

# Signalentropie und Thermodynamik von Berechnungen

Sebastian Ott

27. Januar 2005

## Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>2</b>
1.1	Gesetzmäßigkeiten . . . . .	2
<b>2</b>	<b>reversible Logik-Gatter</b>	<b>3</b>
2.1	Hintergrund . . . . .	3
2.1.1	Beispiele . . . . .	3
2.2	Toffoli gate . . . . .	4
2.3	Fredkin gate . . . . .	4
2.3.1	Universalität . . . . .	4
<b>3</b>	<b>Entropie vs. Energie</b>	<b>6</b>
3.1	Einstieg . . . . .	6
3.2	Bits . . . . .	6
3.3	Gatter . . . . .	7
3.4	Systeme . . . . .	9
3.4.1	Beispiel . . . . .	9
<b>4</b>	<b>Outro</b>	<b>10</b>

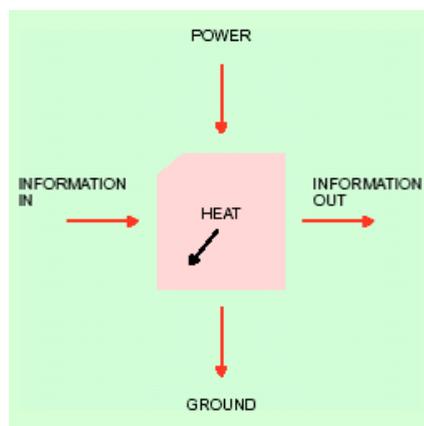
# 1 Einführung

Computer sind thermodynamische Maschinen die Arbeit verrichten, Energie verbrauchen und somit Abwärme produzieren. Moderne Desktop Prozessoren verbrauchen bis zu 100 Watt, komplette Desktop Computer sammt Peripherie leicht über 300 Watt. Andere Regionen erreichen Supercomputer: IBM's ASCI White hat einen Stromverbrauch von 6,2 Megawatt, das würde ausreichen eine Kleinstadt zu versorgen.

All das sind Gründe für Computer, die so wenig wie möglich Energie verschwenden. Betrachten wir einen typischen Chip (Bild 1). Er bezieht Strom von einer Stromquelle welcher gegen Masse abfließt, es werden Informationen verarbeitet und Wärme abgegeben. Für dieses System stellt sich folgende Frage:

*Kann durch die Analyse der INPUT/ OUTPUT-bits auf die thermodynamische Performance des Systems geschlossen werden?*

Abbildung 1: typischer Chip



Es wird also ein Zusammenhang zwischen der Information in einer Funktion und den thermodynamischen Eigenschaften eines Chips, welcher diese implementiert hergestellt.

## 1.1 Gesetzmäßigkeiten

In einem System mit gleichwahrscheinlichen Zuständen  $\Omega$  beträgt die Entropie  $k \log \Omega$ . Reduziert man die Entropie eines Systems um  $dS$ , wird Wärme an die Umgebung abgegeben:  $dQ = TdS$ . Ist der Wert eines Bits unbekannt, ändert sich die logische Entropie beim Löschen des Bits von  $k \log 2$  zu  $k \log 1 = 0$ . Bei allen Betrachtungen gilt das 1. Gesetz der Thermodynamik:

$$\begin{aligned}dU &= dW + dQ \\ &= dW + TdS\end{aligned}$$

Die Änderung der Energie  $dU$  eines Systems ist gleich der Summe aus reversibler Arbeit  $dW$  und der Wärme  $dQ$ , welche irreversibel an die Umgebung abgegeben wurde. Es gilt  $dQ = TdS$ , es wird also die abgegebene Wärme mit der Entropie-Änderung in Verbindung gebracht.

Integriert man das Ganze, erhält man:  $F = U - TS$ . Das ist der Anteil der Energie, welche reversibel zurückgewonnen werden kann um Arbeit zu verrichten.

Desweiteren gilt das 2. Gesetz der Thermodynamik:  $dS \geq 0$ . Wobei für reversible Prozesse gilt:  $dS = 0$  und für irreversible Prozesse:  $dS > 0$ .

## 2 reversible Logik-Gatter

### 2.1 Hintergrund

Ein logisches Gatter  $L$  ist reversibel, falls für jede Ausgabe  $y$  eine eindeutige Eingabe  $x$ , mit  $L(x)=y$ , existiert.

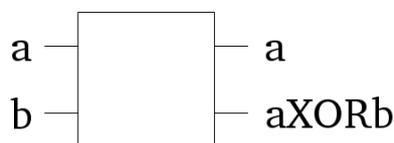
- In 'normalen' Gattern gehen die Eingangszustände verloren; es stecken weniger Informationen im output als im input.
- Aufgrund der thermodynamischen Entropie resultiert dieser Informationsverlust in einem Verlust von Energie in Form von Wärme.
- In reversiblen Gattern werden Ladungen nicht gelöscht und neu erzeugt, sondern nur die Zustände geändert; weil keine Information verloren geht, bleibt die Energie erhalten.

Klar ist, bei reversiblen Gattern muß die Anzahl der Input und Output Bits übereinstimmen.

#### 2.1.1 Beispiele

- 1 bit: NOT, Identität; trivialer Weise reversibel
- 2 bit: AND ist nicht reversibel; bei  $x \text{ AND } y = 0$  hat man keine Information über  $x$  und  $y$  (außer daß  $x$  und  $y$  nicht beide 1 waren)
- reversibles XOR

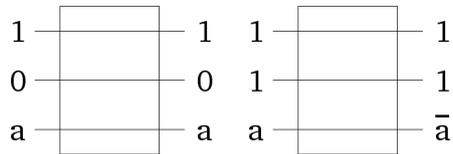
Abbildung 2: reversibles XOR



Die bekanntesten Vertreter reversibler Gatter sind jedoch das Toffoli (Abschnitt: [2.2](#)), bzw. das Fredkin Gatter (Abschnitt: [2.3](#)).

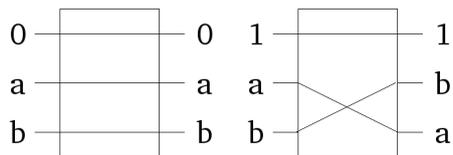
## 2.2 Toffoli gate

Das Toffoli Gatter hat 3 Eingänge, wovon 2 Kontroll-Bits (x und y) sind. Die Kontroll-Bits liegen unverändert am Ausgang wieder an und der 3. Eingang(a) wird negiert, wenn beide Kontroll-Bits 1 sind. Der 3. Ausgang wird also durch die Funktion  $a := aXOR(xANDy)$  beschrieben. (siehe Tabelle: 1)



## 2.3 Fredkin gate

Das Fredkin Gatter hat 3 Eingänge, davon 1 Kontroll-Bit (x). Wird das Kontroll-Bit gesetzt werden die anderen Eingänge vertauscht. Es wird also die Anzahl der 0 und 1 nicht verändert. (siehe Tabelle: 2)



### 2.3.1 Universalität

Die Universalität des Fredkin Gatters läßt sich konstruktiv zeigen, indem man die "Basisfunktionen" AND, OR und NOT mit Fredkin Gattern abbildet. (Bild 3) Es ist also möglich, jede berechenbare Funktion mit Fredkin Gattern abzubilden. Insbesondere ist auch ein reversibler Computer auf Basis von Fredkin Gattern denkbar.

Abbildung 3: AND, OR und NOT mit Fredkin Gattern

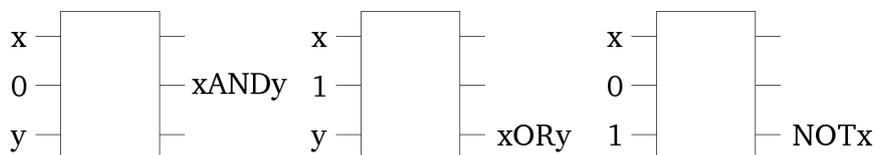


Tabelle 1: Tiffoli Gatter

INPUT			OUTPUT		
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	<b>0</b>	1	1	<b>1</b>
1	1	<b>1</b>	1	1	<b>0</b>

Tabelle 2: Fredkin Gatter

INPUT			OUTPUT		
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	<b>0</b>	<b>1</b>	1	<b>1</b>	<b>0</b>
1	<b>1</b>	<b>0</b>	1	<b>0</b>	<b>1</b>
1	1	1	1	1	1

## 3 Entropie vs. Energie

### 3.1 Einstieg

Ein Bit, welches mit Hilfe eines Kondensators gespeichert wird, benötigt eine Energie von  $CU^2/2$ . Beim Löschen des Bits fließt die Ladung gegen Masse ab und somit wird diese Energie verschwendet. Betrachtet man jetzt noch den Ladevorgang des Kondensators stellt man fest, daß durch das unmittelbare Anlegen einer Spannung nochmals  $CU^2/2$  verschwendet werden. Dies resultiert aus dem Leitungswiderstand. Lädt man den Kondensator jedoch linear mit Rate  $\tau$  auf, reduziert sich diese Energie auf  $CU^2RC/\tau$ .

Das unnötige Löschen von Bits zu verhindern, ist Aufgabe der reversiblen Logik bzw. der 'charge recovery'. Die adiabatische Logik beschäftigt sich damit, diese Änderungen nicht schneller als nötig zu machen.

### 3.2 Bits

Betrachten wir jetzt, wie viele Elektronen nötig sind, um ein Bit zu speichern.

$$C = \frac{Q}{U} = \frac{e * \#e}{U}$$
$$\#e = \frac{C * U}{e} = \frac{(5fF)(3V)}{1.6 * 10^{-19}C} \approx 10^5 \frac{\text{Elektronen}}{\text{bit}}$$

Das entspricht einer Energie von:

$$\frac{1}{2}C * U^2 = \frac{1}{2} * 5fF * (3V)^2$$
$$= 2 * 10^{-14} \frac{J}{\text{bit}} \approx 10^5 \frac{eV}{\text{bit}}$$

Falls  $10^7$  Transistoren diese Energie in jedem Takt nach Masse leiten, entspricht das bei 100MHz einer Energie von:

$$2 * 10^{-14} \frac{J}{\text{bit}} * 10^8 \frac{\text{bits}}{s} * 10^7 \text{Transistoren} = 20W$$

Natürlich ist diese Annahme (alle Transistoren ändern ihren Zustand zu jedem Takt) übertrieben. Im Vergleich ist die thermische Energie eines Gasteilchens bei Raumtemperatur um den Faktor  $10^7$  niedriger:

$$kT = 1.38 * 10^{-23} \frac{J}{K} * 300K \approx 0.02eV$$

Die elektronische Entropie pro Bit läßt sich mit der Sommerfeld Gleichung abschätzen:

$$S \approx \frac{N\pi^2T}{2T_f} k = \frac{10^5\pi^2 300K}{2 * 10^5 K} k = 1480k \approx 10^{-20} \frac{J}{K}$$

bzw. als Wärme:

$$Q = TdS \approx 10^{-18} \frac{J}{\text{bit}}$$

Tabelle 3: 2-Bit Sortierer

IN	OUT
0 0	0 0
0 1	1 0
1 0	1 0
1 1	1 1

### 3.3 Gatter

Wenden wir uns nun Gattern, und zwar am Beispiel des 2-Bit Sortierers (siehe Tabelle: 3) zu. Wenn man nun die (logische) Entropie des Sortierers betrachtet:

$$H = - \sum_{states} p_{state} \log_2(p_{state})$$

$$H_{in} = -4 * \frac{1}{4} \log_2\left(\frac{1}{4}\right) = 2$$

$$H_{out} = -2 * \frac{1}{4} \log_2\left(\frac{1}{4}\right) - \frac{1}{2} \log_2\left(\frac{1}{2}\right) = \frac{3}{2}$$

stellt man fest, daß zwar der Output die gleiche Energie wie der Input hat (gleiche Anzahl von 0,1), aber im Mittel geht jedesmal 1/2 Bit Entropie "verloren". Dieser Entropieverlust resultiert in Wärmeabgabe.

Untersucht man die conventionelle CMOS Implementation des Sortierers (Bild: 4) wird klar, dieser verschwendet Energie bei jeder Zustandsänderung. Bei jedem 0/1 Zustandsübergang muß der FET aufgeladen werden und bei jedem 1/0 Übergang fließt diese Ladung nach Masse ab. Die Anzahl der 0/1, 1/0 Übergänge ist also allein verantwortlich für die verschwendete Energie.

Abbildung 4: 2-Bit Sortierer in CMOS

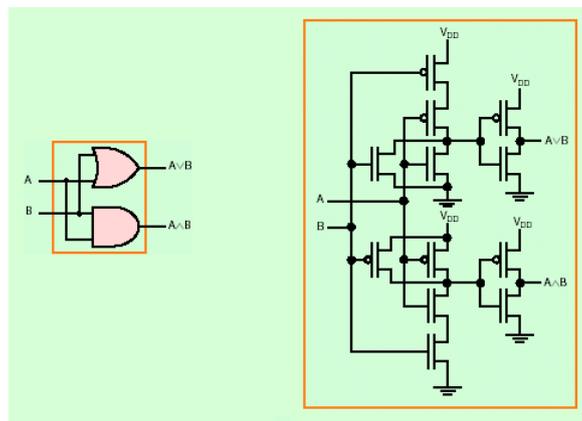
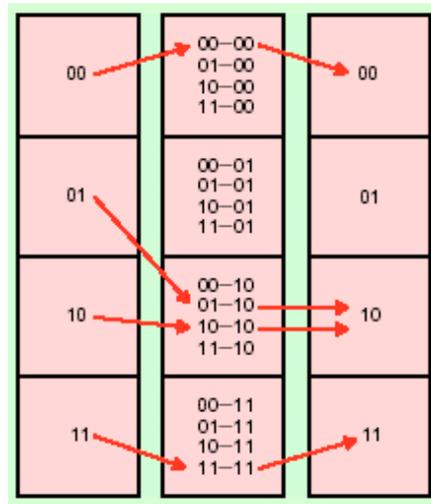


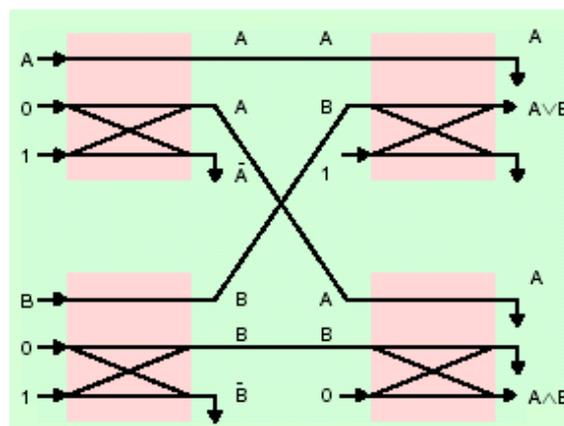
Abbildung 5: 2-Bit Sortierer - Zustände



Betrachten wir nun die möglichen Zustände (Bild 5). Der 2-Bit Sortierer hat 4 Input und 4 Output Zustände. Das macht 16 mögliche Input-Output Paare (auch wenn manche nicht vorkommen). Die Inputzustände werden unabhängig von den Outputs gelöscht, also werden die 4 Pfade zu einem gemerged; das kostet  $\log 4 = 2\text{bits}$  Entropie.

Das eigentliche Problem ist, daß irreversible Gatter für eine reversible Aufgabe benutzt werden. Keines der Gatter kann für sich allein feststellen, ob die Funktion, die sie gemeinsam berechnen, reversibel ist.

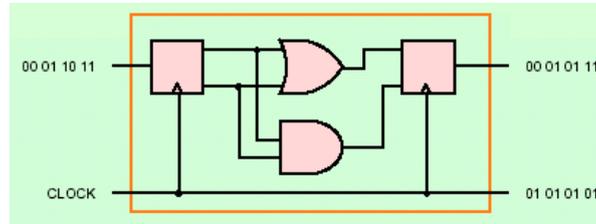
Abbildung 6: 2-Bit Sortierer mit Fredkin Gattern



### 3.4 Systeme

Betrachtet werden jetzt sequenzielle Gatter, bzw. die Unterschiede zum kombinatorischen Fall.

Abbildung 7: getakteter 2-Bit Sortierer



Das Gatter sortiert die zeitliche Ordnung von Bit-Paaren (Bild 7). Sieht man sich nun an, wie die Bit-Strings zeitlich vorhersagbar sind, stellt man fest, daß bei einem zufälligen Input zwar die Anzahl von 0,1 im Output gleich, aber der Output nicht mehr zufällig ist (insbesondere treten bestimmte Bitkombinationen nicht mehr auf: 10).

Betrachten wir nun den Input( $x_n$ ), bzw. den Output( $y_n$ ) zum Zeitpunkt  $n$ . Da das Gatter irreversibel ist, gibt es Information in  $x_n$ , die nicht aus  $y$  bestimmt werden kann. Der Output hat also eine geringere Entropie, aufgrund dessen tritt ein Energieverlust auf. Berechnen wir also jetzt  $I_{reverse}$ , die Information, die im Schnitt vom Input  $x_n$  verloren geht:

$$I_{reverse} = H(x_n, \dots, x_{n+d}; y_n, \dots, y_{n+d}) - H(x_{n+1}, \dots, x_{n+d}; y_n, \dots, y_{n+d})$$

$$H(x_n, \dots, x_{n+d}; y_n, \dots, y_{n+d}) = H(x_n) + H(x_{n+1}, \dots, x_{n+d}; y_n, \dots, y_{n+d})$$

also gilt:  $I_{reverse} = H(x_n)$ .  $\rightarrow$  Alle Bits in  $x_n$  werden vom System verarbeitet. Bezieht man jetzt die Zukunft von  $x_n$  mit ein und bildet:

$$I_{forward} = H(x_n, \dots, x_{n-d}; y_n, \dots, y_{n-d}) - H(x_n, \dots, x_{n-d}; y_{n-1}, \dots, y_{n-d})$$

kann man aus der Differenz  $I_{reverse} - I_{forward}$  auf den Informationsfluß schließen den das System im Schnitt verarbeitet. Aus diesem Entropieverlust läßt sich auf den Energieverlust schlußfolgern.

#### 3.4.1 Beispiel

Sehen wir uns nun einen Chip an, der von einem  $100\text{Mbit/s}$  Netzwerk angesteuert wird. Die Kapazität des Netzwirkabels beträgt ca.  $30\text{pF/ft}$ , die Ausbreitungsgeschwindigkeit ist ungefähr  $1.5\text{nsec/ft}$ . Daraus folgt die Kapazität je Bit:

$$\frac{1}{10^8 \frac{\text{bits}}{\text{sec}} * 1.5 * 10^{-9} \frac{\text{sec}}{\text{ft}}} * 30 \frac{\text{pF}}{\text{ft}} \approx 200 \frac{\text{pF}}{\text{bit}}$$

Bei  $3V$  entspricht das einer Energie von:

$$\frac{1}{2} * 200^{-12} F * (3V)^2 \approx 10^{-9} \frac{J}{bit}$$

bzw:

$$10^{-9} \frac{J}{bit} * 10^8 \frac{bits}{s} = 0.1W$$

## 4 Outro

Zusammengefaßt haben wir verschiedene Ansichten des Energie/Entropie Budgets eines Chips betrachtet:

- zum einen die freie Energie der Input/Output Bits, welche zurück gewonnen werden kann,
- zum anderen die Informationsdifferenz zwischen Input und Output, welche als eine Schranke für die thermalen Eigenschaften eines Chips gesehen werden kann.

## Literatur

1. N. Gershenfeld: *Signal entropy and the thermodynamics of computation*
2. Wikipedia: [www.wikipedia.org](http://www.wikipedia.org)