Material Informatik

Rechnen in Materialien

Stefan Bosse
Universität Bremen, Fachbereich Inforamtik, Bremen, Germany
29.4.2020
sbosse@uni-bremen.de

1. Überblick

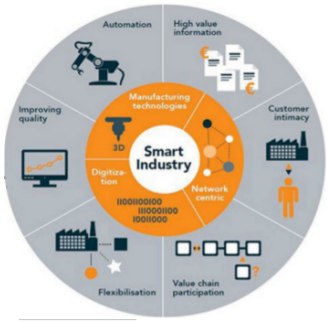
---

2. Das Internet der Dinge und Sensorische Materialien

2.1. Grundlagen und Anwendungen

IoT Einsatz in Fabriken

- Einsatz in Produktion und Fertigung für:
  - Vorhersehbare Wartung
  - Optimierung der Fertigung
  - Belegung der Maschinen
  - Planung von Arbeitern
  - Planung von Logistik
  - Energiemanagement
  - Ressourcenmanagement



[Naeem, IoTRadio, 2016]

Mit Verweis auf das zweite Dokument iot.pdf!

3. Eingebettete Systeme und Datenverarbeitung

3.1. Rechnerarchitektur - klassisch sequenziell!

Von-Neumann/Harvard Rechner

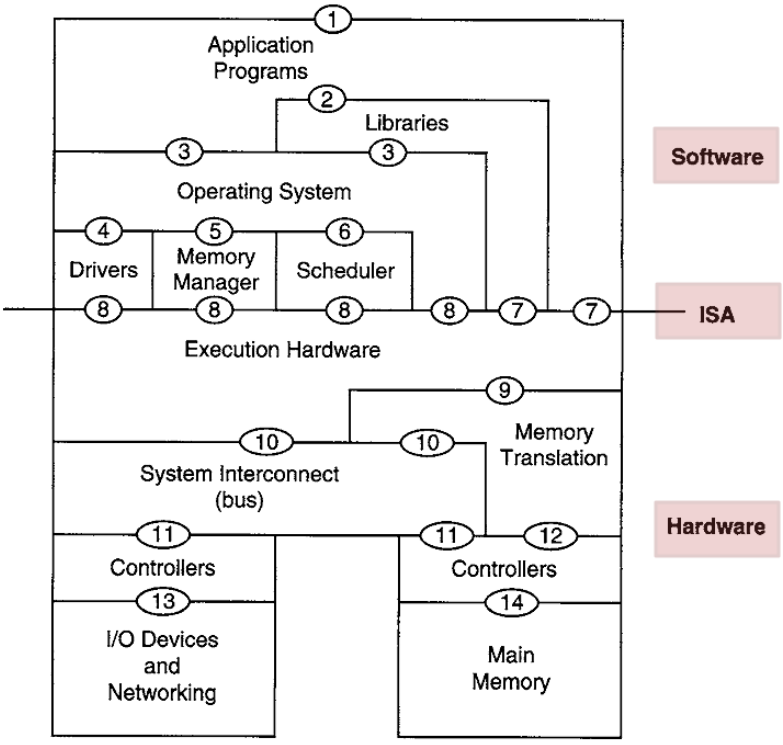


Figure 1. Computersystemarchitekturen. Implementierungsebenen kommunizieren vertikal über die gezeigten Schnittstellen. (Nach Glenford Myers, 1982) [A]

3.2. Rechnerarchitektur - zukünftig verteilt/parallel!

Zellulärer Automat

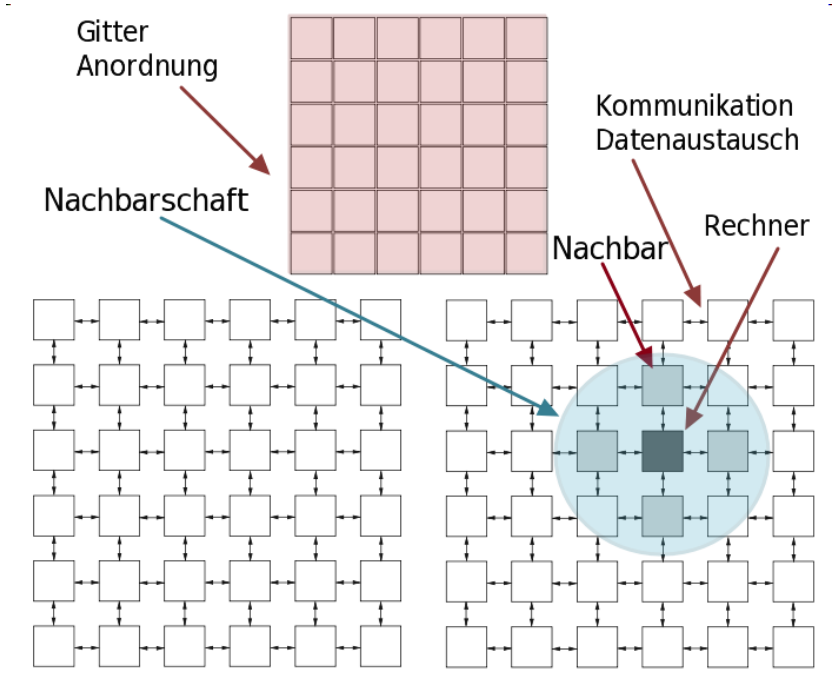


Figure 2. Zellulärer Automat

3.3. Materialinformatik

Bedeutung:

1. Rechnen in Materialien (Verteilte Datenverarbeitung in Materialien)
2. Rechnen von Materialien (Data Science, KI, .., um neue Materialien zu entwickeln)

➤ Normalerweise wird die Berechnung von Sensorerfassung und Steuerung getrennt

➤ Smarte Materialien stellen die enge Verbindung zwischen

- ❏ Berechnung,
- ❏ Kommunikation,
- ❏ Sensorerfassung und

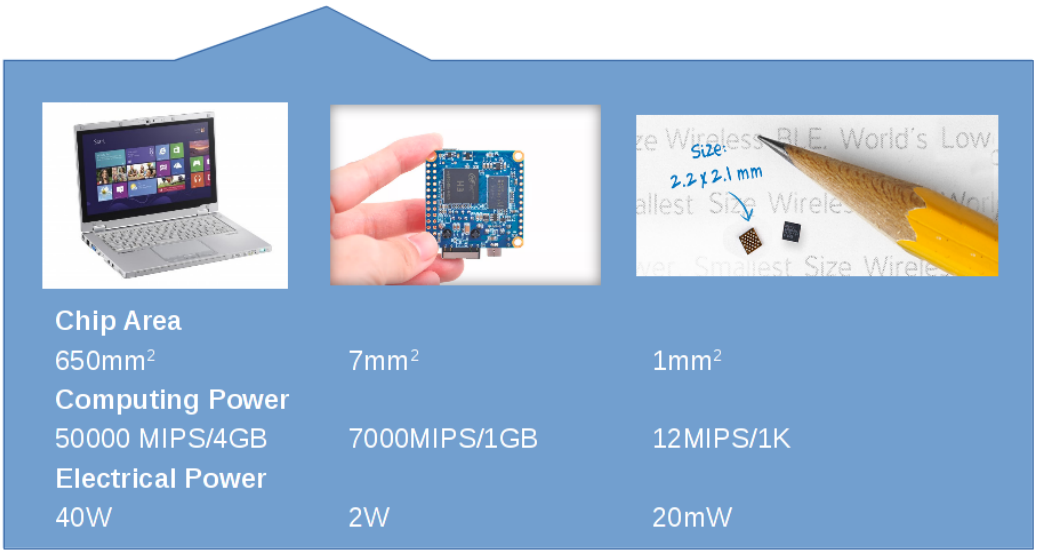
- ❏ Steuerung mit lose gekoppelten Nano-Computern dar.

➤ Algorithmische Skalierung und Verteilung sind erforderlich

- ❏ Verteilte Systeme können als eine große Maschine behandelt werden

➤ Mooresches Gesetz sagte starke Miniaturisierung voraus

➤ Jedoch die Informatik und deren Methoden konnten nicht immer folgen → Lücke zwischen Hardware und Software



Eine normalisierte Recheneffizienz eines Computers (nur unter Berücksichtigung der Datenverarbeitungseinheit) kann definiert werden durch:

$$\epsilon_C = \frac{C}{AP}$$

A

Chipfläche in mm^2

C

Rechenleistung in Mega Instructions per Second (MIPS) - oder besser in Kilo *Dhrystones/s* (KDS)!

P

Elektrische Leistungsaufnahme in W

Die Recheneffizienz kann verwendet werden, um verschiedene Computer und Geräte zu vergleichen, d.h. einen Skalierungsfaktor anzugeben:

$$s = \frac{\epsilon_1}{\epsilon_2}$$

- Neben der reinen Rechenleistung sind noch Speicher und Kommunikationsfähigkeit einer Rechneranlage wichtige Kenngrößen, so dass sich zwei weitere normierte Recheneffizienzen ergeben:

$$\epsilon_{CM} = \frac{CM}{AP}$$
$$\epsilon_{CMD} = \frac{CMD}{AP}$$

M

Gesamter Speicher in Mega Bytes (MB), beinhaltet RAM, ROM, Register

D

Gesamte Kommunikationsfähigkeit als Datendurchsatz (Mega Bit/s)!

Maßzahlen der Rechenleistung

- Einfachste Maßzahl ist die Anzahl der Integer- oder Fliesskommaoperationen pro Zeiteinheit (MIPS/FLOPS)
  - Aber nur der eigentliche Kern des Mikroprozessors wird dabei erfasst - Speicher, Speicherhierarchie und Kommunikationsfähigkeit fehlen
- Ein Programm besteht i.A. aus den folgenden High-level Operationen:
  - Berechnung (skalar, vektoriell)
  - Speicherallokation, Objekterzeugung (Code und Daten)
  - Funktionsaufrufe
  - Erzeugung, Zugriff, und Freigabe verschiedener Objekte mit unterschiedlichen “Speicherabdruck”: Arrays, Strings, Records, Funktionen, Methodische Objekte mit Prototypen (Klassen)

⇒ **Dhrystone** Benchmark umfasst alle oben genannten Operationen!

Chip Fläche

Item Size in rbe	
1 register bit (**rbe**)	1.0 **rbe**
1 static RAM bit in an on-chip cache	0.6 **rbe**
1 DRAM bit	0.1 **rbe**
rbe corresponds to (in feature size: f)	1 $\mathbf{rbe} = 675f^2$
Item A corresponds to 1 mm^2 with $f = 1\mu$.	Size in A Units
$1A$ or about	= $f^2 \times 10^6$ (f in microns) $\approx$ 1481 **rbe**

Figure 3. Normierte Maßzahlen für die Chip Fläche (A/rbe) [H]

$1A$ or about	= $f^2 \times 10^6$ (f in microns) $\approx$ 1481 **rbe**
A simple integer file (1 read + 1 read/write) with 32 words of 32 bits/word or about	 = 1444 **rbe** $\approx 1A$ ($= 0.975A$)
A 4KB direct mapped cache or about	= 23,542 **rbe** $\approx 16A$
Generally a simple cache (whose tag and control bits are less than 1/5th the data bits) uses	 $= 4A/KB$

Figure 4. Normierte Maßzahlen für die Chip Fläche von Speicher (A/rbe) [H]

Simple processors (approximation)	
A 32-bit processor (no cache and no floating point)	$= 50A$
A 32-bit processor (no cache but includes 64-bit floating point)	$= 100A$
A 32b (signal)processor, as above, with vector facilities but no cache or vector memory	$= 200A$
Area for inter-unit latches, buses, control and clocking	allow an additional 50% of the processor area.

Figure 5. Normierte Maßzahlen für die Chip Fläche von Prozessoren (A/rbe) [H]

Xilinx FPGA	
A slice (2 LUTs + 2 FFs + MUX)	= 700 **rbe**
A configurable logic block (4 slices) Virtex 4	= 2,800 **rbe** $\approx 1.9A$
A 18Kb BlockRAM	= 12,600 **rbe** $\approx 8.7A$
An Embedded PPC405 core	$\approx 250A$

Figure 6. Normierte Maßzahlen für die Chip Fläche von FPGAs (A/rbe) [H]

Feature size (in micron)	Number of A per mm^2
1.000	1.00
0.350	8.16
0.130	59.17
0.090	123.46
0.065	236.69
0.045	493.93

Figure 7. Zusammenhang der Technologiegröße f (Fertigungsauflösung/Transistordimension) mit der normierten Fläche A [H]

➤ Hitachi entwickelte bereits 2006 einen 0.15 x 0.15 Millimeter großen, 7.5 μm dicken Microchip der:

- ❏ Drahtlose Kommunikation und Energieversorgung via RFID → (2.45 GHz Mikrowelle) ermöglichtet, einen
- ❏ 128 Bit ROM Speicher, und einen
- ❏ einfachen Mikroprozessor enthielt.

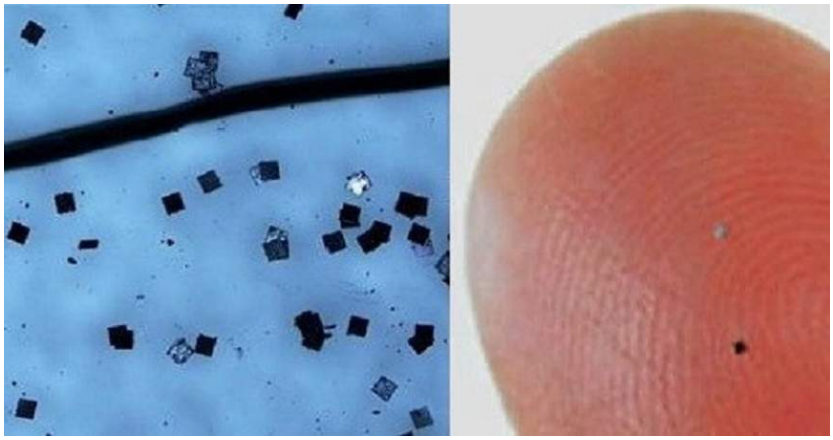


Figure 8. Es geht noch kleiner → Smart Dust reloaded (by Hitachi)

Aufgabe

1. Mache eigene Messungen mit dem *dhrstone* benchmark Test auf verschiedenen Rechnern und Virtuellen Maschinen.
2. Stelle eine Tabelle zusammen mit gängigen Computern (mobil, Desktop, Server, eingebettete Rechner, Nanorechner) mit den Daten MIPS/DPS, Speicher, Kommunikation (abgeschätzt) sowie Chip Fläche und el. Leistungsbedarf
3. Berechne die verschiedenen ϵ Parameter und vergleiche …

3.4. Entwurf Eingebetteter Systeme

Softwareentwurf

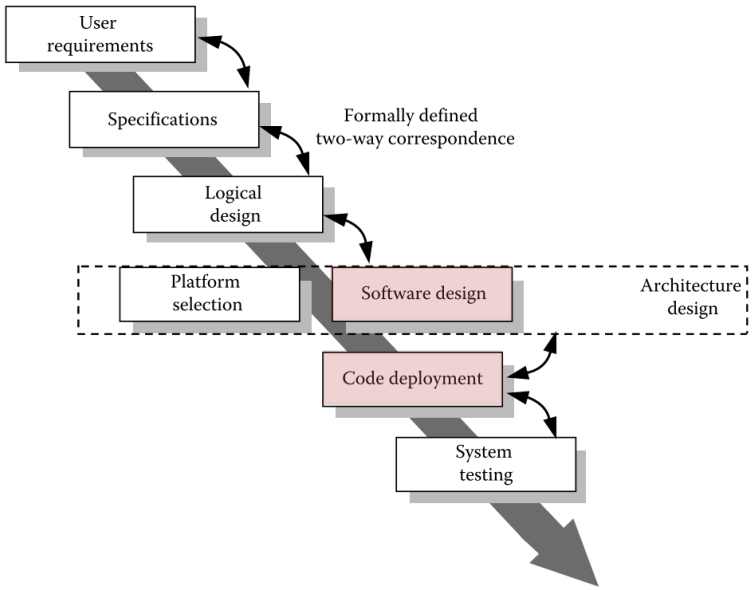


Figure 9. Entwurfsebenen von Eingebetteten Systemen auf Softwareebene

Hardwareentwurf

➤ Der Entwurf Eingebetteter Systeme geht immer mehr Richtung Hardware-Software Co-Design und System-on-Chip Architekturen

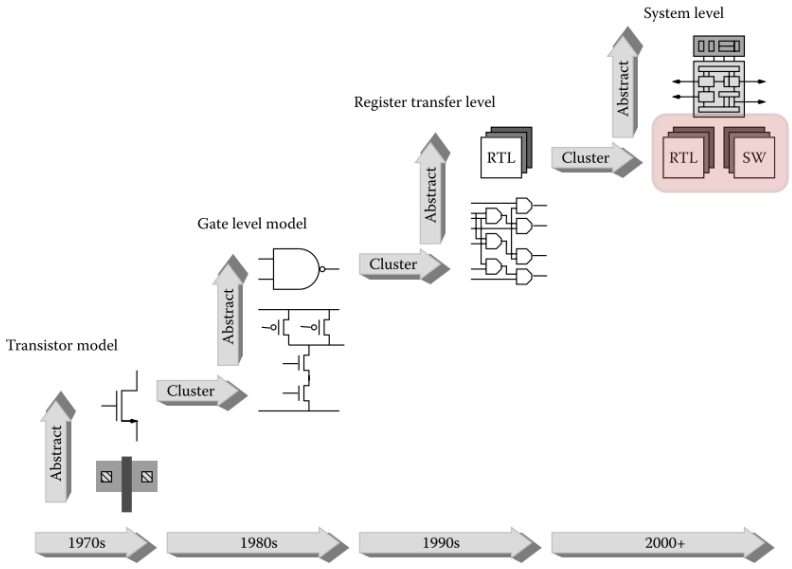


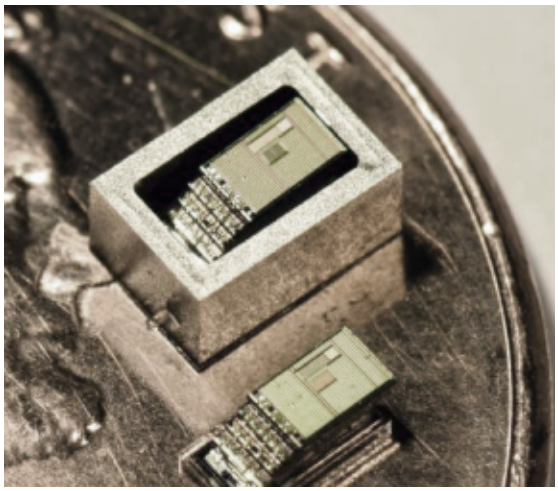
Figure 10. Entwurfsebenen von Eingebetteten Systemen auf Hardwareebene

3.5. Rechnertechnologien

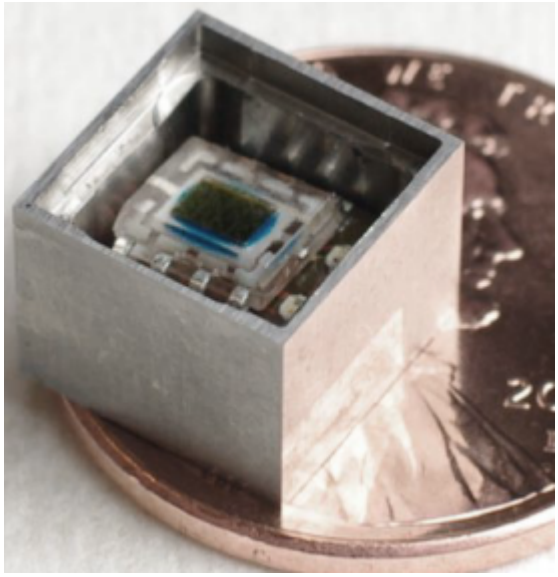
Existierende “Nano”-Computer:

- ➤ Smart Dust → Vision Millionen lose gekoppelter Nano-Computer
  - ❏ Eingebettet in Materialien
  - ❏ Auf Oberflächen verstreut
  - ❏ Dispergiert in Flüssigkeiten, Folien, ..
  - ❏ ungefähr 10mm^3 Volumen

Micro Mote M3



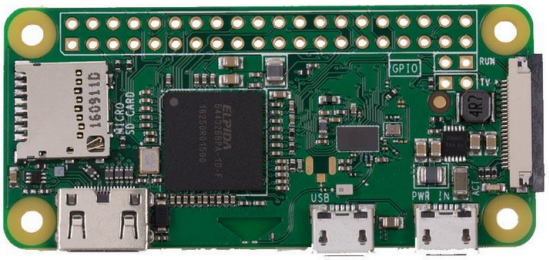
ELM System



Existierende Kleinstrechner für Sensornetzwerke mit Drahtloskommunikation (WLAN, Bluetooth)

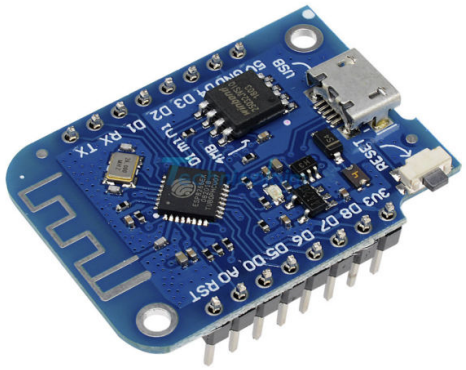
Raspberry PI Zero

- 32-bit RISC ARM Prozessor
- 512MB RAM, 16GB ROM
- 1W
- 70x30mm



ESP8266

- 32-bit RISC Tensilica Xtensa Diamond Standard
- 128kB RAM, 16MB ROM
- 0.15W
- 30x20 mm



	Micro Mote (M3)	**ELM System**	**Atmel Tiny 20**	**Freescale KL03**	**ARM Cortex Smart Phone**
Processor	Arm Cortex M0	C8051F990 (SL)	AVR	Arm Cotex M0+	Arm Cortex A9
Clock	740kHz max.	32kHz	-	48MHz	1GHz
CPU Chip Area	0.1mm^2	9mm^2	1mm^2	4mm^2	7mm^2/ROM
Sensors	Temperature		-	-	Temp, Light, Sound, Accel., Press., Magn.
Communication	900MHz radio, optical	optical	electrical	-	3G/4G, WLAN, USB, Blue-tooth, NFC
Harvester, Battery	Solar cell, Thin film	Solar cell, Coin	-	-	-
Power Consumption	70mW / CPU	160mW / CPU	20mW	3mW @ 48MHz	100mW avg.,
Manufacturing	180nm CMOS	-	-	-	40nm CMOS
Package	Wire bonded	Silicon Stack	PCB	Single Chip	Single Chip
Computing Eff. ϵ_C	**150**	**0.02**	**0.6**	**4.0**	**0.53**

Smart Dust (2000)

- ➤ Kommunikation: Optisch, Laser, LED
- ➤ Energieversorgung: Optisch, Fotozelle
- ➤ Energiespeicher: Dünnfilmbatterie
- ➤ Sensorik: Temperatur, Licht

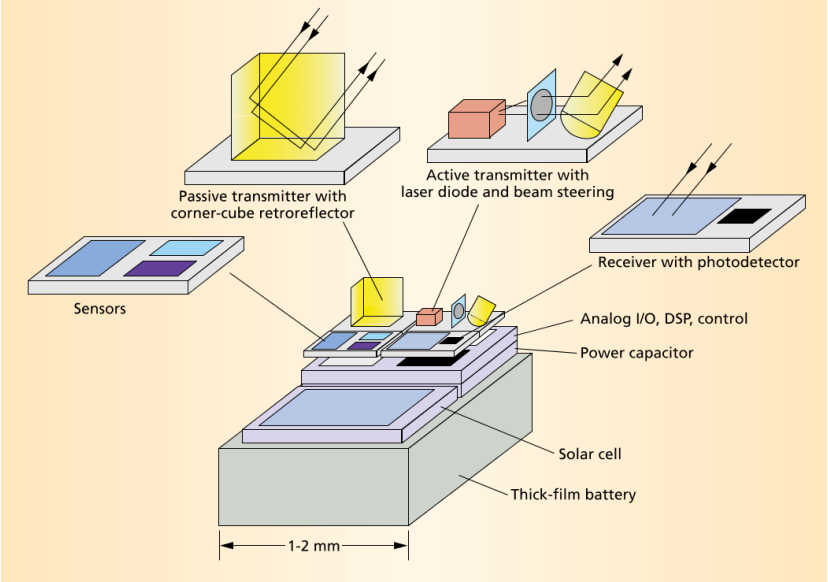
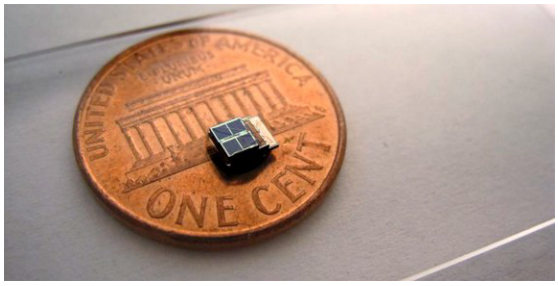


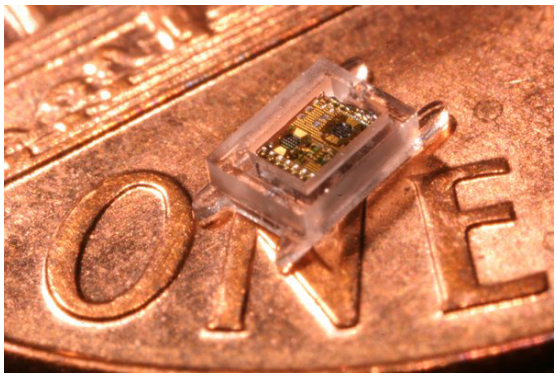
Figure 11. Aufbau einer Smart Dust Mote [Smart Dust: Warneke et al., 2001]

3.6. Rechnertechnologien

Medizinische Implantate (2011)

- University of Michigan (abgeleitet von M3 micro mote Architektur)
- f=100kHz, M=1kB, A=0.1mm^2, Drucksensor, hybrider Aufbau, Dünnfilmbatterie, Solarzelle, Kommunikation (kurzreichweitig <10 cm)(https://www.tested.com/tech/photography/1908-this-sensor-is-actually-a-tiny-computer-for-your-eyeball)





3.7. Algorithmen

- Zurück zu den Algorithmen: Wenn pro Rechnerzelle weniger als !MIPS Rechenleistung und weniger als 1kB Speicher zur Verfügung steht müssen sich die Alogorithmen und Konzepte ändern → Algorithmische Skalierung
- Verwendung der eingangs eingeführten Zellulärer Automaten als verteilte Rechneranlage!
- Die Rechnerzellen arbeiten dabei direkt auf den Sensordaten und führen einfachste Berechnungen mit Nachbardaten aus

Beispiel: Bildverarbeitung auf ZA

- Die einzelnen Pixel des Bilds (Sensoren) sind den Zellenrechnern zugeordnet
- Typische Operationen: Rauschunterdückung, Kantenfilter, Histogrammberechnung usw.
- Eine Zelle verändert immer nur seinen eigenen Datenwert durch Anwendung einfacher Regeln der Menge Φ mit Nachbarzellendaten

Das Problem: Anders als bei klassischen Algorithmen sind diese Regeln nicht bekannt! Diese Transformationsregeln werden daher häufig gelernt (d.h. die geeigneten Regeln z.B. für Kantentendetektion aus der Menge aller möglichen Kombination von Pixeloperationen ausgewählt)

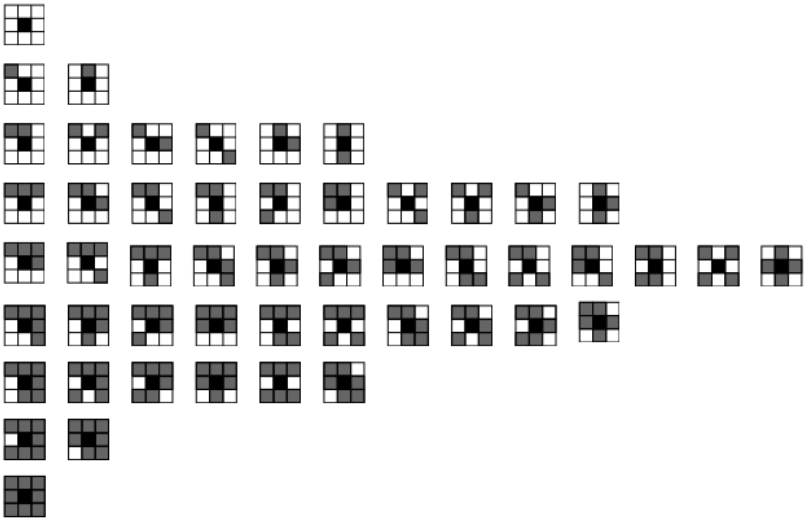


Figure 12. Alle 51 möglichen Muster (Regeln) damit ein Pixel seinen Wert (binär, SW Bild) invertiert (Mooresche Nachbarschaft mit 8 Nachbarn); gezeigt sind nur zentrale schwarze Pixel, durch Invertierung weitere 52 Regeln für weiße Pixel[Rosin,2002]

Rauschunterdrückung

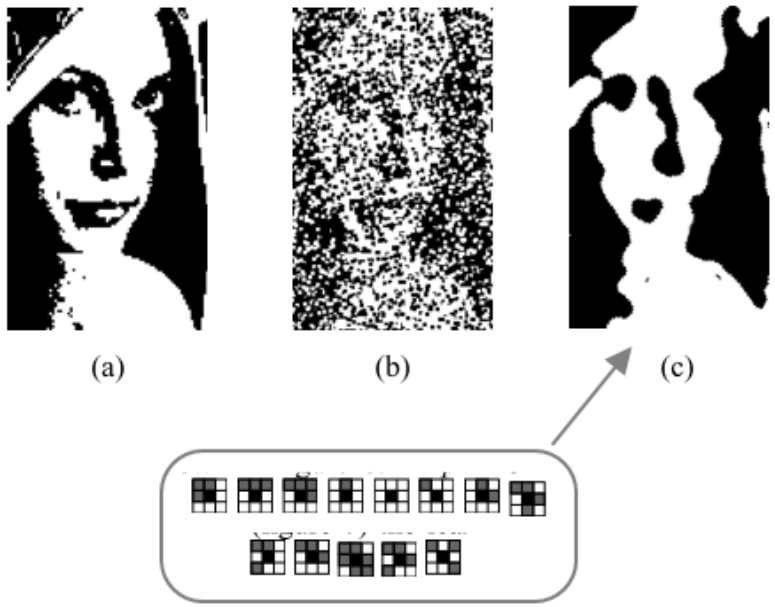


Figure 13. Rauschunterdückung durch einen binären ZA: (a) Originalbild (b) Verrauschtes Bild (c) Entrauschtes Bild mit dem unten gezeigten Regelsatz [Rosin,2002]

Kantendetektion

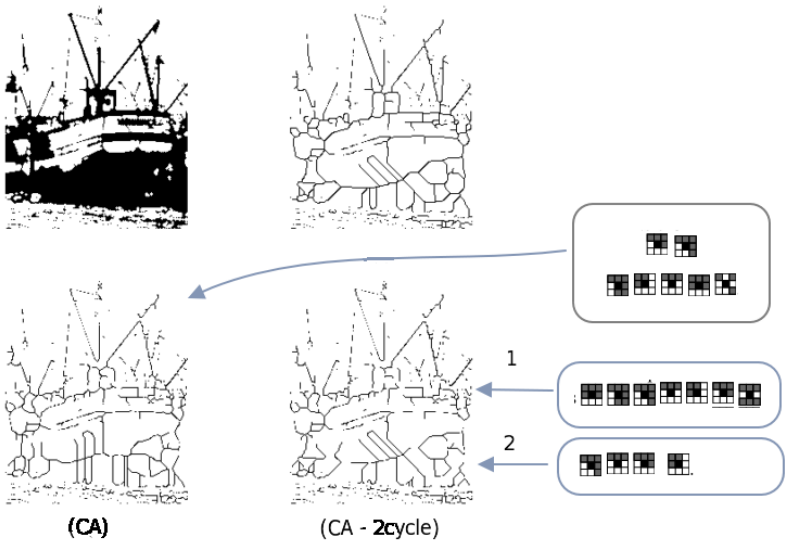


Figure 14. Kantendetektion (oben,links) Originalbild (oben,rechts) Konv. Alg. (unten, links) CA - ein Zyklus ZA (unten rechts) zwei Zyklen CA [Rosin,2002]

4. Zusammenfassung

- ➤ Sensorische Materialien für die materialintegrierte Schadendsdetektion benötigen neue Konzepte der Datenverarbeitung
- ➤ Es gibt Ähnlichkeiten zu dem Internet der Dinge
- ➤ Sensorische Materialien können zusammen mit dem IoT in Cloud Umgebungen zusammengefasst werden
- ➤ Verteilte Datenverarbeitung ist der einzige Weg für skalierbare Sensornetzwerke mit Millionen von Kleinstrechnern
- ➤ Zelluläre Automaten (obwohl schon lange bekannt) bieten einen Lösungsansatz