CSN-101 (Introduction to Computer Science and Engineering)

Lecture 21: CS vs CE; Emerging trends and applications of CSE; Algorithms for Microfluidic Biochips

Dr. Sudip Roy

Assistant Professor

Department of Computer Science and Engineering

Piazza Class Room: https://piazza.com/iitr.ac.in/fall2019/csn101

[Access Code: csn101@2019]

Moodle Submission Site: https://moodle.iitr.ac.in/course/view.php?id=45

[Enrollment Key: csn101@2019]

Plan for Lecture Classes in CSN-101 (Autumn, 2019-2020)

Week	Lecture 1 (Monday 4-5 PM)	Lecture 2 (Friday 5-6 PM)
1	Evolution of Computer Hardware and Moore's Law, Software and Hardware in a Computer	Computer Structure and Components, Operating Systems
2	Computer Hardware: Block Diagrams, List of Components	Computer Hardware: List of Components, Working Principles in Brief, Organization of a Computer System
3	Linux OS	Linux OS
4	Writing Pseudo-codes for Algorithms to Solve Computational Problems	Writing Pseudo-codes for Algorithms to Solve Computational Problems
5	Sorting Algorithms – Bubble sort, selection sort, and Search Algorithms	Sorting Algorithms – Bubble sort, selection sort, and Search Algorithms
6	C Programming	C Programming
7	Number Systems: Binary, Octal, Hexadecimal, Conversions among them	Number Systems: Binary, Octal, Hexadecimal, Conversions among them
8	Number Systems: Negative number representation, Fractional (Real) number representation	Boolean Logic: Boolean Logic Basics, De Morgan's Theorem, Logic Gates: AND, OR, NOT, NOR, NAND, XOR, XNOR, Truth-tables
9	Computer Networking and Web Technologies: Basic concepts of networking, bandwidth, throughput	Computer Networking and Web Technologies: Basic concepts of networking, bandwidth, throughput
10	Different layers of networking, Network components, Type of networks	Network topologies, MAC, IP Addresses, DNS, URL
11	Different fields of CSE: Computer Architecture and Chip Design	Different fields of CSE: Data Structures, Algorithms and Programming Languages
12	Different fields of CSE: Database management	Different fields of CSE: Operating systems and System softwares
13	Different fields of CSE: Computer Networking, HPCs, Web technologies	Different Applications of CSE: Image Processing, CV, ML, DL
14	Different Applications of CSE: Data mining, Computaional Geometry, Cryptography, Information Security	Different Applications of CSE: Cyber-physical systems and IoTs

ETE

ETE

Computer Science:

- Computer Science students learn how to build computer systems, and how to solve problems on computers and other electronic technologies using data storage and processing. Computer science students learn a variety of computer languages and computer environments, which helps them master a range of skills – from creating computer graphics, through developing and analysing numerical and mathematical algorithms and complex networks, operating systems, and building and storing databases, to improving human-computer interactions.

Computer Science:

- Computer Science is the study of how data and instructions are processed, stored, communicated by computing devices. A modern descendant of Applied Mathematics and Electrical Engineering, Computer Science deals with algorithms for processing data, the symbolic representation of data and instructions, the design of instruction languages for processing data, techniques for writing software that process data on a variety of computing platforms, protocols for communicating data reliably and securely across networks, the organization of data in databases of various types and scales, the emulation of human intelligence and learning through computer algorithms, statistical modelling of data in large databases to support inference of trends, and techniques for protecting the content and authenticity of data. Therefore, computer scientists are scientists and mathematicians who develop ways to process, interpret, store, communicate, and secure data.

Computer Engineering:

- Computer Engineering students, on the other hand, are somewhere between computer science and electrical engineering. Therefore, you'll probably find system operations and computer architecture courses in a computer engineering degree as well. However, computer engineering programs focus on the development, prototyping and design of both software and hardware, as well as the integration of the two. As a result, they put a big emphasis on the physics and manufacturing of physical devices and integrated circuits. Computer engineering students learn to master robotics, pattern recognition, speech processing and so much more.

Computer Engineering:

- Computer Engineering is the marriage of Computer Science and Electrical Engineering. It focuses on computing in all forms, from microprocessors to embedded computing devices to laptop and desktop systems to supercomputers. As such, it concerns the electrical engineering considerations of how microprocessors function, are designed, and are optimized; how data is communicated among electronic components; how integrated systems of electronic components are designed and how they operate to process instructions expressed in software; and how software is written, compiled, and optimized for specific hardware platforms. Therefore, computer engineers are electrical engineers who specialize in software design, hardware design, or systems design that integrates both.

Computer Engineering vs. Computer Science:

	Computer Science	Computer Engineering
Learning Material	• Computer languages and environments. • Develop and analyze numerical and mathematical algorithms. • Build and store databases. • Improve human-computer interactions.	• Develop both software and hardware, and integrate the two. • Emphasis on physics and manufacturing of physical devices and integrated circuits. • Master robotics, pattern recognition, speech processing
Job Opportunities	• Computer and app programming. • Develop network systems & databases. • Specialize in automation. • Develop websites. • Run quality assurance tests (QA).	• Develop and manufacture aerospace, automotive, fuel, water, medical & telecommunications systems and devices. • Develop computer architecture systems and equipment.
Further Education	• Specialize in technology (information security & systems, databases, optimization, artificial intelligence). • MBA • PhD	• Specialize in an industry. • Specialize in technology (robotics, computer networks, network development). • MBA • PhD

Computer Engineering vs. Computer Science:

- The key difference is that **computer science focuses on software**, while **computer engineering is more about the hardware**.

Computer Engineering Major

Core Curriculum:

- Electronic Circuit Design
- Principles of Modern Physics
- Artificial Intelligence & Robotics
- Essential Software Development
- Computer Architecture and Design
- Very-Large-Scale Integration (VLSI)
- Microprocessor Design & Interfacing
- Embedded Computer System Design
- Digital Signal Processing

Computer Science Major

Core Curriculum:

- Advanced Software Development
- Computational Problem-Solving
- Data Structures and Algorithms
- Operating System (OS) Design
- Advanced Mathematics inc. Calculus, Logic, Probability and Algebra
- User Interface/Experience (UI/UX) Design
- Human-Computer Interaction (HCI)
- Data Security/Defensive Programming

Computer Engineering vs. Computer Science:

Computer Engineering

Marketable Electives/Concentrations:

- Computer Aided Drafting & Design (CADD)
- Mobile Device Engineering
- Sustainable "Green" Energy
- Biomedical Engineering
- Game Hardware Design
- User Interface Design
- Micro Electro-Mechanical Systems (MEMS) Engineering

Popular Career Paths:

- Electrical Engineer
- Electronics Architect
- Computer Hardware Engineer
- Mechanical Engineer
- Microchip Architect
- Mobile Device Engineer
- Quality Control Engineer

Avg. Starting Salary:

- $70,400 per year (U.S.)

Computer Science

Marketable Electives/Concentrations:

- Mobile Application Development
- Data Mining and Business Intelligence
- Web and Multimedia Design
- Digital Security/Cryptography
- Cloud Computing
- Software Testing
- Virtualization
- Game Theory

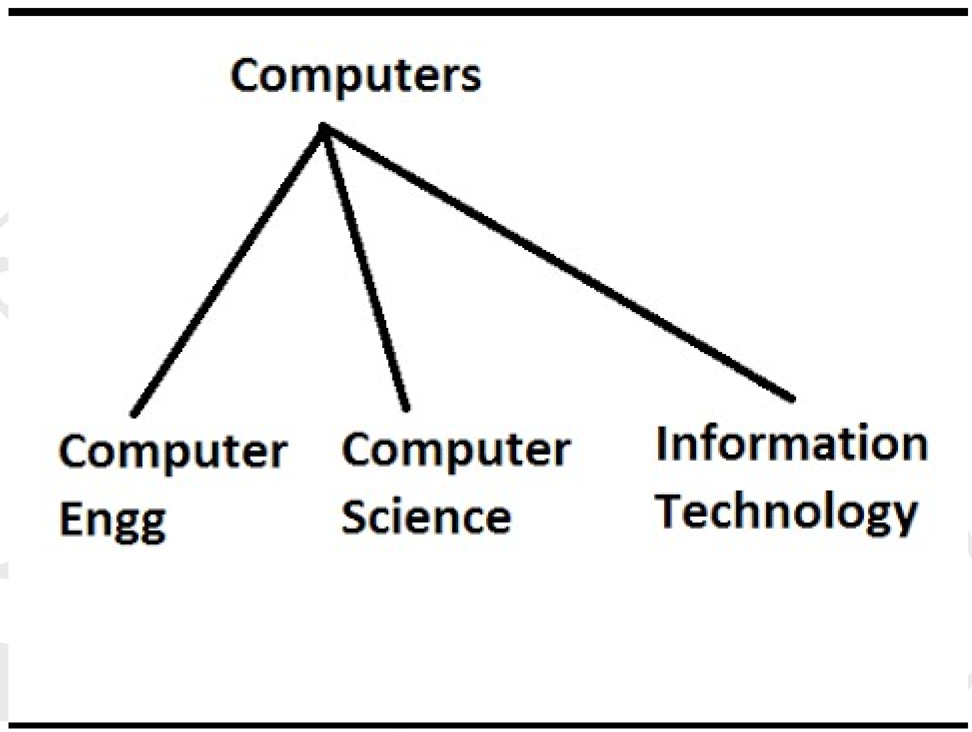
Popular Career Paths:

- Software Engineer
- Mobile App Developer
- UI/UX Designer
- Computer Scientist
- Wed Designer
- Data Scientist
- Systems Analyst

Avg. Starting Salary:

- $64,400 per year (U.S.)

CE, CS and IT:


Computers
Computer Engg
Computer Science
Information Technology

BTech Curriculum:

1st Year

S. No.	Subject Code	Course Title
1.	MA-001	Mathematics-I
2.	PH-005	Electrodynamics and Optics
3.	CE-105	Introduction to Environmental Studies
4.	HS-101A	Communication Skills (Basic)
	HS-101B	Communication Skills (Advance)
5.	HS-002	Ethics and Self Awareness
6.	CS-101	Introduction to Computer Science and Engineering
7.	CS-103	Fundamentals of Object Oriented Programming
		T
1.	MA-010	Optimization Techniques
2.	PH-006	Quantum Mechanics and Statistical Mechanics
3.	EC-104	Digital Logic Design
4.	CS-102	Data Structures
5.	CS-106	Discrete Structures
6.	EC-102	Fundamentals of Electronics
		T

2nd Year

S. No.	Subject Code	Course Title
1.	MI-106	Engineering Thermodynamics
2.	EC-203	Signals and Systems
3.	CS-221	Computer Architecture and Microprocessors
4.	CS-261	Data Structures Laboratory
5.	CS-291	Object Oriented Analysis and Design
6.	HS-ELE	HSS Elective Course*
		T
1.	MT-105	Electrical and Electronic Materials
2.	CS-212	Design and Analysis of Algorithms
5.	CS-232	Operating Systems
3.	CS-252	System Software
4.	CS-254	Software Engineering
6.	EC-252	Digital Electronic Circuits Laboratory
7.	HS-ELE	HSS Elective Course*
		T

BTech Curriculum:

3rd Year

S. No.	Subject Code	Course Title
1.	CS-341	Computer Networks
2.	CS-351	Data Base Management Systems
3.	CS-353	Theory of Computation
4.	CS-361	Computer Networks Laboratory
5.	CS-ELE1	Departmental Elective Course-I
6.	OEC/BM-ELE	Open Elective Course/Management Studies Elective Course*
7.	CS-391	Technical Communication
		T
1.	CS-300	Industry-oriented Problem / Lab-based Project / Software Engineering-based Project
2.	CS-312	Principles of Programming Languages
3.	CS-352	Compiler Design
4.	CS-362	Compiler Laboratory
5.	CS-ELE2	Departmental Elective Course-II
6.	OEC/BM-ELE	Open Elective Course/Management Studies Elective Course*
7.	MSC1/DHC1	Minor Specialization Course-I / Departmental Honours Course-I
8.	CS-399	Educational Tour

4th Year

S. No.	Subject Code	Course Title
1.	CS-ELE3	Departmental Elective Course-III
2.	CS-ELE4	Departmental Elective Course-IV
3.	CS-499	Training Seminar
4.	CS-400A	B.Tech. Project
5.	MSC2/DHC2	Minor Specialization Course-II Departmental Honours Course-II (optional)
6.	MSC3/DHC3	Minor Specialization Course-III / Departmental Honours Course-III (optional)
1.	CS-ELE5	Departmental Elective Course-V
2.	CS-ELE6	Departmental Elective Course-VI
3.	CS-400B	B.Tech. Project (Contd. from Autumn Semester)
4.	MSC4/DHC4	Minor Specialization Course-IV / Departmental Honours Course-IV (optional)
5.	MSC5/DHC5	Minor Specialization Course-V / Departmental Honours Course-V (optional)

Core Subjects of Computer Science:

- Computer Architecture and Microprocessors
- Data Structures
- Discrete Mathematics
- C Programming
- Object Oriented Programming
- Design and Analysis of Algorithms
- Operating Systems
- Computer Networks
- Software Engineering
- System Softwares
- Theory of Computation
- Formal Languages and Automata
- DBMS
- Compiler Design
- Principles of Programming Languages

BTech Curriculum: Elective Courses

- Probability Theory for Computer Engineers
- Artificial Intelligence
- Embedded Systems
- Computer Graphics
- Digital Image processing
- Machine Learning
- Information Retrieval
- Mobile & Pervasive Computing
- Principle of Digital Communication*
- Network Programming
- Software Project Management
- Data Mining
- Information Theory*
- Computer Vision
- Human Computer Interaction
- Information and Network Security
- Wireless Networks*
- Parallel and Distributed Algorithms
- Bio-informatics
- VLSI Design*
- Social Network Analysis
- Intrusion Detection Systems
- Quantum Computing
- Cloud Computing
- Cognitive Systems
- Multimedia Technologies
- Structure of Information Network
- Advanced Graph Theory
- Advanced Automata Theory
- Advanced Database management Systems
- Advanced Computer Architecture
- Advanced Algorithms
- Advanced Computer Networks
- Modelling & Simulation
- High Performance Computing
- Formal Methods and Software verification
- Advanced topics in Software Engineering
- Advanced Data Mining
- Logic and Automated Reasoning
- Big Data Analytics
- Computational Geometry
- Distributed Systems
- Algorithms and Foundations for Chip Design

Topics of Term Projects: Emerging Technologies of CSE:

1. Artificial Intelligence
2. Approximation Algorithms
3. Audio Processing
4. Automated Reasoning
5. Big Data Analytics
6. Bioinformatics
7. Bio-inspired computing
8. Cloud Computing
9. Cognitive Systems
10. Combinatorial Optimization
11. Compilers
12. Computational Biology
13. Computational Chemistry
14. Computational Geometry
15. Computational Physics
16. Computer Graphics
17. Computer Vision
18. Computer-Aided-Design (CAD)
19. Cryptography
20. Cyber security
21. Cyber-Physical Systems
22. Data Mining
23. Data Structures
24. Data Warehousing
25. Digital Communication
26. Digital Geometry
27. Digital Signal Processing
28. Distributed Algorithms
29. Distributed Computing
30. Distributed Systems
31. Edge Computing
32. Electronic Design Automation (EDA)
33. Embedded Systems
34. Evolutionary computing
35. Field-programmable gate-arrays (FPGAs)

Topics of Term Projects: Emerging Technologies of CSE:

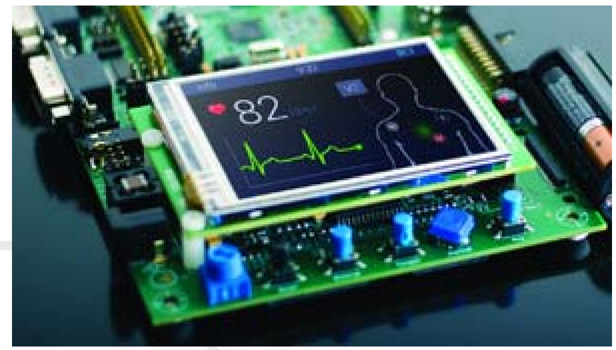
36. Files vs. Databases
37. Fog Computing
38. Formal Languages and Automata
39. Formal methods
40. Functional programming
41. Fuzzy Logic
42. Game theory
43. Genetic Algorithm
44. Geographical Information Systems (GISs)
45. Graph Theory
46. High Performance Computing (HPC)
47. Human Computer Interaction
48. Image Processing
49. Information and Coding Theory
50. Information and Network Security
51. Information Retrieval
52. Internet-of-Things (IoTs)
53. Intrusion Detection Systems
54. Job Scheduling
55. Large-Scale Networking
56. Logic programming
57. Machine Learning
58. MANET (Mobile-Area Network)
59. Mathematical Logic
60. Microprocessors and Microcontrollers
61. Mobile Computing
62. Modelling and Simulation
63. Multimedia Technologies
64. Number theory
65. Numerical analysis
66. Online Algorithms
67. Operation Research
68. Optical Computing
69. Optimization Techniques

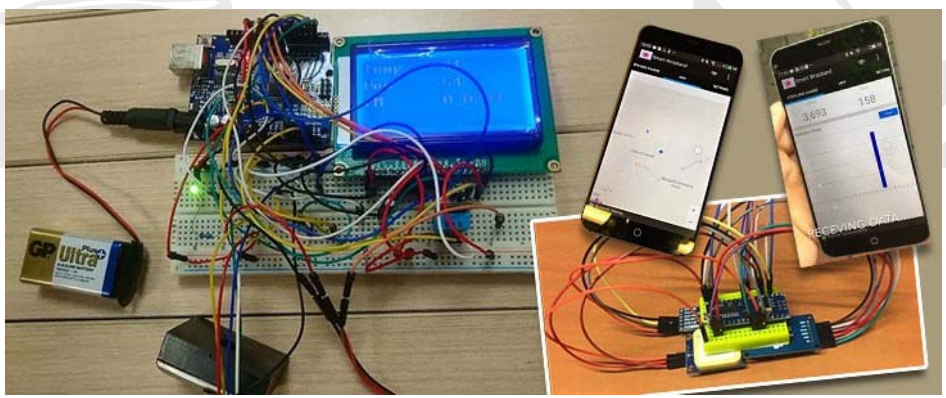
Topics of Term Projects: Emerging Technologies of CSE:

70. Parallel Algorithms
71. Parallel Computing
72. Pattern Recognition
73. Pervasive Computing
74. Principles of Programming Languages
75. Procedural programming
76. Quantum Computing
77. Reverse engineering
78. Robotics
79. Smart Systems
80. Social Area Networks
81. Soft computing
82. Software Engineering
83. Software Project Management
84. System Softwares
85. Testing and Verification of Hardwares
86. Testing and Verification of Softwares
87. Theory of Computation
88. VANET (Vehicular-Area Network)
89. Very-Large-Scale-Integration (VLSI) Circuits
90. Video Processing
91. Wireless Networks
92. Wireless Sensor Networks (WSNs)

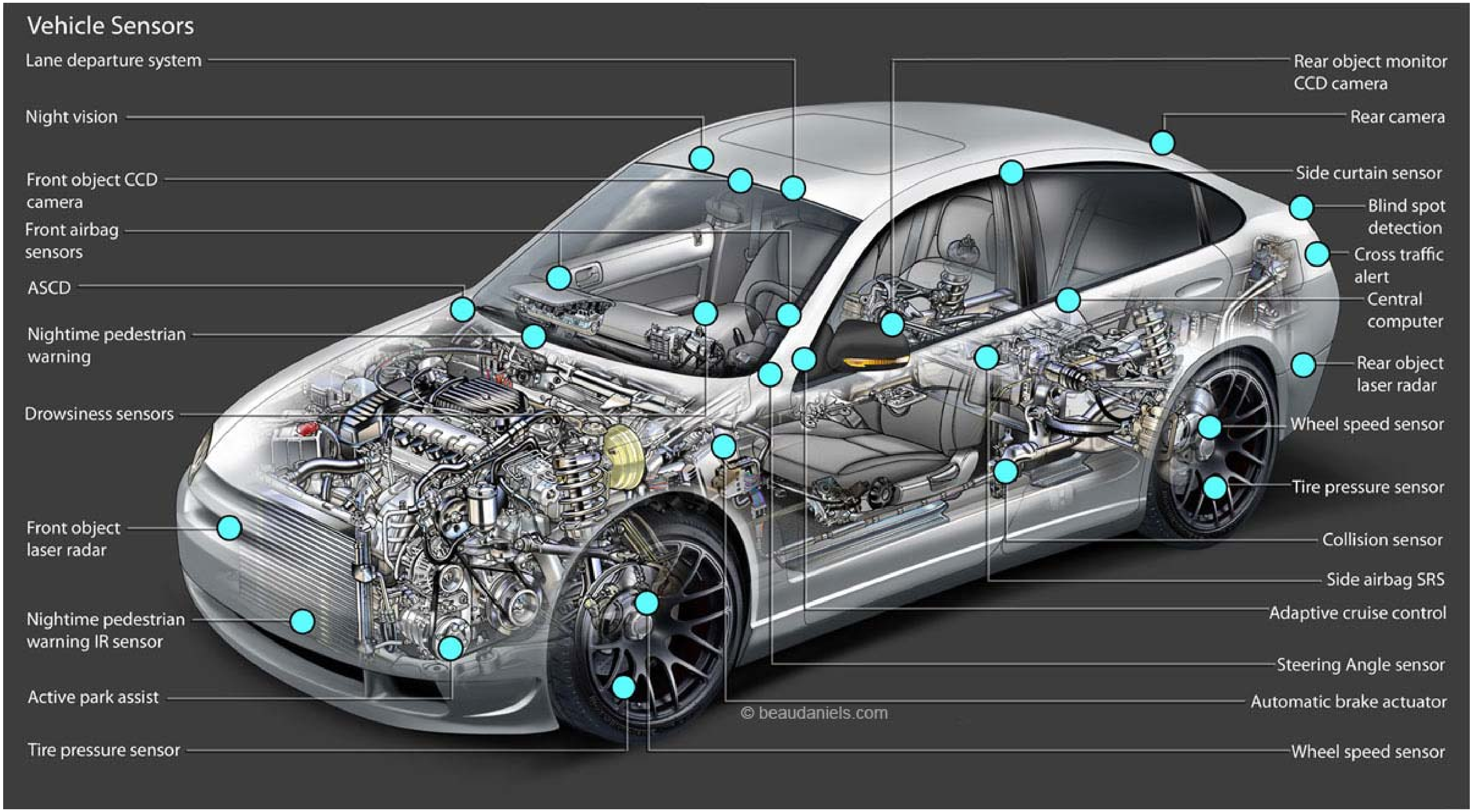
Embedded Systems:

- An **embedded system** is a computer **system** with a dedicated function within a larger mechanical or electrical **system**, often with real-time computing constraints. It is **embedded** as part of a complete device often including hardware and mechanical parts. **Embedded systems** control many devices in common use today.



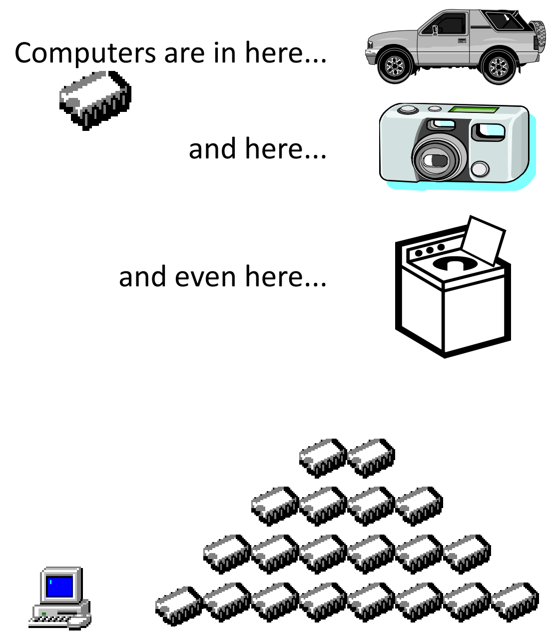


Sensors and ECUs in Modern Cars:



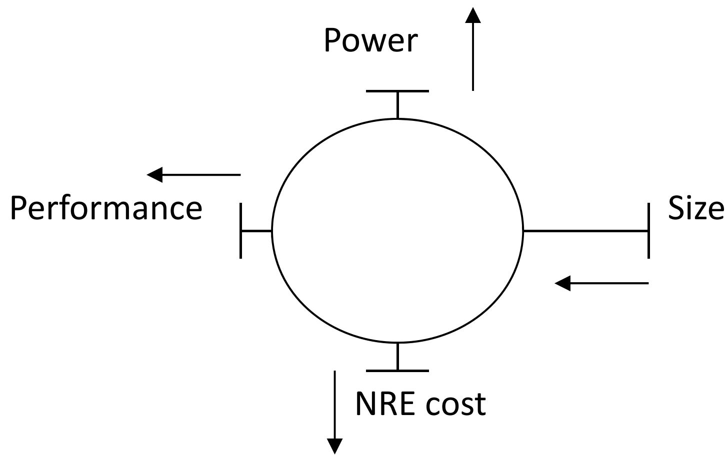
Embedded Systems: overview

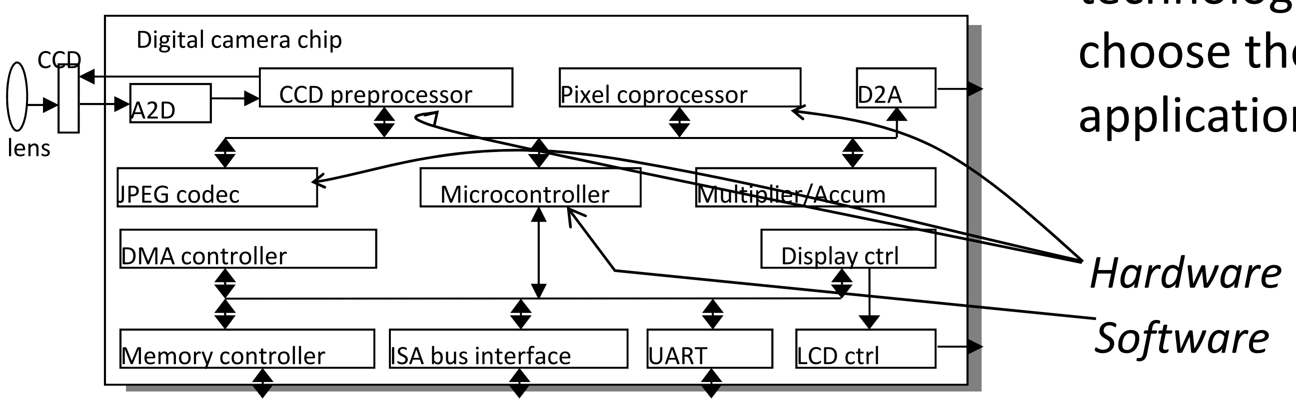
- Embedded computing systems
  - Computing systems embedded within electronic devices
  - Hard to define. Nearly any computing system other than a desktop computer
  - Billions of units produced yearly, versus millions of desktop units
  - Perhaps 50 per household and per automobile



Lots more of these, though they cost a lot less each.

Design metric competition: improving one may worsen others





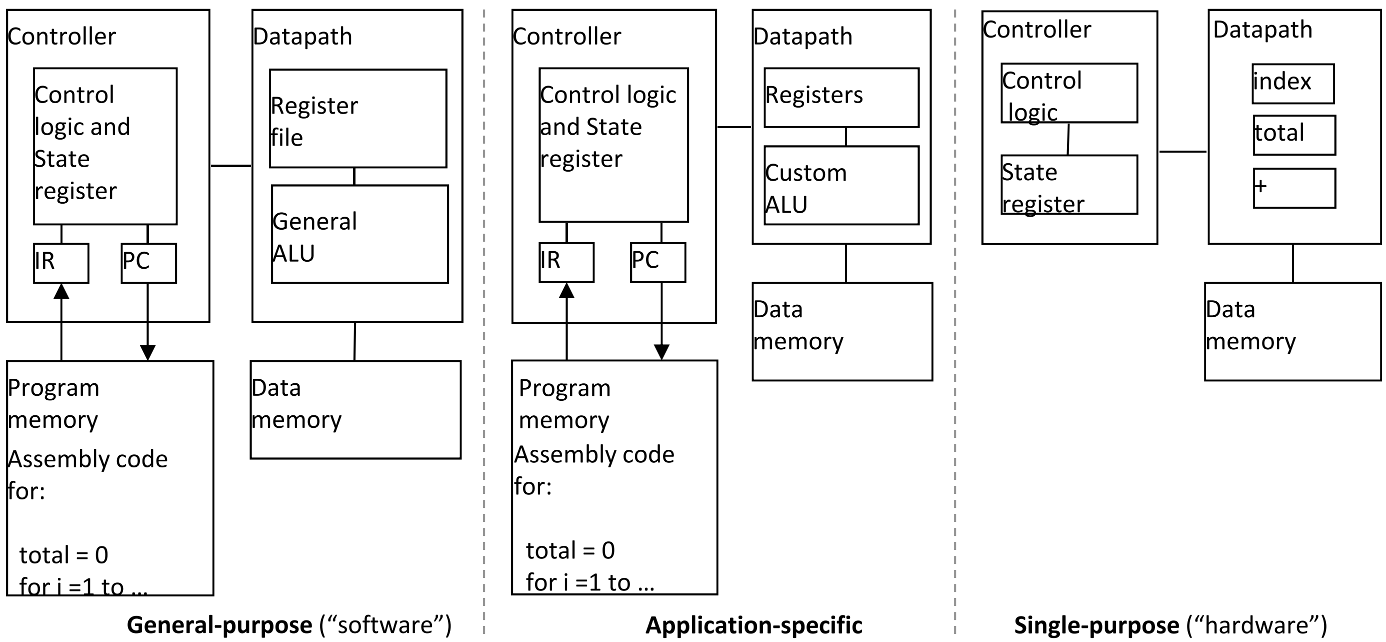
- Expertise with both **software and hardware** is needed to optimize design metrics
  - Not just a hardware or software expert, as is common
  - A designer must be comfortable with various technologies in order to choose the best for a given application and constraints

Three key embedded system technologies

- Technology
  - A manner of accomplishing a task, especially using technical processes, methods, or knowledge
- Three key technologies for embedded systems
  - Processor technology
  - IC technology
  - Design technology

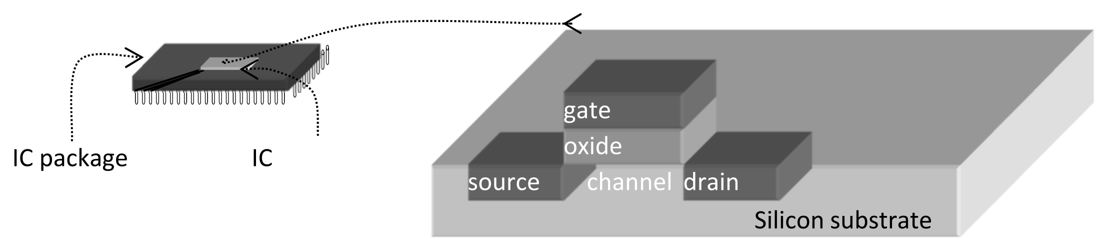
Processor technology

- The architecture of the computation engine used to implement a system's desired functionality
- Processor does not have to be programmable
  - "Processor" *not* equal to general-purpose processor



IC technology

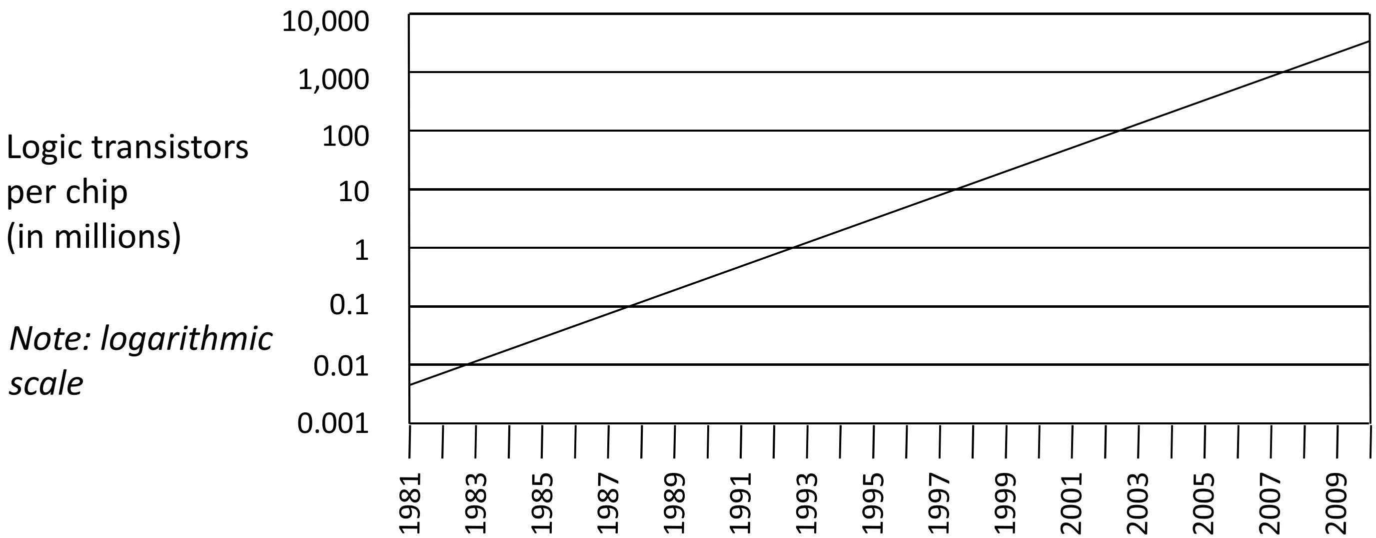
- The manner in which a digital (gate-level) implementation is mapped onto an IC
  - IC: Integrated circuit, or “chip”
  - IC technologies differ in their customization to a design
  - IC’s consist of numerous layers (perhaps 10 or more)
    - IC technologies differ with respect to who builds each layer and when



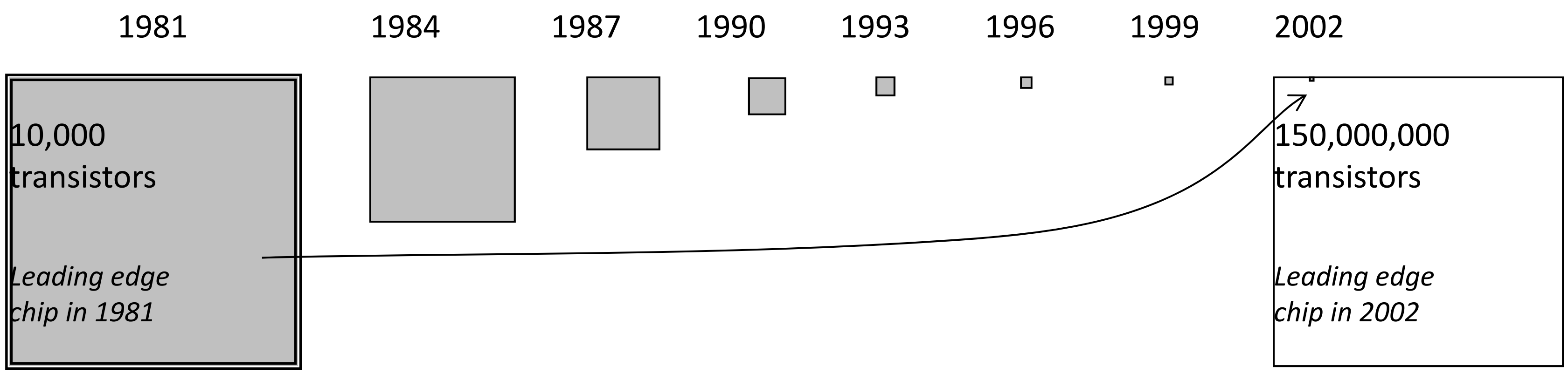
Moore's law

- The most important trend in embedded systems
  - Predicted in 1965 by Intel co-founder Gordon Moore

<u>IC transistor capacity has doubled roughly every 18 months for the past several decades</u>



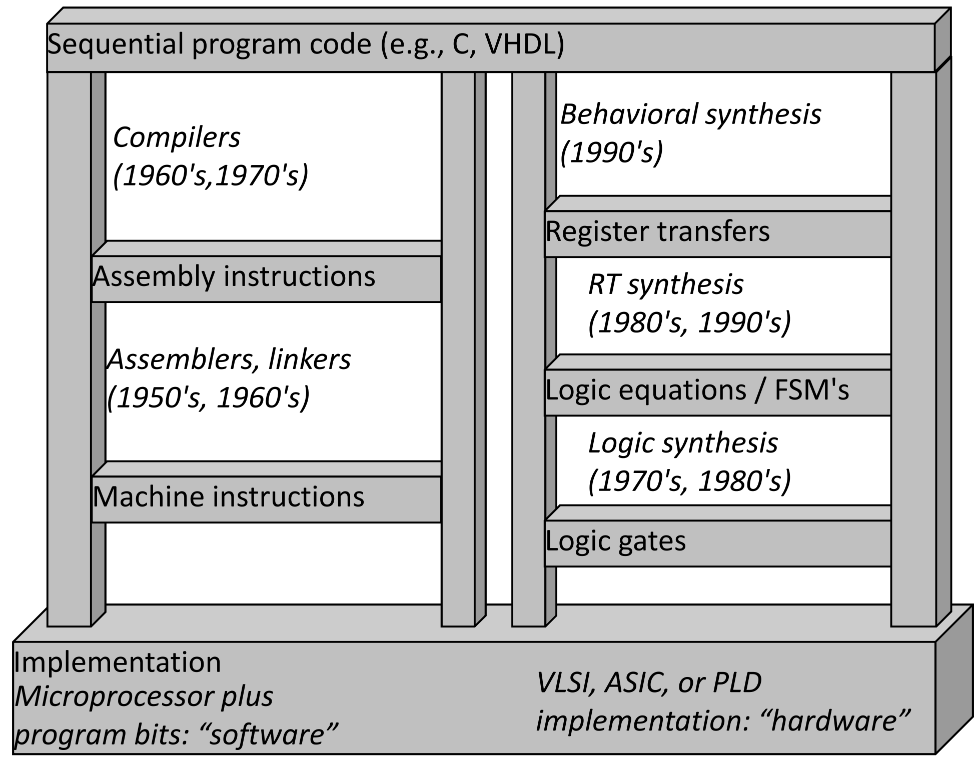
Graphical illustration of Moore's law



- Something that doubles frequently grows more quickly than most people realize!
  - A 2002 chip can hold about 15,000 1981 chips inside itself

Hardware-Software Co-design ladder

- In the past:
  - Hardware and software design technologies were very different
  - Recent maturation of synthesis enables a unified view of hardware and software
- Hardware/software "codesign"



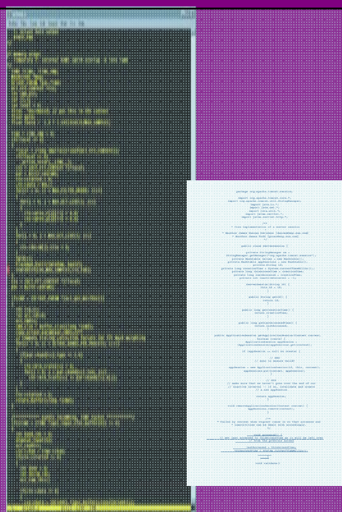
The choice of hardware versus software for a particular function is simply a tradeoff among various design metrics, like performance, power, size, NRE cost, and especially flexibility; there is no fundamental difference between what hardware or software can implement.

Cyber-Physical Systems:

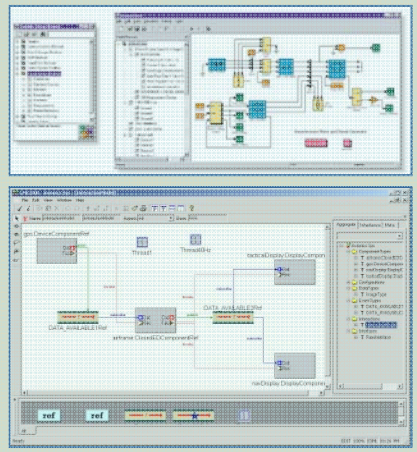
- A **Cyber Physical System** (CPS) is a mechanism controlled or monitored by computer-based algorithms, tightly integrated with internet and its users. It is an engineered **system** that are build from and depend upon, the seamless integration of computational algorithms and **physical** components.

Why is CPS Hard?

Software



Control



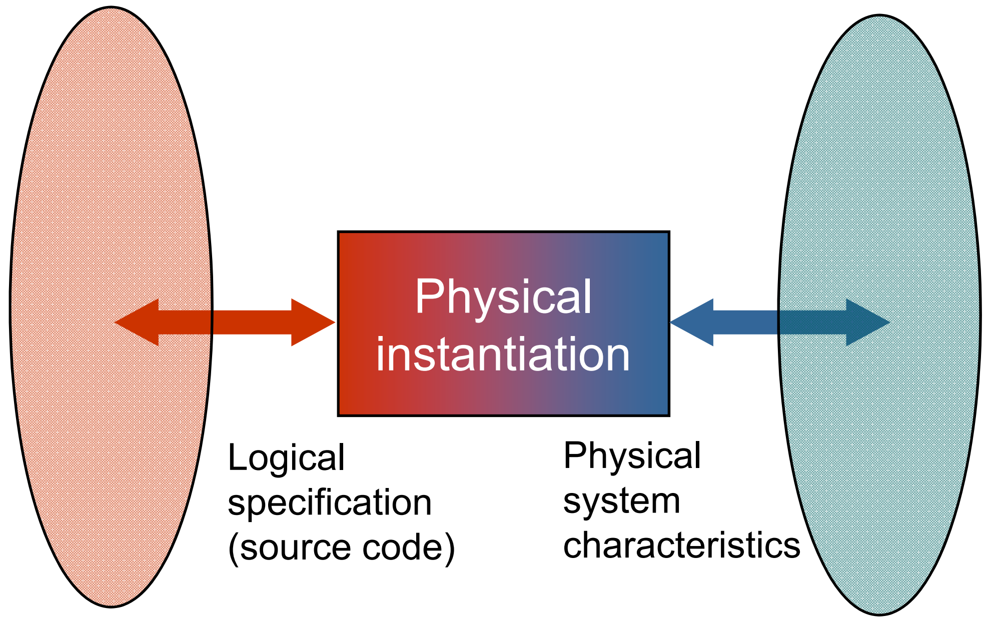
Systems

Crosses Interdisciplinary Boundaries

- Disciplinary boundaries need to be realigned
- New fundamentals need to be created
- New technologies and tools need to be developed
- Education need to be restructured

Heterogeneity and Modeling Languages

Computing System Composition Domain



Physical System Composition Domain

- "Cyber" Models
- Modeling Languages
  - Structure
  - Behaviors
- Mathematical Domains
  - traces/state variables
  - no reference semantics or "semantic units"

- Physical Models
- Modeling Languages
  - Structure
  - Behaviors
- Physical Laws
  - Physical variables
  - Physical Units

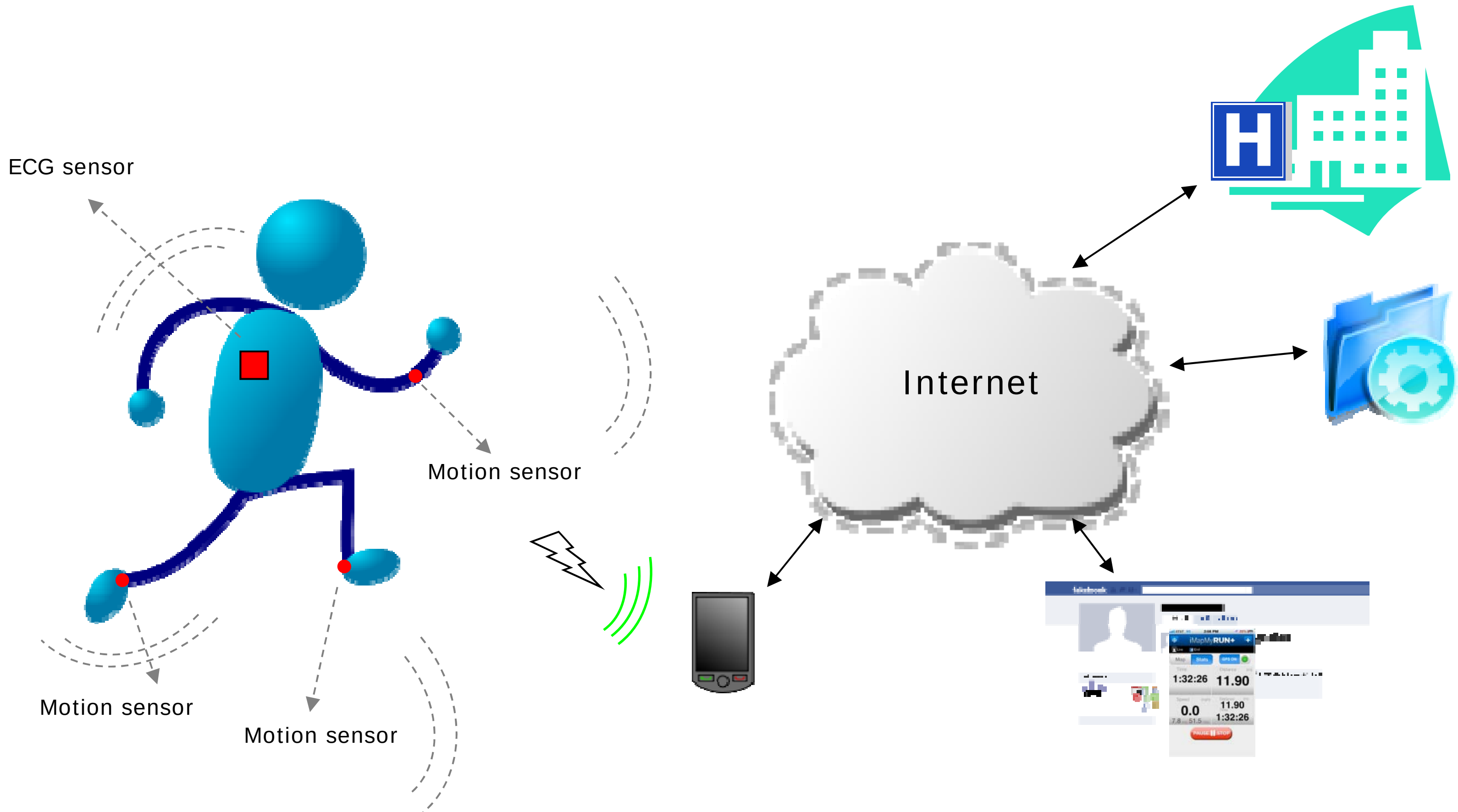
Internet-of-Things (IoTs):

- According to Wikipedia, IoT refers to the interconnection of uniquely identifiable embedded computing-like devices within the existing Internet infrastructure. Typically, IoT is expected to offer advanced connectivity of devices, systems, and services that goes beyond machine-to-machine communications (M2M) and covers a variety of protocols, domains, and applications. The interconnection of these embedded devices (including smart objects), is expected to usher in automation in nearly all fields, while also enabling advanced applications like a Smart Grid.
- Things, in the IoT, can refer to a wide variety of devices such as heart monitoring implants, bio-chip transponders on farm animals, automobiles with built-in sensors, or field operation devices that assist fire-fighters in search and rescue. Current market examples include smart thermostat systems and washer/dryers that utilize WiFi for remote monitoring.

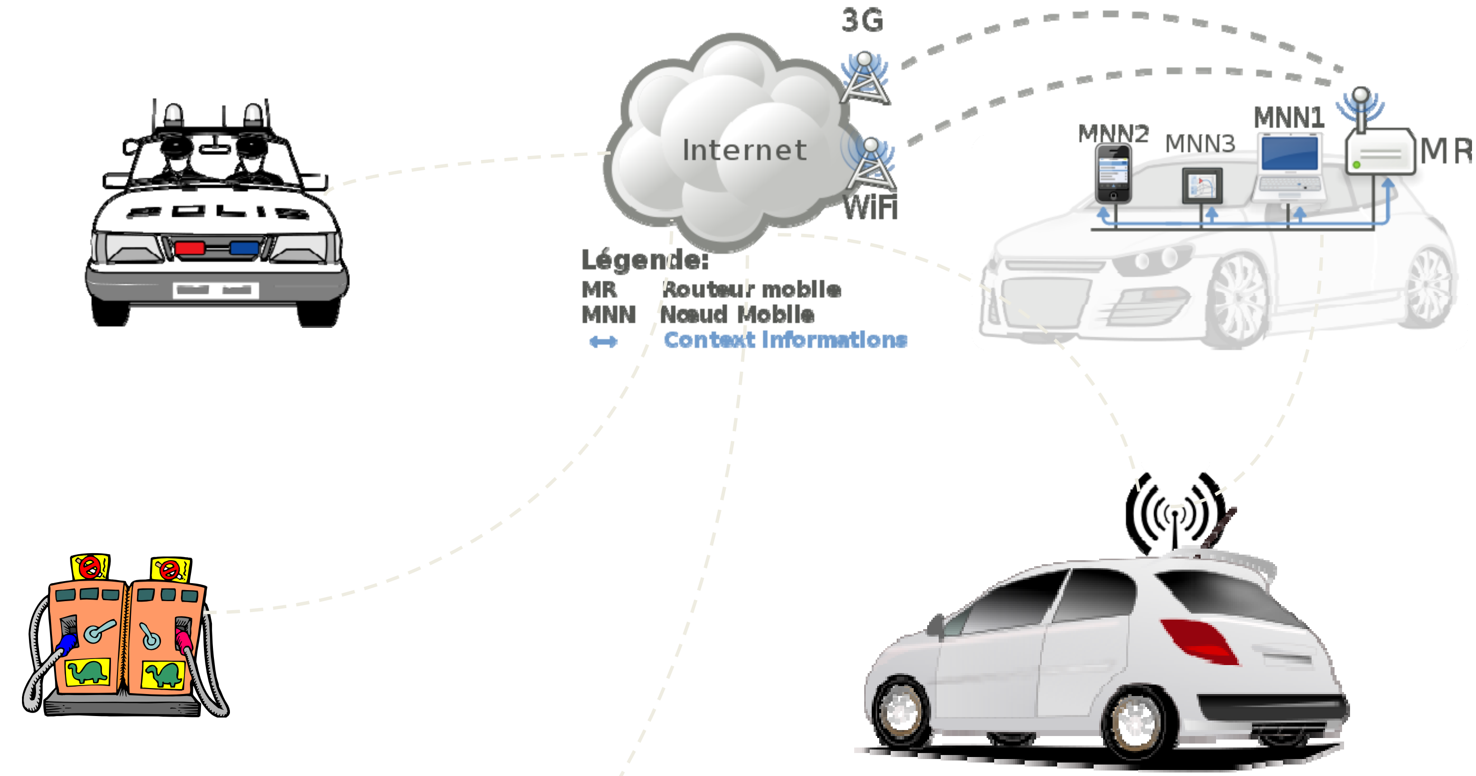
What are the major components of IoT?

1. Sensors/Actuators
2. Communication between servers or server platforms
3. Server/Middleware Platforms
4. Data Analytics Engines
5. Apps (iOS, Android, Web)

People Connecting to Things

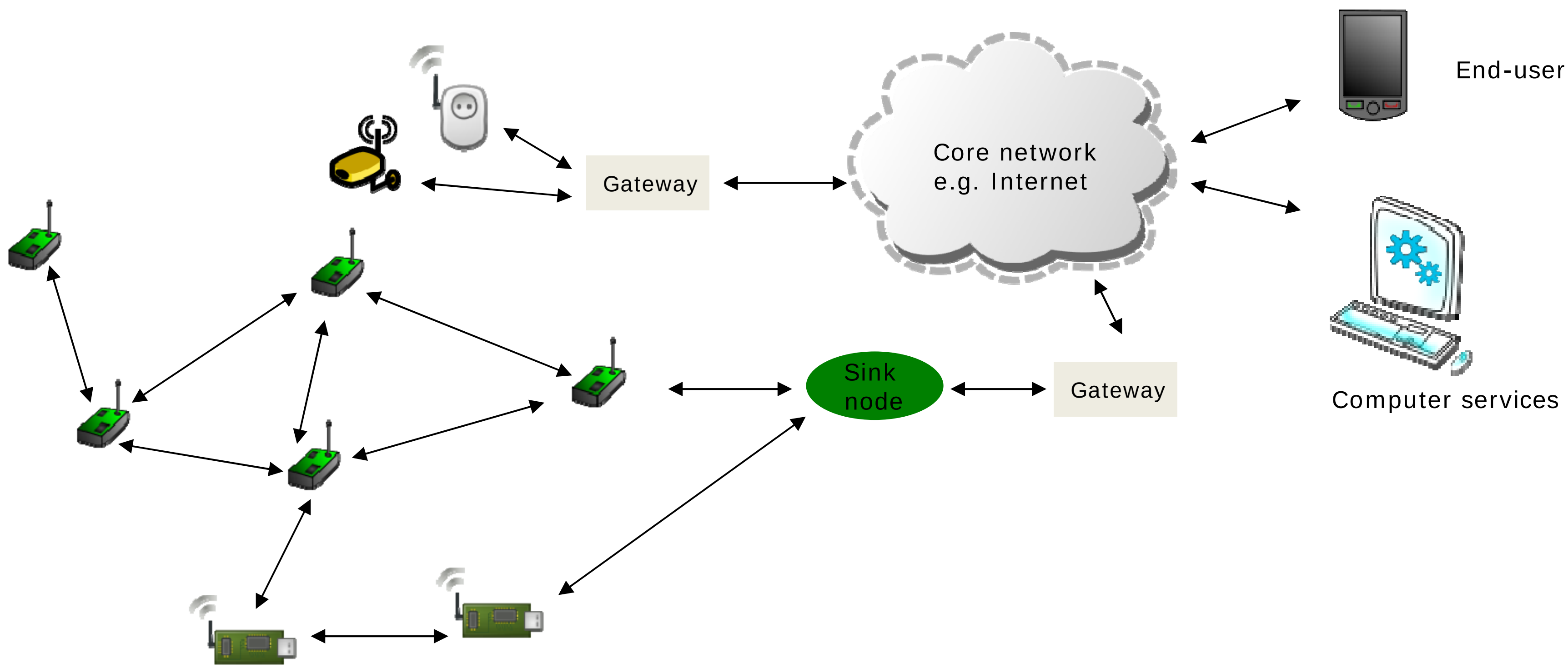


Things Connecting to Things



- Complex and heterogeneous resources and networks

Wireless Sensor Networks (WSN)



- The networks typically run Low Power Devices
- Consist of one or more sensors, could be different type of sensors (or actuators)

Communication Technologies for IoT:

1. Bluetooth
2. Zigbee
3. Z-Wave
4. NFC (Near-Field Communication)
5. RFID
6. WiFi
7. 2G/3G/LTE
8. Wibro/Mobile WiMax
9. PLC (Power Line Communication)
10. Ethernet
11. Many more…

"Thing" connected to the internet

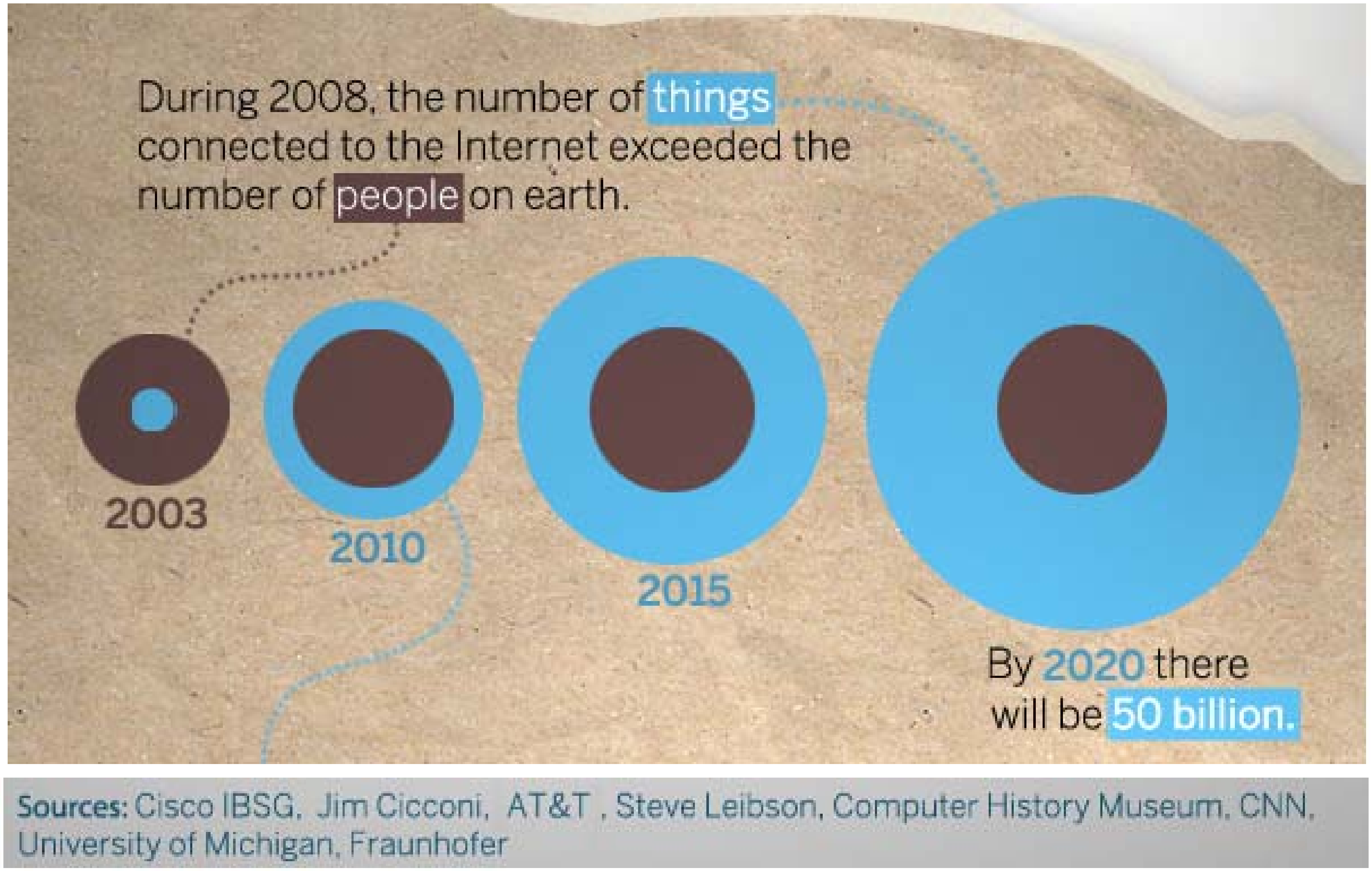
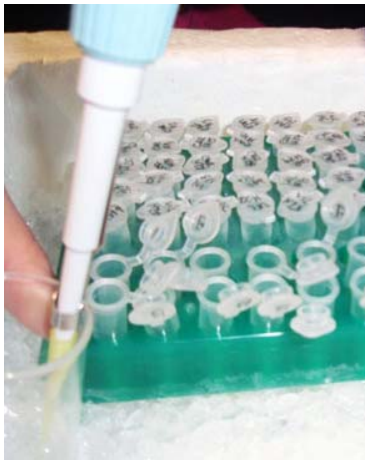


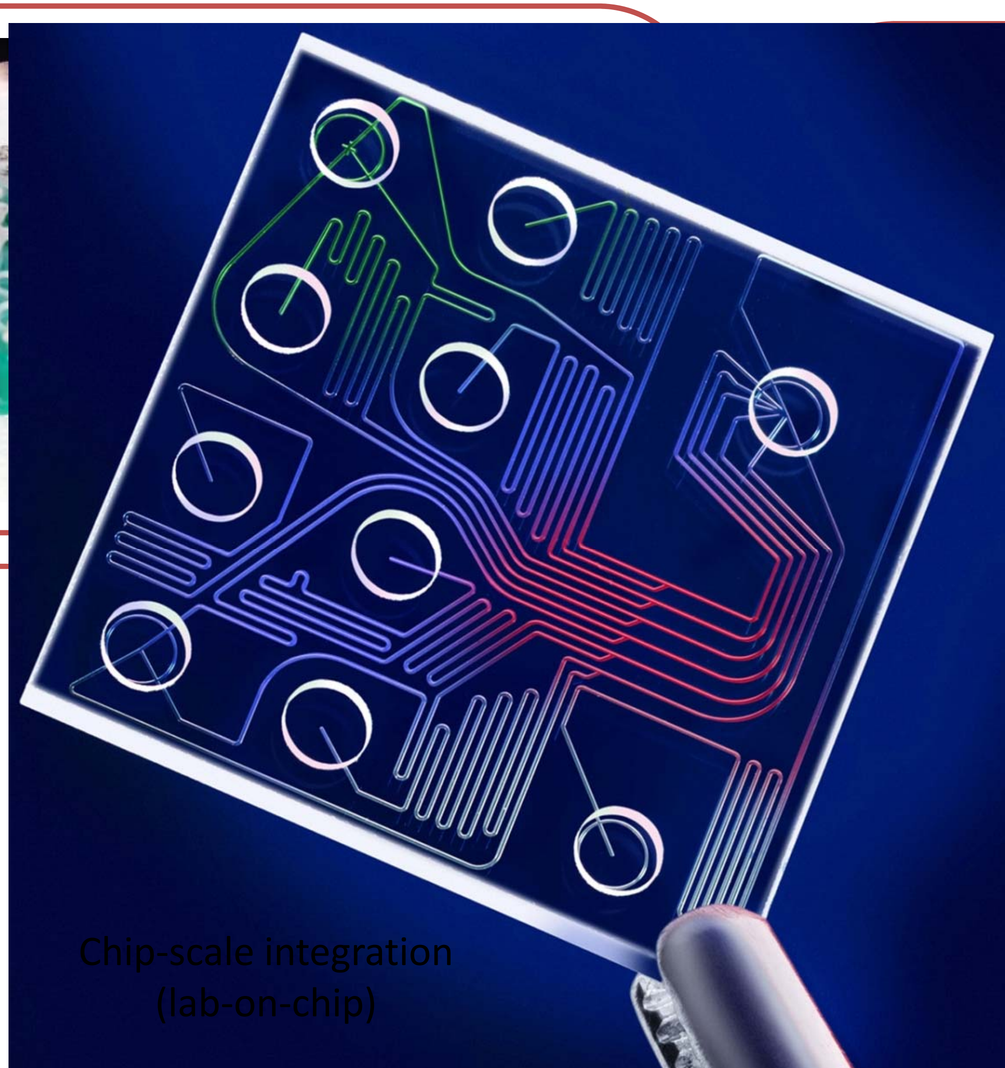
Image Courtesy: : CISCO

Microfluidic Biochips and its Design Algorithms

Laboratory-On-a-Chips (LOCs) or Biochips



Skilled Technicians

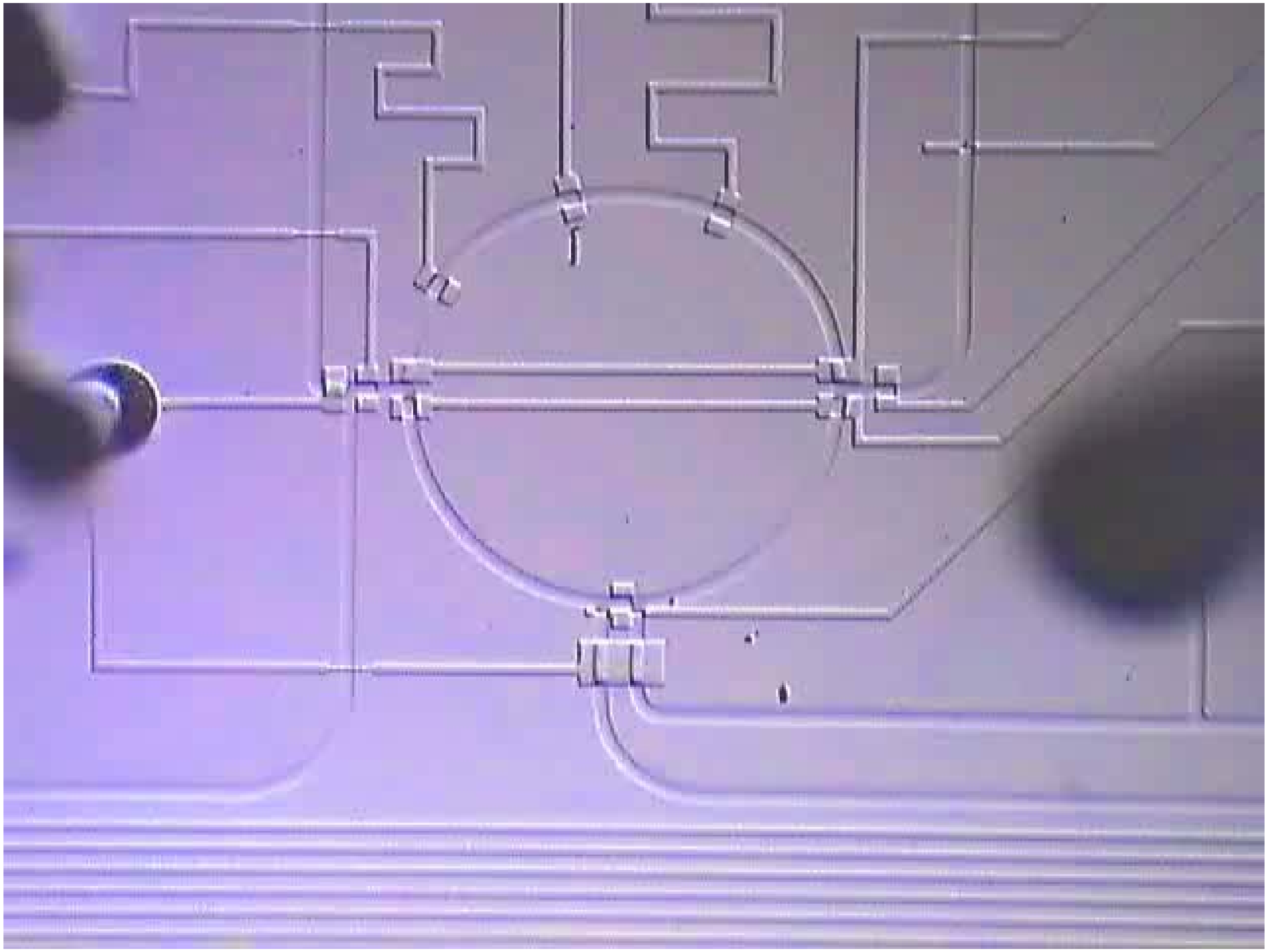


Bulky Equipment

Two Kinds of Lab-on-a-Chips or Microfluidic Biochips:

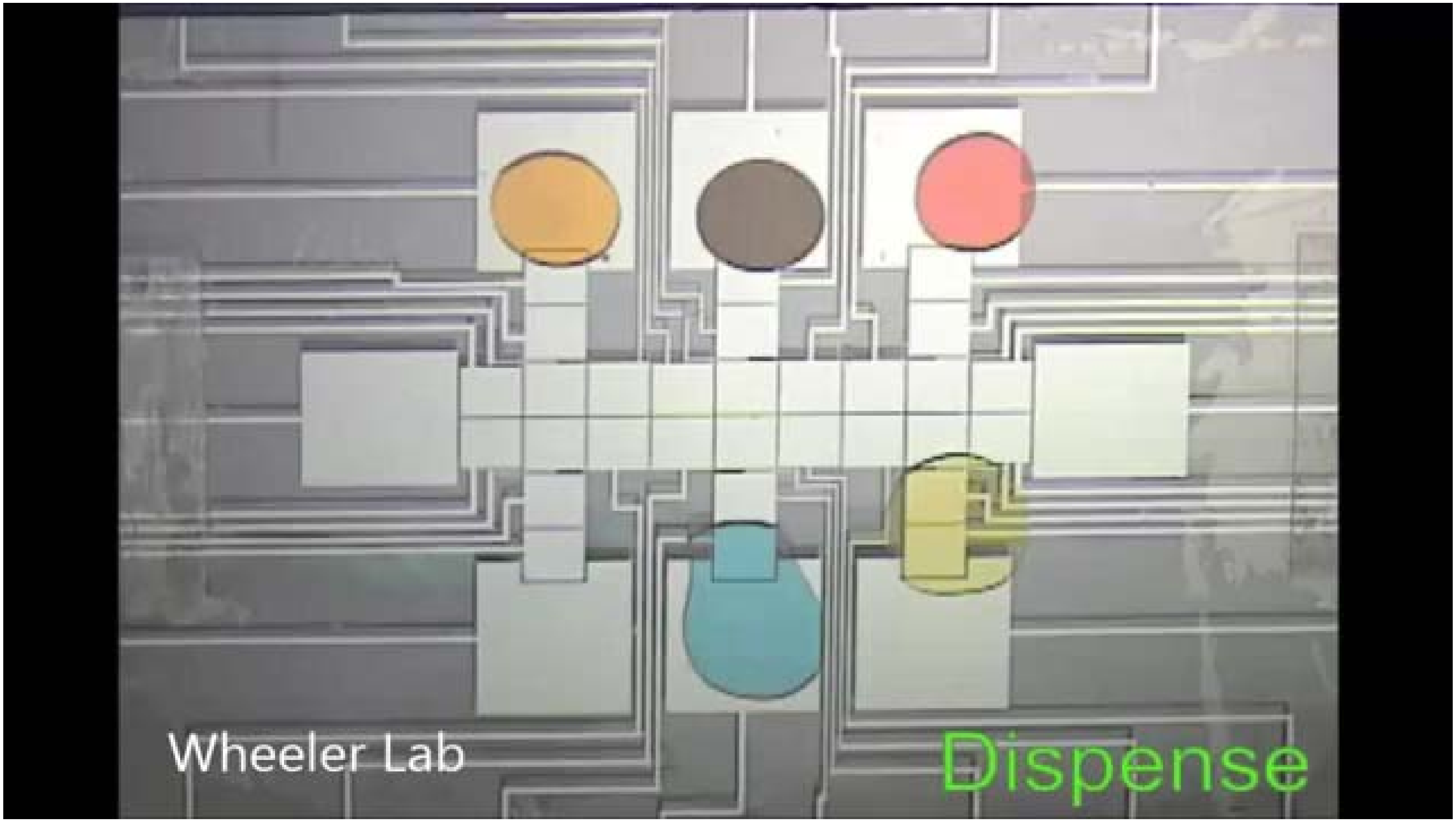
- Continuous-Flow Microfluidic (CMF) Biochips
- Digital Microfluidic (DMF) Biochips

Continuous-Flow Microfluidic Biochips:



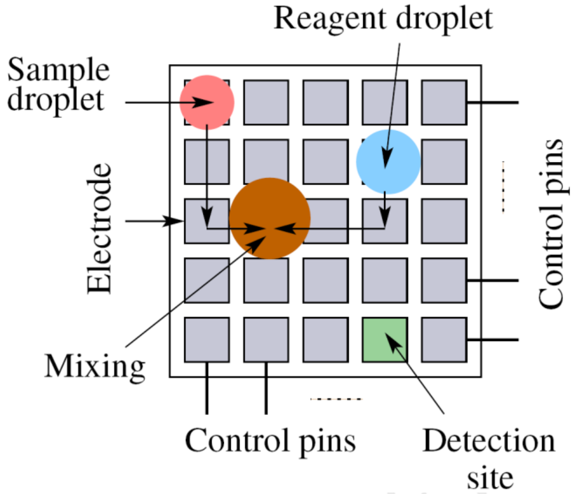
[W. Thies et al. (MIT, USA), Natural Computing, 2008]

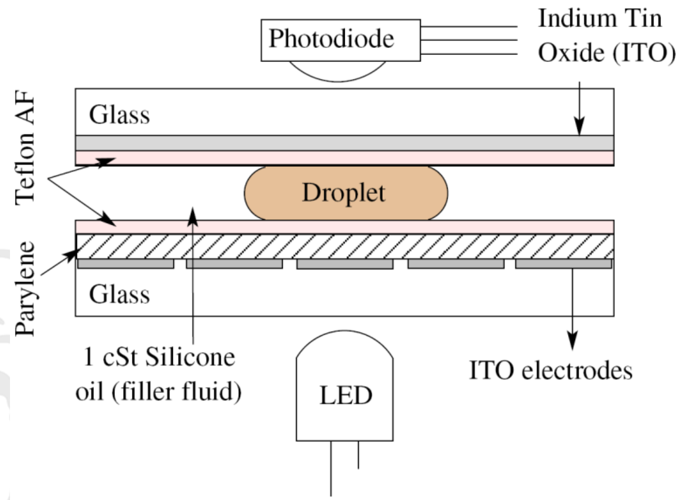
Digital Microfluidic Biochips:



Aaron Wheeler, Wheeler Microfluidics Laboratory, University of Toronto, Canada

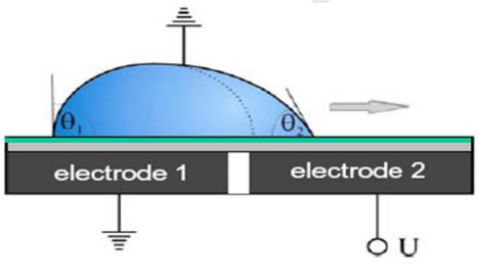
Digital Microfluidic Biochips:



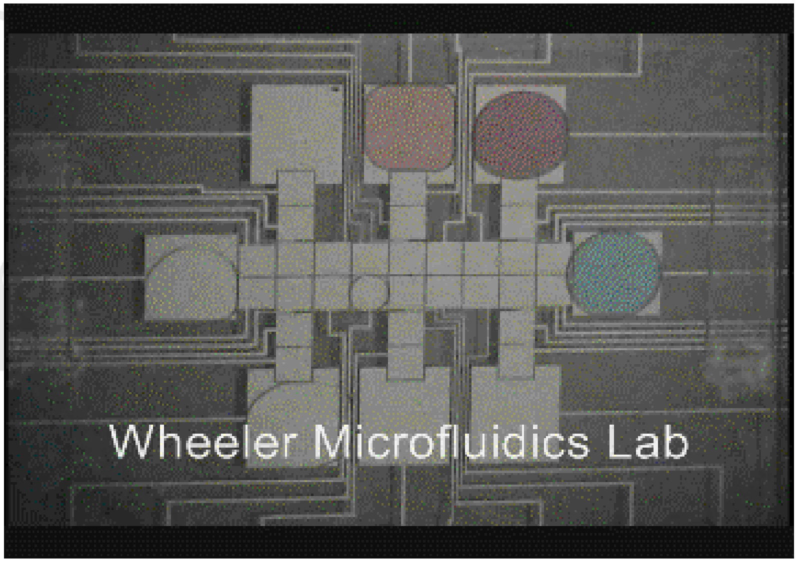


- Young-Lippmann equation (1948)

$$\cos\theta = \cos\theta_0 + \frac{1}{\gamma_{lv}}\frac{cV^2}{2}$$



Electrowetting-on-Dielecctric (EWOD)



Bioassay Description Language:

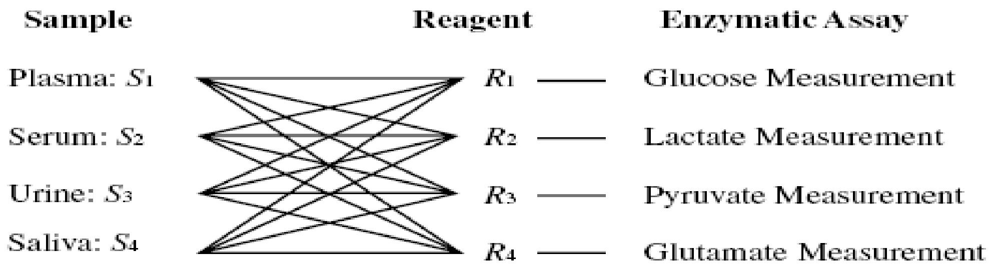
BioCoder: A programming language for standardizing and automating biology protocols

```
#include "BioCoder.h"
#define X 60

void main()
{
        start_protocol("Blackburn - Yeast Colony PCR");

        Fluid naoh = new_fluid("0.02M NaOH");
        Fluid q_solution = new_fluid("Q-solution", "5X");
        Fluid pcr_buffer = new_fluid("PCR buffer", "10X");
        Fluid dntp = new_fluid("dNTPs", "10mM each");
        Fluid f_primer = new_fluid("Forward primer", "100M");
        Fluid r_primer = new_fluid("Reverse primer");
        Fluid taq = new_fluid("Taq");
        Fluid water = new_fluid("ddH<sub>2</sub>O");
        Solid colony = new_solid("a small colony");

        Container pcr_tube1 = new_container(STERILE_PCR_TUBE);
        Container pcr_tube2 = new_container(STERILE_PCR_TUBE);
        Container pcr_tube3 = new_container(STERILE_PCR_TUBE);

        //Yeast Cell Lysis
        //   1. Aliquot 10uL of 0.02M NaOH into PCR tubes.
        rst_step("Yeast Cell Lysis");
           t_sub_step();
              sure_fluid(naoh, vol(10, UL), pcr_tube1);

                 sing a sterile pipette tip, pick a small colony and resuspend in NaOH.
                   If the solution is cloudy, you've added enough cells.
                     I have been told adding too much yeast can inhibit the reaction.
                 ub_step();

n.");
C for 10 minutes.
ep on ice or freeze for
n.");
eep on ice or freeze for
q, water);
vol(0.2, UL), vol(0.1, UL),
channel pipette is helpful

        comment("A multichannel pipette is helpful here.");

        //   4. Run the following PCR cycle:
        //        1. 5 min at 95C
        //        2. 30 cycles of:
        //             1. 10 sec at 95C
        //             2. 10 sec at 50C (or appropriate annealing temperature)
        //             3. 1 min/kbp at 72C (I generally do 30 sec)
        //        3. 10 min at 72C (I don't think this step is critical)
        next_sub_step();
        pcr_init_denat(pcr_tube3, 95, time(5, MINS));
        thermocycler(pcr_tube3, 30, 95, time(10, SECS), 50, time(10, SECS), 72, time(X, SECS),
NORMAL);
        comment("Elongation time : 1 min/kbp.I generally do 30 sec elongation.");
        pcr_final_ext(pcr_tube3, 72, time(10, MINS), 4);
        comment("I don't think this step is critical.");

        end_protocol();
}
```

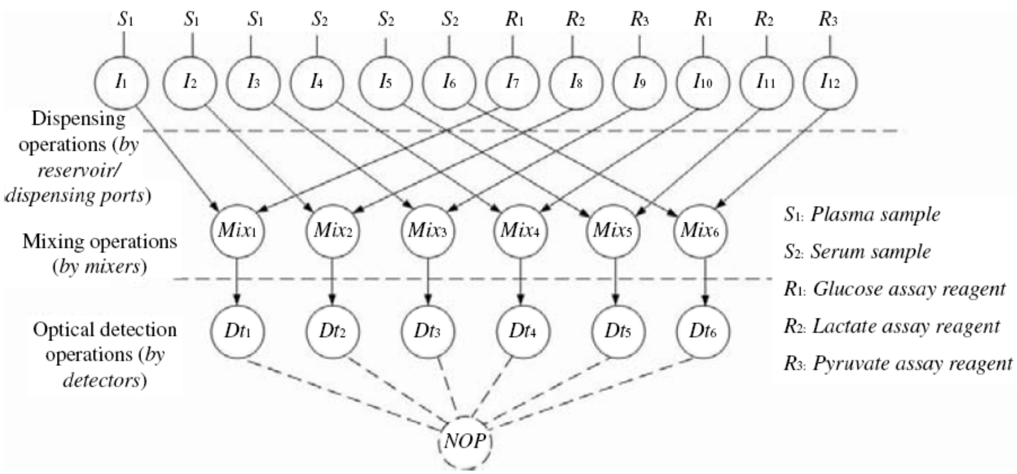

```
#include "BioCoder.h"
#define X 60

void main()
{
        start_protocol("Blackburn - Yeast Colony PCR");

        Fluid naoh = new_fluid("0.02M NaOH");
        Fluid q_solution = new_fluid("Q-solution", "5X");
        Fluid pcr_buffer = new_fluid("PCR buffer", "10X");
        Fluid dntp = new_fluid("dNTPs", "10mM each");
        Fluid f_primer = new_fluid("Forward primer", "100M");
        Fluid r_primer = new_fluid("Reverse primer");
        Fluid taq = new_fluid("Taq");
        Fluid water = new_fluid("ddH<sub>2</sub>O");
        Solid colony = new_solid("a small colony");
```


BioCoder Project, Microsoft Research India, Dec, 2009.

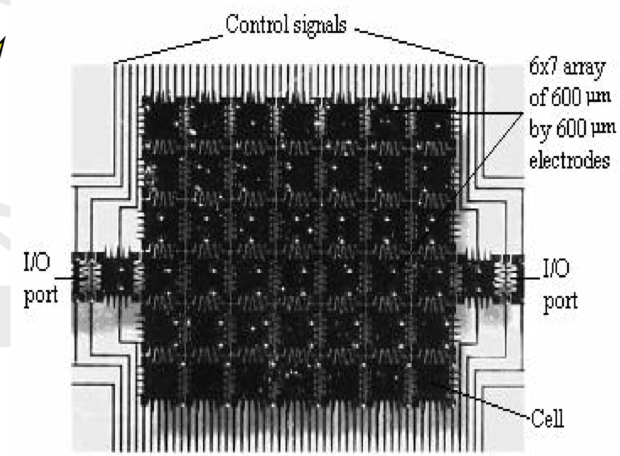
DMF Biochip Design:



Example bioassay:
Multiplexed *In-Vitro* Diagnostics



Sequencing graph model

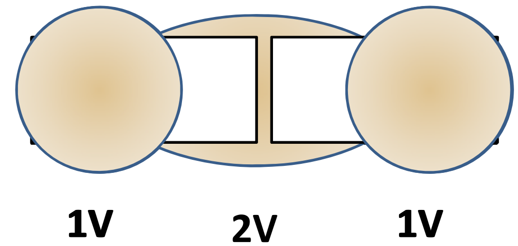


Time step	Actuation Sequences
0	0010101010001010101010101
1	1000100010001010001110100
2	0010100000000110001100100
3	⋮

Dilution of a Fluid

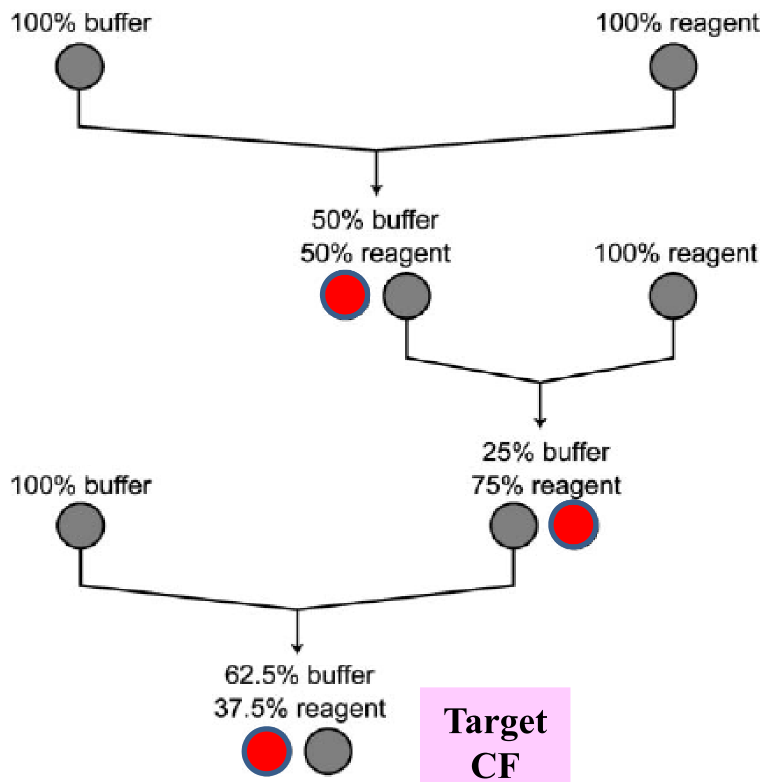
Dilution and Mixing: Problem Formulation

Example: Dilution of biosamples / chemical reagents – target concentration factor $C_T = (3/8) = 37.5\%$ of a biosample



(1:1) Mixing Model

Sequencing Graph (DAG):



Target CF

Mix

Split

Mix/Split cycle

Mix

: Waste Droplet

Split

Mix

Split

W. Thies et al., Natural Computing, May, 2008.

Dilution and Mixing – Concentration Factor

$C_T = 69.92\%$

$C_T =$ Sample : Buffer

$= 7 : 10$

$$C_T = \frac{179}{256}$$

$C_T = 0.6992$

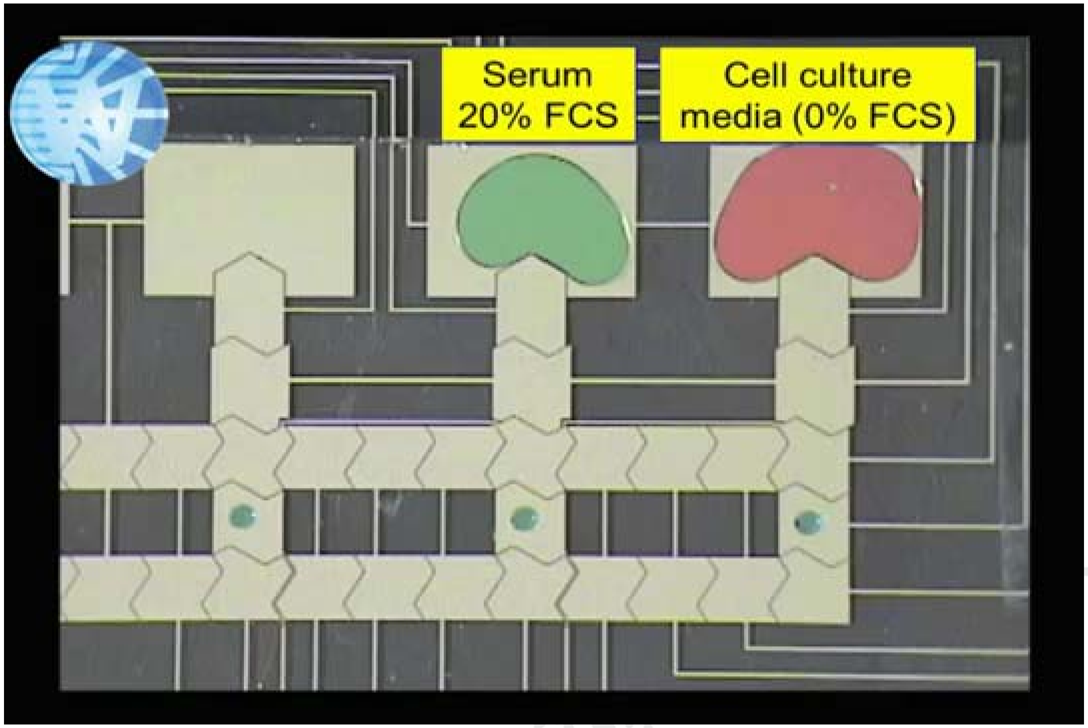
$$C_T = \frac{7}{10}$$

Only two extreme CFs are supplied:

Sample with 100% concentration or $\frac{1024}{1024}$

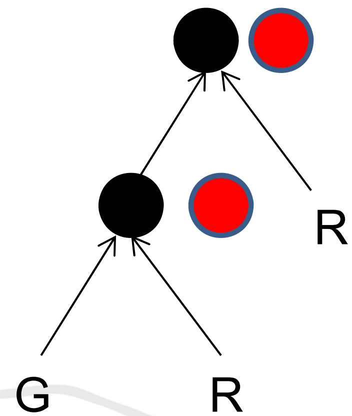
Buffer solution with 0% concentration or $\frac{0}{1024}$

Dilution on DMF Biochip:



On-Chip Dilution

Courtesy: Wheeler Microfluidic Lab,
University of Toronto, 2013.



$$CF(G) = \frac{10\% + 0\%}{2}$$

$$CF(G) = \frac{20\% + 0\%}{2}$$

T1 = (G+R)/2 :-> CF(G)=1/2

T2 = (G+3R)/4 :-> CF(G)=1/4

Dilution Algorithm: twoWayMix

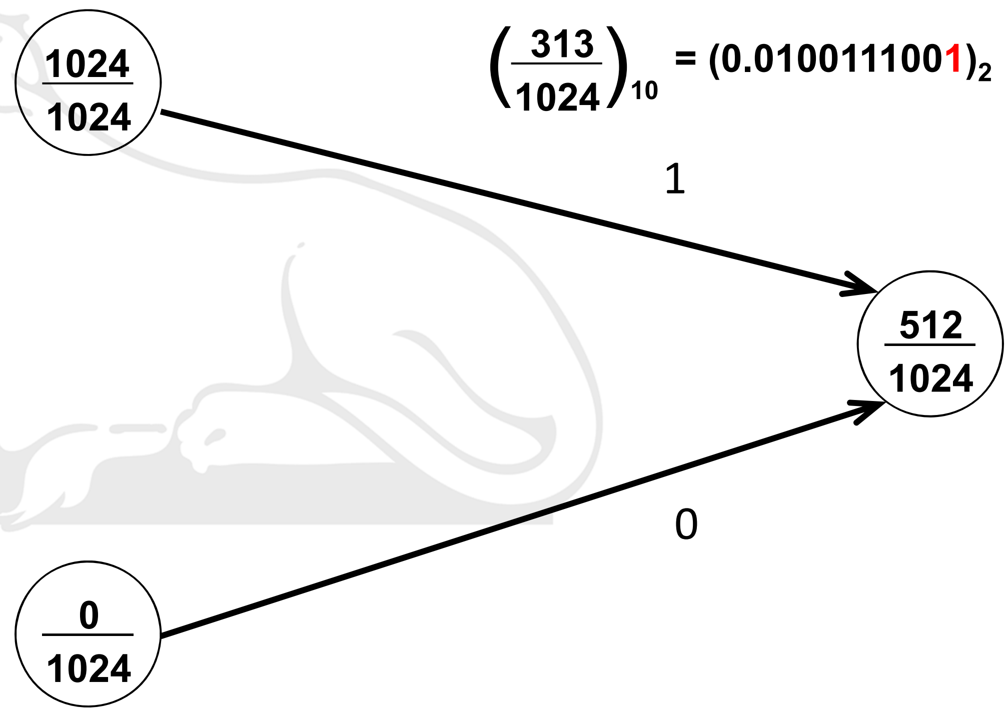
Two extreme CFs are supplied:

Sample with 100% concentration or $\frac{1024}{1024}$ or CF = 1.0

Buffer solution with 0% concentration or $\frac{0}{1024}$ or CF = 0.0

W. Thies et al. (MIT), Natural Computing, 2008.

$$\left(\frac{313}{1024}\right)_{10} = (0.0100111001)_2$$



Dilution Algorithm: twoWayMix

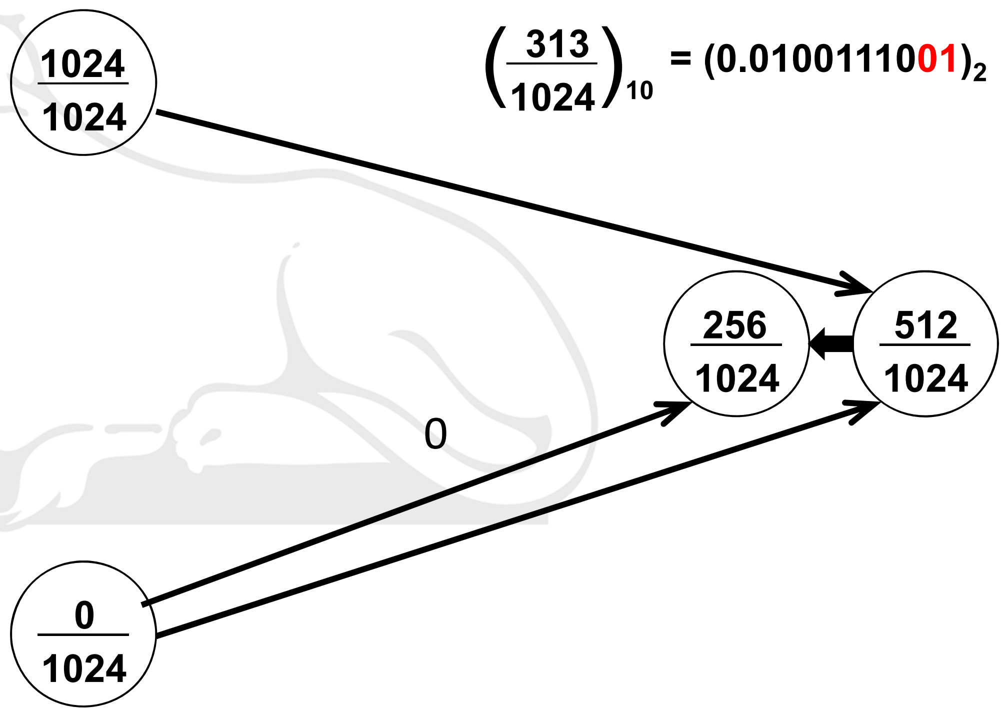
Two extreme CFs are supplied:

Sample with 100% concentration or $\frac{1024}{1024}$ or CF = 1.0

Buffer solution with 0% concentration or $\frac{0}{1024}$ or CF = 0.0

W. Thies et al. (MIT), Natural Computing, 2008.

$$\left(\frac{313}{1024}\right)_{10} = (0.0100111001)_2$$



Dilution Algorithm: twoWayMix

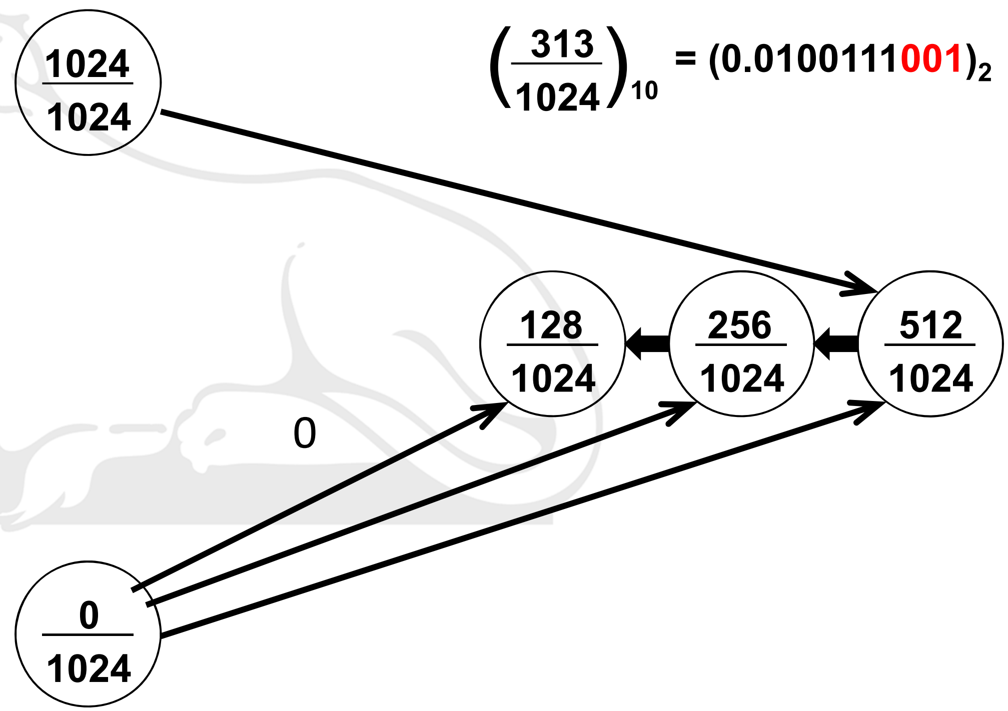
Two extreme CFs are supplied:

Sample with 100% concentration or $\frac{1024}{1024}$ or CF = 1.0

Buffer solution with 0% concentration or $\frac{0}{1024}$ or CF = 0.0

W. Thies et al. (MIT), Natural Computing, 2008.

$$\left(\frac{313}{1024}\right)_{10} = (0.0100111001)_2$$



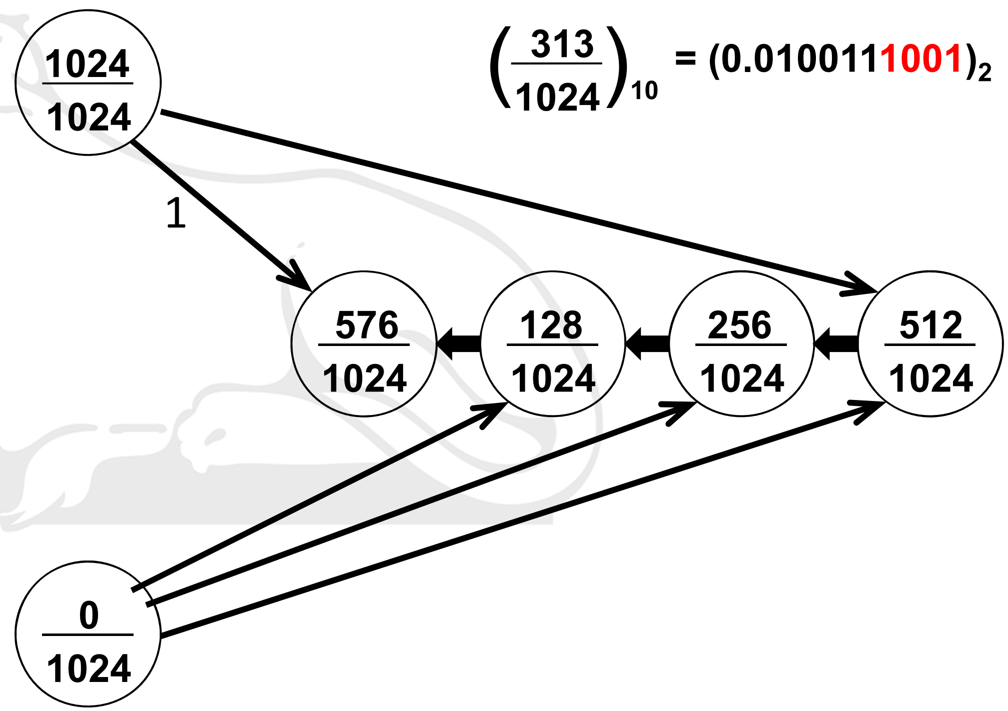
Dilution Algorithm: twoWayMix

Two extreme CFs are supplied:

Sample with 100% concentration or $\frac{1024}{1024}$ or CF = 1.0

Buffer solution with 0% concentration or $\frac{0}{1024}$ or CF = 0.0

W. Thies et al. (MIT), Natural Computing, 2008.



Dilution Algorithm: twoWayMix

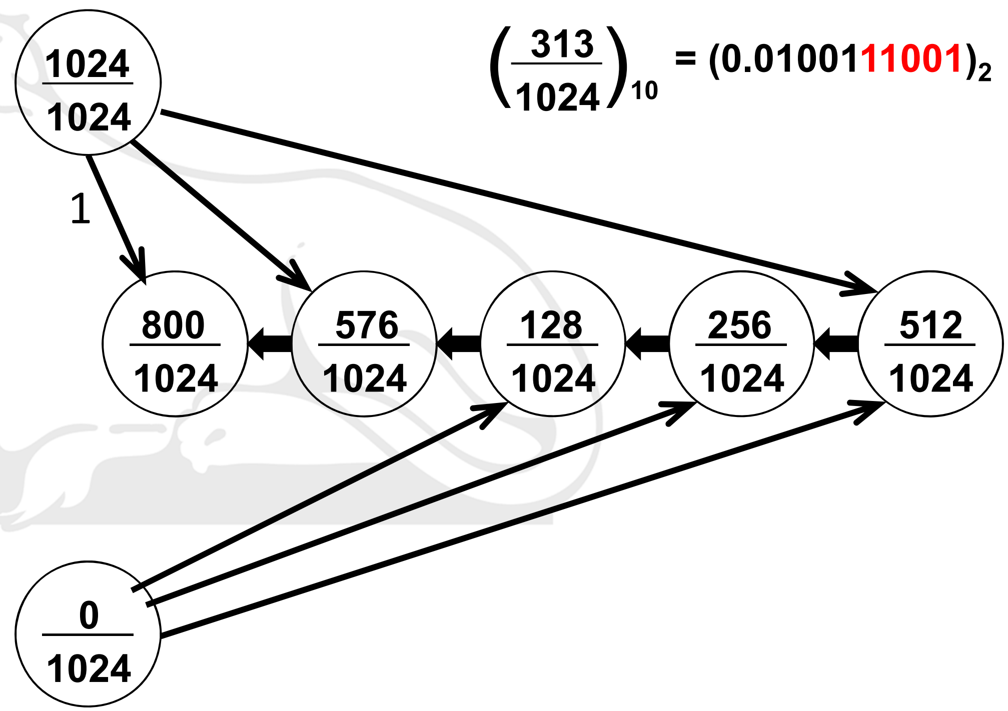
Two extreme CFs are supplied:

Sample with 100% concentration or $\frac{1024}{1024}$ or CF = 1.0

Buffer solution with 0% concentration or $\frac{0}{1024}$ or CF = 0.0

W. Thies et al. (MIT), Natural Computing, 2008.

$$\left(\frac{313}{1024}\right)_{10} = (0.0100111001)_2$$

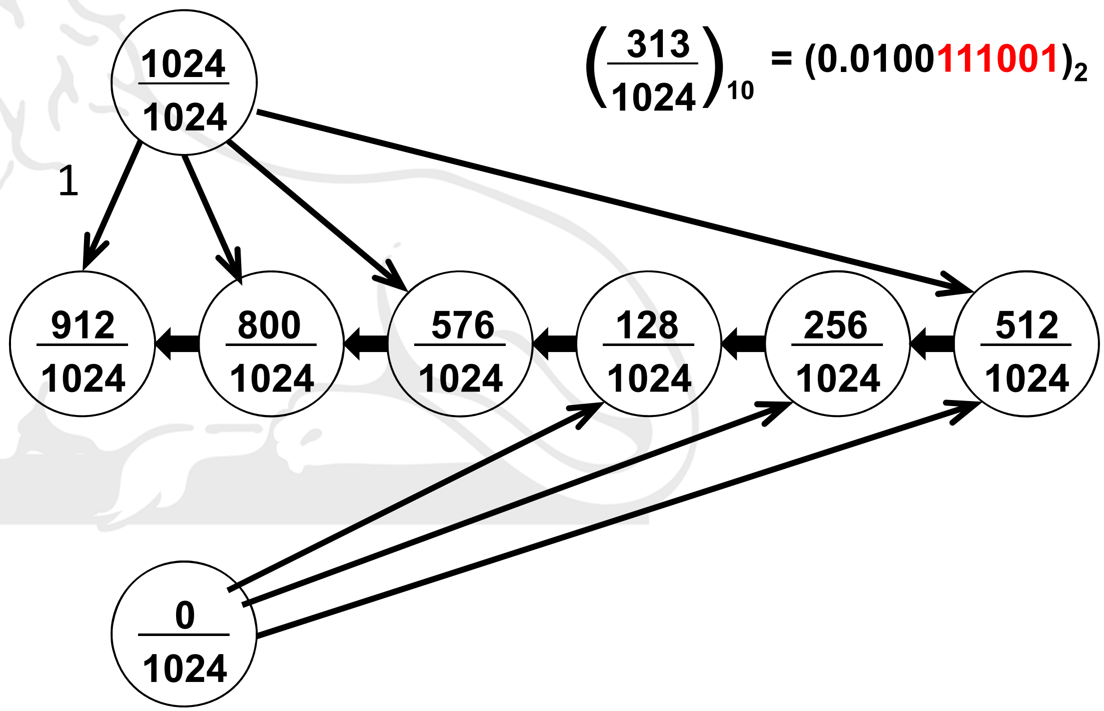


Dilution Algorithm: twoWayMix

Two extreme CFs are supplied:
Sample with 100% concentration or $\frac{1024}{1024}$ or CF = 1.0
Buffer solution with 0% concentration or $\frac{0}{1024}$ or CF = 0.0

W. Thies et al. (MIT), Natural Computing, 2008.

$\left(\frac{313}{1024}\right)_{10} = (0.0100111001)_2$



Dilution Algorithm: twoWayMix

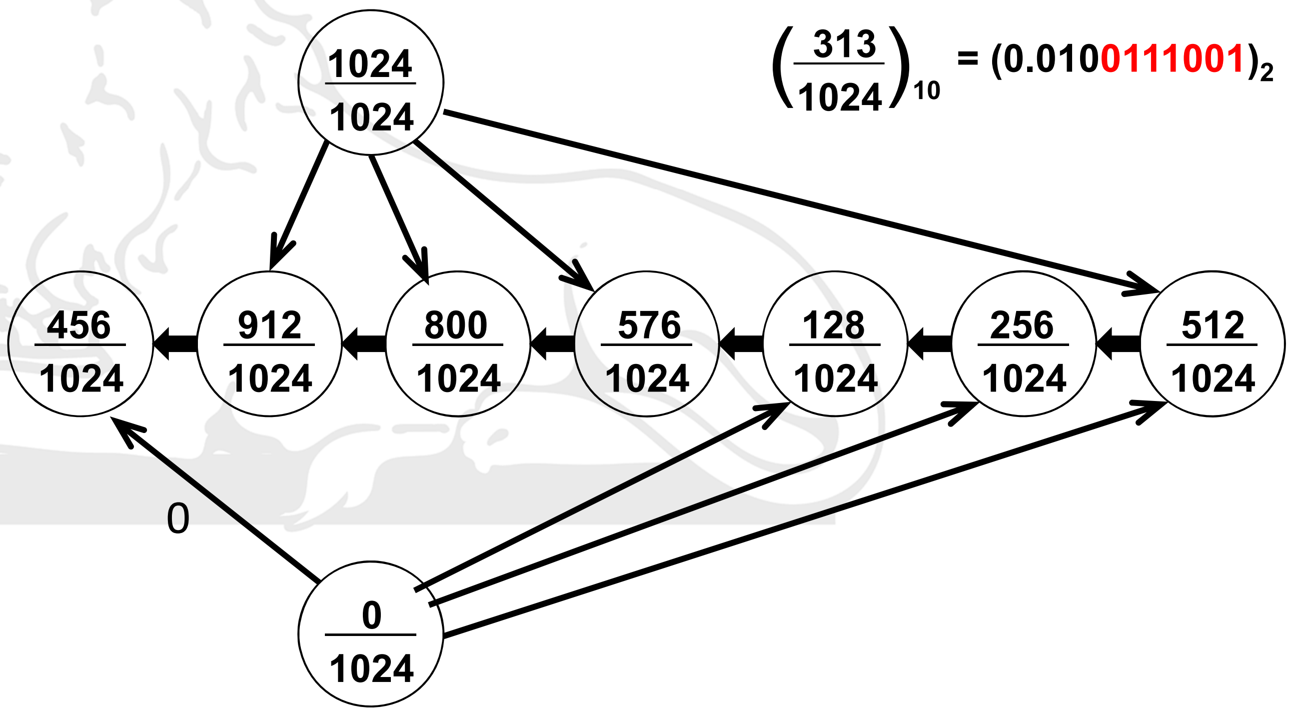
Two extreme CFs are supplied:

Sample with 100% concentration or $\frac{1024}{1024}$ or CF = 1.0

Buffer solution with 0% concentration or $\frac{0}{1024}$ or CF = 0.0

W. Thies et al. (MIT), Natural Computing, 2008.

$\left(\frac{313}{1024}\right)_{10} = (0.0100111001)_2$



Dilution Algorithm: twoWayMix

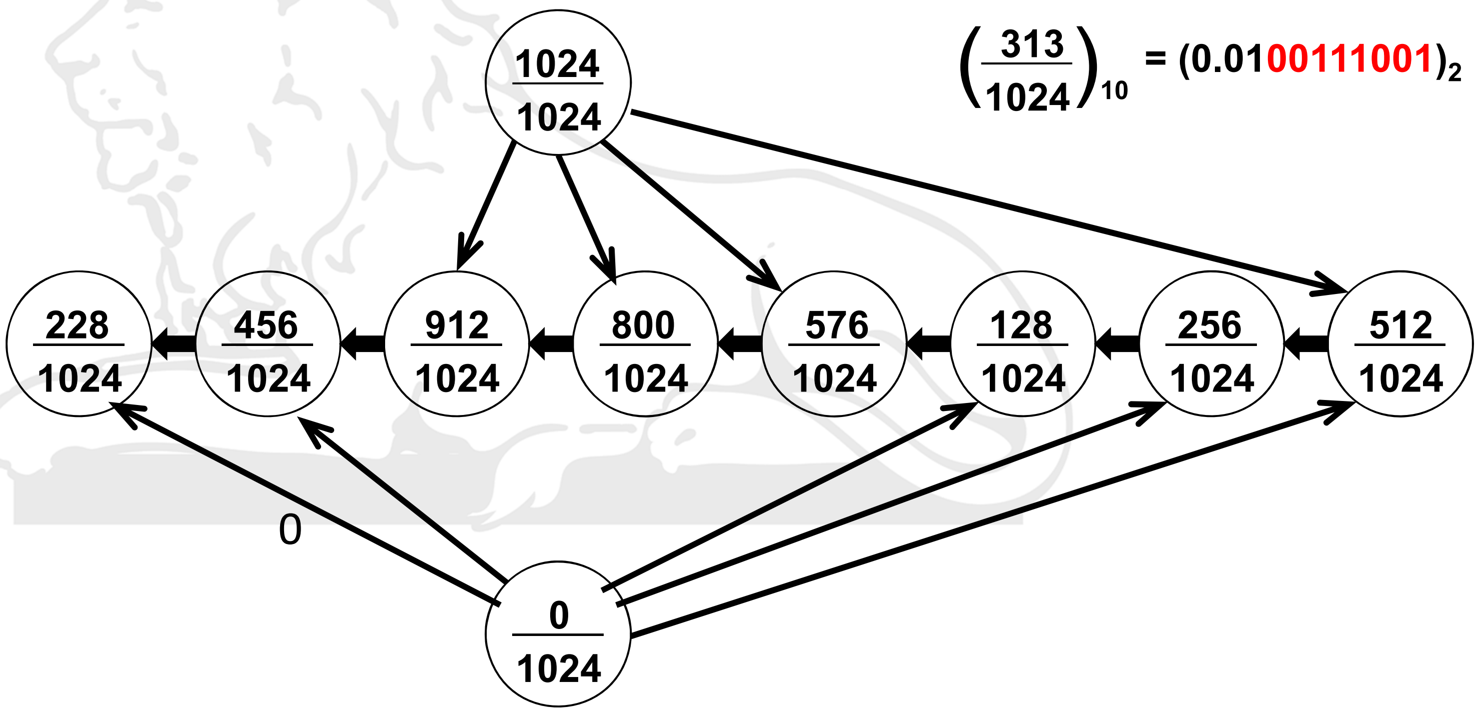
Two extreme CFs are supplied:

Sample with 100% concentration or $\frac{1024}{1024}$ or CF = 1.0

Buffer solution with 0% concentration or $\frac{0}{1024}$ or CF = 0.0

W. Thies et al. (MIT), Natural Computing, 2008.

$\left(\frac{313}{1024}\right)_{10} = (0.0100111001)_2$

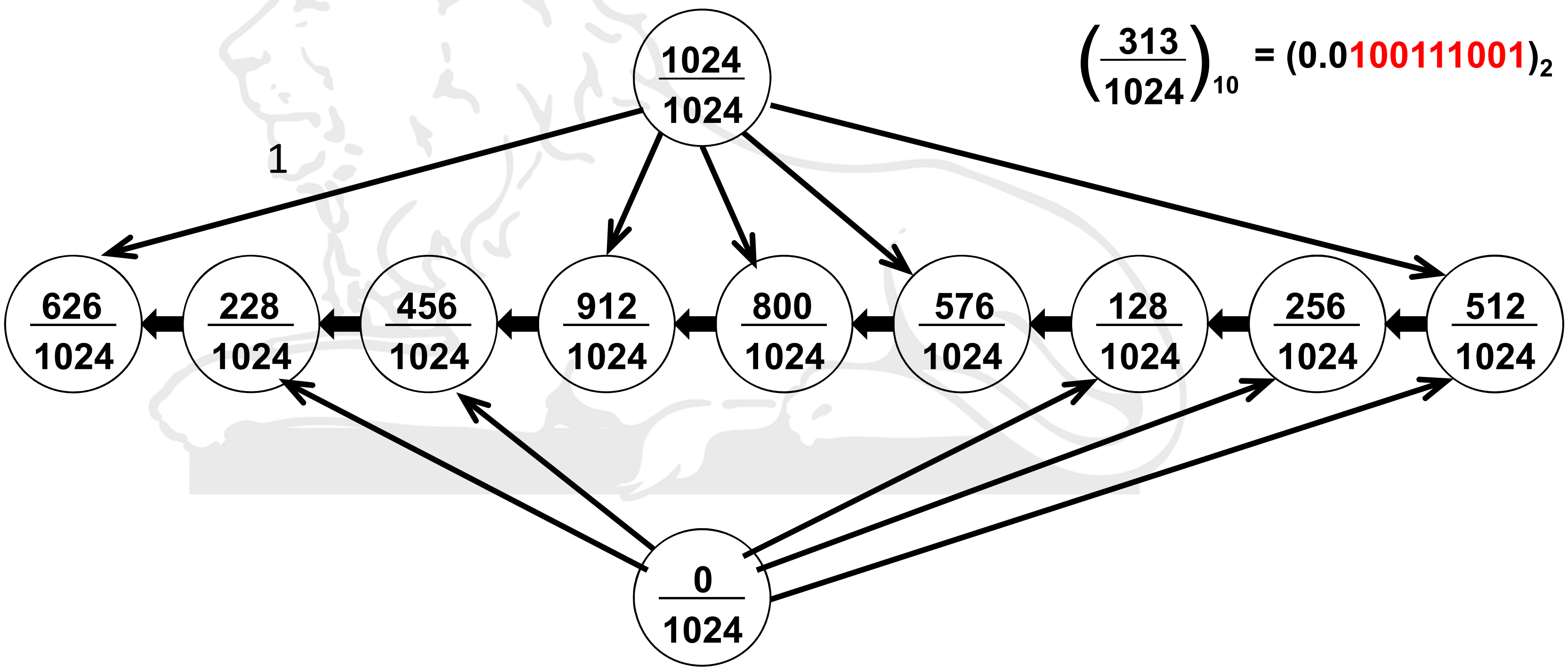


Dilution Algorithm: twoWayMix

Two extreme CFs are supplied:
Sample with 100% concentration or $\frac{1024}{1024}$ or **CF = 1.0**
Buffer solution with 0% concentration or $\frac{0}{1024}$ or **CF = 0.0**

W. Thies et al. (MIT), Natural Computing, 2008.

$$\left(\frac{313}{1024}\right)_{10} = (0.0100111001)_2$$



Dilution Algorithm: twoWayMix

Two extreme CFs are supplied:

Sample with 100% concentration or $\frac{1024}{1024}$ or CF = 1.0

Buffer solution with 0% concentration or $\frac{0}{1024}$ or CF = 0.0

W. Thies et al. (MIT), Natural Computing, 2008.

$$\left(\frac{313}{1024}\right)_{10} = (0.0100111001)_2$$