Physics Letters A 167 (1992) 255–260
North-Holland

PHYSICS LETTERS A

Any nonlinear gate, with linear gates, suffices for computation

Seth Lloyd

Complex Systems Group (T-13) and Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Received 25 February 1992; revised manuscript received 20 May 1992; accepted for publication 22 May 1992
Communicated by A.R. Bishop

Devices that process signals in a linear fashion alone cannot be assembled into a general-purpose computer. But if a set of devices capable of realizing linear operations on continuous or discrete signals is supplemented by any device whose output is a nonlinear function of its input, the resulting set forms a basis for digital computation.

1. Introduction

Any combination of linear functions is itself a linear function: the set of linear functions is closed under composition. As a result, linear effects such as the interference of light or quantum-mechanical superposition of states cannot on their own produce even so simple a nonlinear device as a switch. Unless supplemented by devices whose output is a nonlinear function of their inputs, devices that process signals in a linear fashion cannot induce signals to interact, and do not allow the construction of a general-purpose computer. Mechanical switches, electro-mechanical relays, vacuum tubes and transistors have supplied the required nonlinearity for past and present computers. Many other nonlinear effects have been proposed as a physical basis for computation, including parametric excitation [1,2], tunnel diodes [3], the Josephson effect [4,5] optical bistability [6–8], the Kerr effect [9,10], the Aharonov–Bohm effect [11–13], etc. (for a review of different proposals for devices for the basis of computation, see refs. [14,15]). The question then arises, exactly what nonlinear effects suffice for computation?

The answer is that any will do in principle. If a set of devices realizing linear operations on continuous or discrete signals is supplemented by any device whose output is a nonlinear function of its inputs, the resulting set allows the construction of AND, OR and NOT gates and suffices for digital computation. So, for example, since an interferometer can add amplitudes coherently, in a regime in which coherent linear amplification and attenuation of light signals can be accomplished, virtually any nonlinear optical effect in principle allows the construction of an optical computer.

In practice, of course, not all nonlinear devices are good candidates for actually constructing a digital computer. In particular, some devices may generate large amounts of noise, or may amplify small deviations from the correct signal values, leading to poor signal stability. Nevertheless, a device whose output is a sufficiently well-behaved (i.e., continuous and three times differentiable) nonlinear function of its input can in theory be combined with linear devices to insure signal stability for small amounts of noise.

2. Constructing AND, OR and NOT gates

Suppose that we have at our disposal three different types of linear logic gates: (1) gates that produce a

constant output, (2) gates that multiply their inputs by constants, and (3) gates that add their inputs together (fig. 1). By taking these gates with different constants (which for discrete signals must be confined to discrete values), and connecting them using wires and fan-outs (gates that provide multiple copies of their inputs), we can produce a circuit that realizes any linear function, $l(x, y, ..., z)=\alpha x+\beta y+...+\gamma z+\delta$. (Here, the term "linear" will be used to refer to any function that can be built up from the composition of linear functions. Strictly speaking, such functions should be termed "affine".) Linear functions are the only sort of functions that these gates can produce.

To give rise to computation, the set of linear operations must be supplemented by nonlinear operations. Suppose that we have available a logic gate with n inputs $(x_1, x_2, ..., x_n)\equiv \boldsymbol{x}$, where $n\geqslant 1$, and output $f(\boldsymbol{x})$. Then $f(\boldsymbol{x})$ is linear if and only if the finite-difference expression for the second derivative of f vanishes everywhere. That is, $f(\boldsymbol{x})$ is linear if and only if

$$\frac{f(\boldsymbol{x}+\delta_i\boldsymbol{i}+\delta_j\boldsymbol{j})-f(\boldsymbol{x}+\delta_i\boldsymbol{i})-f(\boldsymbol{x}+\delta_j\boldsymbol{j})+f(\boldsymbol{x})}{\delta_i\delta_j}=0 \tag{1}$$

for all inputs $\boldsymbol{x}$, for all unit vectors $\boldsymbol{i}, \boldsymbol{j}$ in the space of inputs (a unit vector $\boldsymbol{i}$ is a set of inputs $(i_1, i_2, ..., i_n)$ whose length $\boldsymbol{i}\cdot\boldsymbol{i}\equiv i_1^2+i_2^2+...+i_n^2=1$) and for all δ_i, $\delta_j\neq 0$. Equation (1) states simply that $f(\boldsymbol{x})$ is linear if and only if the points $(\boldsymbol{x}+\delta_i\boldsymbol{i}+\delta_j\boldsymbol{j}, f(\boldsymbol{x}+\delta_i\boldsymbol{i}+\delta_j\boldsymbol{j}))$ all lie in a plane (or for $n=1$, on a line), for all $\boldsymbol{x}, \boldsymbol{i}, \boldsymbol{j}, \delta_i, \delta_j$.

If our logic gate is nonlinear, then there exist some $\boldsymbol{x}_0$, some $\boldsymbol{i}, \boldsymbol{j}$ and some δ_i, $\delta_j\neq 0$ such that the expression in eq. (1) fails to vanish when evaluated at $\boldsymbol{x}_0$. Combining our nonlinear gate with linear gates, we can realize a circuit whose inputs are x, y and whose output is $A(x, y)$, where

$$A(x, y)\equiv\frac{f(\boldsymbol{x}_0+x\delta_i\boldsymbol{i}+y\delta_j\boldsymbol{j})-f(\boldsymbol{x}_0)-[f(\boldsymbol{x}_0+\delta_i\boldsymbol{i})-f(\boldsymbol{x}_0)]x-[f(\boldsymbol{x}_0+\delta_j\boldsymbol{j})-f(\boldsymbol{x}_0)]y}{f(\boldsymbol{x}_0+\delta_i\boldsymbol{i}+\delta_j\boldsymbol{j})-f(\boldsymbol{x}_0+\delta_i\boldsymbol{i})-f(\boldsymbol{x}_0+\delta_j\boldsymbol{j})+f(\boldsymbol{x}_0)}. \tag{2}$$

A diagram for this circuit in the case $n=1$ appears in fig. 2. Note that the nonlinear gate is used only once in this circuit, to effect the operation that corresponds to the first term in the numerator of eq. (2). The remainder of the gates in the circuit are linear, either adding together inputs, supplying constant outputs, or multiplying inputs by constants whose values are determined by the parameters of the nonlinear gate.

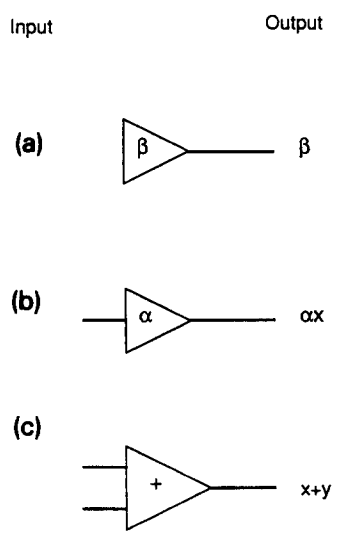

Fig. 1. Linear logic gates: (a) a gate that gives the constant output β, (b) a gate that multiplies its input by a constant α, (c) a gate that adds its inputs together.

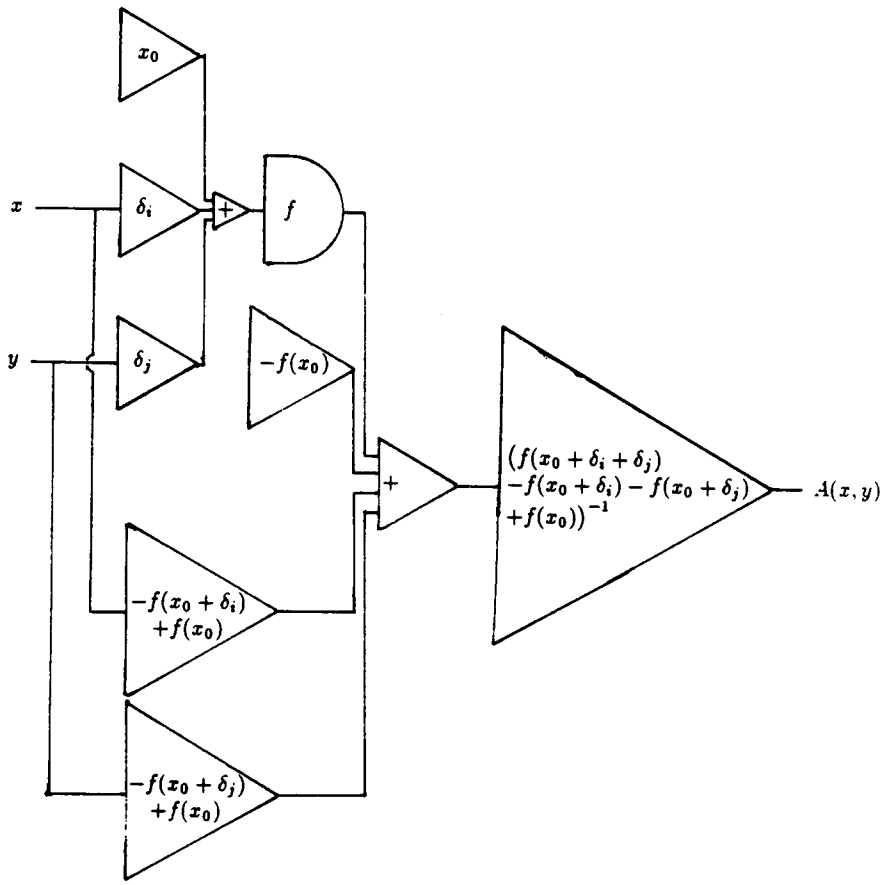

Fig. 2. A circuit that realizes an AND gate out of linear logic gates together with a single nonlinear gate.

Now, combining linear logic gates with the circuit that embodies the function $A(x)$, we can construct a gate with inputs x, y and output

$$O(x, y) = 1 - A((1-x), (1-y)) . \tag{3}$$

It is easy to see that when the inputs x, y are restricted to the values 0, 1, the gate with output $A(x, y)$ is an AND gate and the gate with output $O(x, y)$ is an OR gate. Linear gates alone suffice to construct the one-input gate with output $N(x) = 1 - x$, which for $x = 0$, 1 realizes a NOT gate.

The only restriction on $f(x)$ was that it be nonlinear. (Of course, f should also be bounded at the point of nonlinearity: it cannot be a quasi-function like the Dirac δ-function.) We can combine *any* physically realizable nonlinear gate with linear gates to construct AND and OR gates in principle, and use linear gates to construct a NOT gate. Since AND, OR and NOT, together with connecting wires and fan-outs, suffice to construct a digital computer, any nonlinearity suffices for computation.

3. Signal stability and noise

Any nonlinear device can be combined with linear devices to realize AND, OR and NOT gates: therefore

any nonlinearity suffices for computation in principle. Of course, not every nonlinear effect is a good candidate for actually constructing computers. In particular, the above discussion assumes perfect signal stability and zero noise, conditions that are unlikely to be met with in real devices [14,15]. Von Neumann [16] showed how noisy components can be assembled into a reliable computer by multiplexing – circuitry is set up to perform all operations in parallel, and to "vote" for the correct result. Signal stability, however, can be a problem for the given scheme for constructing logic gates, even in the absence of noise. The scheme works by restricting attention to two particular signal amplitudes, 0 and 1, and performing discrete binary logic using those amplitudes. If the actions of the nonlinear logic gates $A(x, y)$ and $O(x, y)$ of eqs. (2) and (3) amplify small deviations away from the values 0, 1 for x and y, then a circuit constructed of many such gates will not function.

For nonlinear devices whose output is a sufficiently well-behaved function of their input, however, the convexity properties of the function can be exploited to provide stable signals. We now turn to a method for constructing logic gates from an arbitrary nonlinear device, that automatically guarantees the stability of signals.

To give rise to computation, the set of linear operations must be supplemented by nonlinear operations. A simple step function suffices [17]. Suppose that we have a gate that takes the input x to $\sigma(x) = -1$ if $x<0$, 0 if $x=0$, and 1 if $x>0$ (fig. 3). Taken together with the linear operations above, this gate allows the construction of AND, OR and NOT gates as follows:

Let 1 correspond to True, -1 to False. Negation is then equivalent to the linear operation of multiplying by -1. Addition taken together with the step function can be combined to give the nonlinear AND and OR gates, since

$$x \text{ AND } y = \sigma(-1+x+y), \qquad x \text{ OR } y = \sigma(1+x+y)\,. \tag{4}$$

Since the gates AND, OR and NOT taken together form a basis for universal computation, the linear operations taken together with the step function σ also make up a basis for universal computation.

The step function effects computation by turning continuous, linear dynamics into binary, nonlinear dynamics. It has the advantage of aiding signal stability, since all input signals except for the single unstable point $x=0$ are transformed into only two possible output signals. Although an arbitrary nonlinear function need not resemble the step function anywhere on its domain, any operation whose output is a well-behaved (i.e., three times differentiable) nonlinear function of its input may be combined with itself and with linear operations to give a new operation that approximates the step function well enough over a finite domain to allow computation.

Consider an operation whose output is an arbitrary three times differentiable nonlinear function $f(x)$ of its input over some interval. Consider first the case in which $f'''(x_0) \neq 0$ for some x_0 in the interval. Since $f(x)$

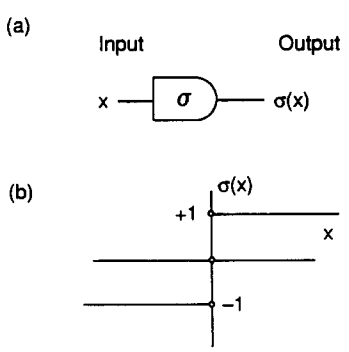

Fig. 3. A nonlinear gate (a) that realizes the step function σ (b).

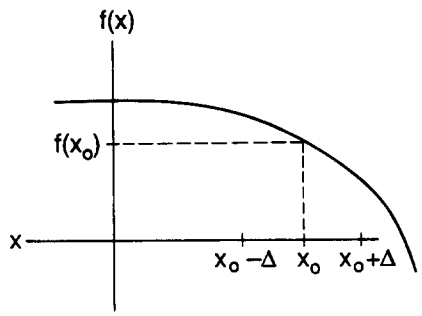

Fig. 4. A nonlinear function $f(x)$. $f'''(x)$ is negative in the neighbourhood of x_0, and $f(x)$ grows increasingly concave down in that neighbourhood.

is $C^{(4)}$, there exists a finite region $[x_0-\Delta, x_0+\Delta]$ about x_0 such that $f'''\neq 0$ and $f'''(x)$ is the same sign as $f'''(x_0)$ for all $x\in[x_0-\Delta, x_0+\Delta]$ (fig. 4). Either $f'''(x_0)$ or $-f'''(x_0)$ is negative: let us assume that $f'''(x_0)<0$.

Using linear operations, the function $g(x)=f(x_0+x)-f(x_0-x)$ can be constructed. We have $g(0)=g''(0)=0, g'(0)=2f'(x_0), g'''(0)=2f'''(x_0)$, and $g(x)$ is concave up in the region $[-\Delta, 0)$ and concave down in the region $(0, \Delta]$ (fig. 5).

From $g(x)$ we can construct, once more using linear operations, a function $h(x)$ that has the same convexity properties as $g(x)$, but for which $h(\Delta)=\Delta$, and $h'(\Delta)=0$ (fig. 6a):

$$h(x)=\{g(x)-[g(\Delta)/\Delta]x\}[g(\Delta)/\Delta-g'(\Delta)]^{-1}+x\,. \tag{5}$$

Now look at the operation obtained by applying h to itself N times (fig. 6b). The points Δ and $-\Delta$ are stable fixed points of h, and 0 is an unstable fixed point. By taking N large enough, we can make $h^{(N)}(x)=h(h(...(h(x))))$ as close to a step function as we like over a finite range. The resulting step function normalizes signals within some small, but finite range of Δ, $-\Delta$ to the values Δ, $-\Delta$. By putting N gates that effect $h(x)$ together in series we get a single gate that effects $h^{(N)}(x)$, which we can put together with linear gates to construct AND and OR gates, which together with the linear NOT and fan-out gates can be used to construct a computer whose signals are stable in the absence of noise. Von Neumann's multiplexing technique can be used in principle to make such a computer reliable in the face of noise whose average signal height is much less than Δ. In practice, threshold logic schemes of this sort are difficult to make function in a reliable manner even in the presence of moderate noise [14,15].

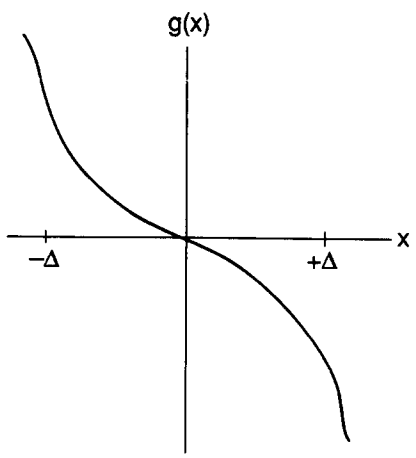

Fig. 5. Graph of $g(x)=f(x_0+x)-f(x_0-x)$.

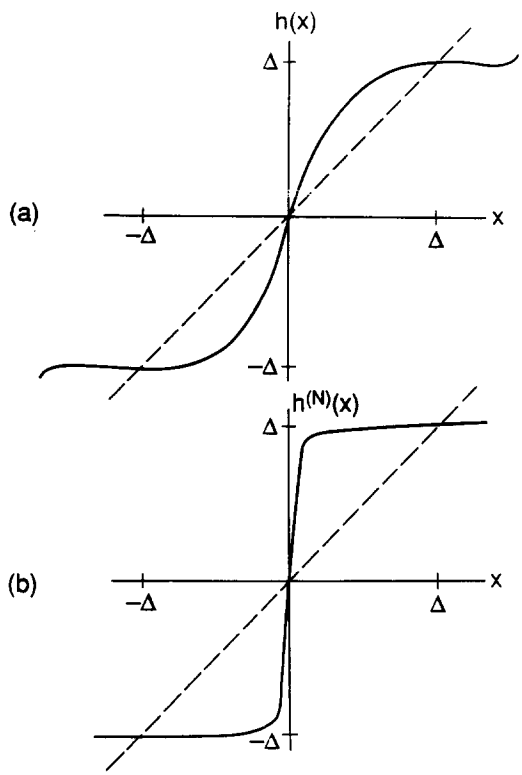

Fig. 6. (a) Graph of $h(x)=\{g(x)-[g(\Delta)/\Delta]x\}[g(\Delta)/\Delta-g'(\Delta)]^{-1}+x$. (b) Graph of h iterated N times. By taking N large, $h^{(N)}$ can be made arbitrarily close to a step function.

If the third derivative of f vanishes everywhere (i.e., $f(x)$ is a parabola), then a function $f_1(x)$ with non-vanishing third derivative can be obtained easily by iterating f: $f_1(x) = f(f(x))$, and the resulting function used to construct a step function.

4. Discussion

Together with linear devices, any device whose output is a nonlinear function of its inputs suffices as a basis for computation. While not every nonlinear effect is suitable for actually constructing a working computer, no such effect is ruled out a priori. In addition, the convexity properties of a well-behaved nonlinear function can in theory be exploited to improve signal stability.

References

[1] J. von Neumann, Non-linear capacitance or inductance switching, amplifying and memory organs, US Patent 2 815 488.
[2] E. Goto, J. Electr. Comm. Eng. Japan 38 (1955) 770.
[3] S.P. Gentile, Basic theory and application of tunnel diodes (Van Nostrand Reinhold, New York, 1962).
[4] V. Kose, ed., Superconducting quantum electronics (Springer, Berlin, 1989).
[5] J.H. Greiner et al., IBM J. Res. Dev. 24 (1980) 195.
[6] H.M. Gibbs, P. Mandel, N. Peyghambarian and S.D. Smith, eds., Optical bistability III (Springer, Berlin, 1986).
[7] H. Haug and L. Bányai, eds., Optical switching in low-dimensional systems (Plenum, New York, 1988).
[8] R. Landauer, in: Optical information processing, eds. Y.E. Nesterikhin, G.W. Stroke and W.E. Kock (Plenum, New York, 1976).
[9] Y. Yamamoto, M. Kitagawa and K. Igeta, in: Third Asia/Pacific Physics Conference, eds. C.N. Ying, Y.W. Chan, K. Young and A.F. Leung (World Scientific, Singapore, 1988).
[10] G.J. Milburn, Phys. Rev. Lett. 62 (1989) 2124.
[11] S. Datta, M.R. Melloch, S. Bandyopadyay and M.S. Lundstrom, Appl. Phys. Lett. 48 (1986) 487.
[12] S. Washburn, H. Schmid, D. Kern and R.A. Webb, Phys. Rev. Lett. 59 (1987) 1791.
[13] S. Bandyopadyay and W. Porod, Appl. Phys. Lett. 53 (1988) 2323.
[14] R.W. Keyes, Science 230 (1985) 138.
[15] R. Landauer., in: Nanostructure physics and fabrication, eds. M.A. Reed and P.K. Wiley (Academic Press, New York, 1989).
[16] J. von Neumann, Probabilistic logics and the synthesis of reliable organisms from unreliable components, lectures delivered at the California Institute of Technology (1952).
[17] S.-T. Hu, Threshold logic (Univ. of California Press, Berkeley, 1965).