



CHARITÉ CAMPUS BENJAMIN FRANKLIN

Institut für
**Medizinische Informatik,
Biometrie und Epidemiologie**

Univ.-Prof. Dr. rer. nat. Thomas Tolxdorff

Informationscodierung

Prof. Dr. Thomas Tolxdorff

Vorlesung an der Charité - Universitätsmedizin Berlin

- Biologische Computer: DNA-Computing
- Erste praktische Anwendung
- Vor- und Nachteile von DNA-Computern
- Halbleiterbasierte Computer
- Informationscodierung mit Bits & Bytes
- Vergleich DNA- / Halbleitercomputer
- Zusammenfassung

Wenn Sie diese Vorlesung absolviert haben, dann können Sie:

- die Grundlagen des Rechnens mit DNA erklären und Vor- und Nachteile des DNA-Computers benennen,
- Information und Daten definieren und Unterschiede zwischen analogen und digitalen Daten erläutern,
- beschreiben, warum das Binärsystem und normierte Zeichensätze zur Datenrepräsentation im traditionellen Computer eingesetzt werden.

- DNA ist Träger der **Erbinformation**
- DNA ist in nahezu **allen** Lebewesen zu finden
- **Doppelhelicale** Struktur
- Durch **Basenpaarung** redundante Information

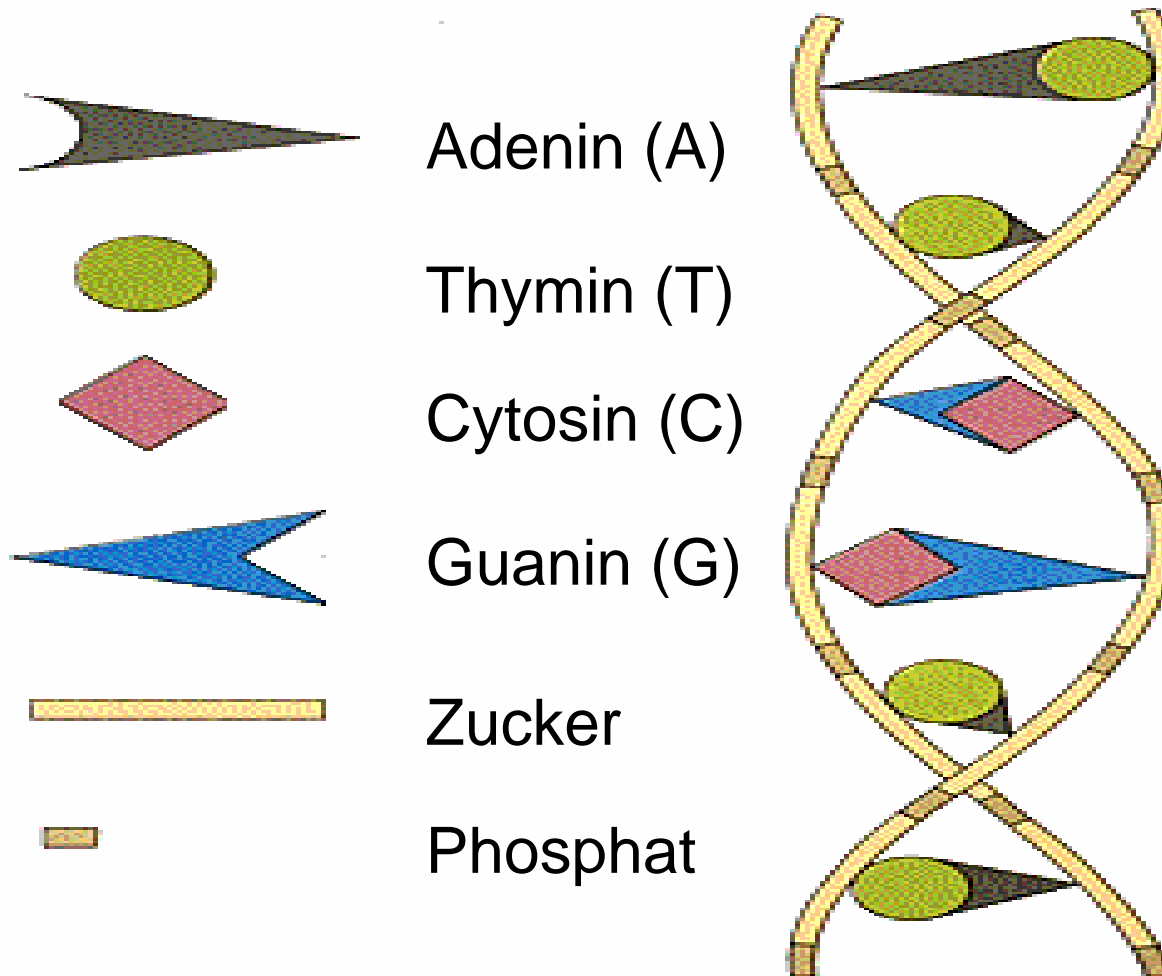
- Datenträger
- **4** elementare Informationselemente (2 Purin und 2 Pyrimidin-Basen)
- 100.000-fach höhere **Datendichte** als traditionelle Datenträger
- 1 Basenpaar entspricht einer Raid-Einheit (Datensicherung aufgrund redundanter Information)

- **Doppelhelix**
- **Zucker-Phosphat Einheiten** bilden das außen liegende **Rückgrat** der gewundenen Einzelstränge
- Pyrimidin- und Purin-Basen liegen im **Inneren**
- Basen sind durch **Wasserstoffbrücken** verbunden
- Basenpaare stehen **senkrecht** zur Helixachse

Struktur eines DNA-Doppelstranges

7

Biologische Computer: DNA-Computing



Struktur eines DNA-Doppelstranges

8

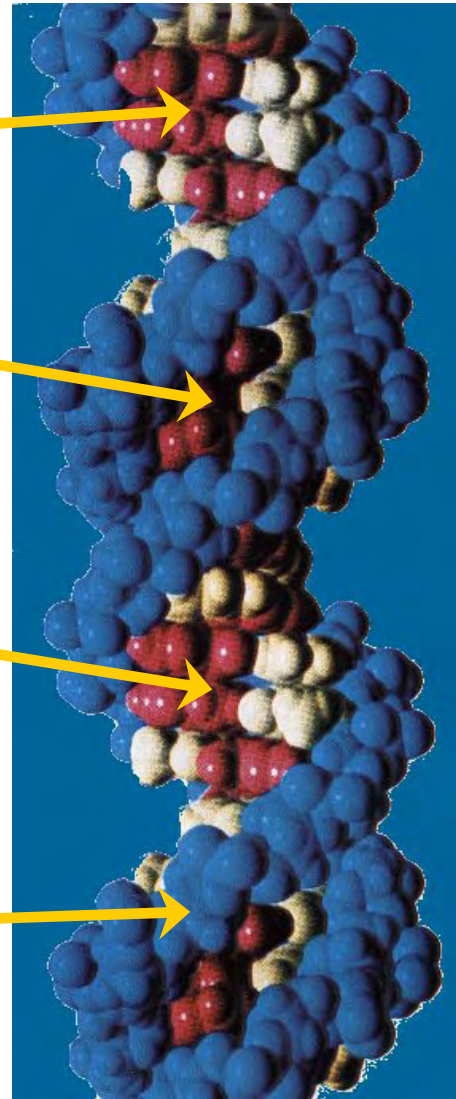
Biologische Computer: DNA-Computing

Basenpaare

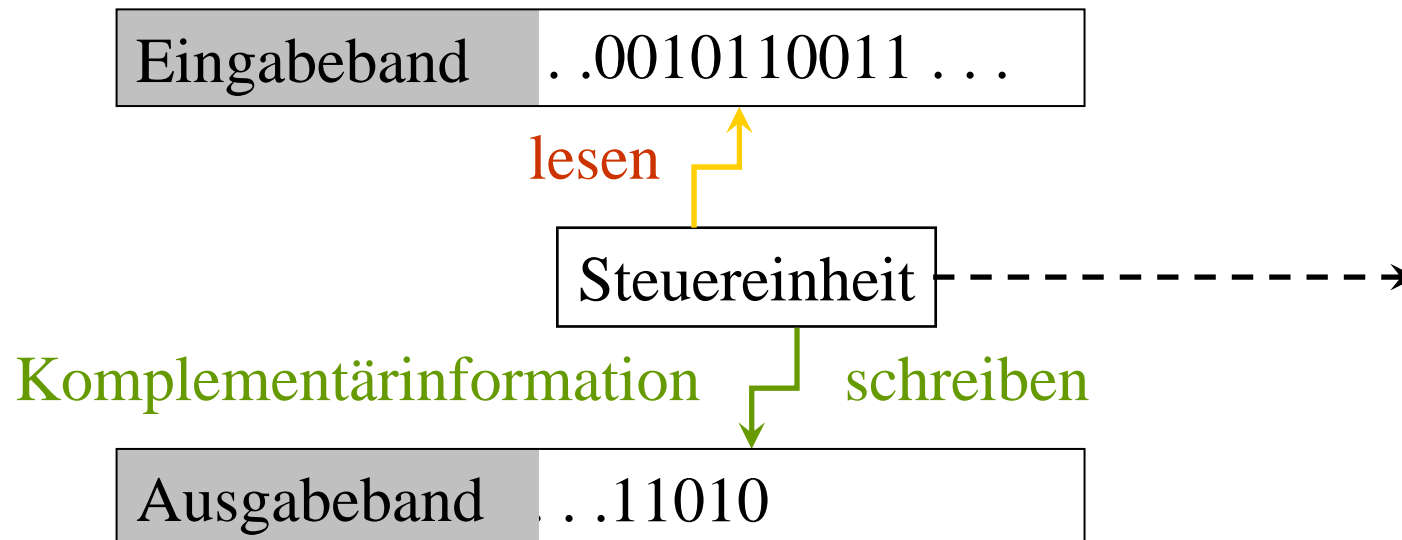
**kleine
Furche**

**große
Furche**

**Zucker-
Phosphat
Rückgrat**



- 1936 Analyse des Begriffs der **Berechenbarkeit** (10 Jahre vor den ersten Computern)
- **Hypothetische** Rechenmaschinen wurden erdacht → Beispiel einer Turingmaschine:



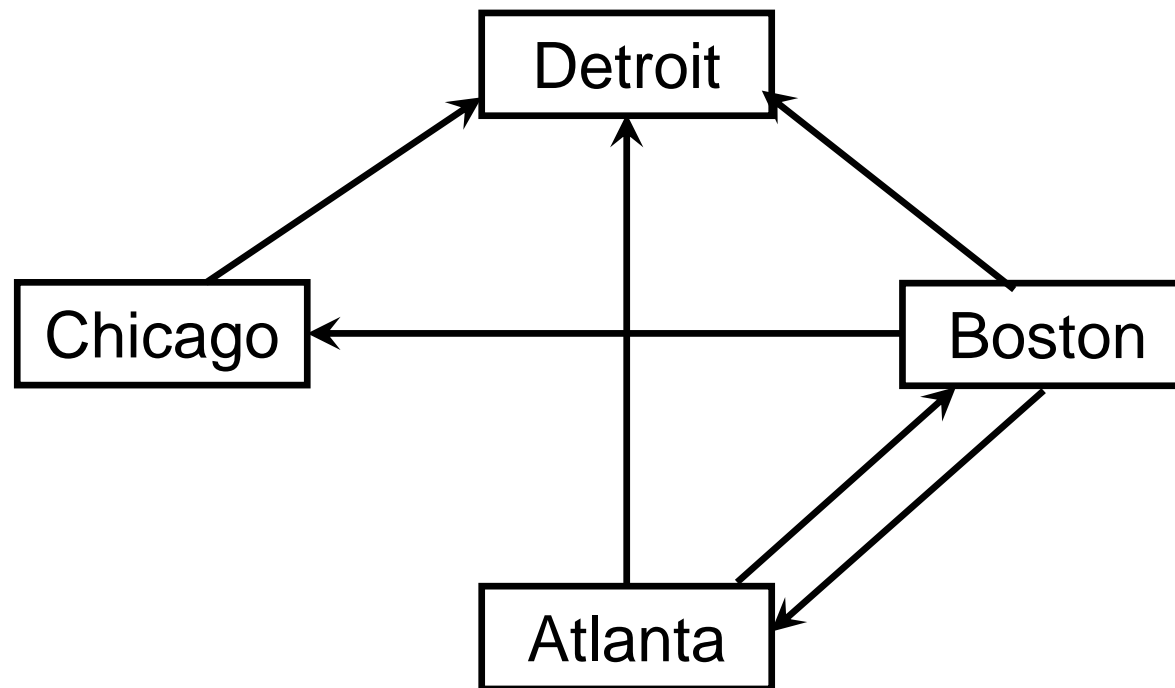
- Church'sche These:
selbst **einfache** Rechenautomaten
wie die Turing Maschine sind
berechnungsuniversell

- *in-vivo* kopiert DNA-Polymerase die Basenabfolge in einen Komplementärstrang
 - Turing-Maschine
 - Church'sche These
 - **MIT DNA LÄSST SICH RECHNEN !**

- **spontane** Paarung komplementärer Basen
- **automatisierte** DNA-Synthese und Analyse
- Molekularbiologische Verfahren:
Enzymbaukasten: Polymerasen, Ligasen, . . .
Gelelektrophorese
Affinitätsprüfungen

DNA-Computer: erste praktische Anwendung

Beispiel Hamilton'sche Wege oder
„Problem des Handlungsreisenden“:
einmaliger Besuch jeder Stadt bei **kürzestem** Weg



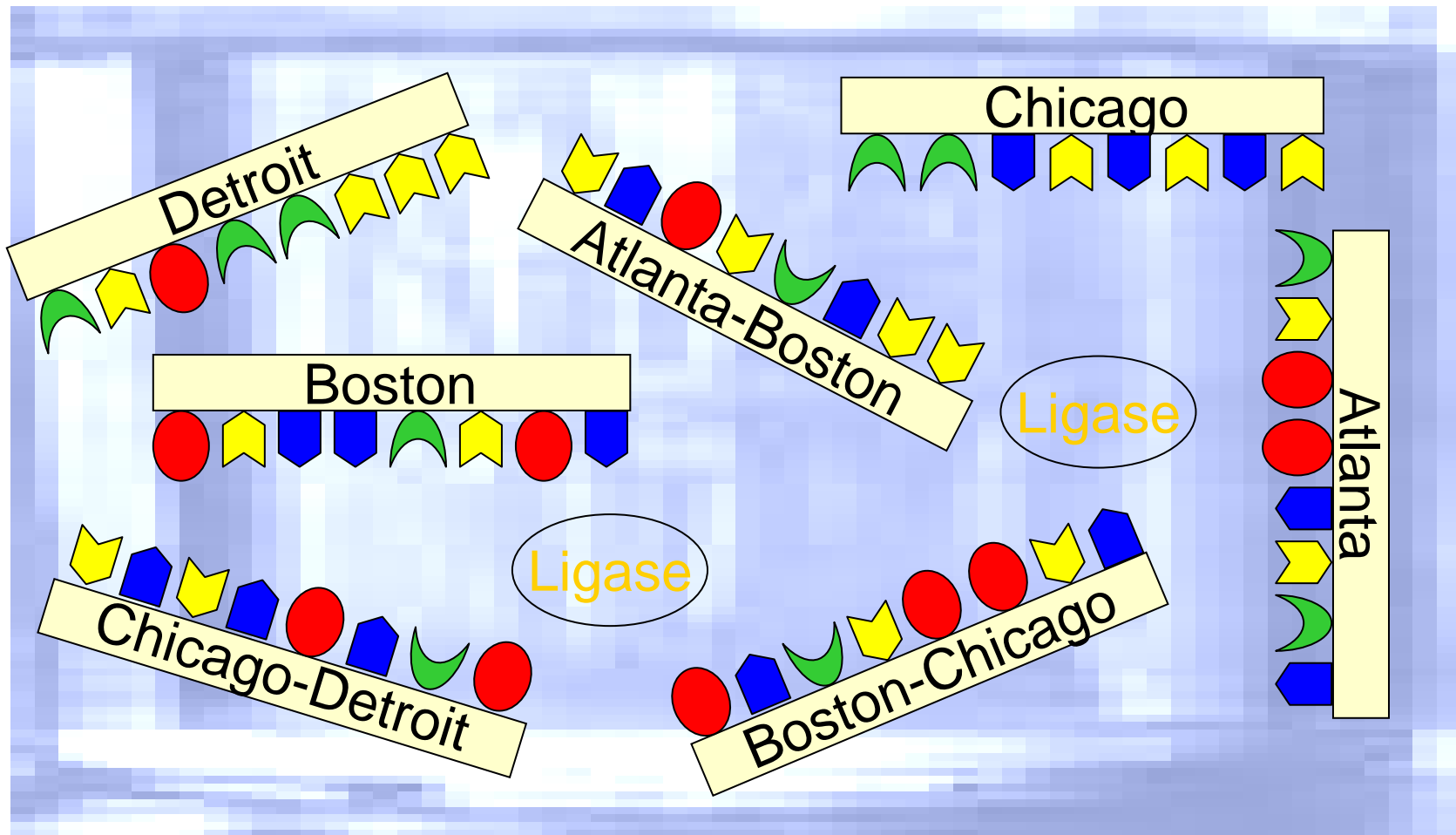
1. Erzeuge **alle** möglichen Wege durch die Landkarte (Graph).
2. Wähle Verbindungen aus, die mit dem richtigen **Startort** beginnen und dem richtigen **Zielort** enden.
3. Wähle alle Verbindungen mit der **richtigen** Anzahl von Städten aus.
4. Wähle die Verbindungen aus, die jede Stadt **genau** einmal enthalten.

- Gesucht: Weg Atlanta → Detroit
- jede Stadt durch Abfolge von 8 Basen kodiert:
Beispiel: Atlanta: **ACTTGCAG**
Boston: **TCGGACTG**
- Städteverbindungen: komplementäre Abfolge der letzten vier Basen des Ausgangsorts und der ersten 4 Basen des Zielorts:
Beispiel: Atlanta → Boston: **CGTCAGCC**

- DNA-Sequenzen für Städte und Städteverbindungen werden **maschinell** synthetisiert
- **automatisiert**
- **zuverlässig**
- **billig**

Schritt 2: Naßchemie im Reagenzglas

DNA-Computer: erste praktische Anwendung

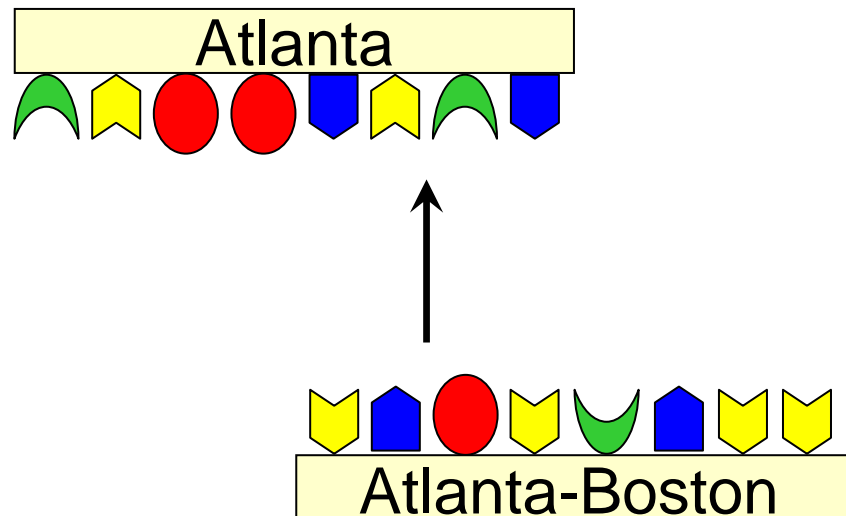


Adenin Thymin Guanin Cytosin

Schritt 2: Naßchemie im Reagenzglas

18

DNA-Computer: erste praktische Anwendung



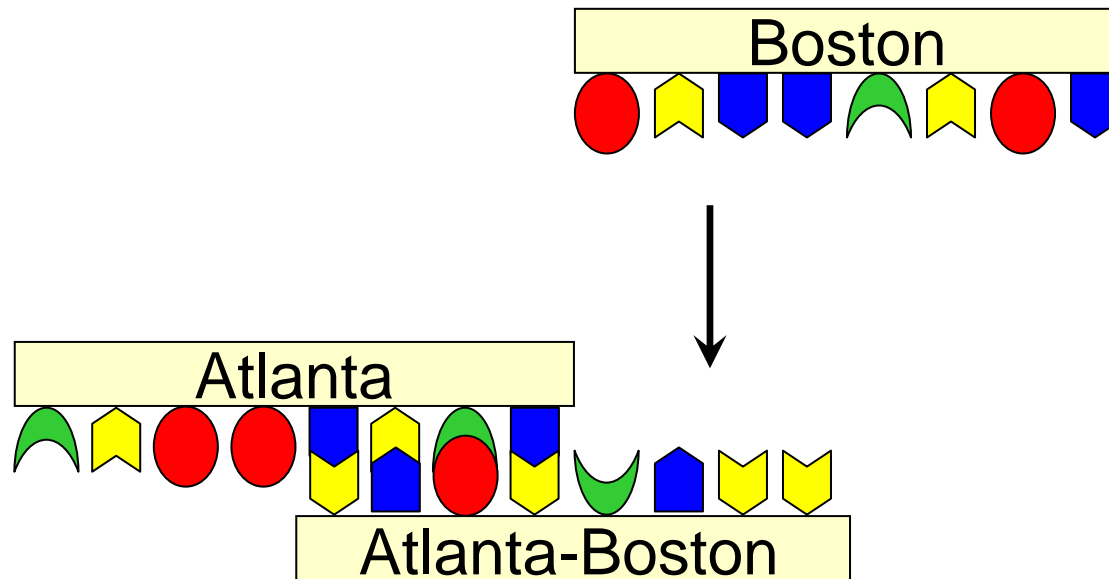
aber:

hohe **Selektivität** der **spontanen** Basenpaarung

Schritt 2: Naßchemie im Reagenzglas

19

DNA-Computer: erste praktische Anwendung

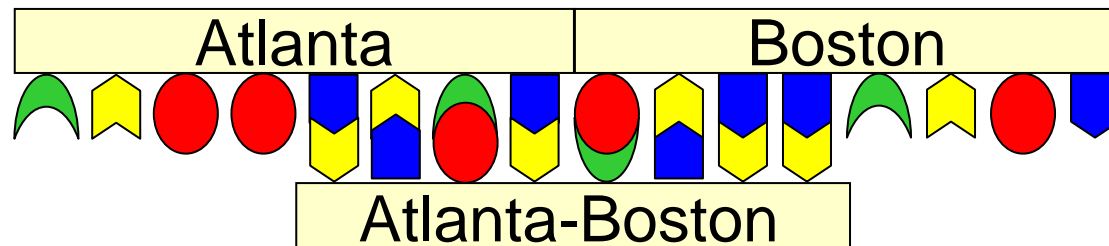


Verknüpfung zweier Städte durch **Verbindungswege**

Schritt 2: Naßchemie im Reagenzglas

20

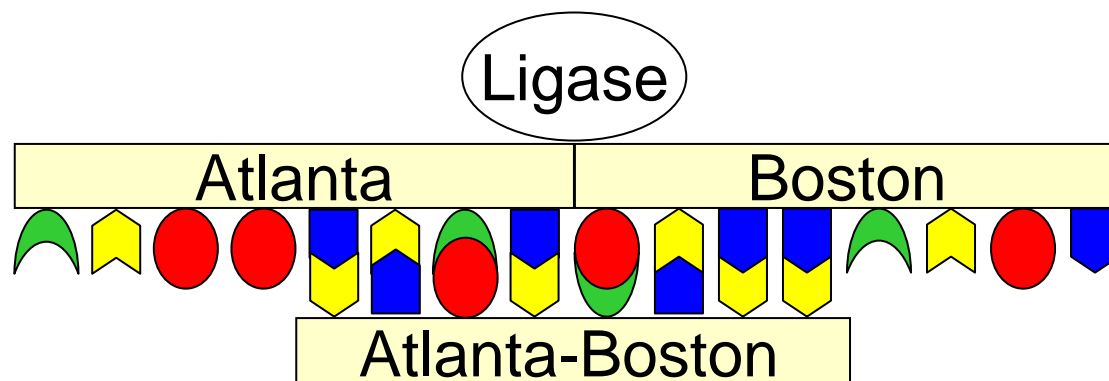
DNA-Computer: erste praktische Anwendung



Schritt 2: Naßchemie im Reagenzglas

21

DNA-Computer: erste praktische Anwendung

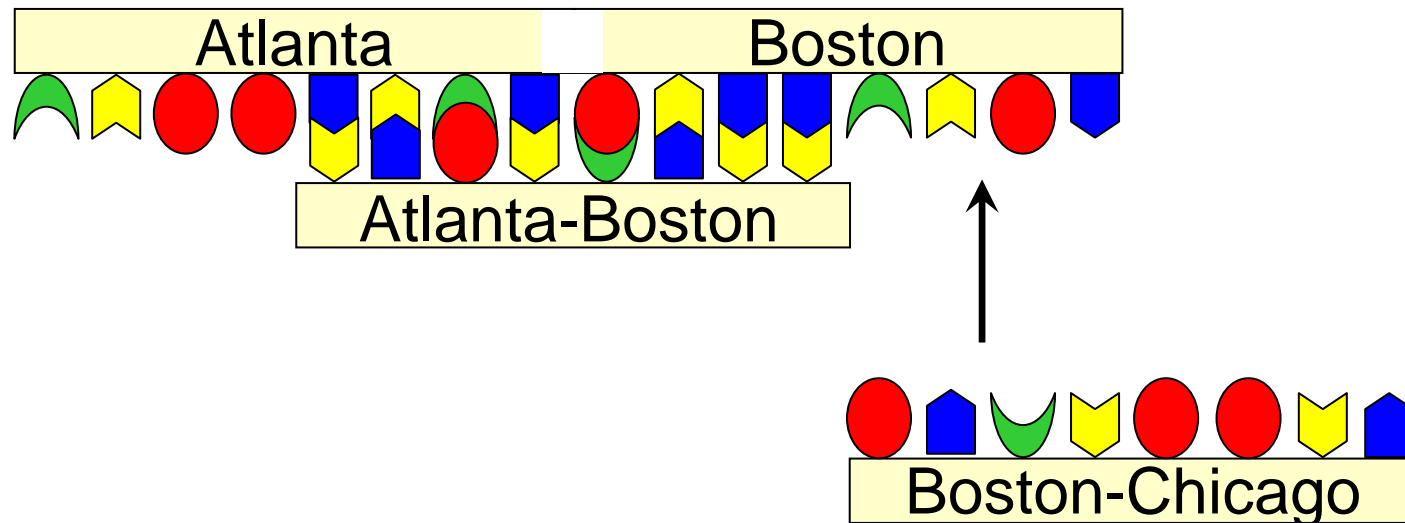


Ligasen: Enzyme die zwei DNA-Stränge **verknüpfen**

Schritt 2: Naßchemie im Reagenzglas

22

DNA-Computer: erste praktische Anwendung

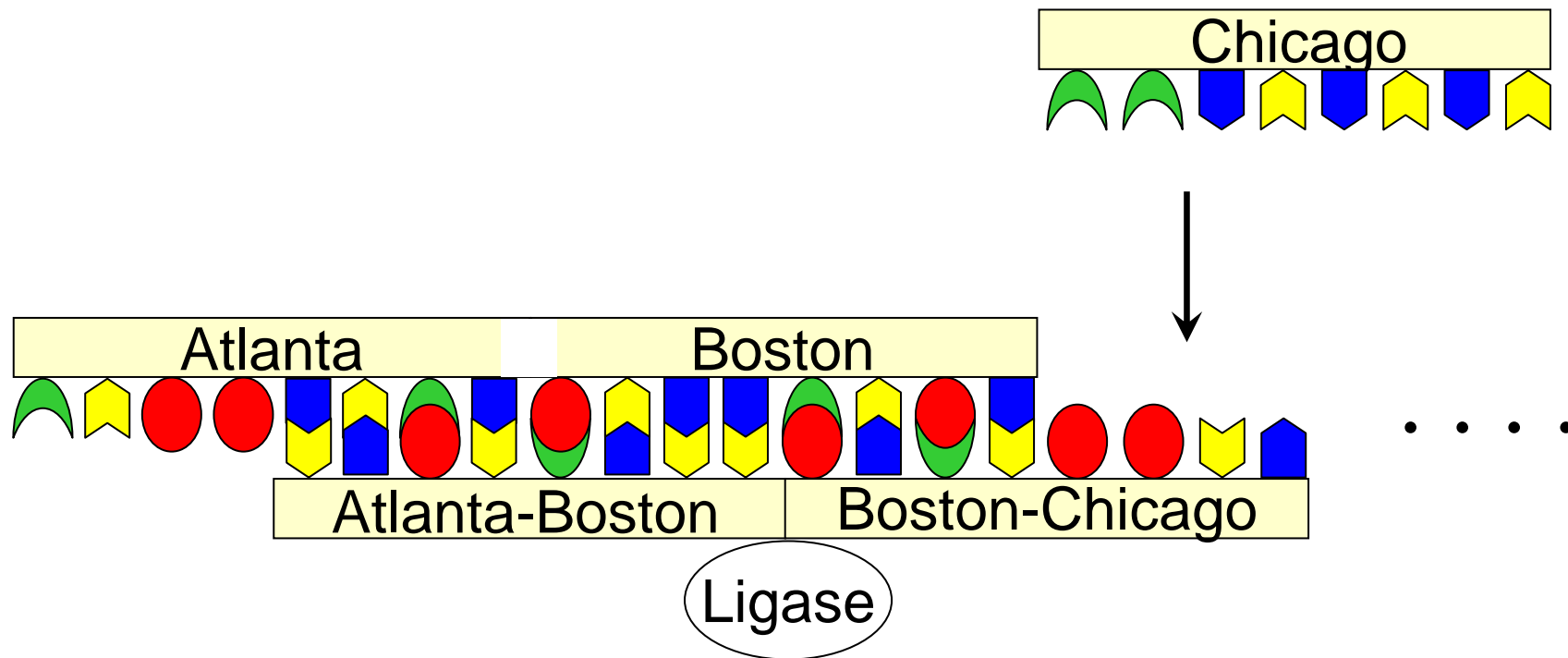


Anlagerung des nächsten Verbindungswegs

Schritt 2: Naßchemie im Reagenzglas

23

DNA-Computer: erste praktische Anwendung



Wiederholung der einzelnen Reaktionsschritte

- Eine Vielzahl **beliebiger** Reiseverbindungen ist entstanden
- Es existieren **wenig richtige** und **viel falsche** Lösungen

- **Hydrolyse** der Doppelhelix
- **Vervielfältigung** von Einzelsträngen mit richtigem Start und Zielort
- Gelelektrophorese zum **Abtrennen** der Reaktionsprodukte mit der richtigen Städteanzahl
- Affinitätsprüfung zum Nachweis des **einmaligen** Besuchs jeder Stadt

- **Sequenzanalyse** des durch Aufarbeitung isolierter DNA-Einzelstränge
- Ergibt für das gezeigte Beispiel als Reiseroute:
Atlanta – Boston – Chicago – Detroit

- Hardware:
Für die eigentliche Berechnung: **Ligase, DNA**
Zur Synthese und Analyse:
DNA-Synthesizer, DNA-Analyzer, vielfältige
Reinigungsverfahren der Biochemie, Computer
- Software:
DNA

- hohe Rechengeschwindigkeit durch **massiv parallele** Datenverarbeitung
- hohe **Informationsdichte**:
1g DNA entspricht 1.000.000.000.000 CD's
- hohe **Energieeffizienz**

- **Naßchemie** des Berechnungsvorgangs:
 - kein kontinuierlicher Rechenbetrieb
- Extrem **langwierige** Aufbereitung zur Ermittlung der Rechenergebnisse
- **Fehleranfällig**, DNA ist nicht unveränderlich (Mutation als Voraussetzung für Evolution)
- Die Berechnungszeit zur Lösung eines Problems wächst in DNA-Computern nicht exponentiell aber die **Menge** benötigter DNA tut es!

Hamilton'scher Wege im Falle 200 Städte:

notwendige DNA-Menge ist mit der Erdmasse vergleichbar!

- Vermeidung der Naßchemie und Reduktion des Aufarbeitungsaufwandes durch Chips mit DNA-Strängen, die selektiv die Rechenlösung binden
- **Automatisierung** der Aufarbeitung durch Fortschritte in DNA-Aufarbeitung und DNA-Manipulation
- allgemeine Fortschritte in der **Nanotechnologie** durch interdisziplinäre Zusammenarbeit in der weiteren Entwicklung von DNA-Computern

Der DNA-Computer steht noch ganz am Anfang. Einige Zitate zur Entwicklung des traditionellen Halbleitercomputers geben Hoffnung:

“I think there is a world market for maybe five computers”

Thomas Watson Senior,
Chairman of IBM, 1943

“There is no reason anyone would want a computer in their home”

Ken Olsen
President, Chairman and
founder of Digital, 1977

Mechanische Rechner: Zahnräder

- 1642** Blaise Pascal:
Addieren und Subtrahieren
- 1670** Gottfried Wilhelm von Leibniz:
Multiplizieren und Dividieren
- 1820** Charles Babbage:
Difference Engine, Addition und
Subtraktion fest programmiert
- 1834** Charles Babbage: Analytical Engine
Über Lochkarten frei programmierbar

Elektromechanische Relais

1938 Konrad Zuse:
Z1 - der erste Digital-computer der Welt.

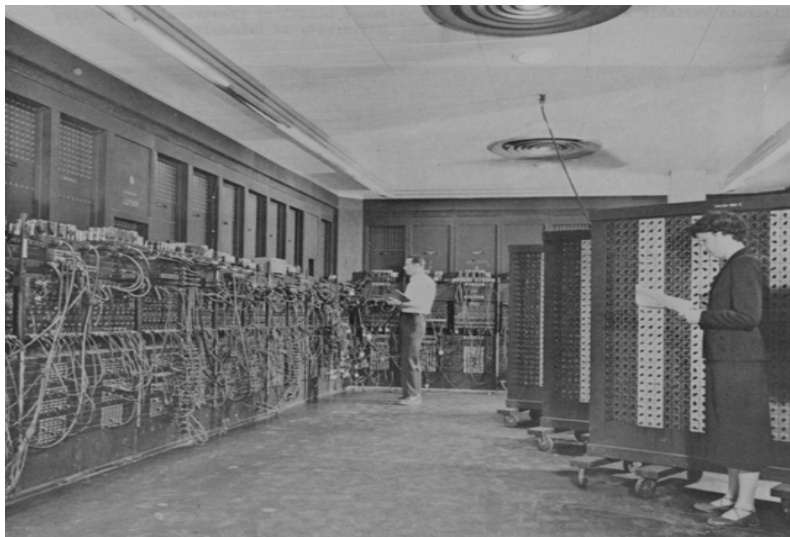
1939 Howard Aiken:
Mark I - arbeitet mit Relais und wurde mittels eines perforierten Bandes gesteuert.
Speicherkapazität: 72 Zahlen (23 Dezimalstellen)
Rechenleistung: eine Multiplikation in 10 Sekunden
Technik: ca. 5000 Relais



Konrad Zuse

Vakuumpipen

1946 ENIAC I: schnellster Rechner seiner Zeit



Entwicklungszeit: 1943 - 1946

Technik: 18.000 Röhren

Gewicht: 30 Tonnen

Länge/Breite: 30m / 3m

Ein-/Ausgabe: Lochkarten

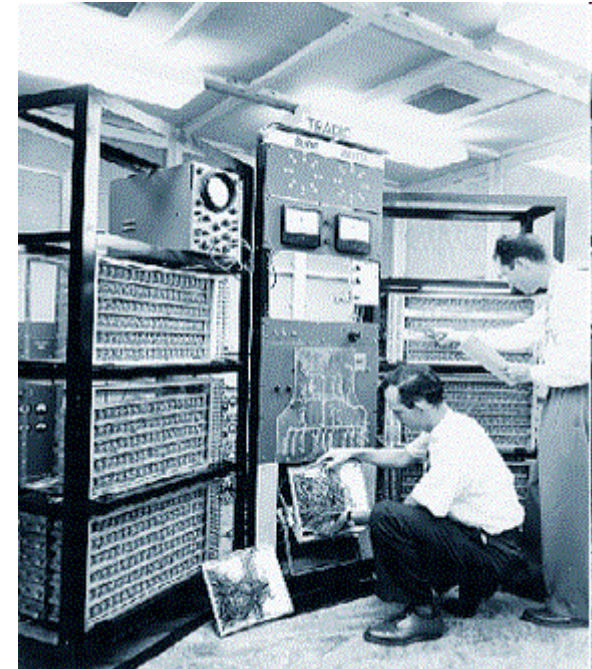
Geschwindigkeit: 5.000

Operationen pro Sekunde

Erster „bug“ am 9. Sept. 1945 um 15:45 Uhr: Eine Motte verklemmt sich in einem Relais eines Mark II Rechners.

Der Transistor

23. Dezember **1947**:
in den **Bell Laboratories** wird
der erste Transistor erfolgreich
getestet. Kommerzielle
Transistoren werden in
Glasröhren montiert eingebaut.

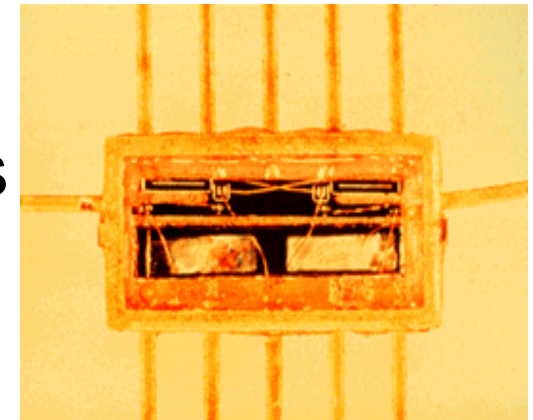


1955 TRADIC:

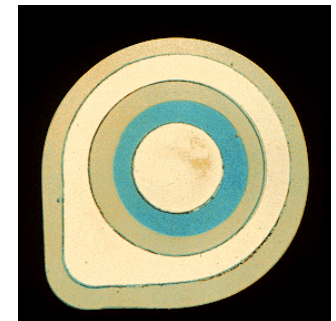
der erste voll transistorisierte Computer der Welt.
Er besitzt 800 Transistoren und ist für die damalige
Zeit sehr klein.

Der integrierte Schaltkreis

1958 Der erste integrierte Schaltkreis (IC) wird von Texas Instruments hergestellt. Seine auf einem Stück Halbleiter realisierten Komponenten sind noch mit Drähten verbunden.



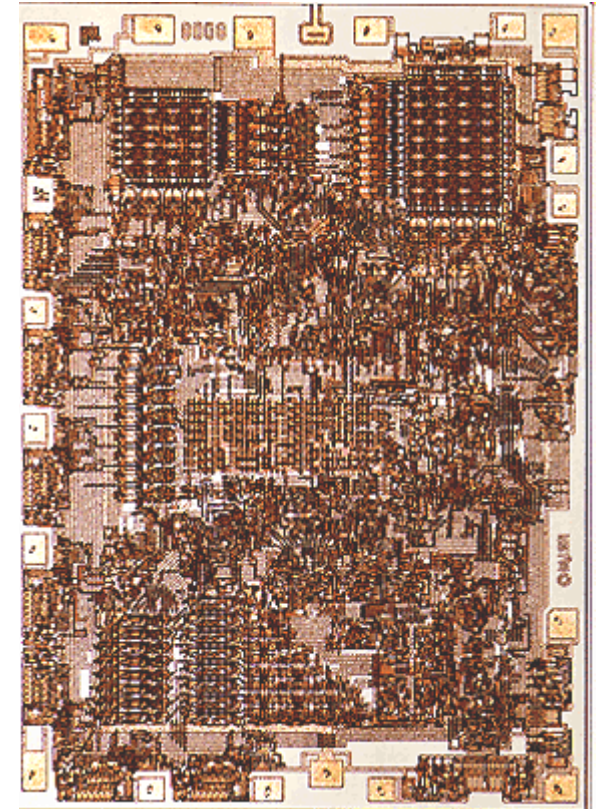
1959 Der erste Integrierte Schaltkreis ohne interne Verkabelung.



Mikroprozessoren

1972: 8008 von Intel, ein 8-bit Mikroprozessor. Erstmals konnten mit einer Wortbreite von 8 bit alle Buchstaben und Zahlen direkt verarbeitet werden.

| | |
|-------------------------|--|
| <i>Transistoren:</i> | <i>3300</i> |
| <i>Taktfrequenz:</i> | <i>1300 KHz</i> |
| <i>Befehle:</i> | <i>50</i> |
| <i>Geschwindigkeit:</i> | <i>100.000 Instruktionen pro Sekunde</i> |





100 000 000 Transistoren, Touchscreen, WLAN

- Information:
 - Angaben über Sachverhalte
 - Gekennzeichnet durch Form (Syntax) und Inhalt (Semantik)

- Daten:
 - Repräsentieren Information in einer maschinell verarbeitbaren Form
 - Die Syntax muß spezifiziert werden

- **Analoge Daten:**
- Repräsentation durch **kontinuierliche Funktionen**
 - Darstellung erfolgt durch eine physikalische Größe, die sich entsprechend den abzubildenden Vorgängen **stufenlos** ändert

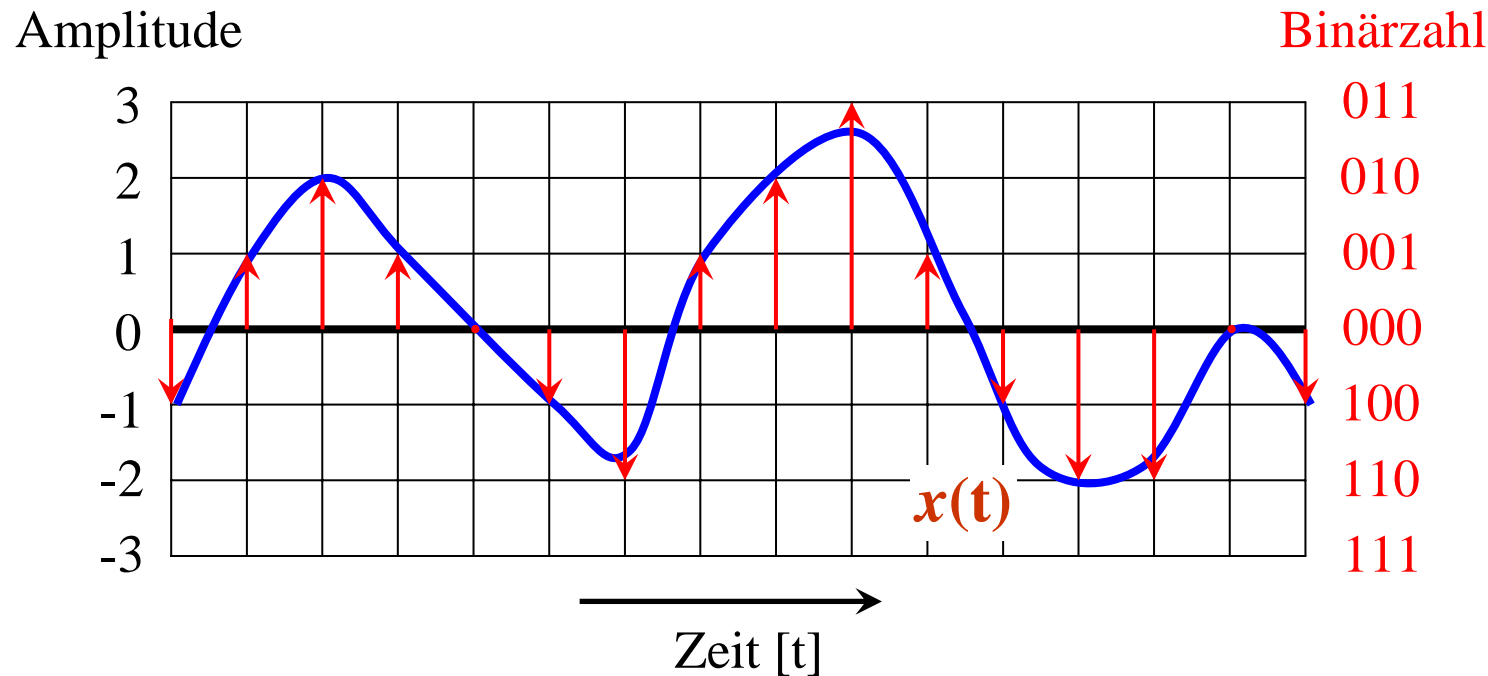
Beispiele:

- Klassische Kamera
- Thermometer mit Quecksilbersäule
- Speicherung von Musik auf einer Vinylplatte

- **Digitale Daten:**
 - Repräsentation durch **Zeichen**
 - Zeichen:= Element aus einer zur Darstellung von Information **vereinbarten** endlichen Anzahl von verschiedenen Elementen (Zeichenvorrat)

Beispiele:

- Digitalkamera
- Speicherung von Musik auf einer CD
- digitale Anzeige von Zeit, Temperatur, Gewicht



Analogdaten

wertkontinuierlich
zeitkontinuierlich

Digitaldaten

wertdiskret
zeitdiskret

= Quantisierung

= Abtastung

- **Abtastung (Sampling):**
Analoges Signal muß mit einer Frequenz abgetastet werden, die mindestens doppelt so hoch ist wie die Frequenz der Bandbreite des analogen Signals (Nyquist-Theorem)

Beispiele:

- *Sprachaufzeichnung im Spektrum von 300-3400 Hz:
Abtastrate mindestens 6800 Hz, in der Praxis 8 kHz*
- *CD-Qualität: 44,1 kHz*
- *Video: 10.3 MHz (25 Bilder · 720 Linien · 576 Zeilen)*

- Digitale Daten können komprimiert werden:
 - benötigen weniger Speicherplatz
 - Kapazität von Übertragungswegen steigt
- Digitale Daten können bei Übertragung von Störungen „**gesäubert**“ werden
 - verbesserte Qualität der Datenübertragung
- Problem:
 - **Ungenau** bei großen Digitalisierungsstufen
→ schlechtere Qualität als analoge Daten

- Menschen rechnen im Dezimalsystem
Wieso: 10 Finger?
10 verschiedene Ziffern (0,...,9)
Nachteil: schlecht in Hardware realisierbar
- **Der Computer rechnet im Binärsystem**
 - Das Binärsystem kennt nur zwei Zustände
an (**1**) und *aus* (**0**)
 - Rechenoperationen sind auch im Binärsystem mit lediglich 2 Ziffern gültig

- Es kann einfach in Hardware realisiert werden:

Schalter: ein / aus

Transistor: Spannung, keine Spannung

Festplatte: Magnetisierung Nord, Süd

RAM: Kondensator geladen, nicht geladen

ROM: Leitung verbunden, nicht verbunden

*RAM: **R**andom **A**ccess **M**emory, flüchtiger Speicher mit wahlfreiem Zugriff, z.B. Hauptspeicher*

*ROM: **R**ead **O**nly **M**emory, nichtflüchtiger Nurplesepeicher, z.B. Aufnahme von fest verdrahteten Computerprogrammen*

- Verwendung des Binäralphabets:

| Anzahl Bits | Bitkombinationen | Abbildbare Zustände |
|-------------|--|------------------------------|
| 1 | 0 1 | 2 2^1 |
| 2 | 00 01 10 11 | 4 2^2 |
| 3 | 000 001 010 011 100 101 110 111 | 8 2^3 |
| 4 | 0000 0001 0010 0100 1000 1001 1010 1100 0101 0110 0011 1110 1101 1011 0111 1111 | 16 2^4 |

Maßgrößen für Bits & Bytes

48

Informationscodierung mit Bits & Bytes

| | | |
|----------------|-------------------|---|
| 1 Kilobit | 1 Kbit = 2^{10} | = 1.024 Bits |
| 1 Megabit | 1 Mbit = 2^{20} | = 1.048.576 Bits |
| 1 Byte = 8 Bit | | |
| 1 Kilobyte | 1 KB = 2^{10} | = 1.024 Bytes |
| 1 Megabyte | 1 MB = 2^{20} | = 1.024 KB = 1.048.576 Bytes |
| 1 Gigabyte | 1 GB = 2^{30} | = 1.024 MB = 1.073.741.824 Bytes |
| 1 Terabyte | 1 TB = 2^{40} | = 1.024 GB = 1.099.511.627.776 Bytes |
| 1 Petabyte | 1 PB = 2^{50} | |

- Mit n Bit können 2^n Zeichen dargestellt werden
 - 26 Großbuchstaben: mindestens 5 Bit erforderlich ($2^5 = 32$)
 - Unter Einbezug von Kleinbuchstaben, Sonderzeichen, etc. sind 7 Bit sinnvoll ($2^7 = 128$)
 - Konventionen notwendig zur Zuordnung von Zeichen → Bitmuster

American Standard Code for Information Interchange

(In etwa: Amerikanischer Standard-Code für Informationsaustausch)

ASCII-Code umfaßt folgende Zeichentypen:

Zeichen 0 bis 31: Steuerzeichen

Zeichen 32 bis 63: Algebraische Zeichen

Zeichen 64 bis 95: Großbuchstaben

Zeichen 96 bis 127: Kleinbuchstaben

Zeichen 128 bis 255: Sonderzeichen

1963 verabschiedet durch die American Standards Organization

“Alpha“ im ASCII-Zeichensatz

54

Informationscodierung mit Bits & Bytes

| Zeichen (Schrift) | Bitmuster (Binär) | ASCII-Wert (Dezimal) |
|----------------------|----------------------|-------------------------|
|----------------------|----------------------|-------------------------|

| | | |
|---|----------|----|
| A | 01000001 | 65 |
|---|----------|----|

| | | |
|---|----------|-----|
| I | 01101100 | 106 |
|---|----------|-----|

| | | |
|---|----------|-----|
| p | 01110000 | 112 |
|---|----------|-----|

| | | |
|---|----------|-----|
| h | 01101000 | 104 |
|---|----------|-----|

| | | |
|---|----------|----|
| a | 01100001 | 97 |
|---|----------|----|

- Unicode enthält Zeichen oder Elemente aller bekannten Schriftkulturen und Zeichensysteme verwendet 2 Byte Binärcode, von den $2^{16} = 65536$ darstellbaren Zeichen sind zur Zeit 38885 Elemente spezifiziert.
- Universal Character Set (UCS, ISO 10646)
 - beruht auf Zeichenwerten des Unicode
 - Verwendung z.B. in HTML-Seiten zur Darstellung von Sonderzeichen: `@:= @`

Repräsentationsgröße von Datentypen

56

Informationscodierung mit Bits & Bytes

| Bezeichnung | Größe | Darstellung | Verwendungszweck |
|-------------|--------|---------------------------------------|--|
| Bit | 1 Bit | $2^1 = 2$ | Wahrheitswert |
| Byte | 8 Bit | $2^8 = 256$ | Schriftzeichen, Wahrheitswert, kleine Ganzzahlen |
| Halbwort | 16 Bit | $2^{16} = 65.536$ | Internationaler Zeichensatz, kleine Ganzzahlen |
| Wort | 32 Bit | $2^{32} = 4.294.967.296$ | Ganzzahlen, Gleitkommazahlen, Speicheradressen |
| Doppelwort | 64 Bit | $2^{64} = 18.446.744.073.709.551.616$ | Gleitkommazahlen hoher Genauigkeit, Verweise in Dateien > 4 GB |

Bedeutung der Datentypen für Programmiersprachen:

- Angabe des Wertebereichs
→ Menge möglicher Ausprägungen
- Zulässige Operationen auf den Wertebereichen
- Datentypüberprüfung
- Möglichkeit der Typumwandlung

DNA-Computer / traditioneller Computer

58

Vergleich DNA- / Halbleitercomputer

| | DNA Computer | IC-basierter Computer |
|----------------------------|-----------------|--------------------------|
| Datenrepräsentation | 4 Zustände | 2 Zustände |
| Datendichte | extrem hoch | gering |
| Rechengeschwindigkeit | sehr hoch | hoch |
| Energiebedarf | gering | hoch |
| Datenverarbeitungsschritte | unautomatisiert | automatisiert |
| Recheneinsatz | speziell | universell |

- DNA-Computer: innovativer Denkansatz
- praktisch noch nicht anwendbar
- neue Impulse für die Nanotechnologie
- herkömmliche Computer: digitale Daten, Datenrepräsentation durch diskrete Zustände
- Informationsrepräsentation durch standardisierte Zeichensätze
- darzustellende Information bestimmt Datentyp
- berechnungsuniversell