CSN-101 (Introduction to Computer Science and Engineering)

Lecture 22: 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]

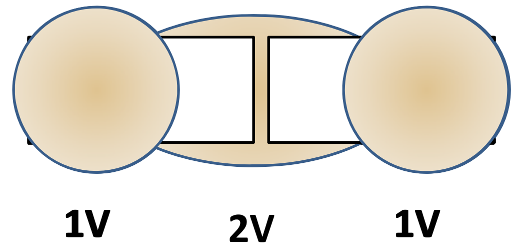
Moodle Submission Site: https://moodle.iitr.ac.in/course/view.php?id=45

[Enrollment Key: csn101@2019]

Dilution of a Fluid

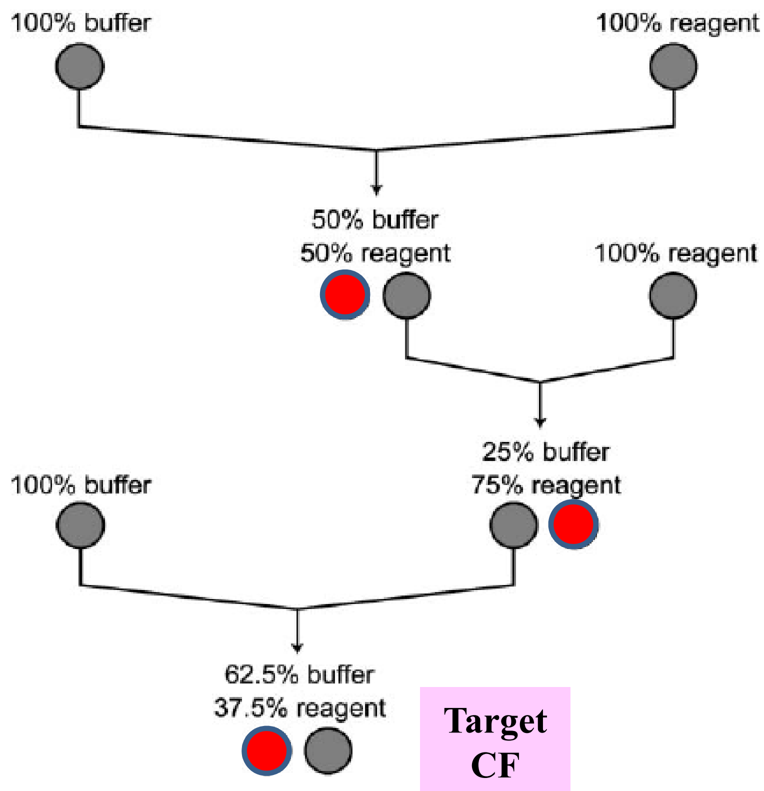
Dilution and Mixing: Problem Formulation

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):



Mix
Split
Mix/Split cycle
Mix
Split
Mix
Split

: Waste Droplet

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

Dilution and Mixing – Concentration Factor

$C_T = 69.92\%$

$C_T =$ Sample : Buffer

$= 7 : 10$

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

$C_T = 0.6992$

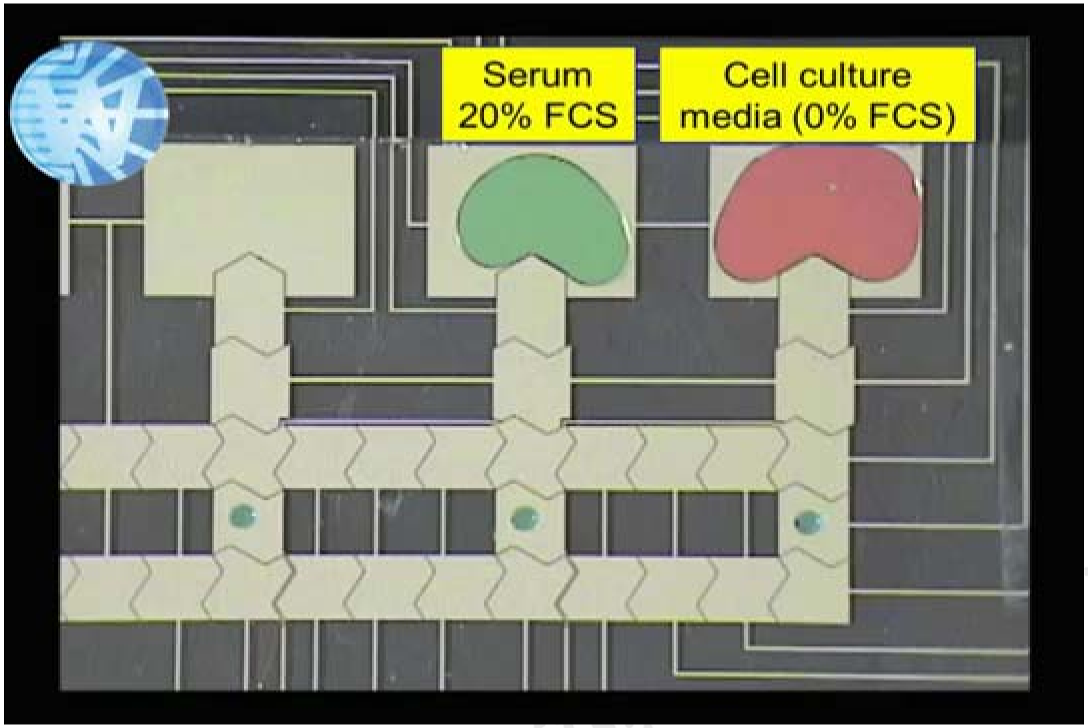
$$C_T = \frac{7}{10}$$

Only two extreme CFs are supplied:

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

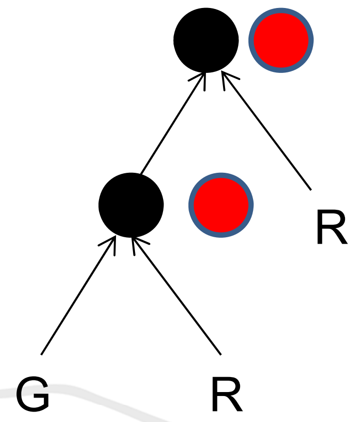
Buffer solution with 0% concentration or $\frac{0}{1024}$

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

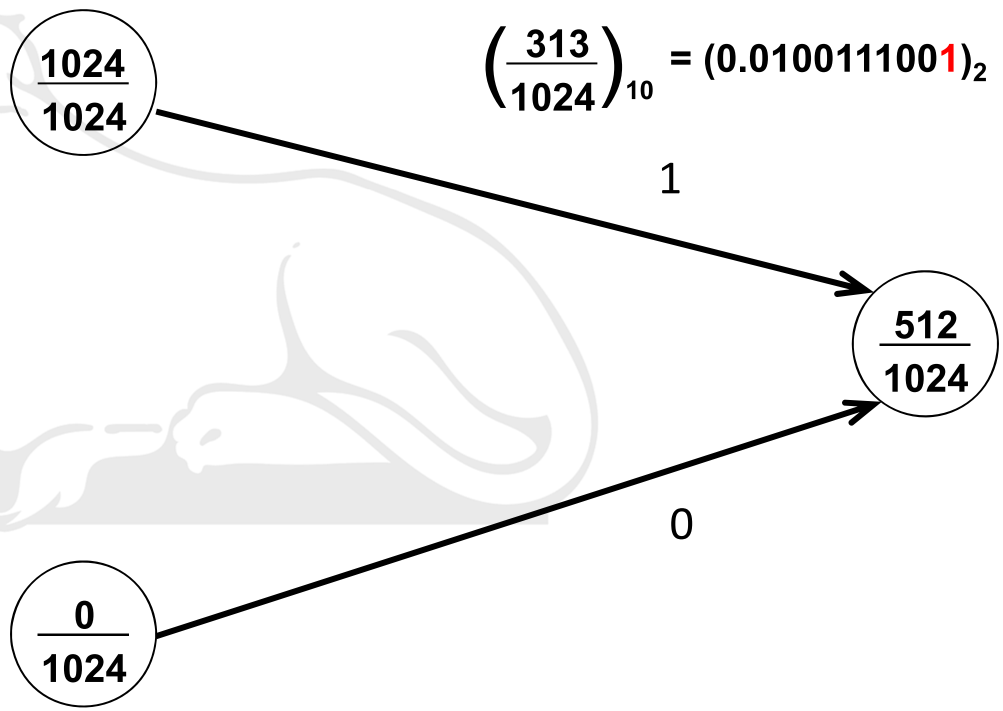
T2 = (G+3R)/4 :-> CF(G)=1/4

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

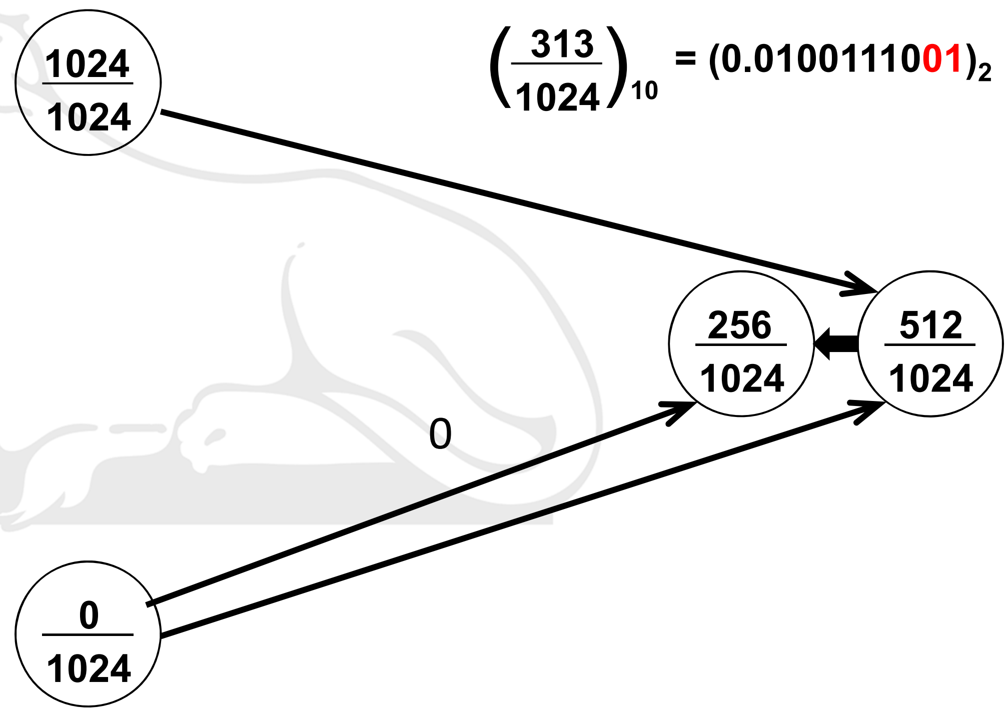
Two extreme CFs are supplied:

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

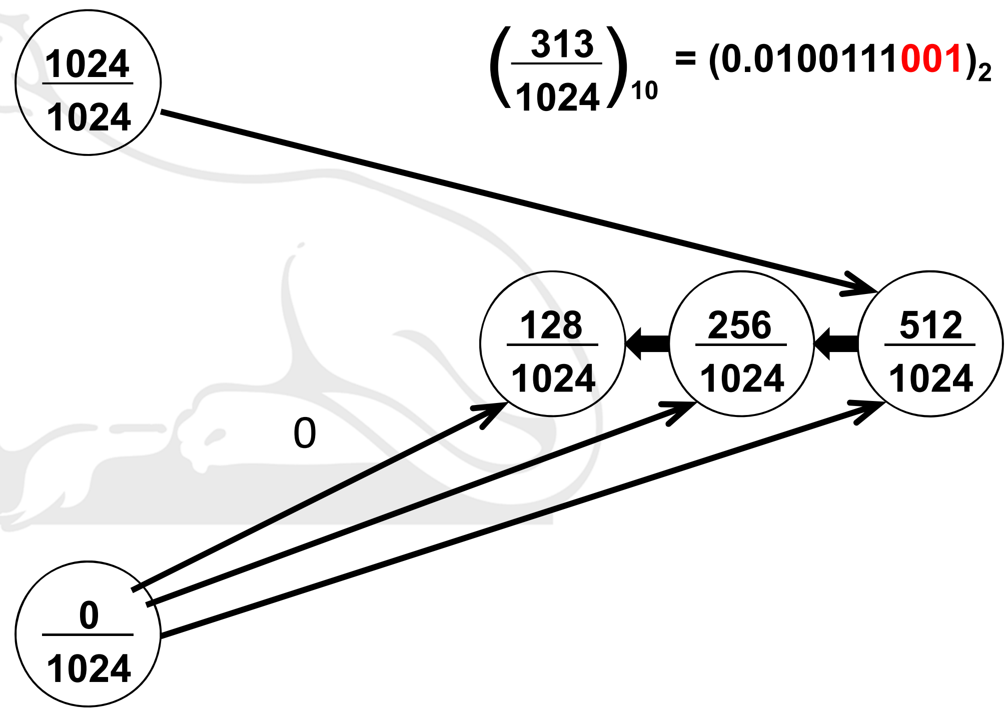
Two extreme CFs are supplied:

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

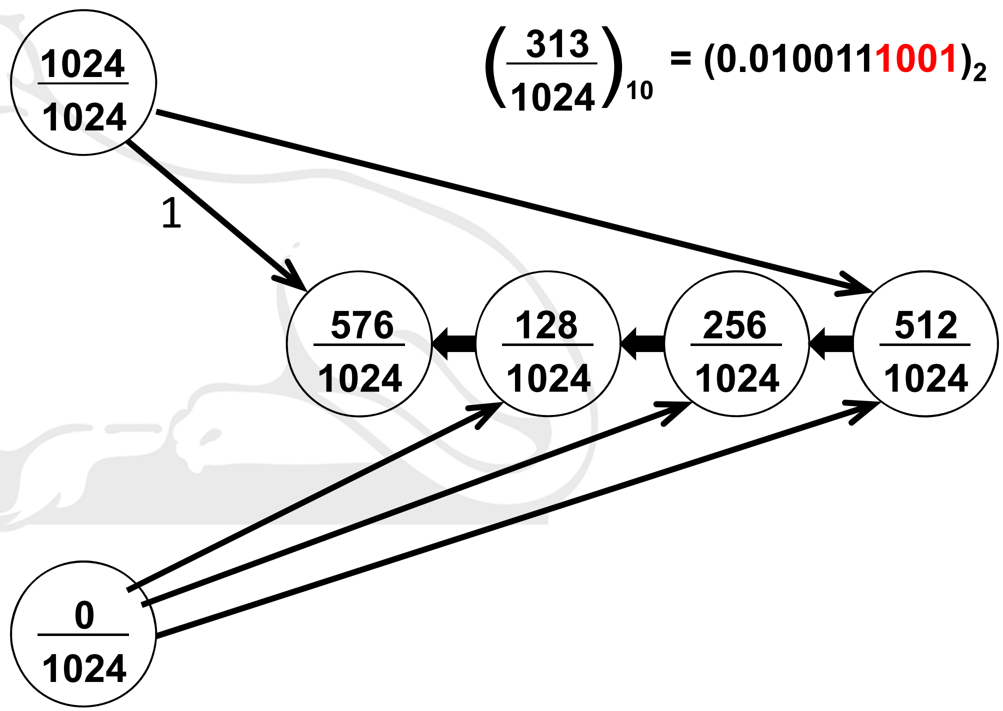
Two extreme CFs are supplied:

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.

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

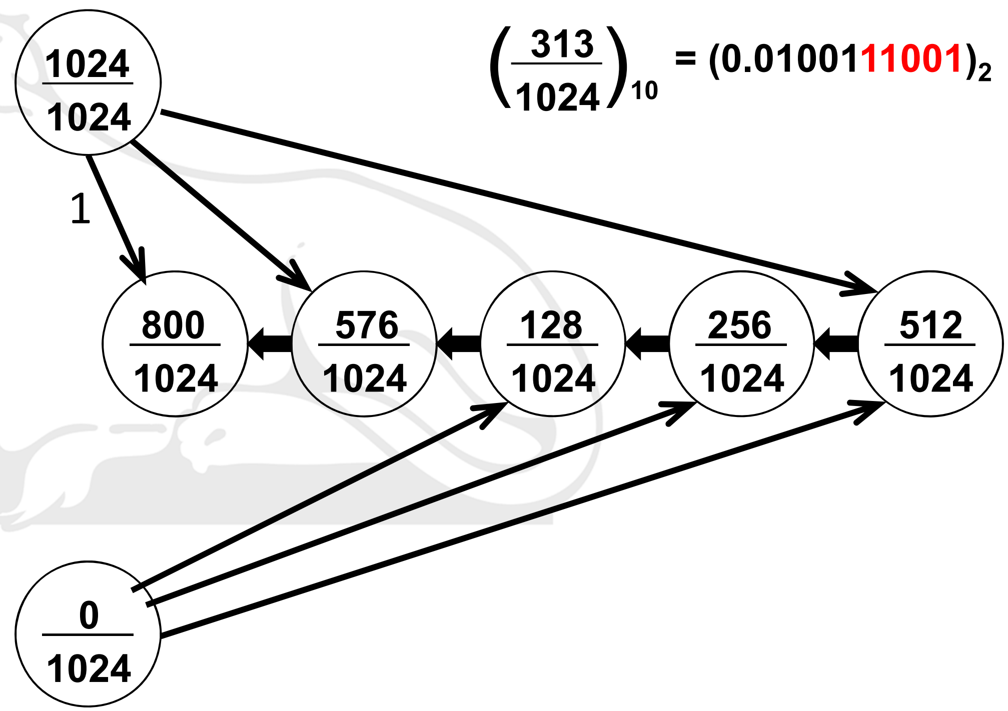


Dilution Algorithm: twoWayMix

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.

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



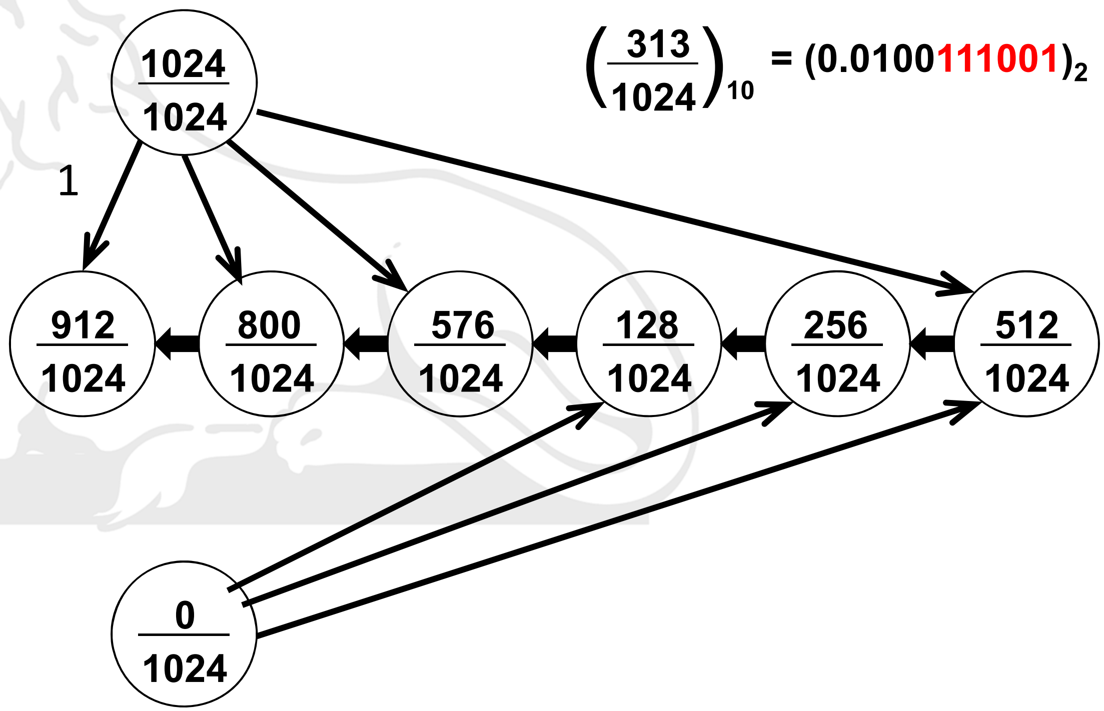
Dilution Algorithm: twoWayMix

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

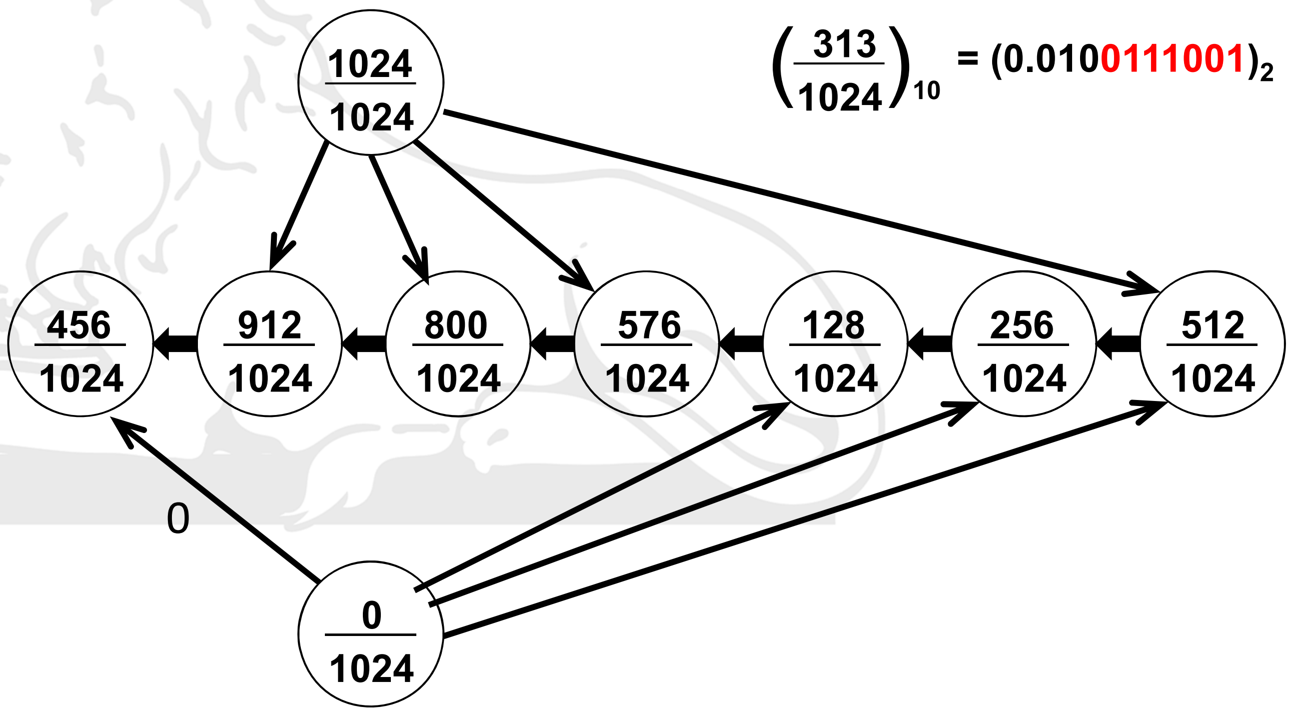
Two extreme CFs are supplied:

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.

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

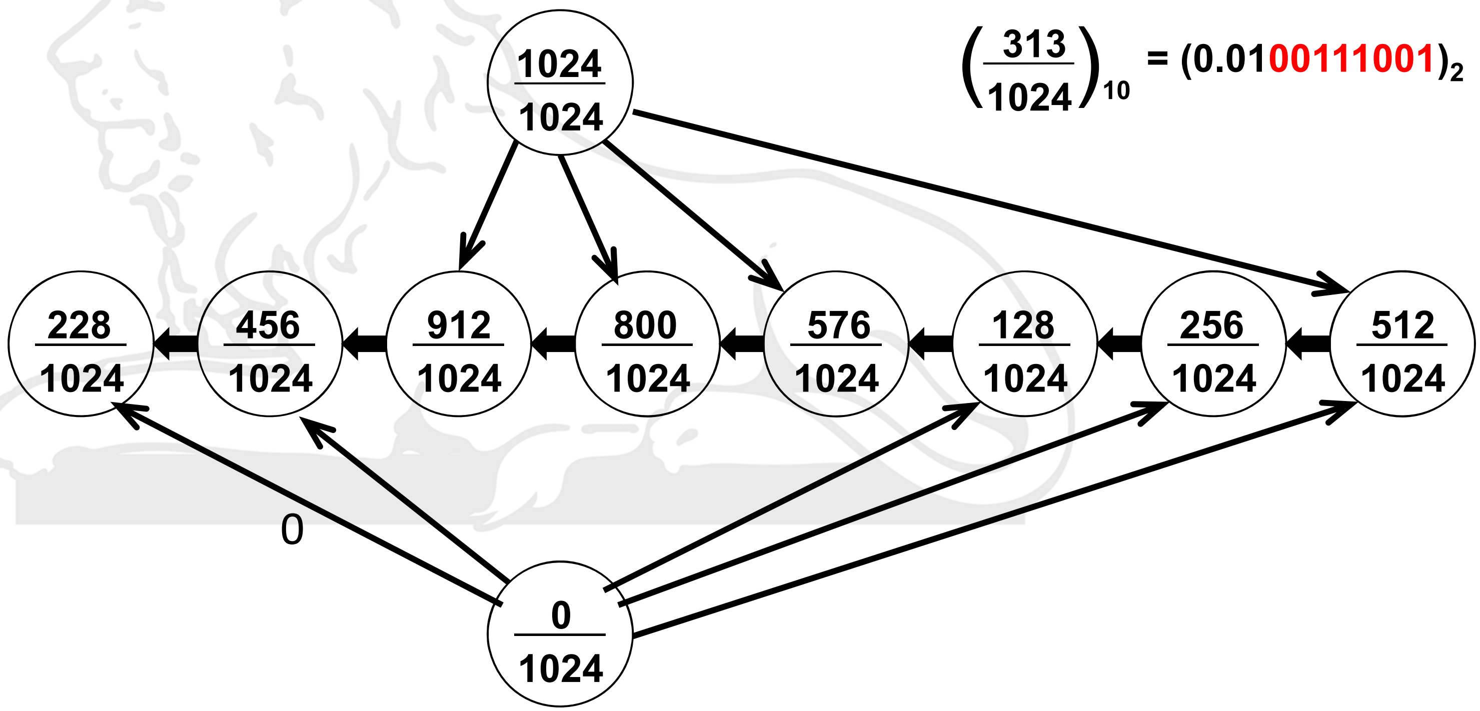


Dilution Algorithm: twoWayMix

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

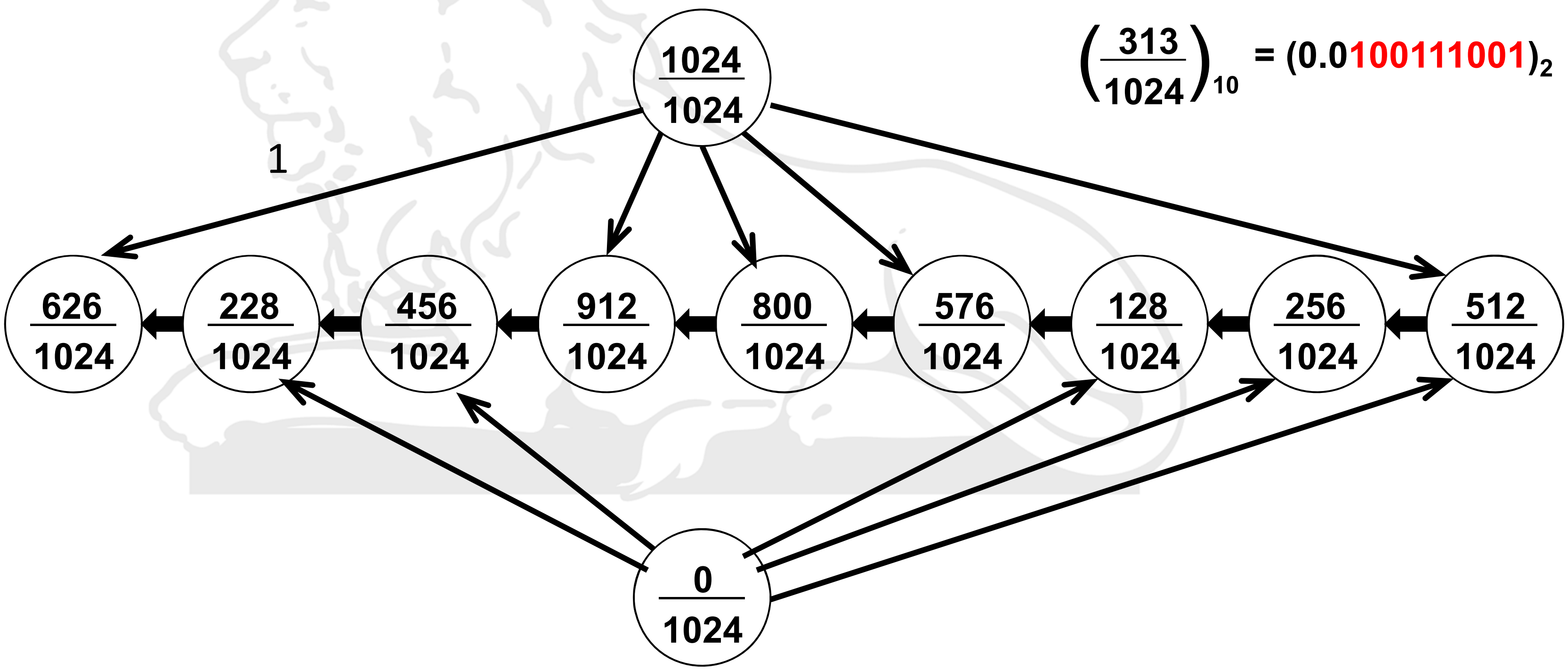
Two extreme CFs are supplied:

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.

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



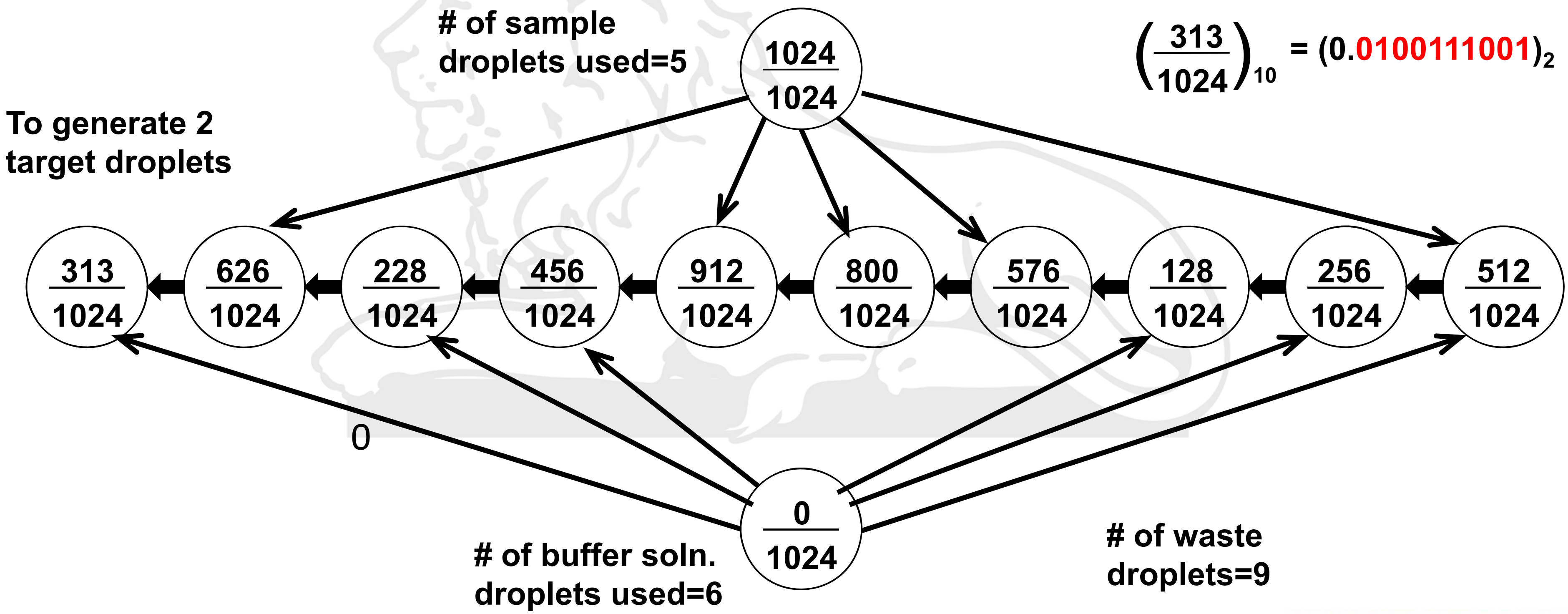
Dilution Algorithm: twoWayMix

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.



Mixing of Several Fluids

Mixing on Biochip:

Courtesy:
IMNF Lab, University of Texas Arlington, 2013

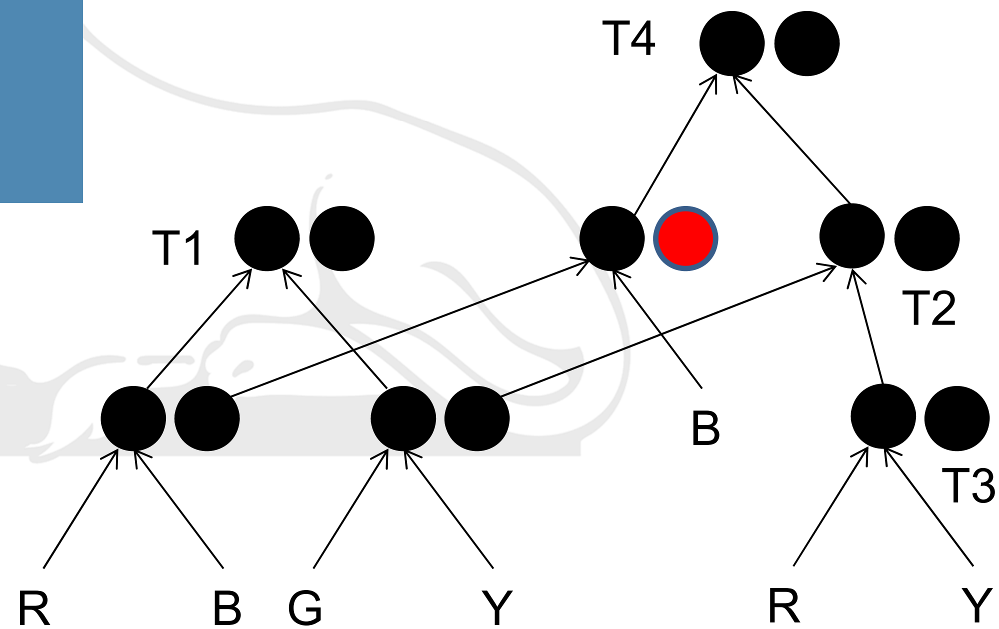
On-Chip Mixing

T1 = R:G:B:Y = 1:1:1:1

T2 = R:G:Y = 1:1:2

T3 = R:Y = 1:1

T4 = R:G:B:Y = 2:1:3:2



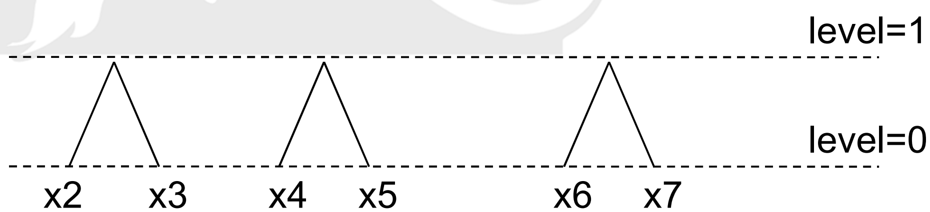
Mixing Algorithm: MinMix

Thies et al., Natural Computing, 2008

Target ratio of volumes of x1, x2, x3, x4, x5, x6, x7 = 2 : 3 : 5 : 7 : 11 : 13 : 87

Target Ratio =
2 : 3 : 5 : 7 : 11 : 13 : 87

x_1 : 0 0 0 0 0 1 $\boxed{0}_2$
x_2 : 0 0 0 0 0 1 $\boxed{1}_2$
x_3 : 0 0 0 0 1 0 $\boxed{1}_2$
x_4 : 0 0 0 0 1 1 $\boxed{1}_2$
x_5 : 0 0 0 1 0 1 $\boxed{1}_2$
x_6 : 0 0 0 1 1 0 $\boxed{1}_2$
x_7 : 1 0 1 0 1 1 $\boxed{1}_2$



Mixing Algorithm: MinMix

Thies et al., Natural Computing, 2008

Target ratio of volumes of x1, x2, x3, x4, x5, x6, x7 = 2 : 3 : 5 : 7 : 11 : 13 : 87

Target Ratio =
2 : 3 : 5 : 7 : 11 : 13 : 87

x_1 : 0 0 0 0 0 1 0_2

x_2 : 0 0 0 0 0 1 1_2

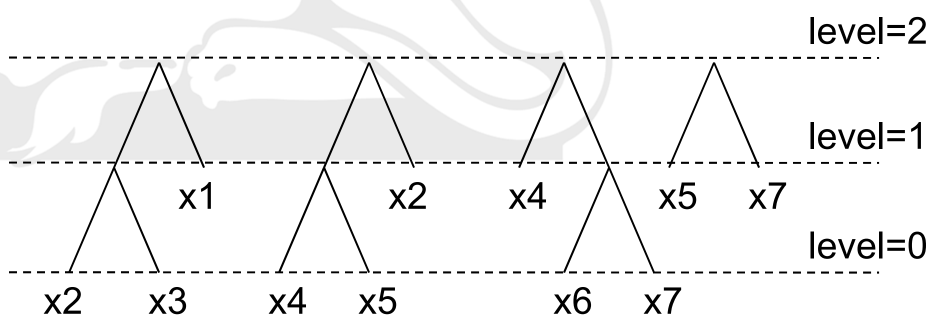
x_3 : 0 0 0 0 1 0 1_2

x_4 : 0 0 0 0 1 1 1_2

x_5 : 0 0 0 1 0 1 1_2

x_6 : 0 0 0 1 1 0 1_2

x_7 : 1 0 1 0 1 1 1_2



Mixing Algorithm: MinMix

Thies et al., Natural Computing, 2008

Target ratio of volumes of x1, x2, x3, x4, x5, x6, x7 = 2 : 3 : 5 : 7 : 11 : 13 : 87

Target Ratio =

2 : 3 : 5 : 7 : 11 : 13 : 87

x_1 : $0\ 0\ 0\ 0\ 0\ 1\ 0_2$

x_2 : $0\ 0\ 0\ 0\ 0\ 1\ 1_2$

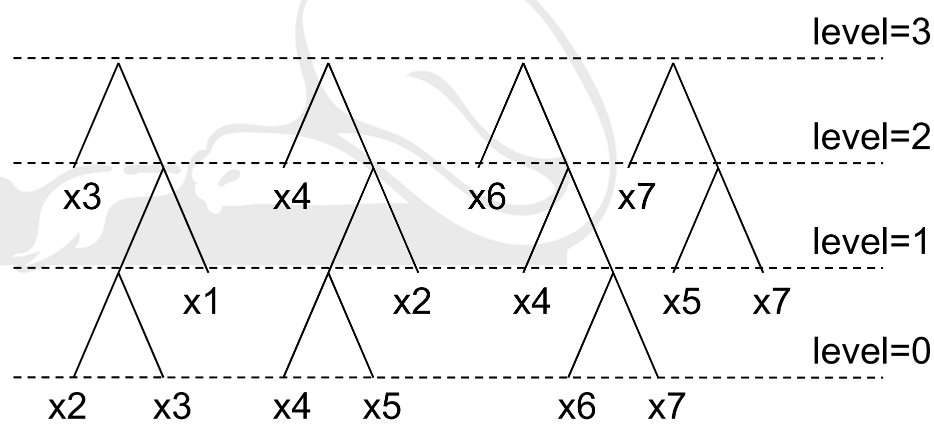
x_3 : $0\ 0\ 0\ 0\ 1\ 0\ 1_2$

x_4 : $0\ 0\ 0\ 0\ 1\ 1\ 1_2$

x_5 : $0\ 0\ 0\ 1\ 0\ 1\ 1_2$

x_6 : $0\ 0\ 0\ 1\ 1\ 0\ 1_2$

x_7 : $1\ 0\ 1\ 0\ 1\ 1\ 1_2$



Mixing Algorithm: MinMix

Thies et al., Natural Computing, 2008

Target ratio of volumes of x1, x2, x3, x4, x5, x6, x7 = 2 : 3 : 5 : 7 : 11 : 13 : 87

Target Ratio =

2 : 3 : 5 : 7 : 11 : 13 : 87

x_1 : $0\,0\,0\,0\,0\,1\,0_2$

x_2 : $0\,0\,0\,0\,0\,1\,1_2$

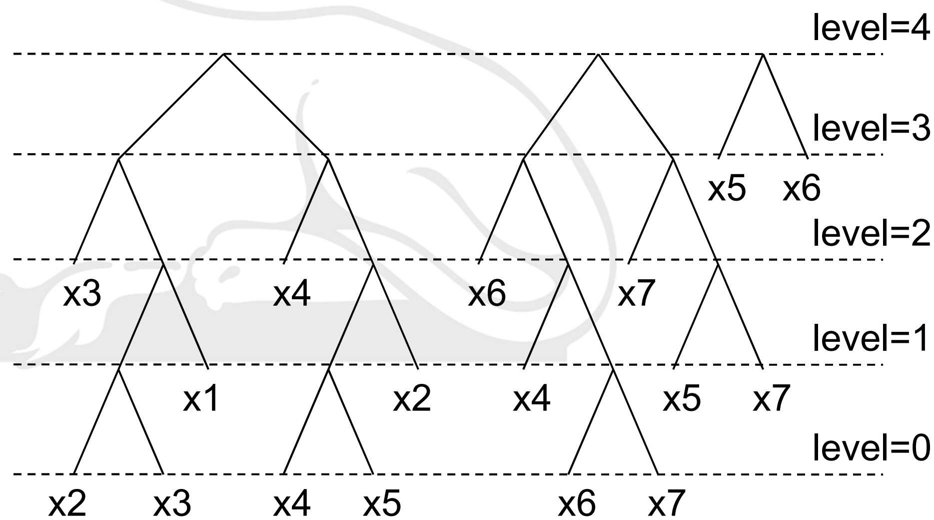
x_3 : $0\,0\,0\,0\,1\,0\,1_2$

x_4 : $0\,0\,0\,0\,1\,1\,1_2$

x_5 : $0\,0\,0\,1\,0\,1\,1_2$

x_6 : $0\,0\,0\,1\,1\,0\,1_2$

x_7 : $1\,0\,1\,0\,1\,1\,1_2$



Mixing Algorithm: MinMix

Thies et al., Natural Computing, 2008

Target ratio of volumes of x1, x2, x3, x4, x5, x6, x7 = 2 : 3 : 5 : 7 : 11 : 13 : 87

Target Ratio =

2 : 3 : 5 : 7 : 11 : 13 : 87

x_1 : $0\,0\,\boxed{0}\,0\,0\,1\,0_2$

x_2 : $0\,0\,\boxed{0}\,0\,0\,1\,1_2$

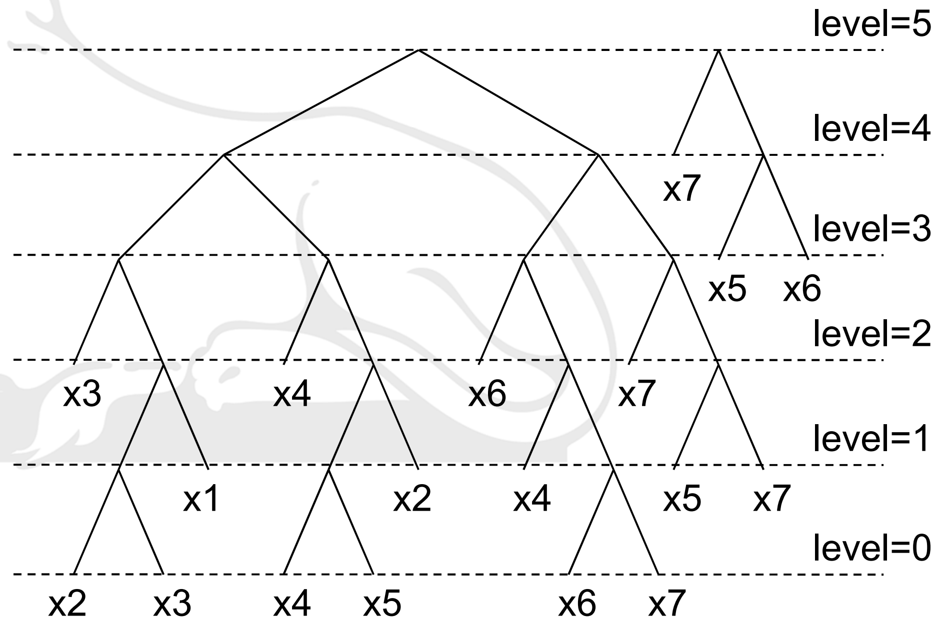
x_3 : $0\,0\,\boxed{0}\,0\,1\,0\,1_2$

x_4 : $0\,0\,\boxed{0}\,0\,1\,1\,1_2$

x_5 : $0\,0\,\boxed{0}\,1\,0\,1\,1_2$

x_6 : $0\,0\,\boxed{0}\,1\,1\,0\,1_2$

x_7 : $1\,0\,\boxed{1}\,0\,1\,1\,1_2$



Mixing Algorithm: MinMix

Thies et al., Natural Computing, 2008

Target ratio of volumes of x1, x2, x3, x4, x5, x6, x7 = 2 : 3 : 5 : 7 : 11 : 13 : 87

Target Ratio =

2 : 3 : 5 : 7 : 11 : 13 : 87

$\mathbf{x}_1 :\ 0\,0\,0\,0\,0\,1\,0_2$

$\mathbf{x}_2 :\ 0\,0\,0\,0\,0\,1\,1_2$

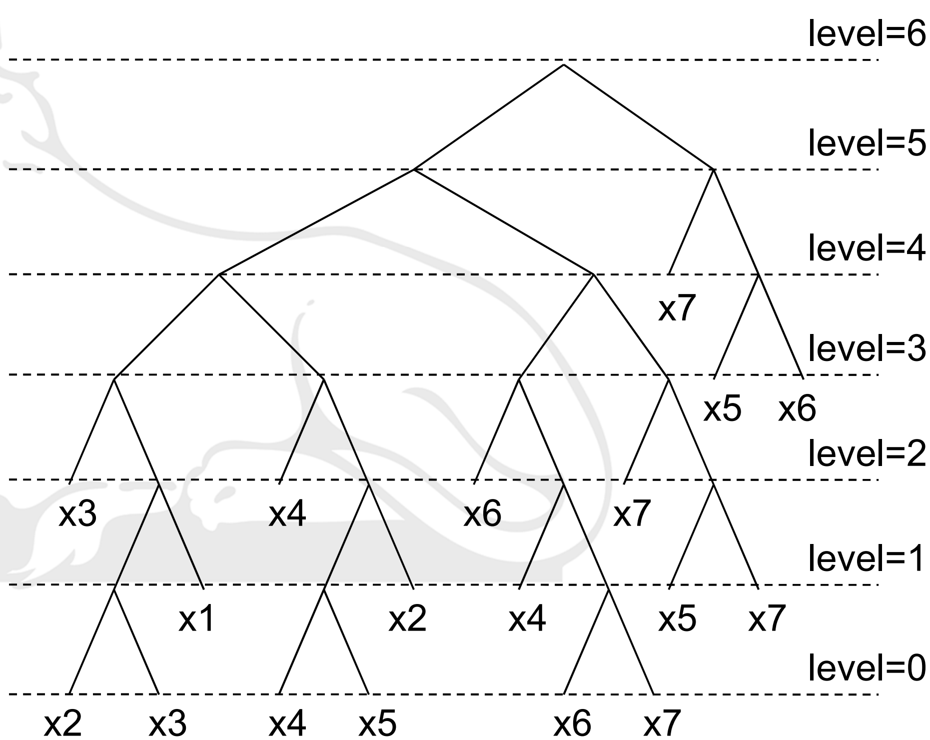
$\mathbf{x}_3 :\ 0\,0\,0\,0\,1\,0\,1_2$

$\mathbf{x}_4 :\ 0\,0\,0\,0\,1\,1\,1_2$

$\mathbf{x}_5 :\ 0\,0\,0\,1\,0\,1\,1_2$

$\mathbf{x}_6 :\ 0\,0\,0\,1\,1\,0\,1_2$

$\mathbf{x}_7 :\ 1\,0\,1\,0\,1\,1\,1_2$



Mixing Algorithm: MinMix

Thies et al., Natural Computing, 2008

Target ratio of volumes of x1, x2, x3, x4, x5, x6, x7 = 2 : 3 : 5 : 7 : 11 : 13 : 87

Target Ratio =

2 : 3 : 5 : 7 : 11 : 13 : 87

$\mathbf{x}_1 :\ \boxed{0}\ 0\ 0\ 0\ 0\ 1\ 0_2$

$\mathbf{x}_2 :\ \boxed{0}\ 0\ 0\ 0\ 0\ 1\ 1_2$

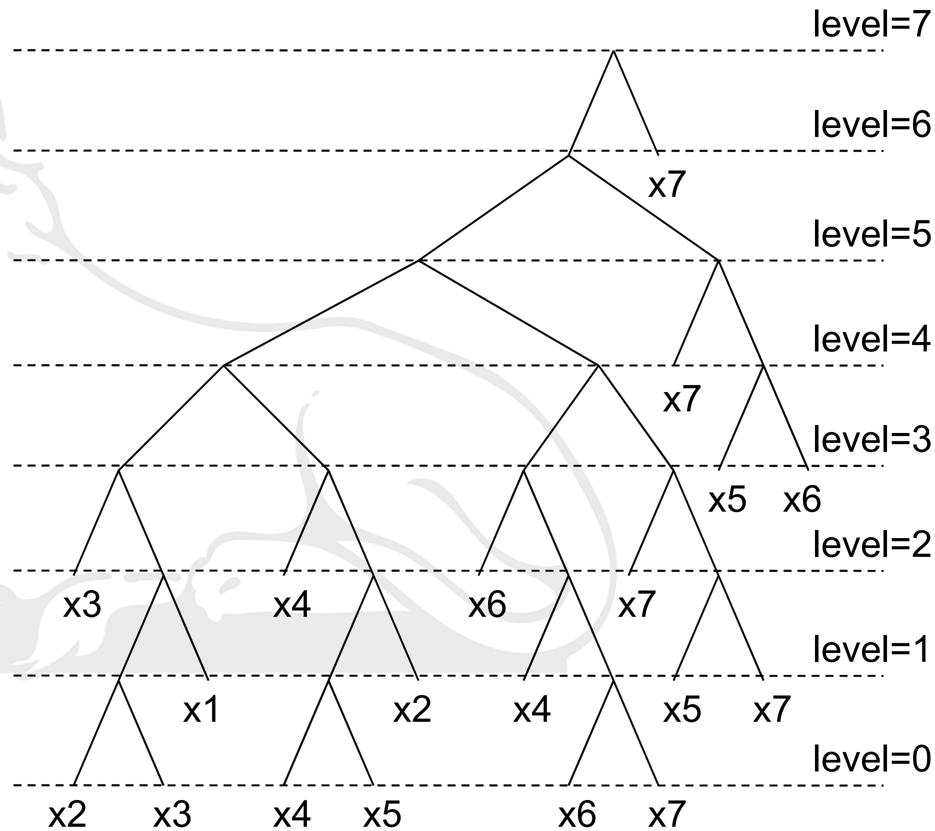
$\mathbf{x}_3 :\ \boxed{0}\ 0\ 0\ 0\ 1\ 0\ 1_2$

$\mathbf{x}_4 :\ \boxed{0}\ 0\ 0\ 0\ 1\ 1\ 1_2$

$\mathbf{x}_5 :\ \boxed{0}\ 0\ 0\ 1\ 0\ 1\ 1_2$

$\mathbf{x}_6 :\ \boxed{0}\ 0\ 0\ 1\ 1\ 0\ 1_2$

$\mathbf{x}_7 :\ \boxed{1}\ 0\ 1\ 0\ 1\ 1\ 1_2$



Mixing Algorithm: MinMix

Thies et al., Natural Computing, 2008

Target ratio of volumes of x1, x2, x3, x4, x5, x6, x7 = 2 : 3 : 5 : 7 : 11 : 13 : 87

MinMix-tree

Target Ratio =

2 : 3 : 5 : 7 : 11 : 13 : 87

x_1 : $0\ 0\ 0\ 0\ 0\ 1\ 0_2$

x_2 : $0\ 0\ 0\ 0\ 0\ 1\ 1_2$

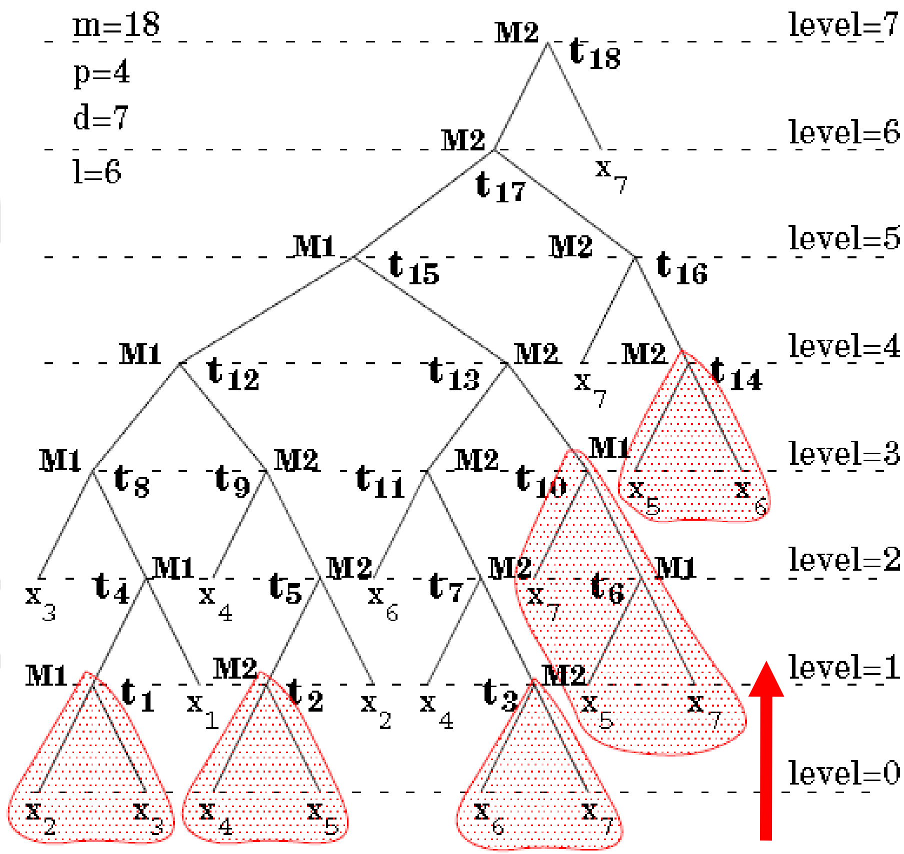
x_3 : $0\ 0\ 0\ 0\ 1\ 0\ 1_2$

x_4 : $0\ 0\ 0\ 0\ 1\ 1\ 1_2$

x_5 : $0\ 0\ 0\ 1\ 0\ 1\ 1_2$

x_6 : $0\ 0\ 0\ 1\ 1\ 0\ 1_2$

x_7 : $1\ 0\ 1\ 0\ 1\ 1\ 1_2$



Discussion on Concentration Factor in Dilution Process and Algorithm for Getting Directed Graph from Binary Fraction

Dilution of a Fluid and Binary Fractions:

(e) Let us consider that you are given some acid solution (also known as sample) and water (also known as buffer) having concentration factors as 100% and 0%, respectively. You are asked to follow the sequence of steps as below (considering unit volume as 10 ml):

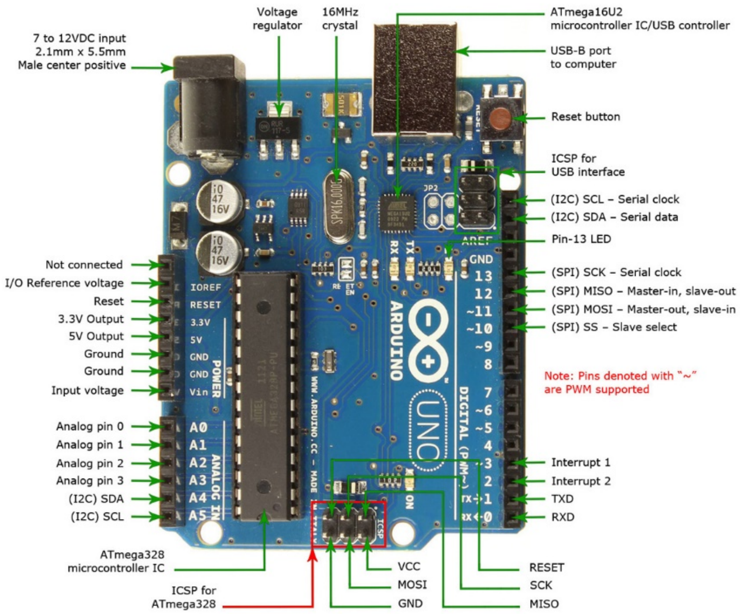
S1: Take one unit volume of sample (S) and one unit volume of buffer (B) in a single container.
S2: Mix the liquid mixture very well and Discard (remove) half of the total volume.
S3: Take one unit volume of S again and pour that in the container.
S4: Mix the liquid mixture very well and Discard (remove) half of the total volume.
S5: Take one unit volume of S again and pour that in the container.
S6: Mix the liquid mixture very well and Discard (remove) half of the total volume.
S7: Take one unit volume of B and pour that in the container.
S8: Mix the liquid mixture very well and Discard (remove) half of the total volume.
S9: Take one unit volume of B and pour that in the container.
S10: Mix the liquid mixture very well and Discard (remove) half of the total volume.
S11: Take one unit volume of S and pour that in the container.
S12: Mix the liquid mixture very well and Discard (remove) half of the total volume.

After Step 12, what is the concentration factor in decimal fraction (in the form of x/y, where x is numerator and y is denominator). Convert that decimal fraction into 6-bit binary number (without binary point).

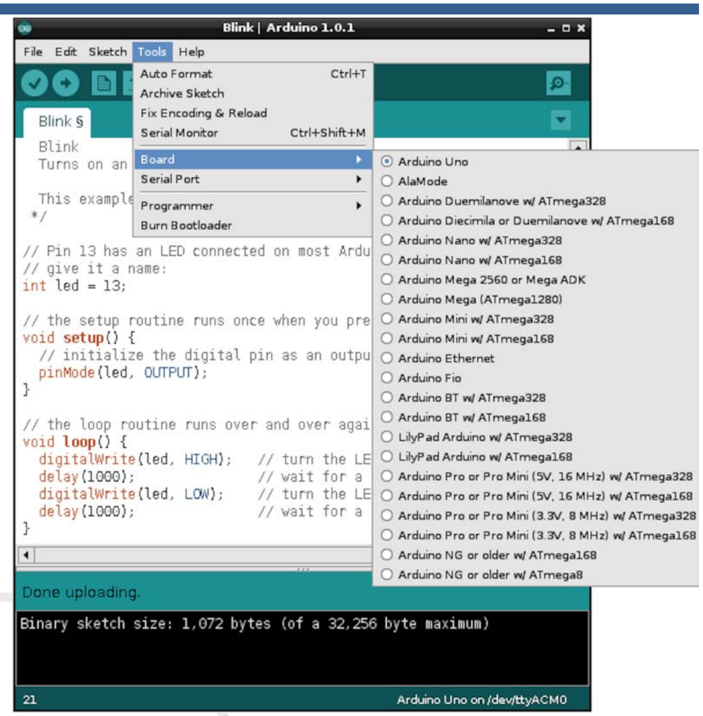
- Homework for CF=111/512 and A:B:C:D = 2:3:5:6

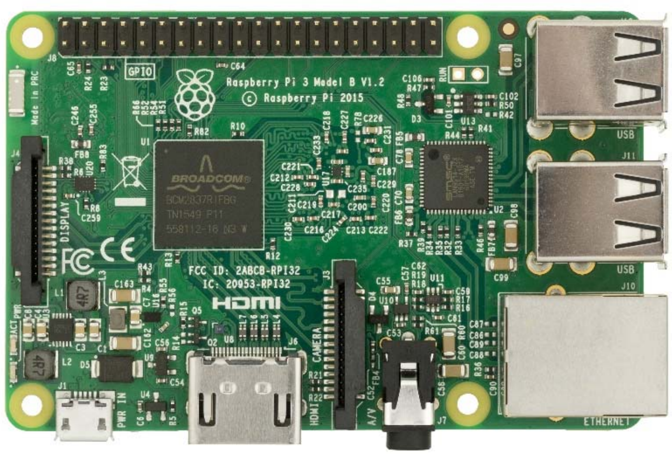
Scope in CoDA Lab

Microcontroller based System Design:

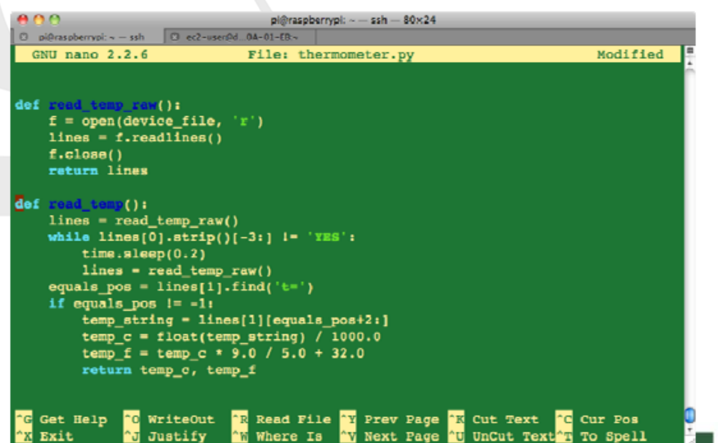


Arduino Uno kit

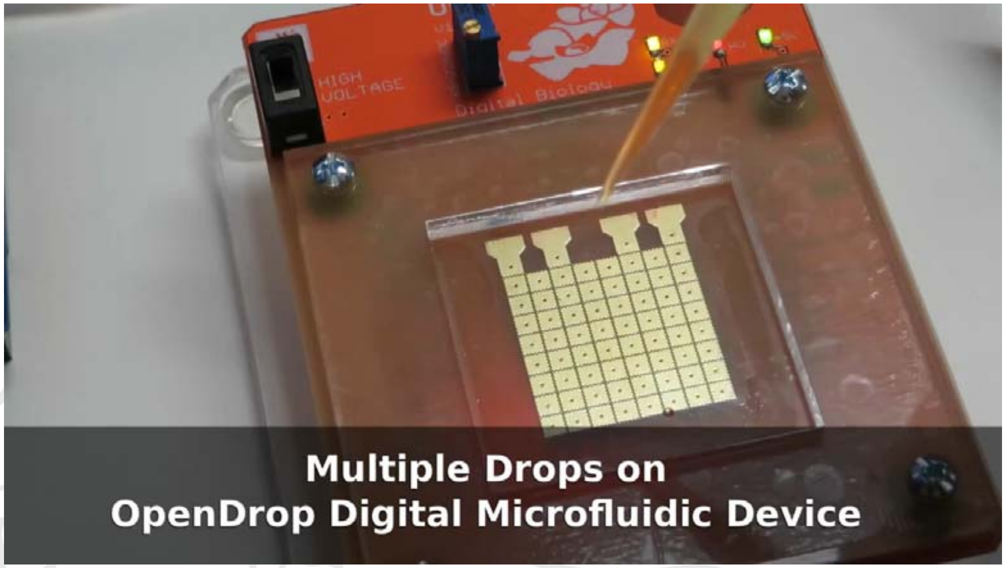




Raspberry Pi kit



OpenDrop Kits by GaudiLabs, Switzerland [May, 2017]:



CSN-101 (Introduction to Computer Science and Engineering)

Lecture 22: Binary Number System, Binary Number Storage and Registers, Boolean Algebra and Logic Gates

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]

Comparing the signed number systems

- *Positive numbers are the same in all three representations*
- Signed magnitude and one's complement have *two* ways of representing 0. This makes things more complicated
- Two's complement has asymmetric ranges; there is one more negative number than positive number. Here, you can represent -8 but not +8
- However, two's complement is preferred because it has only one 0, and its addition algorithm is the simplest

Decimal	S.M.	1's comp.	2's comp.
7	0111	0111	0111
6	0110	0110	0110
5	0101	0101	0101
4	0100	0100	0100
3	0011	0011	0011
2	0010	0010	0010
1	0001	0001	0001
0	0000	0000	0000
-0	1000	1111	—
-1	1001	1110	1111
-2	1010	1101	1110
-3	1011	1100	1101
-4	1100	1011	1100
-5	1101	1010	1011
-6	1110	1001	1010
-7	1111	1000	1001
-8	—	—	1000

Ranges of the signed number systems

- How many negative and positive numbers can be represented in each of the different systems on the previous page?

	Unsigned	Signed Magnitude	One's complement	Two's complement
Smallest	0000 (0)	1111 (-7)	1000 (-7)	1000 (-8)
Largest	1111 (15)	0111 (+7)	0111 (+7)	0111 (+7)

- In general, with n-bit numbers including the sign, the ranges are:

	Unsigned	Signed Magnitude	One's complement	Two's complement
Smallest	0	$-(2^{n-1}-1)$	$-(2^{n-1}-1)$	-2^{n-1}
Largest	2^n-1	$+(2^{n-1}-1)$	$+(2^{n-1}-1)$	$+(2^{n-1}-1)$

Real Numbers:

- *Conversion from real binary to real decimal*

  $-1101.1011_2 = -13.6875_{10}$

  since: $1101_2 = 2^3 + 2^2 + 2^0 = 13_{10}$ and

  $0.1011_2 = 2^{-1} + 2^{-3} + 2^{-4} = 0.5 + 0.125 + 0.0625 = 0.6875_{10}$

- Conversion from real decimal to real binary:

  $+927.45_{10} = +\ 1110011111.01\ 1100\ 1100\ 1100\ \ldots..$

927/2 = 463 + ½ ← LSB	0.45 × 2 = 0.9
463/2 = 231 + ½	0.9 × 2 = 1.8
231/2 = 155 + ½	0.8 × 2 = 1.6
155/2 = 57 + ½	0.6 × 2 = 1.2
57/2 = 28 + ½	0.2 × 2 = 0.4
28/2 = 14 + 0	0.4 × 2 = 0.8
14/2 = 7 + 0	0.8 × 2 = 1.6
7/2 = 3 + ½	0.6 × 2 = 1.2
3/2 = 1 + ½	0.2 × 2 = 0.4
1/2 = 0 + ½	0.4 × 2 = 0.8 ……

Floating-Point Number Formats:

- The term floating point number refers to representation of real binary numbers in computers
- IEEE 754 standard defines standards for floating point representations

- General format

$$\pm 1.\text{bbbbb}_{\text{two}} \times 2^{\text{eeee}}$$

or

$$(-1)^S \times (1+F) \times 2^E$$

- Where
  - S = sign, 0 for positive, 1 for negative
  - F = fraction (or mantissa) as a binary integer, 1+F is called *significand*
  - E = *exponent* as a binary integer, positive or negative (two's complement)

IEEE 754 Floating Point Standard:

- Biased exponent: true exponent range [-127,128] is changed to [0, 255]:
  - *Biased exponent* is an 8-bit positive binary integer.
  - True exponent obtained by subtracting 127_{ten} or 01111111_{two}

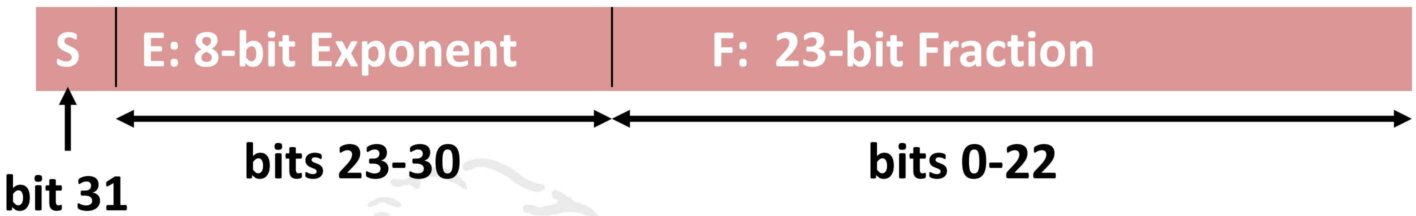
- First bit of significand is always 1:

  $$\pm 1.bbbb \ldots b \times 2^E$$

  - 1 before the binary point is implicitly assumed.
  - Bias = $2^{(k-1)} - 1$, in general

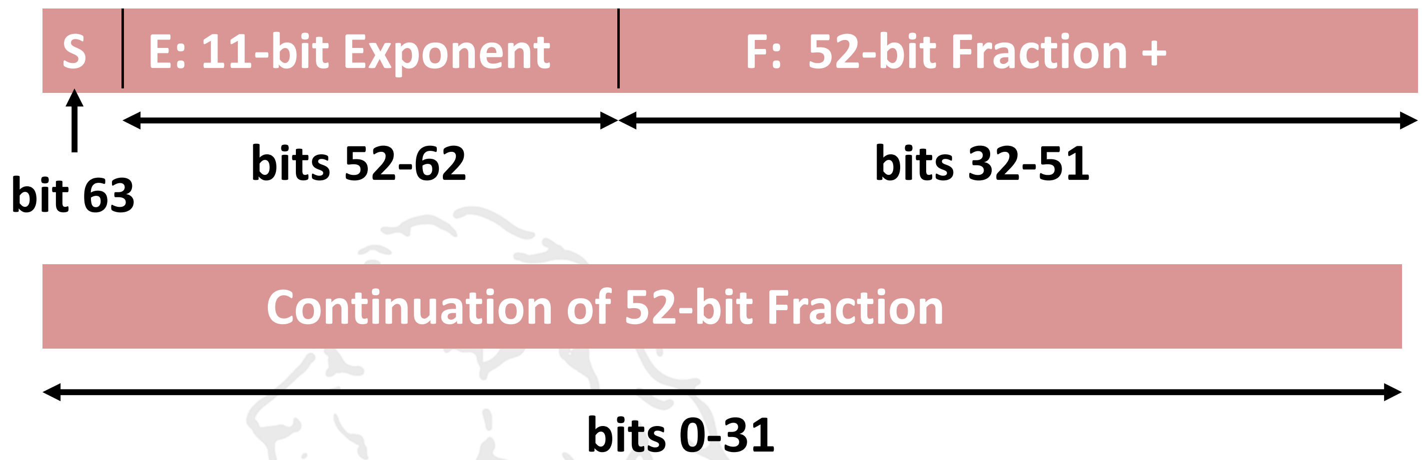
  - Significand field represents 23 bit fraction after the binary point.
  - Significand range is [1, 2), to be exact $[1, 2 - 2^{-23}]$
  - True exponent = biased exponent – 127, for 32-bit representation

Floating-Point Numbers (Single Precision):



- $-127 \leq E \leq 128$, Max $|E| \sim 128$
- Overflow: Exponent requiring more than 8 bits. Number can be positive or negative.
- Underflow: Fraction requiring more than 23 bits. Number can be positive or negative.

Floating-Point Numbers (Double Precision):



- -1023 ≤ E ≤ 1024, Max |E| ~ 1024
- Overflow: Exponent requiring more than 11 bits. Number can be positive or negative.
- Underflow: Fraction requiring more than 52 bits. Number can be positive or negative.

IEEE 754 Floating Point Standard:

(a) Format

- +/- 1.significand x $2^{exponent}$
- Standard for floating point storage
- 32 and 64 bit standards
- 8 and 11 bit exponent respectively
- Extended formats (both mantissa and exponent) for intermediate results

$$1.1010001 \times 2^{10100} = 0\ 10010011\ 10100010000000000000000 = 1.638125 \times 2^{20}$$
$$-1.1010001 \times 2^{10100} = 1\ 10010011\ 10100010000000000000000 = -1.638125 \times 2^{20}$$
$$1.1010001 \times 2^{-10100} = 0\ 01101011\ 10100010000000000000000 = 1.638125 \times 2^{-20}$$
$$-1.1010001 \times 2^{-10100} = 1\ 01101011\ 10100010000000000000000 = -1.638125 \times 2^{-20}$$

Conversion to Decimal:

normalized E
bits 23-30

F
bits 0-22

Sign bit S

1 10000001 01000000000000000000000

- Sign bit is 1, number is negative
- Biased exponent is $2^7+2^0 = 129$
- The number is

$$(-1)^S \times (1 + F) \times 2^{(\text{exponent} - \text{bias})} = (-1)^1 \times (1 + F) \times 2^{(129 - 127)}$$

$$= - 1 \times 1.25 \times 2^2$$

$$= - 1.25 \times 4$$

$$= - 5.0$$

Single-Precision Range

- Exponents 00000000 and 11111111 reserved
- Smallest value
  - Exponent: 00000001
    $\Rightarrow$ actual exponent = 1 – 127 = –126
  - Fraction: 000…00 $\Rightarrow$ significand = 1.0
  - $\pm 1.0 \times 2^{-126} \approx \pm 1.2 \times 10^{-38}$
- Largest value
  - exponent: 11111110
    $\Rightarrow$ actual exponent = 254 – 127 = +127
  - Fraction: 111…11 $\Rightarrow$ significand ≈ 2.0
  - $\pm 2.0 \times 2^{+127} \approx \pm 3.4 \times 10^{+38}$

Double-Precision Range

- Exponents 0000…00 and 1111…11 reserved
- Smallest value
  - Exponent: 00000000001
    $\Rightarrow$ actual exponent = 1 – 1023 = –1022
  - Fraction: 000…00 $\Rightarrow$ significand = 1.0
  - $\pm 1.0 \times 2^{-1022} \approx \pm 2.2 \times 10^{-308}$
- Largest value
  - Exponent: 11111111110
    $\Rightarrow$ actual exponent = 2046 – 1023 = +1023
  - Fraction: 111…11 $\Rightarrow$ significand ≈ 2.0
  - $\pm 2.0 \times 2^{+1023} \approx \pm 1.8 \times 10^{+308}$

Floating Point Arithmetic:

- Addition and subtraction are more complex than multiplication and division
- Need to align mantissas
- Algorithm:
  - Check for zeros
  - Align significands (adjusting exponents)
  - Add or subtract significands
  - Normalize result

Not required for CSN-101 course.
You will learn more about this in CSN-221 course.

Floating Point in C

- C Guarantees Two Levels

  `float`   single precision

  `double`   double precision

- Conversions
  - Casting between `int`, `float`, and `double` changes numeric values
  - `Double` or `float` to `int`
    - Truncates fractional part
    - Like rounding toward zero
    - Not defined when out of range
      - Generally saturates to TMin or TMax
  - `int` to `double`
    - Exact conversion, as long as int has $\leq$ 53 bit word size
  - `int` to `float`
    - Will round according to rounding mode

Binary Storage and Registers

■ Registers

♣ A *binary cell* is a device that possesses two stable states and is capable of storing one of the two states.

♣ A *register* is a group of binary cells. A register with *n* cells can store any discrete quantity of information that contains *n* bits.

n cells ⟹ 2^n possible states

- A binary cell
  - two stable state
  - store one bit of information
  - examples: flip-flop circuits, ferrite cores, capacitor
- A register
  - a group of binary cells
  - AX in x86 CPU
- Register Transfer
  - a transfer of the information stored in one register to another
  - one of the major operations in digital system
  - an example

Transfer of information

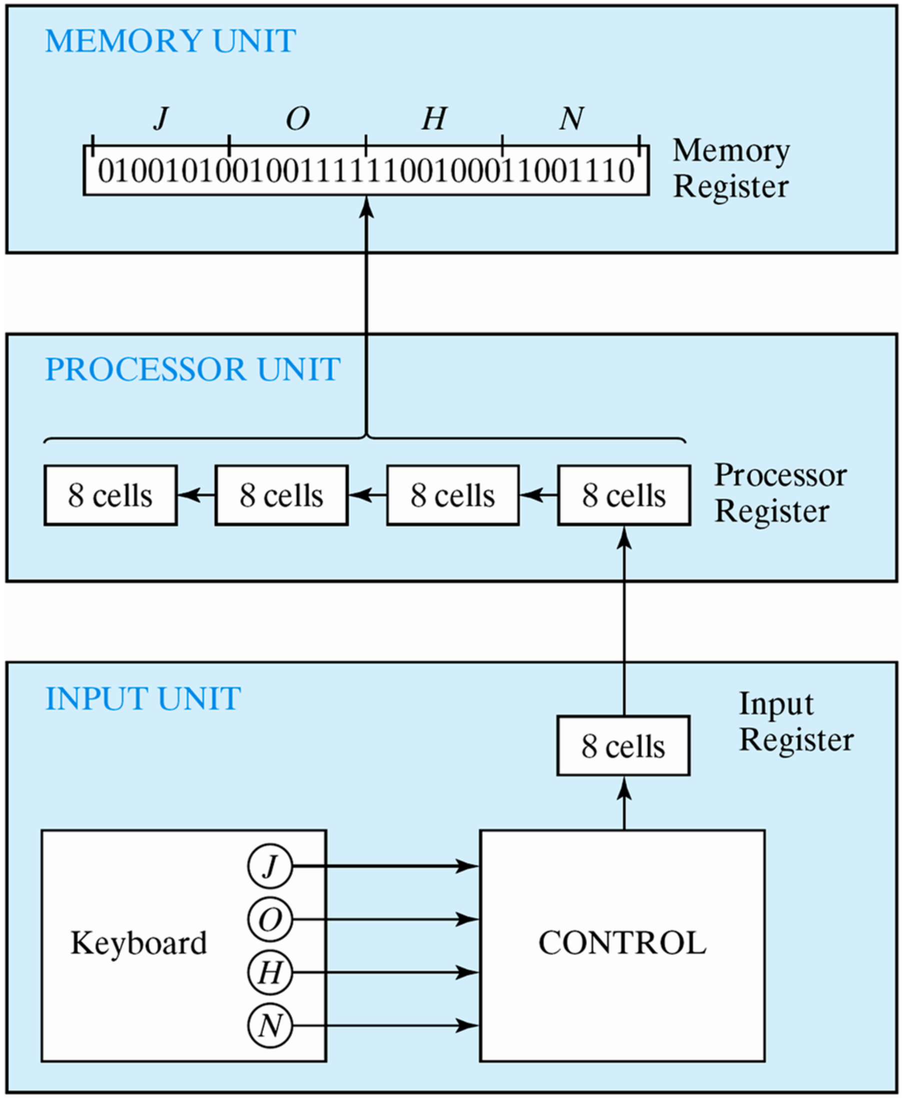


Fig. 1-1 Transfer of information with registers

- The other major component of a digital system
  - circuit elements to manipulate individual bits of information

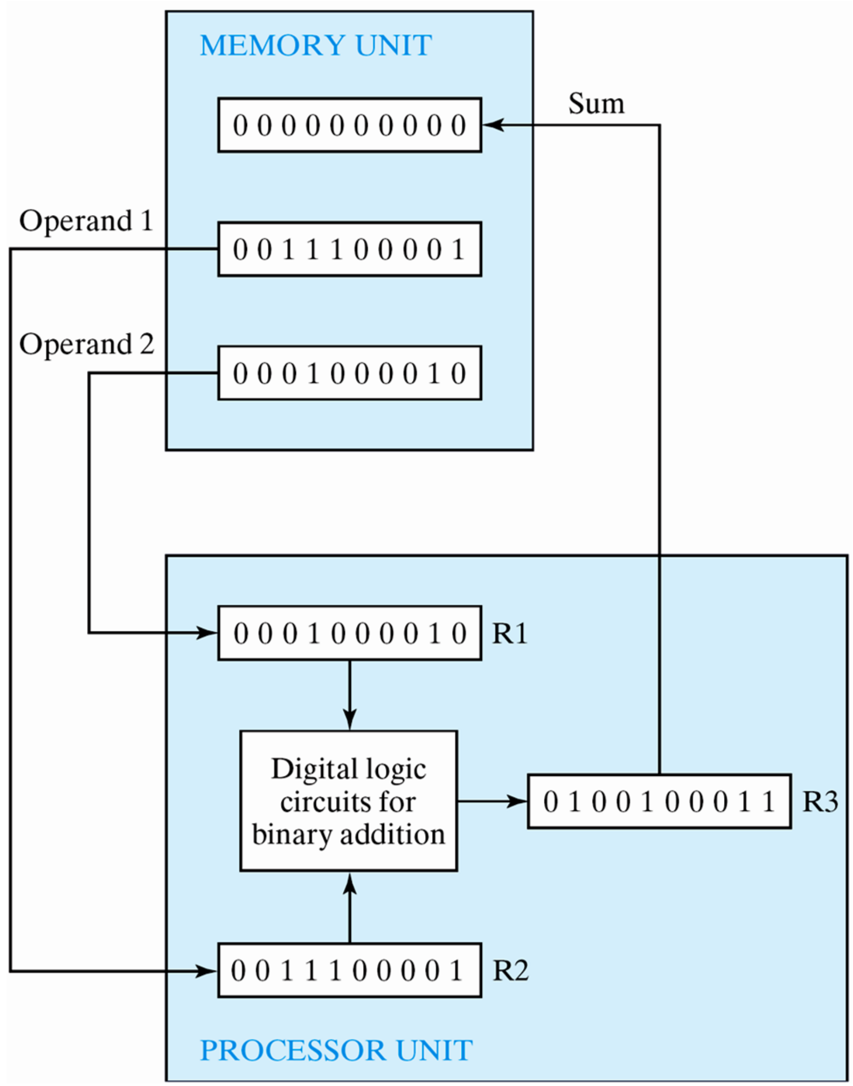


Fig. 1-2 Example of binary information processing

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