References

[A] Atrubin, A.J., "A One-Dimensional Real-Time Iterative Multiplier", *IEEE Trans. on Electronic Computers*, Vol. EC-14, No. 3, June 1965, pp. 394-399.

[C] Cole, S.N., "Real-Time Computation by n-Dimensional Iterative Arrays of Finite-State Machines", *IEEE Trans. on Computers*, Vol. C-18, No. 4, 1969, pp. 349-365.

[CHEP] Commoner, F., Holt, A.W., Even, S., Pnueli, A., "Marked Directed Graphs", *J. of Computer and System Sciences*, Vol. 5, 1971, pp. 511-523.

[D] Cohen, D., "Mathematical Approach to Computational Networks", Information Sciences Inst., ISI/RR-78-73, Nov. 1978. ARPA order No. 2223.

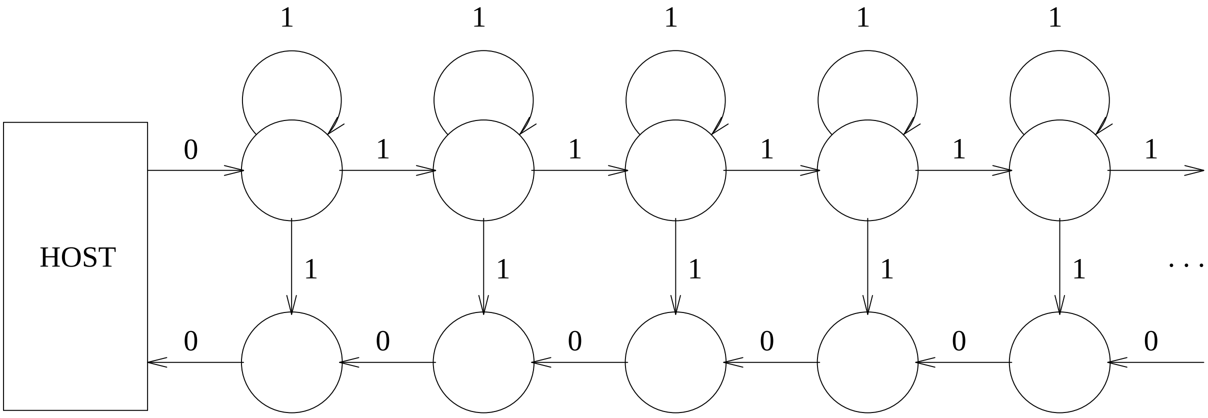
[E] Even, S., "Systolic Modular Multiplication", presented in CRYPT’90. To appear in its proceedings.

[EL] Even, S., and Litman, A., "On the Capabilities of Systolic Systems", to be presented in *3rd Annual ACM Symp. on Parallel Algorithms and Architectures*, Hilton Head, South Carolina, July 21-24, 1991.

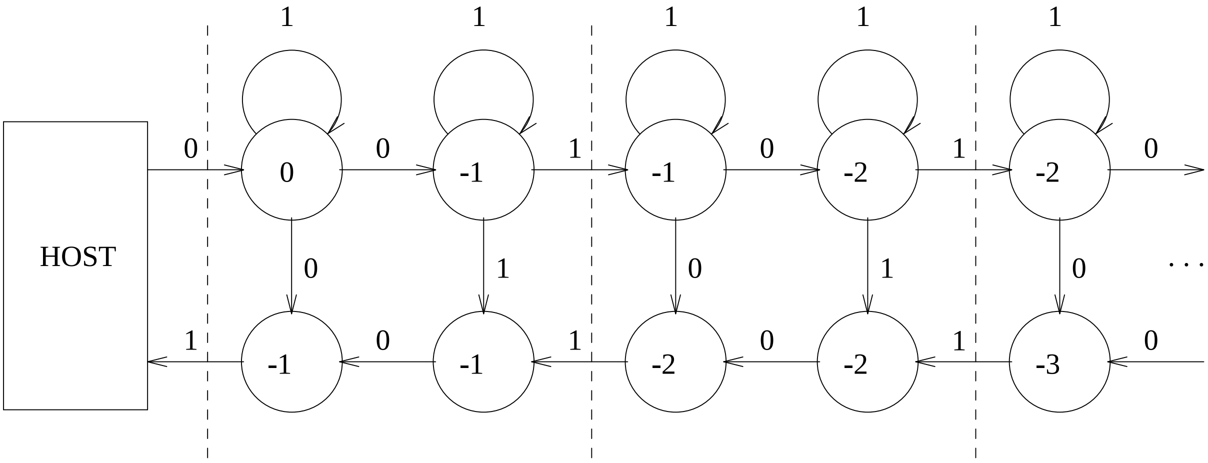
[LS] Leiserson, C.E., and Saxe, J.B., "Optimizing Synchronous Systems", *Twenty-Second Annual Symposium on Foundations of Computer Science*, IEEE, 1981, pp. 23-36. Also, *Journal of VLSI and Computer Systems*, Vol. 1, 1983, pp. 41-67.

[LRS] Leiserson, C.E., Rose, F.M., and Saxe, J.B., "Optimizing Synchronous Circuitry by Retiming", *Third Caltech Conference on Very Large Scale Integration*, ed. R. Bryant, Computer Science Press, 1983, pp. 87-116.

[S] Sieferas, J. I., "Iterative Arrays with Direct Central Control", *Acta Informatica*, Vol. 8, 1977, pp. 177-192.



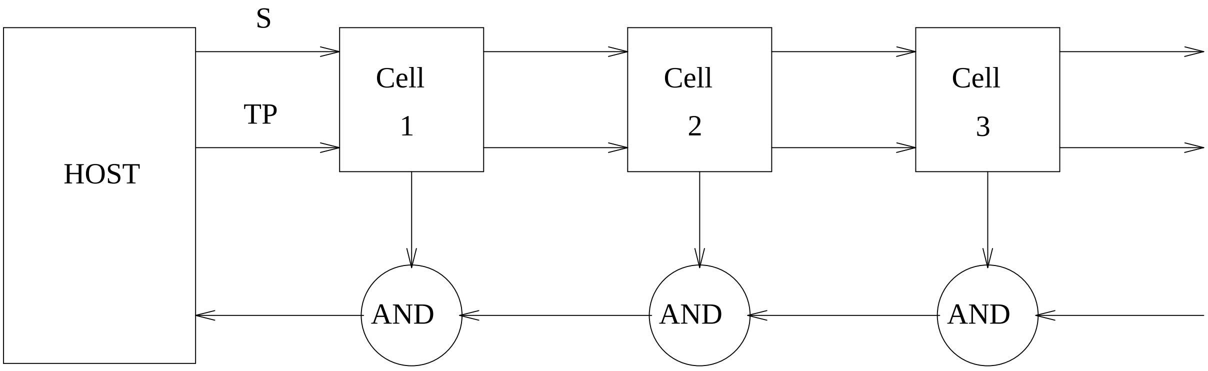
The delays, before retiming



The delays, after retiming

The techniques described above, either for removing the broadcast facility, or removing the instant-accumulation facility (but not both in the same system), are applicable to arrays of any finite dimension. Each new segment of the *k*-dimensional systolic array, after the retiming, consists of 2^k segments of the initial array. Along each axis, the retiming is identical to the 1-dimensional case, and in general, the retiming of a vertex is the sum of the retiming of vertices at its projections on the axes.

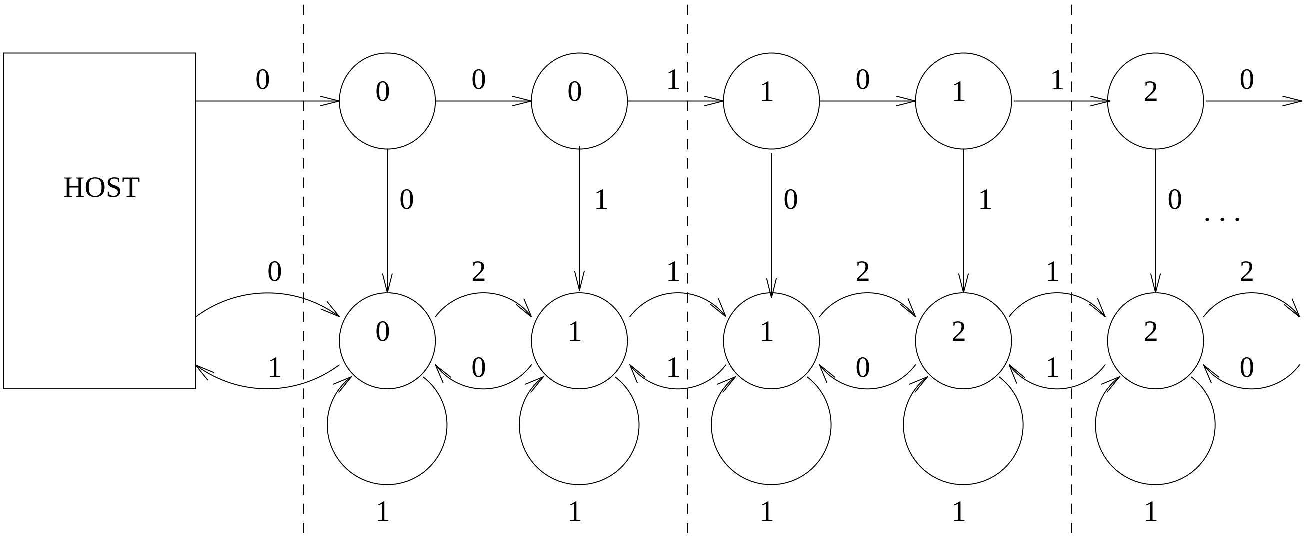
First construct a linear array with instant-accumulation, as follows. Initially, each cell's output to the instant-accumulation mechanism, is "yes"; i.e. from the point of view of the cell, the input sequence read so far by the system is a palindrome. This remains the cell's output until the cell has seen two input symbols. The input symbols are fed, one at a time to the first cell. The first symbol to reach a cell is stored, and is not fed forward. (One can use a TP signal, which is delayed two ticks in each cell, to notify the cell that its "first" symbol is arriving.) Every consecutive symbol is fed forward to the next cell, via a one-tick delay. At each tick, following the storage of the cell's first symbol, the arriving symbol is compared to the stored one. If they are equal, a "yes" is output; if they are different, a "no" is output. These outputs are supplied to the instant-accumulation mechanism, which supplies a "yes" answer to the host, if and only if all the (delayed) output answers of the cells are "yes". The general layout of the array is depicted in the following diagram:



General layout, with instant-accumulation

The next diagram shows, schematically, the number of delays on each edge, before retiming.

Now apply the following retiming: The vertices in the upper level are retimed by 0, -1, -1, -2, -2, ... , while the vertices in the lower level are retimed by -1, -1, -2, -2, -3 ... The resulting number of delays on the edges, as determined by Equation 13, are shown in the following diagram, as well as the division into new segments. Observe that the resulting system is systolic.



The array, after retiming

have achieved our goal of building a systolic array for multiplication. This design is exactly the Atrubin multiplier.

5. Comments on Similar Systems

The retiming used above is useful in other similar situations. For example, one can design a systolic queue (FIFO), or a stack (FILO), by following the same route we have taken above: First allow the broadcast facility, and design the array. Next, use the retiming of the previous section to make the whole system systolic.

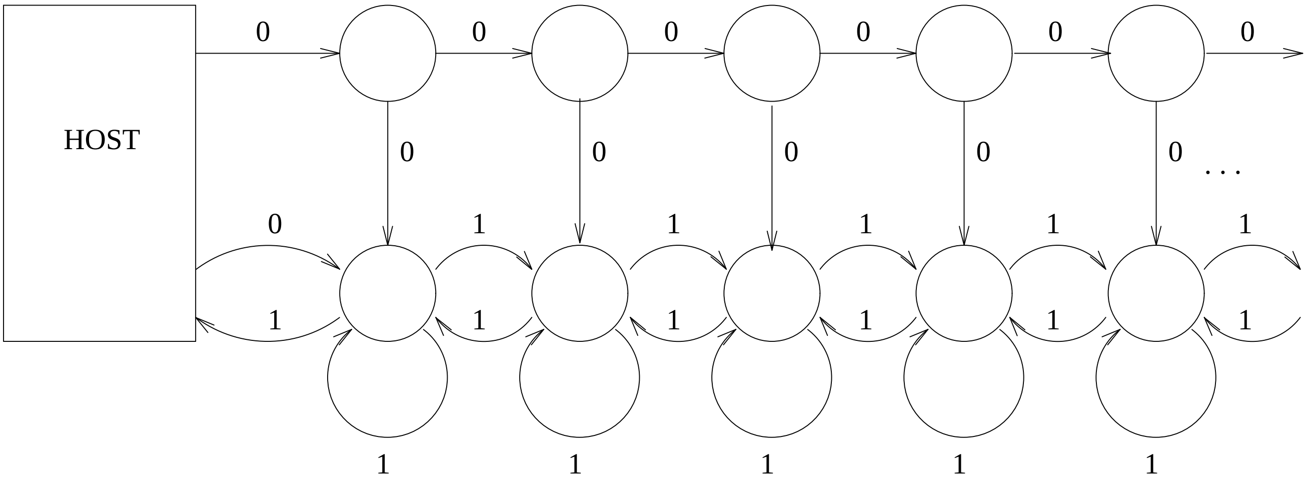
A similar technique is useful in handling a situation which, in a sense, is the opposite of broadcast.

Instant-accumulation is a mechanism which instantaneously presents to the host, via a single edge, a value which depends on values produced for this purpose by all the segments. A common example is an AND function of binary values produced by all the segments. However, the exact computation performed by the instant-accumulation mechanism is immaterial; it may be any commutative and associative multi-variable function of the individual values produced by the segments.

We assume that all the inputs to the instant-accumulation mechanism pass through delays, as they come out of the segments. This is in concert with the assumption that every edge has at least one delay, except those emanating from the host.

Let us describe, briefly, a construction of a linear systolic array to recognize palindromes*. There are several known solutions (see, for example, [C], [S] and [LS]). Our construction differs mainly in the methodology of the construction.

A *palindrome* is a word which is identical to its reversal.



One dimensional array, with broadcast

instantaneous. Thus, the upper level represents the broadcast capability. The logic boxes represented by the vertices in the lower level are all identical. All output lines emanating from these boxes have exactly one delay flip-flop. Thus, the lower level of the system is completely systolic. The self-loops of the vertices in the lower level represent the memory of the individual cells.

Our purpose is to shift the delays (as tokens in a marked graph) in such a way that the array will divide into cells. All cells are to be identical. Each output line of a cell will have at least one delay on it. This transformation will yield a systolic array, although each cell may "contain" more than one stage of the original array. This shift of delays is best described by *retiming*: An assignment of nonnegative integers to the vertices (circles). The meaning of a retiming value $\lambda(v)$ to a vertex v is the number of ticks of the clock by which the operation of the corresponding logic is delayed. (In the marked graph model, this is the number of times vertex v is to be "fired" backwards.)

In the following diagram, the value of $\lambda(v)$ is shown in the circle representing v, and the corresponding delays of the edges are shown next to them. In the upper level the sequence of $\{\lambda(v)\}$ is 0,0,1,1,2,2,... and in the lower it is the same sequence with one zero omitted. The new number of delays, $d'(e)$, on an edge $u \overset{e}{\rightarrow} v$, is determined by the formula

$$d'(e) = d(e) + \lambda(v) - \lambda(u), \tag{13}$$

where $d(e)$ is the previous number of delays on the edge e.

Observe that all the "new" cells, as between the dashed horizontal lines, are identical, and each of the edges emanating from them has a (one tick) delay on it. Thus, the new array is systolic. It is also easy to see that, as far as the host is concerned, nothing has changed; the host will observe the same input-output behavior. (The proof of this claim, for general systolic systems with broadcast, is fairly involved, and can be found in Section 6 of [EL], but is not required in the relatively simple situation at hand.) Thus, we respectively.

Input number 4, by the inductive hypothesis, is equal to 0 for $0 \le t \le i+1$. Beyond some limit, all bits are 0. Thus, the (finite) number fed to the adder, from $t = i$, and on, is

$$\sum_{t=i+2}^{\infty} S_{i+1}(t) \cdot 2^{t-i}.$$

By Lemma 1, there exists an integer N_i, such that for $t > N_i$, $S_i(t) \equiv 0$, and

$$\sum_{t=i+1}^{\infty} S_i(t) \cdot 2^{t-i-1} = A(i) \cdot B(i) + \sum_{t=i+1}^{n-1} A(i) \cdot B(t) \cdot 2^{t-i} + \sum_{t=i+1}^{n-1} B(i) \cdot A(t) \cdot 2^{t-i} + \sum_{t=i+2}^{\infty} S_{i+1}(t) \cdot 2^{t-i}. \tag{11}$$

It remains to show that equation (11) implies equation (8). By the inductive hypothesis,

$$\begin{aligned} \sum_{t=i+1}^{\infty} S_i(t) \cdot 2^{t-i-1} &= A(i) \cdot B(i) + \sum_{t=i+1}^{n-1} A(i) \cdot B(t) \cdot 2^{t-i} + \sum_{t=i+1}^{n-1} B(i) \cdot A(t) \cdot 2^{t-i} + \\ &\quad 2^2 \cdot [\sum_{t=i+1}^{n-1} A(t) \cdot 2^{t-i-1}] \cdot [\sum_{t=i+1}^{n-1} B(t) \cdot 2^{t-i-1}] \\ &= A(i) \cdot B(i) + A(i) \cdot \sum_{t=i+1}^{n-1} B(t) \cdot 2^{t-i} + B(i) \cdot \sum_{t=i+1}^{n-1} A(t) \cdot 2^{t-i} + \\ &\quad [\sum_{t=i+1}^{n-1} A(t) \cdot 2^{t-i}] \cdot [\sum_{t=i+1}^{n-1} B(t) \cdot 2^{t-i}] \\ &= [\sum_{t=i}^{n-1} A(t) \cdot 2^{t-i}] \cdot [\sum_{t=i}^{n-1} B(t) \cdot 2^{t-i}]. \end{aligned}$$

■

4. From Nonsystolic to Systolic

In this section, we shall use a retiming technique. The idea is not new; it can be found in a primitive form in papers from the seventies (see, for example, [CHEP] and [D]). A more explicit description of the technique, and its consequences, were presented in the work of Leiserson et. al. (see, for example, [LS] and [LRS]). We shall follow our own approach, as in [EL].

Consider a general one dimensional array, with broadcast, as depicted in the following diagram.

The vertices (circles) represent memoryless combinational logic boxes. The edges (lines) represent information channels. The integer $d(e)$, next to an edge e, is the number of delay flip-flops (registers) on the corresponding line.

Observe that the nonsystolic multiplication array, of the previous section, fits into the general scheme described by this diagram. Each of the vertices in the upper level represents nothing but a simple junction, which transmits to both outputs (to the right, and down) the information received from its input (from the left); this transmission is

We extend this claim, to include the statement that for all $t \geq 0$, $C_{n-1}^{1}(t) \equiv C_{n-1}^{2}(t) \equiv 0$, and prove it by induction on t, from $t = 0$ and up.

By (6),

$$S_{n-1}(0),\ \tilde{A}_{n-1}(0),\ \tilde{B}_{n-1}(0),\ C_{n-1}^{1}(0),\ C_{n-1}^{2}(0) \equiv 0.$$

For $0 \leq t < n-1$, it is easy to see that inputs 1, 2 and 3 are all 0. Also, by the first claim, $S_n(t) \equiv 0$. Thus,

$$S_{n-1}(t+1) \equiv C_{n-1}^{1}(t+1) \equiv C_{n-1}^{2}(t+1) \equiv 0.$$

For $t = n-1$, the sampling is being performed, but $\tilde{A}_{n-1}(t)$ and $\tilde{B}_{n-1}(t)$ are still 0. Thus, inputs 2 and 3 are still 0. Also, input 4 is always 0. The only input that counts is $A(t) \angle B(t) \angle (t=i)$, which is equal to $A(n-1) \cdot B(n-1)$. Thus,

$$C_{n-1}^{1}(n) \equiv C_{n-1}^{2}(n) \equiv 0,$$

and

$$S_{n-1}(n) = A(n-1) \cdot B(n-1).$$

For $t \geq n$, all 4 inputs are 0. Thus,

$$S_{n-1}(t+1) \equiv C_{n-1}^{1}(t+1) \equiv C_{n-1}^{2}(t+1) \equiv 0.$$

This completes the proof of the basis.

We turn to the proof of the inductive step. The inductive hypothesis is:

(a) There exists an integer N_{i+1}, such that for all $t > N_{i+1}$, $S_{i+1}(t) \equiv 0$.

(b)

$$\sum_{t=i+2}^{\infty} S_{i+1}(t) \cdot 2^{t-i-2} = [\sum_{t=i+1}^{n-1} A(t) \cdot 2^{t-i-1}] \cdot [\sum_{t=i+1}^{n-1} B(t) \cdot 2^{t-i-1}], \tag{10}$$

and for $0 \leq t \leq i+1$, $S_{i+1}(t) \equiv 0$.

Let us consider the situation in the i-th cell. We first examine the time $0 \leq t \leq i$, and claim that

$$S_i(t) \equiv C_i^{1}(t) \equiv C_i^{2}(t) \equiv 0,$$

by induction on t. The basis follows from (6). For any $t < i$, inputs number 1, 2 and 3 are all 0. By the inductive hypothesis, so is input number 4. Thus, the claim follows.

Now consider $t \geq i$.

At $t = i$, input number 1 is equal to $A(i) \cdot B(i)$. For $t > i$, input number 1 is equal to 0. Thus, the number fed sequentially to the adder, by input number 1, starting at time $t = i$, in binary, least significant bit first, is simply $A(i) \cdot B(i)$.

At $t = i$, $A(i)$ and $B(i)$ are sampled, and become the values of $\tilde{A}_i$ and $\tilde{B}_i$, respectively. However, inputs number 2 and 3 are still equal to 0. For $t > i$, they are $A(i) \cdot B(t)$ and $B(i) \cdot A(t)$, respectively. Thus, the numbers they feed, sequentially, starting at $t = i$, are

$$\sum_{t=i+1}^{n-1} A(i) \cdot B(t) \cdot 2^{t-i} \quad \text{and} \sum_{t=i+1}^{n-1} B(i) \cdot A(t) \cdot 2^{t-i},$$

And a similar statement holds for $\tilde{B}_i(t)$.

The lower half of the diagram depicts a serial adder, just as in the previous section. It has 4 input lines and only 2 carries, but this is acceptable for the following reason.

Input number 1 has the value $A(t) \angle B(t) \angle (t=i)$. This value is 0 if $t \neq i$. Input number 2 has the value $\tilde{A}_i(t) \angle (t \neq i) \angle B(t)$. Thus input number 2 (and 3) are never equal to 1, if input number 1 has the value 1, and effectively, the number of input lines is bounded by 3. Therefore, 2 carries suffice.

Our aim is to prove the following equation:

$$\sum_{t=1}^{2n} S_0(t) \cdot 2^{t-1} = [\sum_{t=0}^{n-1} A(t) \cdot 2^t] \cdot [\sum_{t=0}^{n-1} B(t) \cdot 2^t]. \tag{7}$$

Instead, we will first prove the following lemma. Equation (7) will follow from Equation (8), by substituting $i = 0$ and observing that the equality implies that all bits after the $2n$'th, must be 0.

Lemma 2:

For every $n-1 \geq i \geq 0$, the sequence $\{S_i(t)\}$ satisfies the following conditions:

(a) There exists an integer N_i, such that for all $t > N_i$, $S_i(t) \equiv 0$.

(b)

$$\sum_{t=i+1}^{\infty} S_i(t) \cdot 2^{t-i-1} = [\sum_{t=i}^{n-1} A(t) \cdot 2^{t-i}] \cdot [\sum_{t=i}^{n-1} B(t) \cdot 2^{t-i}], \tag{8}$$

and for $0 \leq t \leq i$, $S_i(t) \equiv 0$.

Proof:

First, let us show, by induction on t, that for every $i \geq n$ and for every $t \geq 0$,

$$C_i^1(t) \equiv C_i^2(t) \equiv S_i(t) \equiv 0. \tag{9}$$

The basis, i.e. when $t = 0$, follows from (6). Now assume the statement holds for t, and let us show that it holds for $t+1$.

Input number 1 is $A(t) \angle B(t) \angle (t=i)$. If $t < n$, since $i \geq n$, we have $(t=i) \equiv 0$. If $t \geq n$ then $A(t) \equiv B(t) \equiv 0$. Thus, input number 1 is always 0.

Since $\tilde{A}_i$ and $\tilde{B}_i$ are sampled at $t = i$, and $i \geq n$, it follows that their values are always 0. Thus, input numbers 2 and 3 are always 0. By the inductive hypothesis, $C_i^1(t) \equiv C_i^2(t) \equiv 0$, and also, $S_{i+1}(t) \equiv 0$. Thus, by (1),

$$C_i^1(t+1) \equiv C_i^2(t+1) \equiv S_i(t+1) \equiv 0.$$

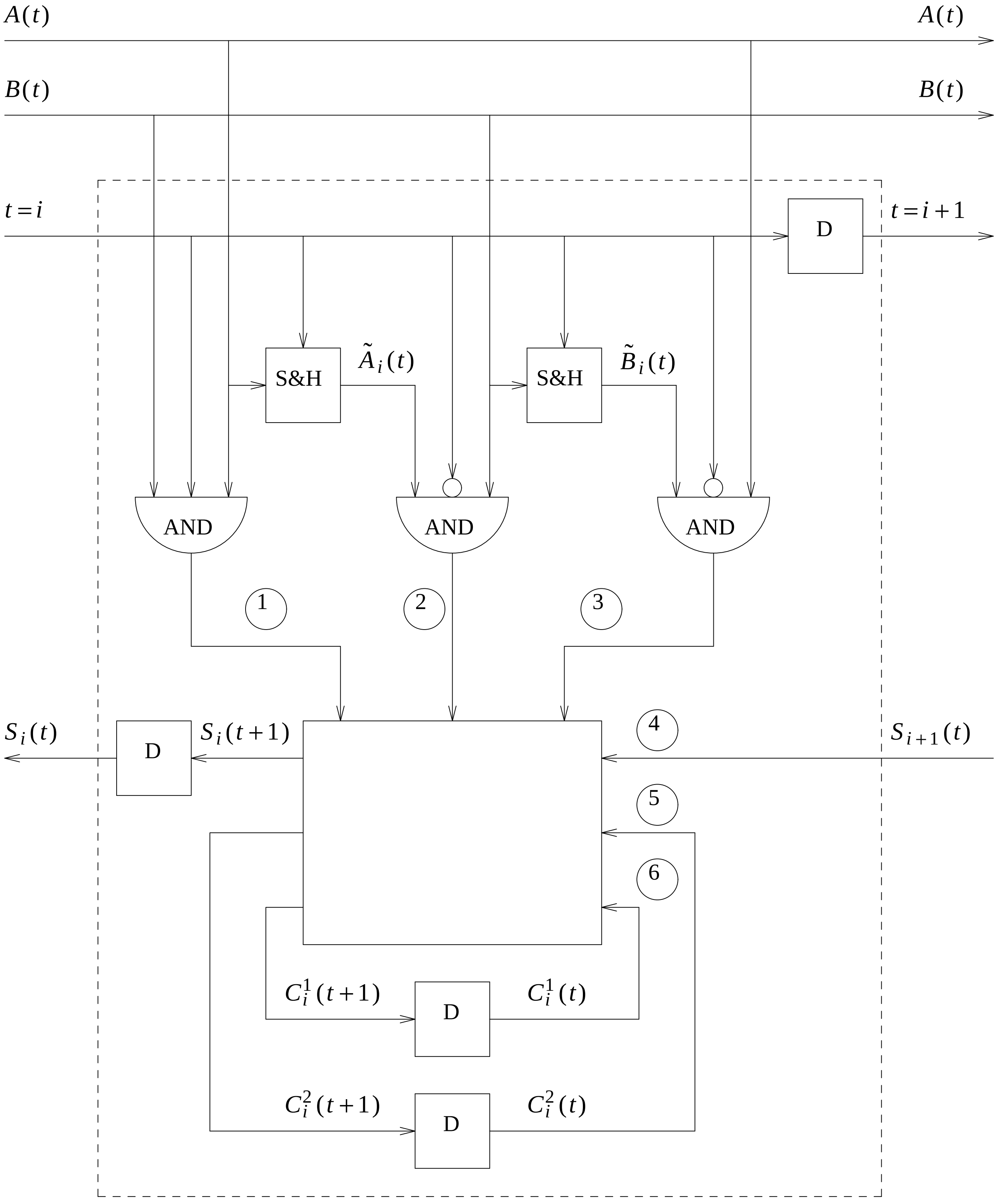
This completes the proof of the first claim.

We now prove the lemma by induction on i, from n-1 down to 0. For the basis it suffices to show that:

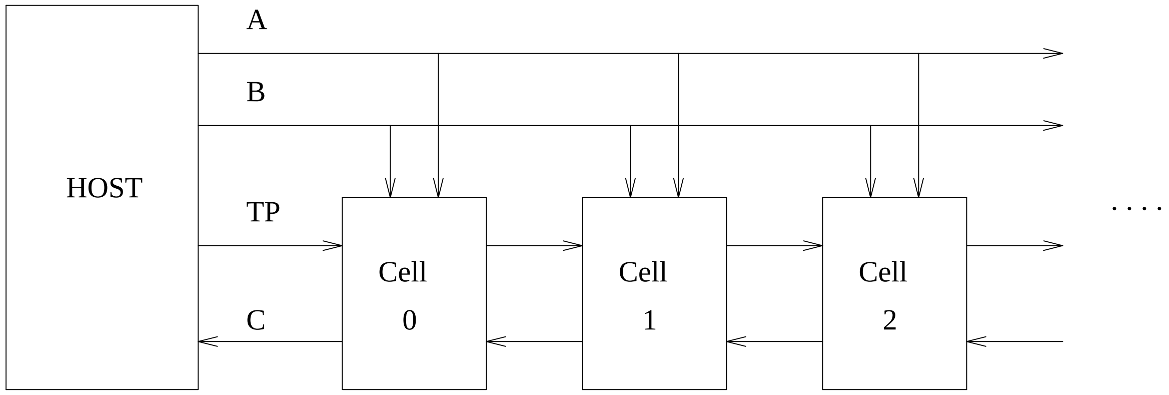
(1) For $0 \leq t < n$, $S_{n-1}(t) \equiv 0$,

(2) for $t = n$, $S_{n-1}(t) = A(n-1) \cdot B(n-1)$,

(3) for $t > n$, $S_{n-1}(t) \equiv 0$.



i-th cell of nonsystolic array



Nonsystolic multiplication array

The host has three binary output channels and one binary input channel. On channel A and B, the two nonnegative n-bit multiplicands are fed to the system, sequentially, least significant bits first, one bit on every tick of the (implied global) clock. The first bits are fed at time $t = 0$, and the last ones at time $t = n - 1$. The following bits are all zeros. On channel TP, a timing pulse is fed to the system at time $t = 0$; i.e. the logical value is 1 at $t = 0$, and 0 at $t \neq 0$. This timing pulse is delayed one unit of time in each of the cells of the system. Thus, the i-th cell gets it at time $t = i$. Cell number 0 of the system, feeds the product of the two multiplicands to the host, via channel C. The $2n$ bits of the product are delivered sequentially, least significant bit first, starting from time $t = 1$ and ending at time $t = 2n$.

The structure of all the cells is identical, and will be described shortly. The cell is a Moore finite automaton; i.e. its outputs go through clocked delay flip-flops. Thus, the outputs of each cell, at time t, depend only on its state, and not on its inputs at time t. The only nonsystolic part in this system is the broadcast channels, A and B. In the following section, it will be shown how the broadcast channels can be removed, to make the system completely systolic.

In addition to the global clock, we assume that there is a reset signal which is fed to all flip flops before $t = 0$. These facilities are not shown in our diagrams. The "hard" reset signal can be replaced by a "soft" one; i.e. one which travels through the system in a systolic manner. More can be found about this feature in [EL].

The structure of a typical cell is shown in the following diagram.

The reset signal assures that

$$S_i(0),\ \tilde{A}_i(0),\ \tilde{B}_i(0),\ C_i^1(0),\ C_i^2(0) \equiv 0. \qquad (6)$$

The *sample and hold* (S&H) are clocked flip-flops which sample the l.h.s. input when the top input is equal to 1. Thus,

$$\tilde{A}_i(t) = \begin{cases} 0 & \text{if } t \le i \\ A(i) & \text{if } t > i \end{cases} .$$

$$S(i+1)\cdot 2^i \ + \ 2^{i+1}\cdot \sum_{d=1}^{l} C^d(i+1) \ = \ \sum_{j=1}^{k} \alpha_j(i)\cdot 2^i \ + \ 2^i\cdot \sum_{d=1}^{l} C^d(i).$$

Adding the latter to the inductive hypothesis, completes the proof of the claim.

Thus, we have

$$\sum_{a=1}^{n} S(a)\cdot 2^{a-1} \ + \ 2^n\cdot \sum_{d=1}^{l} C^d(n) \ = \ \sum_{j=1}^{k} \sum_{b=0}^{n-1} \alpha_j(b)\cdot 2^b. \tag{4}$$

For $t \geq n$, (1) degenerates into

$$S(t+1) \ + \ 2\cdot \sum_{d=1}^{l} C^d(t+1) \ = \ \sum_{d=1}^{l} C^d(t). \tag{5}$$

By induction on t, from n up, it follows (in a manner similar to the proof of the previous claim) that:

$$\sum_{a=1}^{t+1} S(a)\cdot 2^{a-1} \ + \ 2^{t+1}\cdot \sum_{d=1}^{l} C^d(t+1) \ = \ \sum_{j=1}^{k} \sum_{b=0}^{n-1} \alpha_j(b)\cdot 2^b.$$

And all that remains to be shown is that for $t \ = \ n+m-1$, the second term on the l.h.s. is zero. For $t \geq n$, (5) implies that

$$\sum_{d=1}^{l} C^d(t+1) \ \leq \ \frac{1}{2}\cdot \sum_{d=1}^{l} C^d(t).$$

Therefore,

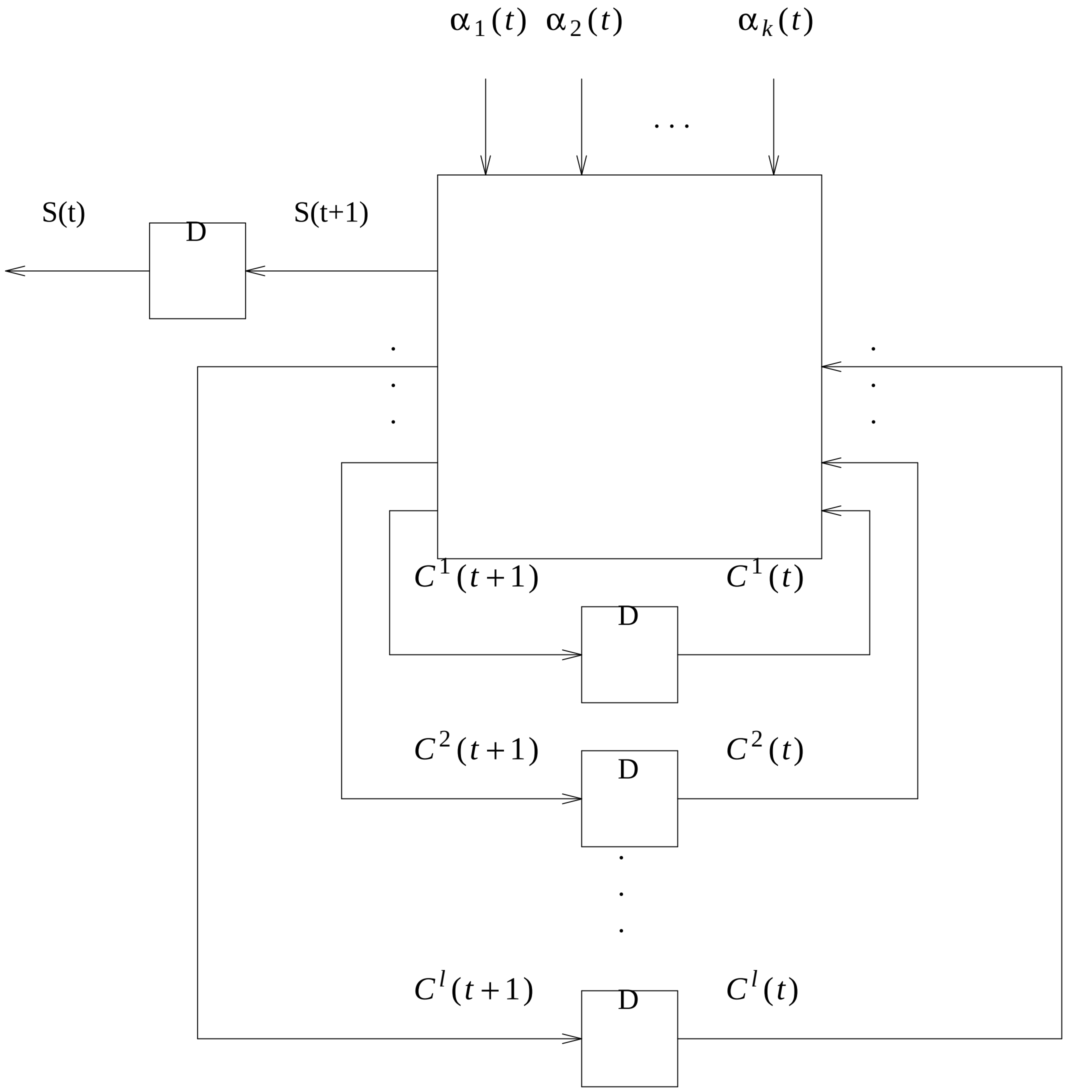
$$\sum_{d=1}^{l} C^d(t+1) \ \leq \ \left\lfloor \ \frac{1}{2^{t+1-n}}\cdot \sum_{d=1}^{l} C^d(n) \ \right\rfloor.$$

Note that $\sum_{d=1}^{l} C^d(n) \leq l$. Thus, if $l < 2^{t+1-n}$ then the r.h.s. is zero. It suffices that $\lceil \log_2 l \rceil < t+1-n$, or $\lceil \log_2 l \rceil \leq t-n$. Therefore, (3) follows, as well as the fact that for $t > n+m$, $S(t) = 0$.

■

3. A Nonsystolic Version of the Multiplier

Let us describe now a nonsystolic version of the multiplier. Its overall structure is depicted in the following diagram.



Serial adder

Proof:

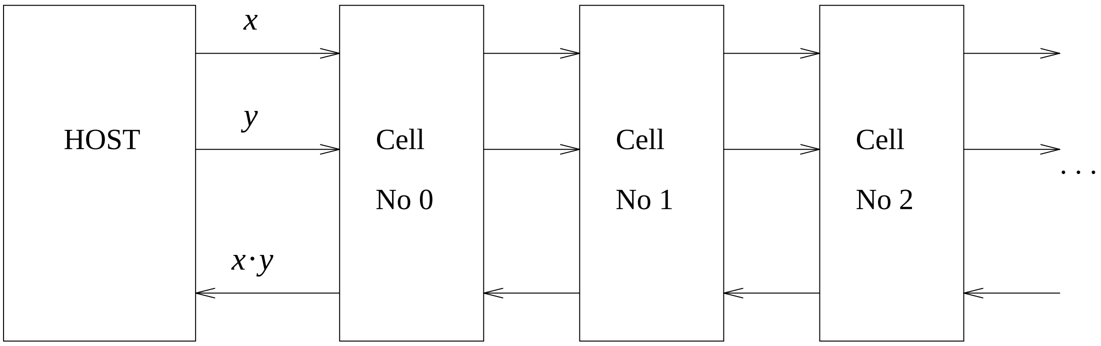
First, by induction on i, from 0 to $n-1$, the following holds:

$$\sum_{a=1}^{i+1} S(a)\cdot 2^{a-1} \;+\; 2^{i+1}\cdot \sum_{d=1}^{l} C^d(i+1) \;=\; \sum_{j=1}^{k} \sum_{b=0}^{i} \alpha_j(b)\cdot 2^b .$$

The basis follows from (1) and (2). The inductive hypothesis is

$$\sum_{a=1}^{i} S(a)\cdot 2^{a-1} \;+\; 2^{i}\cdot \sum_{d=1}^{l} C^d(i) \;=\; \sum_{j=1}^{k} \sum_{b=0}^{i-1} \alpha_j(b)\cdot 2^b .$$

Taking equation (1), for $t\;=\;i$, and multiplying all terms by 2^i, yields:



The Atrubin Multiplication Array

In spite of its reputation, the structure of the Atrubin array has remained a mystery. It is the purpose of this paper to explain this mystery away, and prove the validity of the multiplier. This is done by breaking the design into stages.

First, the operation and validity of a multi-input serial adder is discussed. Next, a simplified version of the multiplier is studied, in which a broadcast capability is used. Finally, by using retiming, the systolic version is achieved.

2. Serial Adder

One of the components used in the design of the multiplier is a serial adder. It is a simple extension of a two-inputs serial full adder. Its structure is depicted in the diagram.

The main logic box computes the outputs $(S(t+1),\ C^1(t+1),\ C^2(t+1),\ldots,\ C^l(t+1))$ from the inputs $(\alpha_1(t),\ \alpha_2(t),\ldots,\ \alpha_k(t))$ to satisfy

$$S(t+1) + 2 \cdot [C^1(t+1) + \cdots + C^l(t+1)] = \alpha_1(t) + \cdots + \alpha_k(t) + C^1(t) + \cdots + C^l(t). \quad (1)$$

Note that the carries are represented in unary. A necessary condition is that $l \geq k-1$. The D boxes are clocked delay flip-flops (registers).

The initial conditions (which can be implemented via a global reset signal for the flip-flops) are:

$$S(0) \equiv C^1(0) \equiv C^2(0) \equiv \cdots \equiv C^l(0) \equiv 0. \quad (2)$$

Lemma 1:

Let the α inputs feed n-bit positive integers, in binary, least significant bits first, and let $m = \lceil \log_2 l \rceil$. The following equation holds:

$$\sum_{a=1}^{n+m} S(a) \cdot 2^{a-1} = \sum_{j=1}^{k} \sum_{b=0}^{n-1} \alpha_j(b) \cdot 2^b, \quad (3)$$

and for $t > n+m$, $S(t) = 0$.

A Systematic Design and Explanation of the Atrubin Multiplier

Shimon Even and Ami Litman

Computer Science Dept., Technion, Haifa, Israel 32000
and
Bellcore, 445 South St., Morristown, NJ 07960-1910

1. Introduction

In 1962, Allan J. Atrubin invented a synchronous system for real-time multiplication of integers. (It was published in 1965, [A].) The host (user) feeds the system two binary encoded multiplicands, x and y, serially, least significant bits first, and the system outputs the product $x \cdot y$, in binary, serially, least significant bit first. Clearly, the time it takes to multiply two n-bit multiplicands is $2 \cdot n$.

Informally, a (finite, or infinite) synchronous system, serving a host, is called *systolic*, if it has the following characteristics. The system consists of *segments*, connected to each other and to the host by communication *lines*. Each segment consists of a modest amount of hardware, which realizes a Moore finite state automaton; i.e. its current output signals, which appear on its output ports, depend only on its present state, and its next state depends on the present state and the present input signals, which appear presently at the input ports. Without loss of generality, we may assume that each output port of a segment is the output port of a (clocked) delay flip-flop. The lines go from one output port to one input port; there is no fan-in or fan-out in these connections.

The Atrubin system is systolic; namely it has no long paths where a rippling effect can happen between clock ticks. Therefore, it allows high clock rates. Furthermore, its segments are all identical, and they are arranged in a linear array. This simplifies the design and production of the system.

It is known how to perform multiplication in $O(log\ n)$ time, but the equipment required is of size $O(n^2)$. Thus, the Atrubin multiplier, which requires equipment of size $O(n)$, remains competitive.

The general layout of the multiplier is depicted in the following diagram. In order to multiply n-bit numbers, the array must consist of at least $\lceil \frac{n}{2} \rceil$ cells. All cells have the same structure. Each consists of a few hundreds transistors, and realizes a finite Moore automaton.

Recently, Even showed how, with the addition of another systolic array, repeated modular multiplication can be computed almost as fast ([E]). The combination of the two arrays may be useful in cryptographic applications.