European Journal of Physics

PAPER

Hydraulic logic gates: building a digital water computer

IOP Publishing European Journal of Physics

Eur. J. Phys. **39** (2018) 025801 (9pp) https://doi.org/10.1088/1361-6404/aa97fc

Hydraulic logic gates: building a digital water computer

Nicolas Taberlet, Quentin Marsal, Jérémy Ferrand and Nicolas Plihon

Univ Lyon, ENS de Lyon, Univ Claude Bernard Lyon 1, CNRS, Laboratoire de Physique, F-69342 Lyon, France

E-mail: nicolas.taberlet@ens-lyon.fr

Received 5 September 2017, revised 16 October 2017
Accepted for publication 3 November 2017
Published 29 January 2018

Abstract

In this article, we propose an easy-to-build hydraulic machine which serves as a digital binary computer. We first explain how an elementary adder can be built from test tubes and pipes (a cup filled with water representing a 1, and empty cup a 0). Using a siphon and a slow drain, the proposed setup combines AND and XOR logical gates in a single device which can add two binary digits. We then show how these elementary units can be combined to construct a full 4-bit adder. The sequencing of the computation is discussed and a water clock can be incorporated so that the machine can run without any exterior intervention.

Keywords: fluidic, hydraulic calculus, fluid computer

(Some figures may appear in colour only in the online journal)

1. Introduction

Physics has been the cornerstone of all major technical advances throughout the history of computing [1], with fundamental concepts emerging from solid and fluid mechanics, electronics and solid-state physics, as well as quantum physics in recent years. In mechanics, pinwheel computers include Pascal's calculator [2] (designed as early as 1642, the Pascaline was the first calculator to be patented), the arithmometer (the first mass-marketed commercial calculator, about 5500 units were produced and sold world-wide) [3], Babbage's programmable analytical engine (designed in 1834 and considered to be the ancestor of modern computers [4]) or the famous cypher enigma machine, most notably used by Nazi Germany during WWII [5]. In fluid mechanics, processing and computing devices include the Tesla valve [6, 7], pressure-actuated valves acting as analogue transistors, nonlinear amplifiers relying on deflecting a weaker jet striking a stronger jet of fluid [8], digital pneumatic logic gates [9, 10], as well as the MONIAC (a water-based analogue computer designed to model economy) [11, 12]. It is important to note that computing using advanced fluidics systems was deeply studied and implemented in operational systems until the 1970's [13, 14], before fluidics could not compete against electronics. In electronics, vacuum tubes [15] remained fundamental components for electronics throughout the first half of the 20th century [16] until the invention of semiconductors and the discovery of the transistor effect (which was honored with the Nobel Prize in Physics in 1956) [17]. Another leap came with the invention of the integrated circuit (patented in 1959) for which Kilby won the 2000 Nobel Prize in Physics [17]. Finally, the concept of quantum computation has attracted much interest in the past two decades [18–20]. While promising recent technical developments in this field brought the concept closer to reality [21, 22], building a fully functioning large-scale quantum computer still remains a challenge. [23]

One of the most fundamental tasks any computer has to perform is arithmetic operations. Adding binary numbers (a and b) is nothing more than the routine children are taught in elementary school. For example, 1101 + 1001 (i.e. 13 + 9) is computed as follows (starting from the right-most digit):

- 1 and 1 is zero and 1 is carried
- 0 and 1 is zero, which makes 1 with the carry
- 1 and 0 is one,
- 1 and 1 is zero and 1 is carried
- resulting in 10110 (= 22).

In this article we present an easy-to-build hydraulic digital adding machine which relies on simple physical principles proposed by Gitton [24]. The present article reports some results obtained from an advanced undergraduate experimental project intended to describe the possible analogies between digital electronic computing and fluidics. This project provided insights into the analogy between hydraulics and electrokinetics, but also into the translation from theoretical digital computation to its physical implementation. An introduction to the notions related to clock requirement in digital computing and to computation error handling may also be grasped. It was conducted as a series of four 4 h-long lab projects in preparation for the International Physicists Tournament http://iptnet.info, a world-wide competition for undergraduate students. Unlike the typical physics exam, the problems must not only be presented, but also challenged and reviewed by the other participants allowing students to respectively assume the roles of researchers, referees and editors. In addition to the challenge that the tournament represents, it provides students with an exciting and eye-opening experience in which they learn how to design experiments with the aim of solving physics problems, and to constructively criticise scientific solutions. This article reports some of the results obtained from a series of six 4 h-long labs, where students solved the question of the origin of the flickers as a problem-based learning. The organisation of the article is the following: we first explain (section 2) how using a siphon, a slow drain and a overflowing gauge one is able to construct AND and XOR logic gates in one single device. In section 3 we explain how these processing units can be arranged so as to form a full binary adding machine, while section 4 describes the experimental 4-bit water computer that we have built.

Table 1. Carry, c_0, and digit, d_0, of the sum of a_0 and b_0. In logical operations: $c_0 = a_0$ AND b_0 while $d_0 = a_0$ XOR b_0.

a_0	0	0	1	1
b_0	0	1	0	1
c_0	0	0	0	1
d_0	0	1	1	0

2. Adding two digits

2.1. Logical operations

Adding the first digit of each number (a_0 and b_0) yields the first digit of their sum (d_0) and the first carry (c_0). The carry is 1 if and only if $a_0 = b_0 = 1$ while the first digit of the sum is 1 if and only if either a_0 or b_0 is 1 (not both) (see table 1). In terms of logical operations this can be summarised as

$$c_0 = a_0 \mathrm{AND} b_0 \qquad d_0 = a_0 \mathrm{XOR} c_0. \tag{1}$$

2.2. Hydraulic AND and XOR gates

The addition of the first two digits therefore only requires an AND and XOR gate. The hydraulic device shown in figure 1 combines both gates in one single apparatus. It consists of a central container into which the content of two smaller cups (representing a_0 and b_0) is simultaneously poured. The cups can be either completely empty (representing a 0) or completely full (representing a 1) and their volume is half that of the central container. A small hole is drilled at the bottom of the main container (bottom-right in figure 1) so that its content can slowly drain out. On the other side (bottom left in figure 1), a large hole is connected to a siphon, whose height is greater than half of that of the main container.

If one and only one of the top cups is full, the water poured into the central container gradually winds up in the bottom-right cup. On the other hand, if both top cups are full, the siphon is almost instantly primed and the content of the main container quickly empties out into the bottom left cup, whose overflow tube ensures that the results remains binary. Overflow is a generic feature of fluidics devices in which vents were widely implemented [14]. Note that even when the siphon is primed and the water is quickly evacuated, a small amount still leaks through the slow drain. The diameters of the drain and siphon must be chosen so that the flow rate through the siphon is far greater.

Therefore, the bottom cups give the results of the logical operation AND (for the left cup) and XOR (for the right cup). The smaller cups (top and bottom) act as memory units while the larger central container incorporates the logic gates.

It is important to note that the gates presented here are basic digital components which do not rely on basic analogue hydraulic components. This is a fundamental difference with electronic digital circuits, designed as the association of several analogue circuits, namely transistors [25].

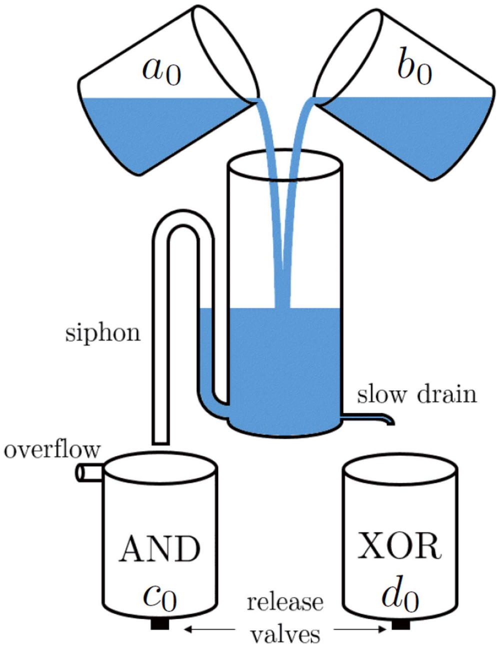

Figure 1. (a) Hydraulic digital arithmetic unit: the content of one (and only one) cup (i.e. 1 + 0) slowly leaks out of the central container through the XOR gate while the content of two cups (1 + 1) primes the siphon and is rapidly evacuated through the AND gate. The smaller cups have a volume of 25 ml while that of the central container is 50 ml, and the height of the siphon is 3/4 of the total height. (b) Sketch of the transfert functions of the AND and XOR gates assuming the volume of input a_0 is exactly V_0 (see text for details).

2.3. Technical details

The central container is a 50 ml plastic test tube, of diameter $D = 27$ mm. The smaller cups (memory units) are made of the same test tube cut in half ($V_0 = 25$ ml). The hole through which water slowly drains out is pierced using an incandescent needle which gives a diameter d of typically 500 μm ($\pm$20%). The siphon and the overflow tube are built using plastic straws (5 mm in diameter). Straws are very convenient as they are made to be bent without buckling or pinching. In this device it took approximatively 2 min for the content of one cup (25 ml) to slowly drain through the XOR gate whereas it takes less than 5 s for the content of two cups (50 ml) to quickly evacuate through the AND gate (the primed siphon).

In order to guide the water gradually leaking through the XOR gate, a straw (see figure 3) is glued around the small hole. A release valve, consisting of a 5 mm rubber stopper, is placed at the bottom of each memory unit for further processing.

Note that capillary effects may cause a number of problems: some liquid can stay trapped in the straws while the flow through the small drain can be hindered. This can be overcome by reducing the surface tension of the liquid. Soap must be avoided since it creates bubbles or foam but adding a small amount of ethanol to the water is enough to considerably help.

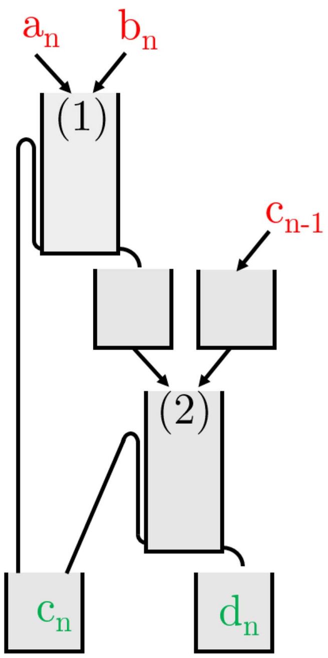

Figure 2. Sketch of a full-adder allowing one to add two digits (a_n and b_n) while taking into account the carry (c_{n-1}) resulting from the earlier stages of computation. The full-adder yields the *n*th digit of the sum (d_n) as well a the *n*th carry (c_n). The computation must be sequenced so as to ensure that the second computation of the stage (2) is triggered only after all previous operations have terminated.

3. Full-adder

3.1. Adding the nth digits

The device sketched in figure 1 allows one to add two 1-bit numbers and yields a 2-bit number and is known as a half-adder. [26] As explained above, it can be used to compute the sum of the first digits (a_0 and b_0). However, when adding any *n*th digits (a_n and b_n), the carry (c_{n-1}) originating from all previous calculations must be taken into account. A full-adder is therefore an operation with three inputs and two outputs: (a_n, b_n, c_{n-1}) → (c_n, d_n).

As illustrated in the introductory addition of 1101 (13) and 1001 (9), a full-adder must first add the *n*th digits together, their XOR result then needing to be added to the $n - 1$th carry. That second operation gives the *n*th digit of the sum (d_n). The AND results of both computations give the *n*th carry. Note that not possible for both AND gates to yield 1 and no further precaution (such as an additional OR gate) needs to be taken. The combination of two half-adders is known as a full-adder and a sketch of a device built with our hydraulic gates is shown in figure 2.

Finally, one important feature of digital computation when cascading several elementary logic gates is the ability of the assembly to retain a small error probability. The simplest example is often provided considering the integrated circuit implementation of an electronic NOT gate (or inverter gate) which involves a transistor—a highly nonlinear component. As a consequence, the transfert function curve giving the output voltage V_{out} as a function of the input voltage V_{in} is nonlinear. The digital state of the output is unambiguously determined even when the value V_{in} is slightly different from the voltages level defining the digital 0 and 1 (which would for instance be respectively 0 and 5 V for TTL circuits). The implementation of the hydraulic adder proposed in this article is based on highly nonlinear implementation of AND and XOR gates. Let us for instance consider the water volume on the output (c_0) of the AND gate, assuming that the input water volume a_0 is exactly V_0. While the water volume of the input b_0 is lower than $V_0/2$, the siphon is not primed and the water volume of the output is 0; when it is larger than $V_0/2$, the siphon is primed and the output fills in very quickly. Due to the overflow gauge, the water volume filling the output is exactly V_0. This nonlinear transfer function ensures no error propagation via the AND gate. We note that, in the version of the adder proposed here, there is no error propagation from the XOR gates (since no results from the XOR gates are used as carries). However, the XOR gate transfer function is also highly nonlinear (an argument similar to the AND gate can easily be drawn). An upgraded version incorporating an overflow valve on the XOR gate result container would even constrain the output volume to be exactly V_0 even if the container of the low-level input is not exactly empty. In conclusion, the implementation proposed here meets the requirement for retaining small errors.

3.2. Clock rate

When using a full-adder, great care must be given to the timing of the operations. The first half-adder (container 1 in figure 2) can perform its computation as soon as the device is started ($t = 0$) but the second computation (container 2 in figure 2) must wait for all previous operations to have been performed. As explained above, the output of a AND gate in our device is given in less than 5 s while that of the XOR gate is given in $T = 2$ min. Therefore, one must ensure that enough time is given for all half-additions to be over before further processing. Therefore, the second operation of the nth full-adder must be performed at time $t = nT$. Our water computer can only achieve a disheartening clock rate of the order of 0.01 Hz, far less than the typical 10^{10} Hz of modern microchips [27]. The actual clock rate of the proposed adder can be derived using simple physical arguments. The largest time scale is that related to the slow drain of a volume V_0 in a container of diameter D through the small straw of diameter d. This corresponds to the time required to empty a container following Torricelli's law if the diameter D is larger than the capillary length and is given by $T = \sqrt{\frac{8V_0}{\pi D^2 g}}\frac{D^2}{d^2}$ where g is the gravity acceleration. Considering $d = 700\ \mu$m leads to $T = 140$ s. It is important to note that this clock rate does not depend on the viscosity nor the surface tension of the fluid. Keeping the geometry constant, speeding up the clock would (unphysically) require to increase the gravity. A more practical way could be to pressurise the top of the containers (though this might lead to practical difficulties) at a higher pressure. It is important to note that the clock estimate discussed here do not hold for very small container sizes of the order of the capillary length. For the sake of simplicity, the version of the full-adder presented here does not include a fully integrated hydraulic clock and relies on the successive manual opening of valves. However, it is in principle possible to build a dedicated clock which uses hydraulically controlled valves and whose sequencing is governed by the associated timing diagrams of the computer [14].

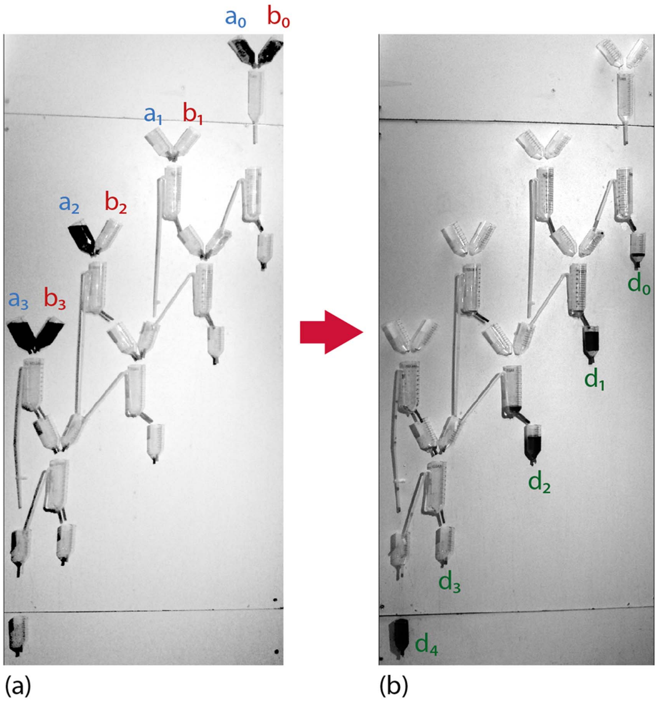

Figure 3. Pictures of our 4-bit hydraulic digital adder made of seven processing units. (a) The initial state shows the two numbers to be added ($a = 1101$) and ($b = 1001$) and (b) the result can be read in the final state ($d = 10110$) after 8 min of computation.

4. 4-bit hydraulic digital computer

4.1. A full 4 bit adder

Figure 3 shows a 4-bit adding machine built by assembling seven half-adders. As explained in section 2, the memory units (the smaller cups) are emptied out into the processing units (the central containers) through 5 mm holes at their bottom when rubber stoppers are removed. The red straws channel the results of the XOR gates into the memory units while the AND gates (the siphons) are made of green straws. For clarity, the water is died blue. The initial state is shown on figure 3(a): $a = 1101$ (=13) and $b = 1001$ (=9). After 8 min, the computation is complete and the result can be read in the bottom memory units (figure 3(b)). As expected the sum is indeed $d = 10110$ (=22).

4.2. Subtraction

The digital hydraulic adder presented above can also be used to perform subtractions (13 – 9 for instance) using a method known as the two's complement [28], that allows one to use signed binary numbers. The two's complement defines negative 9 as $-9 = \text{NOT}(9) + 1$, to which 13 can be added. Table 2 presents the operation. The result must not be interpreted as a 5-digit number since obviously the difference between two 4-digit numbers is less than 15. Instead, the 5th is only an indication of the sign. In the example shown in table 2, the result, 1 0100, must be read as +4.

Table 2. Computation of 13 – 9 using the two's complement method in which −9 is given by NOT(9) + 1. Since the difference cannot exceed $2^4 - 1 = 15$, the 5th digit must be ignored, and the result is 0100 = 4, as expected.

9		1	0	0	1
NOT(9)		0	1	1	0
NOT(9)+1		0	1	1	1
+13		1	1	0	1
−9 + 13	1	0	1	0	0

The NOT operation can be performed using the hydraulic half-adder presented in section 2 since for each digit NOT(a_n) = a_n XOR 1. One simply needs to collect the result of the XOR gate (the slow drain), the b cup being always filled, while the result of the AND gate is irrelevant. The 4-bit adder presented in section 4 can then first be used to add 1 to −9, and then used again to add 13 to the result.

5. Conclusion

In this article, the principle of a digital binary adder was described and the use of the AND and XOR logic gates was explained. We proposed a simple arithmetic unit relying on hydraulic principles which combines both AND and XOR gates in a single device, easily built with plastic test tubes and straws. Arranging seven of these elementary processing units, a functioning hydraulic 4-bit digital adder was constructed. The proposed design can be improved in several ways.

A water-based clock can be built to automatically trigger the computation every two minutes. Weights attached to the two rubber stoppers above the processing unit, through initially loose strings, can be placed on a balance beam with a leaking cup of water acting as a counterweight. After the desired duration, the balance beam tips over, causing the weight to fall and to pull the stoppers out, thereby releasing the water, i.e. triggering the computation. Such a simple independent clock allows the machine to run free of any human intervention during the computing process (which viewers may perceive as a deception).

Moreover, it is well known that during the adding process information is lost (in mathematical terms, addition is not bijective). Obviously, for instance adding 13 and 9 yields the same results as adding 8 and 14. However, in our 4-bit adding machine, a trick might be used to identify the original numbers (a and b) from the result (d). If each initial digit is assigned a specific colour, then analysing the hue of the final digits in the sum could allow one to deduce the value of the two initial numbers.

Acknowledgments

The authors acknowledge support from the University Lyon Claude Bernard, the Société Française de Physique and from the École Normale Supérieure de Lyon and its Physics Department and Laboratoire de Physique.

ORCID iDs

Nicolas Plihon https://orcid.org/0000-0001-8874-3674

References

[1] Wolf M 2016 *The Physics of Computing* (Amsterdam: Elsevier)
[2] Adamson D 1994 *Blaise Pascal: Mathematician, Physicist and Thinker About God* (Berlin: Springer)
[3] Reilly E D 2003 *Milestones in Computer Science and Information Technology* (Westport, CT: Greenwood Publishing Group)
[4] Bowden B V 1953 *Faster than Thought: a Symposium on Digital Computing Machines* (London: Pitman)
[5] Kozaczuk W 1984 *Enigma: How the German Machine Cipher was Broken, and How it was Read by the Allies in World War Two* (*Foreign Intelligence Book Series*) (Lanham, MD: University Publications of America)
[6] Tabeling P 2005 *Introduction to Microfluidics* (Oxford: Oxford University Press)
[7] Angelescu D E 2011 *Highly Integrated Microfluidics Design* (Boston, MA: Artech House)
[8] Hunt T M, Hunt T, Vaughan N D and Vaughan N 1996 *The Hydraulic Handbook* (Amsterdam: Elsevier)
[9] Rhee M and Burns M A 2009 Microfluidic pneumatic logic circuits and digital pneumatic microprocessors for integrated microfluidic systems *Lab Chip* **9** 3131–43
[10] Dewhirst R 1964 Passive pneumatic half-adder with no moving parts *PhD Thesis* Massachusetts Institute of Technology, Department of Mechanical Engineering
[11] Bissell C 2007 Historical perspectives—the Moniac a hydromechanical analog computer of the 1950s *IEEE Control Syst.* **27** 69–74
[12] Colander D 2011 The Moniac, modeling, and macroeconomics *Economia Politica* **28** 63–82
[13] Kirshner J 1975 *Design Theory of Fluidic Components* (New York: Academic)
[14] Belsterling C A 1971 *Fluidic Systems Design* (New York: Wiley Interscience)
[15] Schultz H L and Wadey W G 1951 A laboratory course in electronics *Am. J. Phys.* **19** 214–23
[16] Macdonald S and Braun E 1977 The transistor and attitude to change *Am. J. Phys.* **45** 1061–5
[17] Weste N, Harris D and Banerjee A 2005 CMOS VLSI design *Circuits Syst. Perspect.* **11** 739
[18] Bennett C H and DiVincenzo D P 2000 Quantum information and computation *Nature* **404** 247–55
[19] Mermin N D 2003 From cbits to qbits: teaching computer scientists quantum mechanics *Am. J. Phys.* **71** 23–30
[20] Candela D 2015 Undergraduate computational physics projects on quantum computing *Am. J. Phys.* **83** 688–702
[21] DiCarlo L *et al* 2009 Demonstration of two-qubit algorithms with a superconducting quantum processor *Nature* **460** 240–4
[22] Somaschi N *et al* 2016 Near-optimal single-photon sources in the solid state *Nat. Photon.* **10** 340–5
[23] Williams C P 2010 *Explorations in Quantum Computing* (London: Springer Science & Business Media)
[24] Gitton B Unfortunatly Gitton's design seems to have remained unpublished but a device using his concept was built at the Technisches Museum of Vienna, Austria. https://en.wikipedia.org/wiki/Bernard_Gitton
[25] Horowitz P and Hill W 1989 *The Art of Electronics* (Cambridge : Cambridge Univeristy Press)
[26] Page D 2009 *A Practical Introduction to Computer Architecture* (London: Springer Science & Business Media)
[27] Rodwell M J W 2001 *High-speed Integrated Circuit Technology: Towards 100 GHz Logic* vol 21 (Singapore: World Scientific)
[28] Null L *et al* 2014 *The Essentials of Computer Organization and Architecture* (Sunbury, MA: Jones & Bartlett Publishers)