Zufallszahlen generieren

Was ist Zufall?

- Gibt es überhaupt irgendeinen Zufall?
- Kann man das wissen?

Zufall?

- Von Zufall spricht man, wenn für ein einzelnes Ereignis oder das Zusammentreffen mehrerer Ereignisse keine kausale Erklärung gefunden werden kann. (Wikipedia)

Deterministischer Algorithmus

- Auf eine bestimmte Eingabe folgt immer eine bereits definierte Ausgabe
- z.B. Kaffeeautomat

Randomisierter Algorithmus

- Durch Wahl von zufälligen Zwischenergebnissen (mithilfe von Zufallsbits/-variablen), dennoch determiniert
- Einfacher zu implementieren, spart Rechenzeit
- Darf falsche Ergebnisse liefern (Monte Carlo) bzw. „aufgeben“ (Las Vegas)
- z.B. der Solovay-Strassen-Test

Zufallsexperiment

- Ergebnis des Experiments kann einen randomisierten Algorithmus speisen
- (Nur) Aus Sicht des Durchführers zufällig
  - Pseudozufall
  - Quasizufall
  - Echter Zufall

Pseudozufall

- **Berechneter Zufall (determiniert)**
- Aus Sicht des Beobachters nicht von echtem Zufall zu unterscheiden

- Seed-Algorithmus (Erklärung folgt)

Quasizufall

- **„Gefundener“ Zufall (Compilezeit-Zufall)**
- Greift auf nicht durch den Zufallsgenerator beeinflussbare / vorhersehbare Variablen auf der Maschine zu

- Systemzeit, Netzwerkdurchsatz, Speicherzuweisung

Echter Zufall

- **Kein Beobachter kann ihn vorhersehen (indeterminiert)**
  - Warum?
    - Für ein Kind mag ein Abzählreim zufällig erscheinen, ist aber nur Pseudorandomisiert. ##Hier Illustration: Gleicher Startwert = Gleicher Endwert
    - Für ein Programm mag Netzwerkdurchsatz zufällig erscheinen ##Hier Illustration: Gleiches Netzwerk = Gleicher Endwert
    - Betrachtet ein Beobachter innerhalb des Universums einen möglichen Quantenzustand, so verschwimmt mit schärferer Sicht auf Raum jene auf Zeit (Heisenbergsche Unschärferelation)
      - Selbst wenn es verborgene Parameter gibt, die uns, wie dem Kind beim Abzählreim, nicht bewusst sind, mit denen sich Quantenzustände eindeutig voneinander ableiten ließen, so müsste man einen Computer konstruieren, der jedes Quantum für jedes Chronos betrachten und iterieren müsste – und kann, unter den gleichen physikalischen Bedingungen, somit höchstens in Echtzeit alle Ereignisse im Universum zurückrechnen. Würde man also heute ein Ereignis eines Quantenexperiments vorhersagen wollen, wäre das Ergebnis erst in Milliarden von Jahren zu erwarten.
- Radioaktiver Zerfall, Widerstandsrauschen

Güte

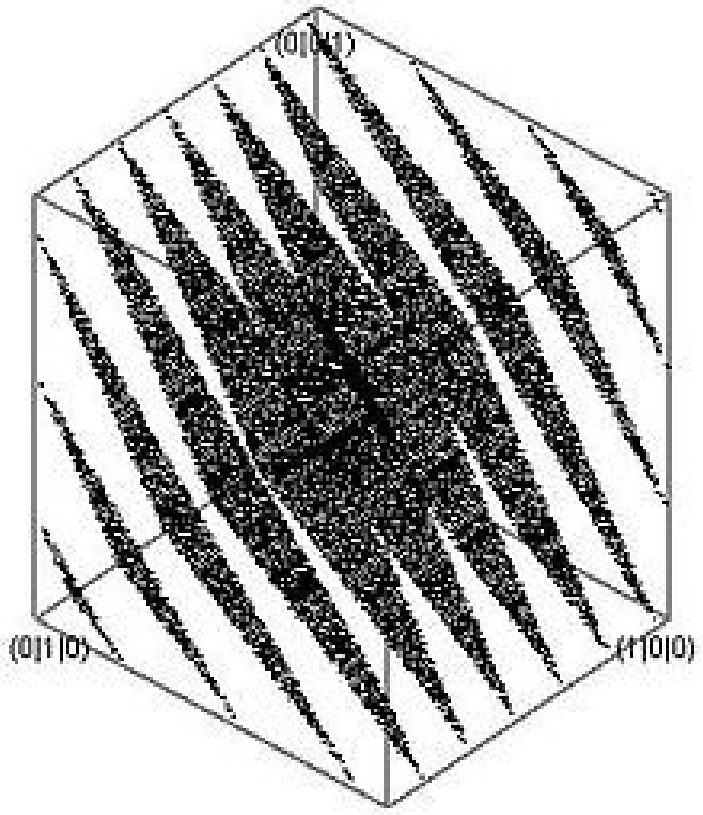
Güte eines Zufallsgenerators ist abhängig von

- Der Verteilung (Normalverteilung, Gleichverteilung, etc.)
- Unabhängigkeit zwischen generierten Zahlen
- Periodenlänge / Wertebereich

Güte - Spektraltest

- Packt *i* zu untersuchende Zufallszahlen in Tupel zusammen und überprüft Gleichverteilung im *i*-Dimensionalen Raum
- So kann Abhängigkeit von aufeinander folgenden Zahlen erkannt und somit die Unberechenbarkeit des Zufalls widerlegt werden

Güte - Spektraltest



Güte - Run-Test

- Testet, ob es Muster bei der Verteilung austeigender und absteigender Zahlen gibt.

1 5 19 9 9 8 7 4 7 4 4 8 6 4 8 9 7 5 3 2 4 5

\+ + - - - - + - + - - + + - - - - + +

Güte – χ^2-Test

- Berechnung einer erwarteten Verteilung und Vergleich mit tatsächlicher Verteilung (Normalverteilung)
- χ^2-Verteilung: n Zufallsvariablen Z_i sind normal-verteilt, χ^2-Verteilung mit n Freiheitsgraden wird dann definiert als

$$Chi = \frac{k}{n} \sum_{1 \le i \le k} \left(b_i^2 \right) - n$$

Güte

Maurers universeller Test:

- Eine Folge von Zufallszahlen kann durch keinen Komprimieralgorithmus komprimiert werden.

Pseudozufall

- Mithilfe eines Startwertes kann eine beliebig lange, periodische Zahlenfolge generiert werden (√n, wo n keine ganze Zahl)
- Wird mit einem anderen Takt, bei der f1/f2 einer irrationalen Zahl entspricht, diese Zahlenfolge eingelesen, kann nicht-Periodität erreicht werden.

Kongruenzgeneratoren

- Modul $m \in \{2, 3, 4, \ldots\}$
- Faktor $a \in \{1, \ldots, m-1\}$
- Inkrement $b \in \{1, \ldots, m-1\}$
- Startwert $y_1 \in \{0, \ldots, m-1\}$

Aus dem Startwert werden dann die weiteren Werte nach folgender Formel (mit $i \geq 2$) berechnet:

$$y_i = (a y_{i-1} + b) \mod m$$

mit *ggt(b, m)*

m^n mögliche Zustände, ungünstig jedoch Periodenlänge 1

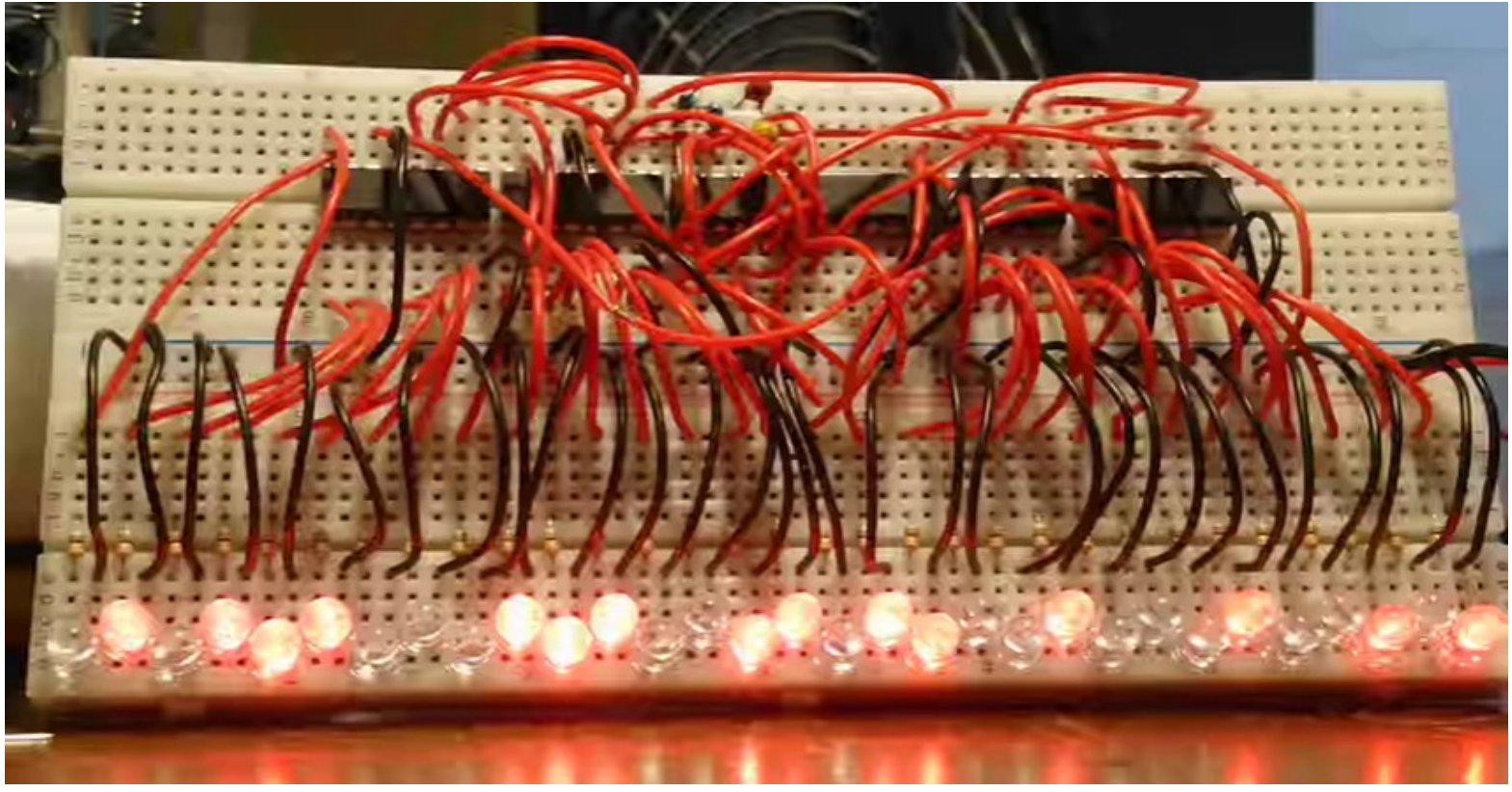
Kongruenzgeneratoren

- Satz von Knuth
- Periodität
- Vorhersehbarkeit

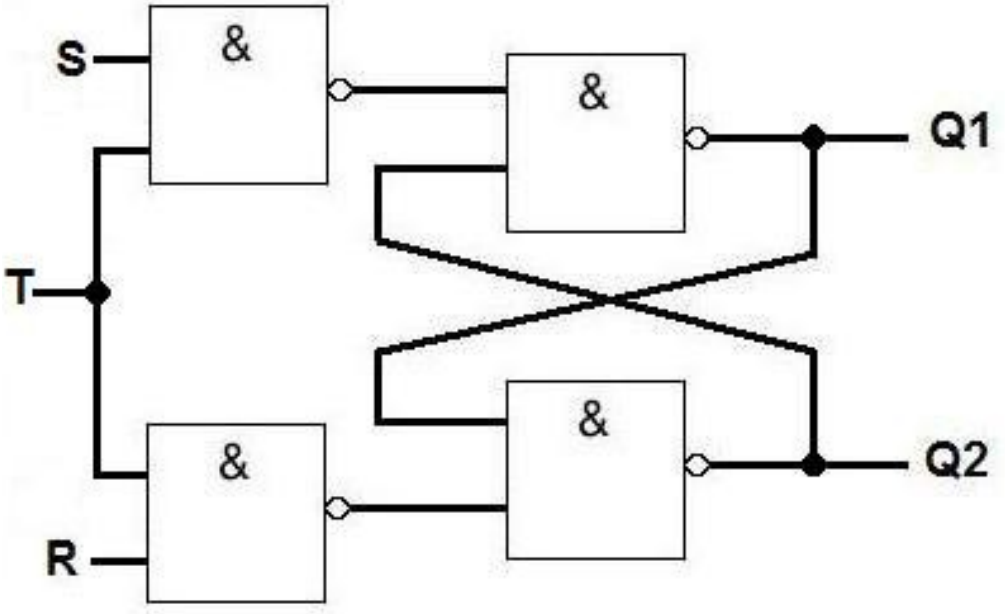
Linear rückgekoppeltes Schieberegister

- Periodenlänge: $2^n - 1$ bei n Bit
- Kann sowohl einfach in TTL, als auch im FPGA oder Software realisiert werden

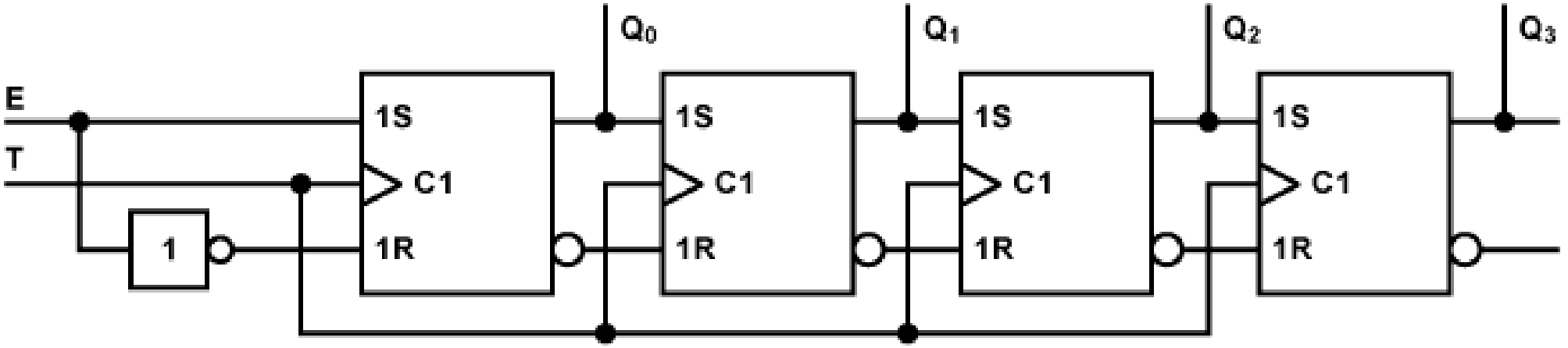
Linear rückgekoppeltes Schieberegister



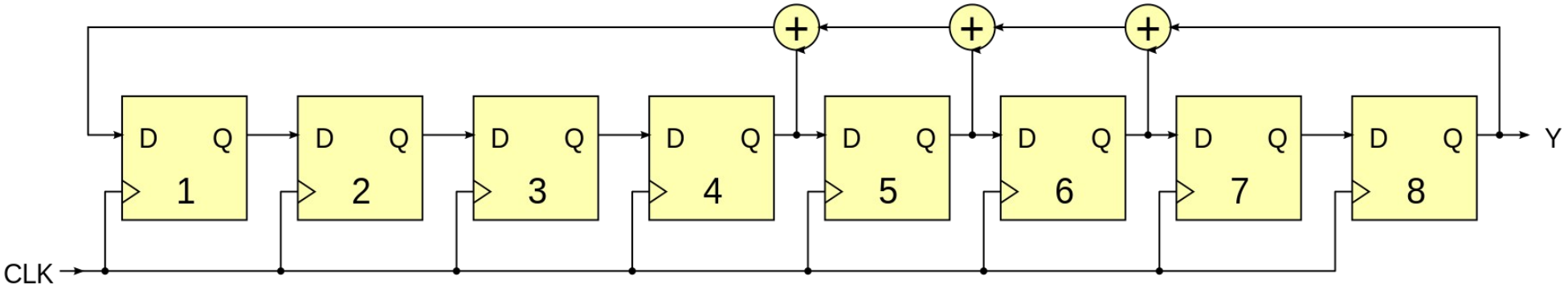
Linear rückgekoppeltes Schieberegister



Linear rückgekoppeltes Schieberegister



Linear rückgekoppeltes Schieberegister



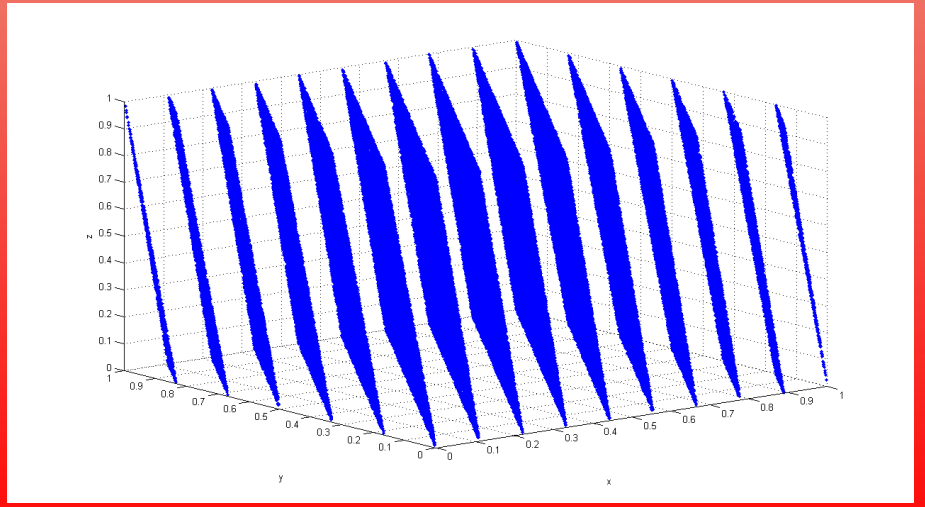
$$x^8 + x^6 + x^5 + x^4 + 1$$

Mersenne-Twister MT19937

- Periodenlänge der Mersenne-Primzahl $2^{19937}-1$
- Mersenne-Primzahl im Dualzahlsystem heterogen
- Absolute Gleichverteilung bis zu n=623

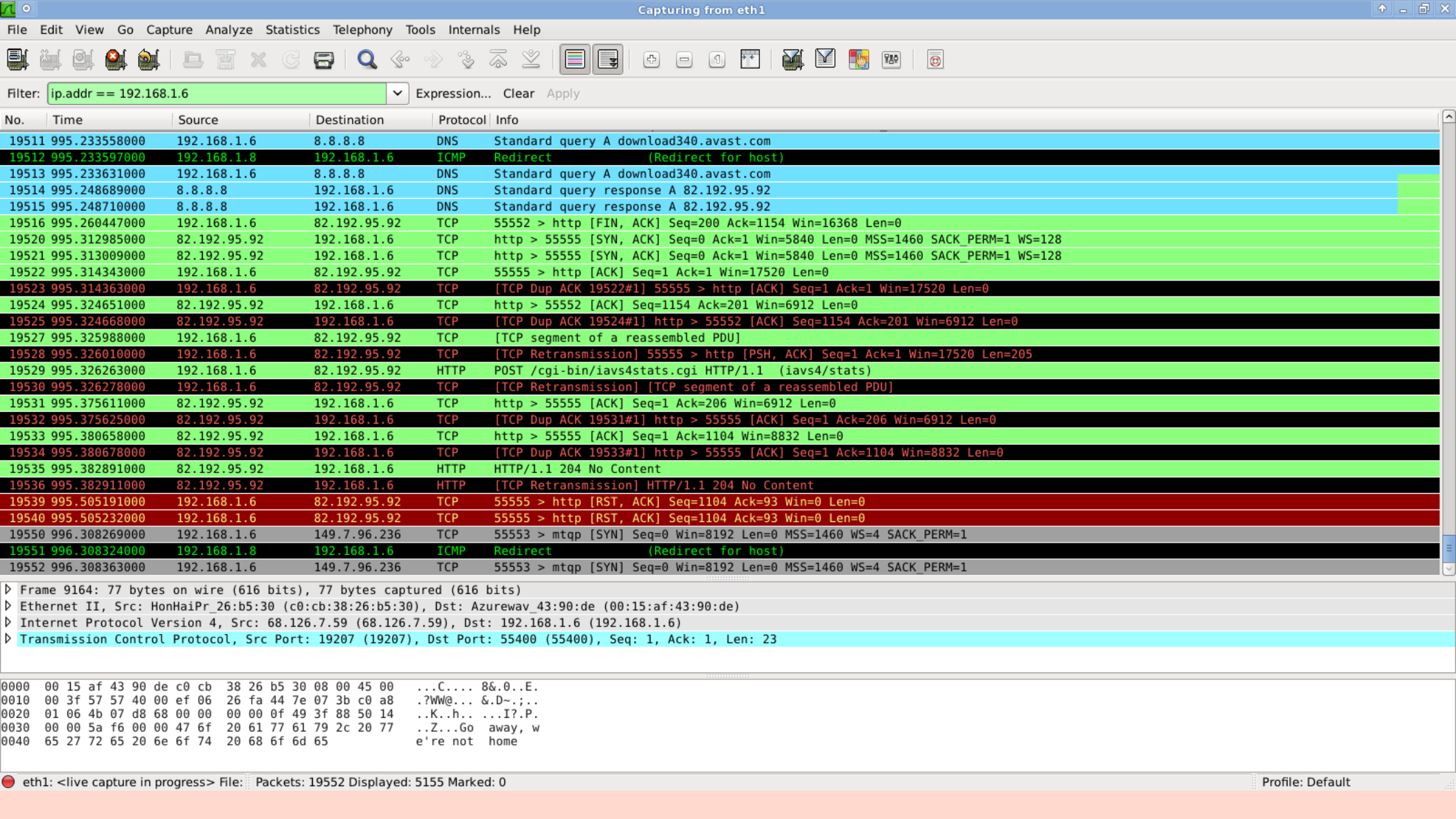
RANDU

- Von IBM in den 60ern entwickelt (Linearer Kongruenzgenerator)
- Spektraltest in 3.er Dimension fehlgeschlagen
- Jede Zahl ist ungereade



Quasizufall

- Kann als Seed benutzt werden
- Unabhängigkeit kann nicht garantiert werden


Capturing from eth1
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: ip.addr == 192.168.1.6
Expression... Clear Apply
No. Time Source Destination Protocol Info
19511 995.233558000 192.168.1.6 8.8.8.8 DNS Standard query A download340.avast.com
19512 995.233597000 192.168.1.8 192.168.1.6 ICMP Redirect (Redirect for host)
19513 995.233631000 192.168.1.6 8.8.8.8 DNS Standard query A download340.avast.com
19514 995.248689000 8.8.8.8 192.168.1.6 DNS Standard query response A 82.192.95.92
19515 995.248710000 8.8.8.8 192.168.1.6 DNS Standard query response A 82.192.95.92
19516 995.260447000 192.168.1.6 82.192.95.92 TCP 55552 > http [FIN, ACK] Seq=200 Ack=1154 Win=16368 Len=0
19520 995.312985000 82.192.95.92 192.168.1.6 TCP http > 55555 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM=1 WS=128
19521 995.313009000 82.192.95.92 192.168.1.6 TCP http > 55555 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM=1 WS=128
19522 995.314343000 192.168.1.6 82.192.95.92 TCP 55555 > http [ACK] Seq=1 Ack=1 Win=17520 Len=0
19523 995.314363000 192.168.1.6 82.192.95.92 TCP [TCP Dup ACK 19522#1] 55555 > http [ACK] Seq=1 Ack=1 Win=17520 Len=0
19524 995.324651000 82.192.95.92 192.168.1.6 TCP http > 55552 [ACK] Seq=1154 Ack=201 Win=6912 Len=0
19525 995.324668000 82.192.95.92 192.168.1.6 TCP [TCP Dup ACK 19524#1] http > 55552 [ACK] Seq=1154 Ack=201 Win=6912 Len=0
19527 995.325988000 192.168.1.6 82.192.95.92 TCP [TCP segment of a reassembled PDU]
19528 995.326010000 192.168.1.6 82.192.95.92 TCP [TCP Retransmission] 55555 > http [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=205
19529 995.326263000 192.168.1.6 82.192.95.92 HTTP POST /cgi-bin/iavs4stats.cgi HTTP/1.1 (iavs4/stats)
19530 995.326278000 192.168.1.6 82.192.95.92 TCP [TCP Retransmission] [TCP segment of a reassembled PDU]
19531 995.375611000 82.192.95.92 192.168.1.6 TCP http > 55555 [ACK] Seq=1 Ack=206 Win=6912 Len=0
19532 995.375625000 82.192.95.92 192.168.1.6 TCP [TCP Dup ACK 19531#1] http > 55555 [ACK] Seq=1 Ack=206 Win=6912 Len=0
19533 995.380658000 82.192.95.92 192.168.1.6 TCP http > 55555 [ACK] Seq=1 Ack=1104 Win=8832 Len=0
19534 995.380678000 82.192.95.92 192.168.1.6 TCP [TCP Dup ACK 19533#1] http > 55555 [ACK] Seq=1 Ack=1104 Win=8832 Len=0
19535 995.382891000 82.192.95.92 192.168.1.6 HTTP HTTP/1.1 204 No Content
19536 995.382911000 82.192.95.92 192.168.1.6 HTTP [TCP Retransmission] HTTP/1.1 204 No Content
19539 995.505191000 192.168.1.6 82.192.95.92 TCP 55555 > http [RST, ACK] Seq=1104 Ack=93 Win=0 Len=0
19540 995.505232000 192.168.1.6 82.192.95.92 TCP 55555 > http [RST, ACK] Seq=1104 Ack=93 Win=0 Len=0
19550 996.308269000 192.168.1.6 149.7.96.236 TCP 55553 > mtqp [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
19551 996.308324000 192.168.1.8 192.168.1.6 ICMP Redirect (Redirect for host)
19552 996.308363000 192.168.1.6 149.7.96.236 TCP 55553 > mtqp [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
Frame 9164: 77 bytes on wire (616 bits), 77 bytes captured (616 bits)
Ethernet II, Src: HonHaiPr_26:b5:30 (c0:cb:38:26:b5:30), Dst: Azurewav_43:90:de (00:15:af:43:90:de)
Internet Protocol Version 4, Src: 68.126.7.59 (68.126.7.59), Dst: 192.168.1.6 (192.168.1.6)
Transmission Control Protocol, Src Port: 19207 (19207), Dst Port: 55400 (55400), Seq: 1, Ack: 1, Len: 23
0000 00 15 af 43 90 de c0 cb 38 26 b5 30 08 00 45 00 ...C.... 8&.0..E.
0010 00 3f 57 57 40 00 ef 06 26 fa 44 7e 07 3b c0 a8 .?WW@... &.D~.;..
0020 01 06 4b 07 d8 68 00 00 00 00 0f 49 3f 88 50 14 ..K..h.. ...I?.P.
0030 00 00 5a f6 00 00 47 6f 20 61 77 61 79 2c 20 77 ..Z...Go away, w
0040 65 27 72 65 20 6e 6f 74 20 68 6f 6d 65 e're not home
eth1: <live capture in progress> File: Packets: 19552 Displayed: 5155 Marked: 0
Profile: Default

Echter Zufall

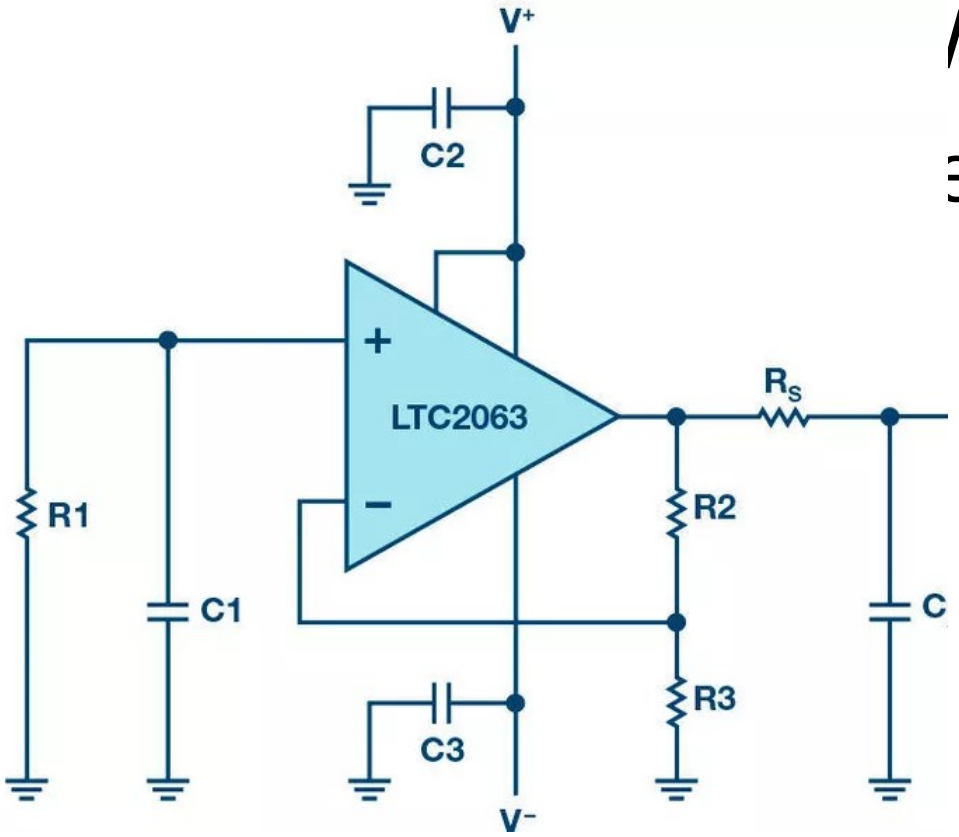
- Kryptographisch sicher
- Höchstmögliche Güte
- Da die Maschine, auf der der Zufall benötigt, niemals idealistisch, d.h. ohne physikalische Hülle, existieren kann, da wir in einem materialistischen Universum leben, manipulieren äußere Einflüsse die Logikschaltung / Eingabegeräte – Unter Quanteneinfluss ein super Zufallszahlengenerator
- Quantencomputer

Echter Zufall

- Thermisches Rauschen / Widerstandsrauschen
- Schrotrauschen / Avalanche-Dioden
- CCD-Rauschen

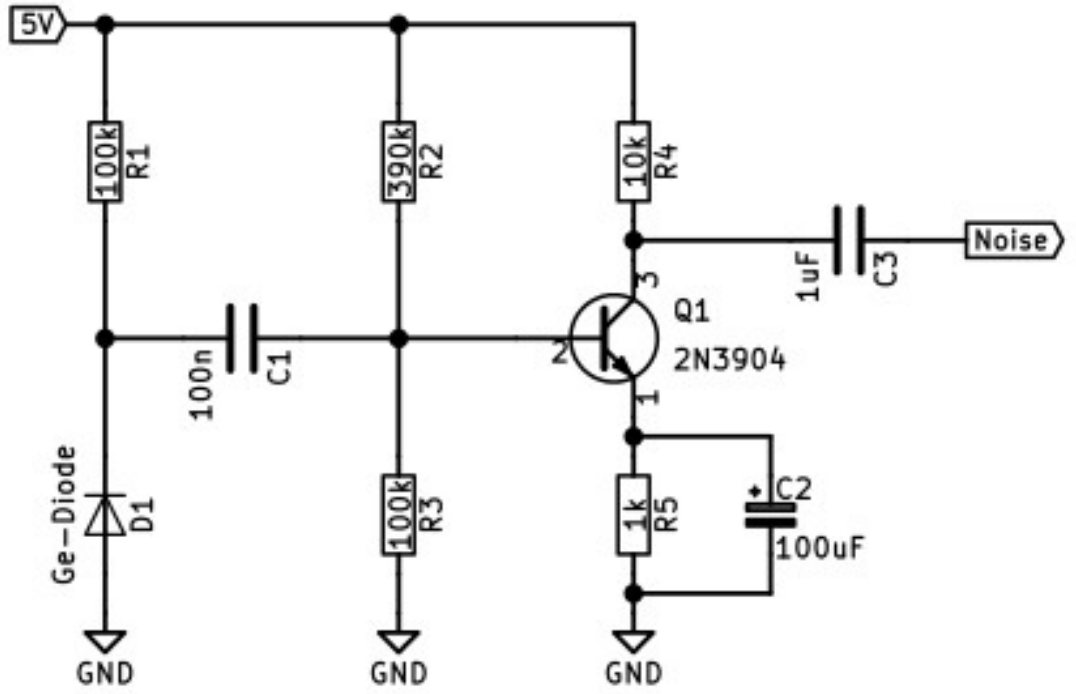
Echter Zufall

- Th /iderstandsrauschen
- So e-Dioden
- C(



Echter Zufall

- Thermische
- Schrotrausc
- CCD-Rausc



/dev/random

- Datei in Unix-artigen Betriebssystemen
- Gewinnt Zufall aus Umgebungsrauschen in Entropiepool

Exkursion! #TakeMeToTheLinux