodenschaltung.

$$I_C = I_S(e^{\frac{V_{BE}}{V_T}} - 1) \tag{9}$$

$$V = V_T \ln \frac{I}{I_S} \tag{10}$$

Aus einem Vergleich mit den Gleichungen (4) und (5) lässt sich schließen, dass I_0/I der Wahrscheinlichkeit $P_X(x = 0)$ und und I_1/I der Wahrscheinlichkeit $P_X(x = 1)$ entspricht. Außerdem muss $\Delta V/V_T$ gleich $L(X)$ sein und $\Delta I/I = \tanh(\Delta V/V)$ dem Softbit $\lambda(X)$ entsprechen. Das linke Transistorenpaar ermöglicht somit die Umwandlung eines Log-Likelihood-Verhältnisses (dargestellt als differentielle Spannung) in Wahrscheinlichkeiten (dargestellt durch Ströme). Das rechte Transistorenpaar aus Transistoren in Diodenschaltung bewerkstelligt die Umkehrfunktion, nämlich eine Umwandlung von Wahrscheinlichkeiten (Ströme) in ein Log-Likelihood-Verhältnis (differentielle Spannung). Eine allgemeine Beschreibung für den nichtbinären Fall kann in [Hag00] nachgelesen werden.

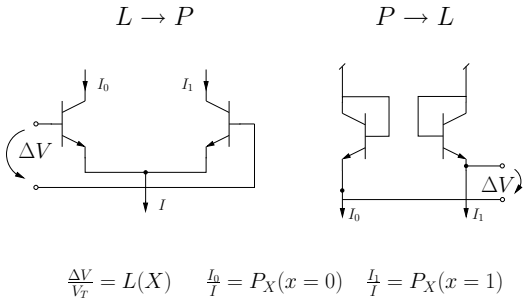

Bild 3: Transistorenpaare.

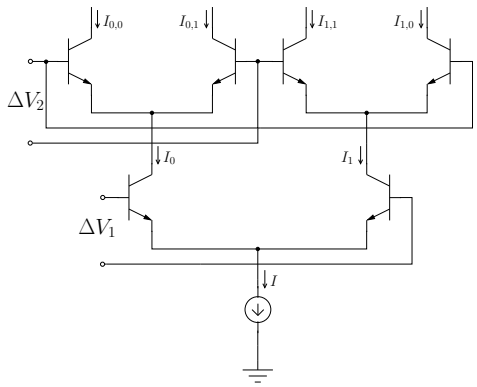

Bild 4: Gestapelte Konfiguration von Transistorenpaaren.

Die Addition zweier statistisch unabhängiger Variablen X_1 und X_2 entspricht einer XOR Verknüpfung im GF(2):

$$X_3 = X_1 \oplus X_2. \tag{11}$$

Die selbe Operation wird von einem binären Trellisabschnitt durchgeführt. In der λ-Domäne

entspricht dies der Multiplikation der Softbits:

$$\lambda(X_3) = \lambda(X_1)\lambda(X_2). \tag{12}$$

Verwendet man die Log-likelihood-Verhältnisse, so ergibt sich:

$$L(X_3) = 2\tanh^{-1}\Big[\tanh\Big(\frac{L(X_1)}{2}\Big)\tanh\Big(\frac{L(X_2)}{2}\Big)\Big]. \tag{13}$$

Diese Gleichung wird mit der „BOX-Plus“ Operation

$$L(X_3) = L(X_1) \boxplus L(X_2) \tag{14}$$

abgekürzt. Für $L(X_3)$ gilt aufgrund von Gleichung (11):

$$L(X_3) = \ln\Big(\frac{P_X(x_3=0)}{P_X(x_3=1)}\Big) = \ln\Big(\frac{P_X(x_1=0)P_X(x_2=0)+P_X(x_1=1)P_X(x_2=1)}{P_X(x_1=0)P_X(x_2=1)+P_X(x_1=1)P_X(x_2=0)}\Big). \tag{15}$$

Betrachtet man eine gestapelte Konfiguration von Transistorenpaaren, wie im Bild 4, erkennt man sofort, dass die Ströme an den Ausgängen, bezogen auf den Nennstrom,

$$\frac{I_{i,j}}{I} = P_X(x_1=i)P_X(x_2=j) \tag{16}$$

jeweils den Ergebnissen aus der Multiplikation zweier Wahrscheinlichkeiten entspricht, so wie sie im Nenner und Zähler der Gleichung(15) zu finden sind. Verbindet man die Ausgänge der gestapelten Konfiguration von Transistorenpaaren geschickt miteinander, so ergibt sich durch die Addition der jeweiligen Teilströme jeweils ein Strom für den Ausdruck im Zähler und Nenner. Erinnern wir uns jetzt an das Transisitorpaar in Diodenschaltung von Bild 3 rechts, das eine Umwandlung der Wahrscheinlichkeiten in ein Log-Likelihood-Verhältnis realisiert. Verwendet man die gewonnenen Summenströme als Eingang für dieses Transistorpaar, hat man die Gleichung (13) bzw. (15) exakt nachgebildet (siehe Kernblockverschaltung vom Typ I im Bild 5).

Die zweite Operation, die wir benötigen, ist die Addition zweier statistisch unabhängiger Bits, die der arithmetischen Addition zweier L-Werte entspricht:

$$L(X_3) = L(X_1) + L(X_2) = \ln\Big(\frac{P_X(x_1=0)P_X(x_2=0)}{P_X(x_1=1)P_X(x_2=1)}\Big). \tag{17}$$

Es ergibt sich die im Bild 5 unter Typ II gezeigte Kernblockverschaltung.

Wir sehen, dass die Grundschaltung zur Addition als auch die Grundschaltung zur BOX-Plus Verknüpfung zweier L Werte aus 9 Transistoren besteht. Im Bild 5 ist zusätzlich die notwendige Eingangs- (links) und Ausgangsstufe (rechts) eingezeichnet. Außerdem werden zusätzliche Dioden verwendet, um den Spannungsabfall zu erhöhen.

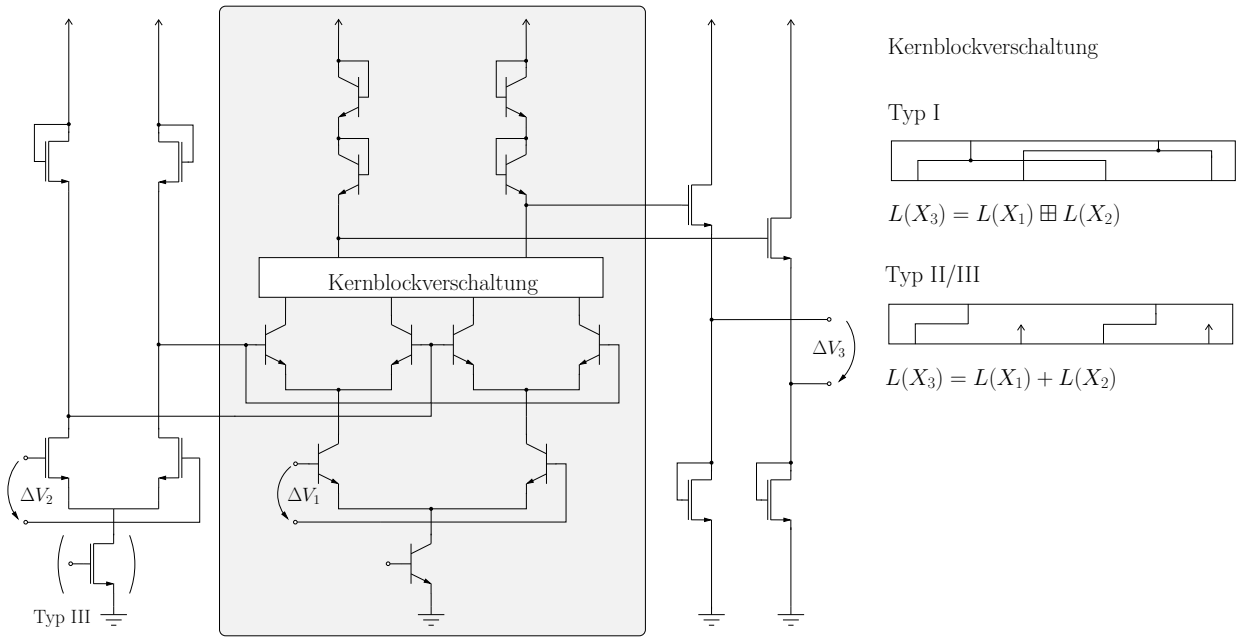

Bild 5: In Abhängigkeit von der Kernblockverschaltung implementiert der Aufbaublock die BOX-Plus Verknüpfung bzw. Addition.

3 Komponenten Decoder

3.1 Basierend auf Tannergraph

Der Tannergraph verdeutlicht graphisch die Prüfstruktur eines Kanalcodes. Die Prüfknoten, mit $\boxplus$ bzw. $\oplus$ bezeichnet, stellen gültige Paritätsprüfgleichungen für den beschriebenen Code dar. In der Domäne der L-Werte entsprechen diese den BOX-Plus Verknüpfungen. In Variablenknoten, mit $\bigcirc$ bezeichnet, finden Additionen statt, die in der Domäne der L-Werte ebenfalls Additionen entsprechen. Im Bild 6 ist ein Graph für den (7,4) Hamming Code dargestellt. Mit den oben beschriebenen Grundschaltungen können wir also den Decoder direkt in die Analogtechnik umsetzen. Nach der Initialisierung mit Kanalwerten führt das analoge Netzwerk die Decodierung durch. Dies geschieht zeitkontinuierlich für alle Bits. Die Geschwindigkeit des Decodierprozesses (Einschwinggeschwindigkeit der analogen Schaltung) wird begrenzt durch den verwendeten Prozess bzw. Technologie. Das Einschwingverhalten wird durch parasitäre Widerstände und Kapazitäten modelliert.

3.2 Basierend auf Trellis

Handelt es sich bei dem Code um einen Faltungscode, so kann der optimale BCJR Algorithmus [Bah74] mittels einer Vorwärts- und einer Rückwärtskette implementiert werden. Jeder Trellisabschnitt setzt sich bei einem Rate $1/N$ Code aus mehreren binären Trellisabschnitten zusammen, wie im Bild 7 zu erkennen ist. Der Trellisabschnitt im unteren Teil des Bildes ist um

90° gedreht um den Zusammenhang zur VLSI-Implemetierung zu verdeutlichen. Es bietet sich an, Tailbiting Codes [And98] zu verwenden, da im Gegensatz zur digitalen Implementierung kein zusätzlicher Rechenaufwand entsteht. Es reicht einfach den letzten Trellisabschnitt mit dem ersten zu verbinden. Nach dem Laden der Kanalwerte schwingt das Netzwerk ein.

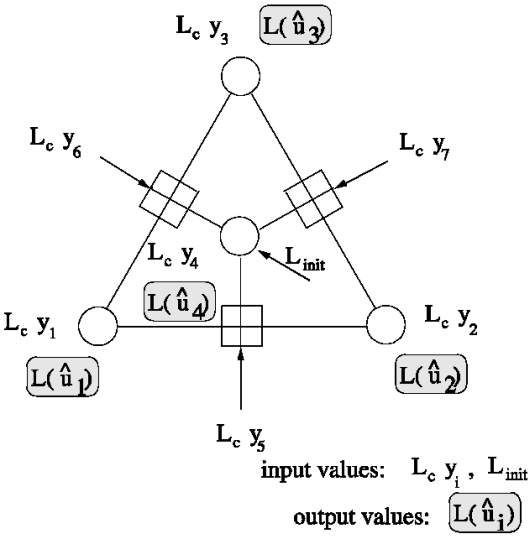

Bild 6: Graph des Decodiernetzwerkes - (7,4) Hamming Code.

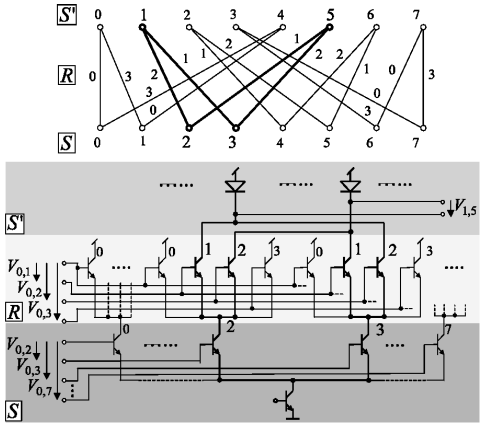

Bild 7: Implementierung eines Trellisabschnitts eines Rate 1/2 Codes mit Gedächtnis 3.

4 Turbo Decoder

Ein klassischer Turbo-Decoder besteht aus einem vertikalen und einem horizontalen Komponentendecoder. Die im Rahmen des Decodierprozesses in einem Komponentendecoder gewonnenene extrinsische Information wird nach einer Verwürfelung durch den Interleaver dem zweiten Komponentendecoder als a priori Information zugeführt. Es ergibt sich ein geschlossener Kreis. Ein solcher Durchlauf wird im Falle von einer digitalen Implemetierung als eine Iteration bezeichnet. In der analogen Welt entspricht der Interleaver einem sinnvollen Verbindungsnetzwerk. Man kann keine Iterationen mehr erkennen, sondern das Netzwerk schwingt nach dem Laden mit Kanalwerten ein und liefert das Decodierergebnis wie nach „unendlich“ vielen Iterationen.

5 Anwedungsmöglichkeiten und Offene Fragen

Hohe Bitraten im Bereich von mehreren Gbit/s werden vor allem bei der magnetischen Speicherung und optischen Übertragung benötigt. Es handelt sich in beiden Fällen um gute Kanäle

im Vergleich zum Mobilfunkkanal, was die Verwendung hochratiger Codes zur Folge hat. Der Trellis solcher Codes lässt sich unter Verwendung der dualen Codes sehr stark vereinfachen. Wir können demzufolge einen Decoder beruhend auf einem Trellis für niederratige Codes einsetzen.

Die Zukunft der analogen Decodierung ist zur Zeit noch ungewiss. Vieles hängt von der Entwicklung im Bereich der analogen und digitalen VLSI ab. Es müssen noch einige offene Fragen geklärt werden, wie z.B. die Realisierung des Interleaver Verbindungsnetzwerkes am Chip, die Rahmensynchronisation, die Rückgewinnung des Symboltaktes, die Seriell/Parallel-Umwandlung der Eingangswerte, das Abspeichern analoger Werte und die Realisierung der Analog/digital-Schnittstelle. Da vor allem Turbo Decoder von kontinuierlichen, nicht quantisierten Werten so viel profitieren, wäre unter Umständen sogar ein komplett analoger Empfänger denkbar. Die Decodiergeschwindigkeit der hoch parallelen analogen Netzwerke ist nur durch die parasitären Widerstände und Kapazitäten eingeschränkt, da es sich um kontinuierliches Einschwingen und keine Iterationen im Netzwerk handelt.

Literatur

[Hag98] J. Hagenauer, „Decoding of Binary Codes with Analog Networks," In *Proc. 1998 IEEE Information Theory Workshop*, San Diego, CA, USA, pp. 13-14, Feb. 1998.

[Loe99] H.-A. Loliger, M. Helfenstein, F. Lustenberger and F. Tarkoey, „Iterative sum-product decoding with analog VLSI," In *Proc. Int. Symp. Inform. Theory*, Cambridge, MA, USA, p.146, Aug. 1998.

[Moe00] M. Moerz, T. Gabara, R. Yan, and J. Hagenauer, „An analog 0.25 μm BiCMOS tailbiting MAP decoder," In *Proc. IEEE International Solid-State Circuits Conference (ISSCC 2000)*, pp. 356-357, San Francisco, CA, Feb. 2000.

[Lus99] F. Lustenberger, M. Helfenstein, H.-A. Loliger, F. Tarkoey and G.-S. Moschytz, „All analog decoder for binary (18,9,5) tailbiting trellis code," In *Proc. of 25th European Solid-State Circuits Conference*, Duisburg, Germany, Sep. 1999.

[Win01] C. Winstead, J. Dai, W. J. Kim, S. Little, Y.-B. Kim, C. Myers and C. Schlegel, „Analog MAP Decoder for (8,4) Hamming code in subthreshold CMOS," In *Proc. Advanced Research in VLSI*, Salt Lake City, pp. 132-147, Mar. 2001.

[Hag00] J. Hagenauer, M. Moerz and E. Offer, „A circuit based representation of analog MAP decoding with binary trellises," in *Proc., 3rd ITG Conference Source and Channel Coding, Munich, Germany*, Jan. 2000.

[Bah74] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, „Optimal decoding of linear codes for minimizing symbol error rate," *Trans. Inform. Theory*, pp. 284-287, Mar. 1974.

[And98] J. Anderson and S. Hladik, „Tailbiting MAP decoders," *IEEE Journal selected Areas in Comm.*, vol. 16, pp. 297-302, Feb. 1998.