1985

On the Existence of Analytical Proofs for VLSI Computational Networks

Rami Melhem

Report Number:

85-509

On the Existence of Analytical Proofs for VLSI Computational Networks

Rami Melhem

Technical Report CSD_TR_509 (March 1985)

Department of Computer Science
Purdue University
West Lafayette, IN 47907

ABSTRACT

In a previous work, it was shown that the verification of a systolic or a dead-lock free self timed network may be achieved by solving a system of sequence equations which models the network. Here, we prove that such a system has always a solution which may be expressed in the sequence notation. In particular, the iteration operator is introduced to compensate for the absence of the time dimension from sequence equations, thus providing a suitable notation for the solution of recursive equations resulting from feed back loops.

Key words and phrases:

Formal verification - Correctness proofs - Self timed Networks - Systolic arrays - Feed back loops - recursive equations.

1. Introduction

Many models have been suggested recently for proving the correctness of systolic and self timed computational networks. These models may be classified into two classes. In the first class, a network is verified by showing that it may be obtained by the application of a 'correctness preserving' transformation to another simpler network (e.g. [3]), to an algorithmic description (e.g. [4,6,11]), or to a program on some abstract sequential model of computation [2]. This approach assumes the existence of such transformations and the availability of correctness proofs in the domains of the transformations.

In the second class, a model is designed specifically for the specification and verification of computational networks [1,8]. In theory, such models may be applied to any systolic or self timed network provided that the operation of each cell in the network may be described by a deterministic function. In practice, however, the verification procedure is feasible only if the interconnection between cells is regular and the operation of each cell is relatively simple.

In this letter, we consider the model presented in [8] and extended in [7] to include systolic networks with memory and multiplexing capabilities. This model is also applied in [9] to the verification of a class of self timed systems. Its basic idea is to represent the consecutive data items that appear on any communication link of a network by an infinite sequence, and to model the computation performed by each cell by operators on sequences. The operation of the network is thus modeled by the system of all sequence equations describing the various cells. The solution of this system is the network I/O description which expresses the output sequences in terms of the input sequences of the network. This allows a computation of the outputs for any given input either analytically or by a computer solver [10].

Let $R_\delta = R \cup \{\delta\}$, where R is the set of data items that may be transmitted on a communication link of the network, and δ is a special symbol called the don't care symbol. Let also $\vec{R}_\delta$ be the set of all sequences defined on R_δ. A sequence operator $\Gamma: [\vec{R}_\delta]^n \to \vec{R}_\delta$ is called a causal operator if the t^{th} element of its image sequence $\eta(t) = [\Gamma(\xi_1, \cdots, \xi_n)](t)$ does not depend on any element $\xi_i(\tau)$, $i=1, \cdots, n$ for $\tau \geq t$. If $\tau \geq t$ is replaced by $\tau > t$, then the operator is called weakly causal. For example, the zero shift operator defined by

$$[\Omega_0^r \eta](t) = \begin{cases} 0 & \text{if } t \leq r \\ \eta(t-r) & \text{if } t > r \end{cases}$$

is causal, and the element-wise operators '*op*' = +, -, * and /, defined by $[\xi \text{ 'op' } \eta](t) = \xi(t) \text{ 'op' } \eta(t)$, are weakly causal. It is argued in [8] and [9] that cells in systolic networks are modeled by causal operators and cells in self timed networks are modeled by either causal or weakly causal operators, depending on the initial status of their output links.

From the above discussion, it is clear that the existence of an analytical proof for a systolic or a self timed network depends on: 1) The ability to model each cell in the network by sequence operators. This is always possible if each cell performs a deterministic function. 2) The existence of analytical solutions to systems of sequence equations. This is the subject of the following sections.

2. Analytical Solutions to Systems of Sequence Equations.

In this section, we use the term sequence equation in a restrictive manner to indicate an equation in which the left side is a sequence and the right side is a sequence expression. This is the only type of equations needed for modeling the operation of computational networks.

In order to discuss systems of equations without referring to the underlying networks, we let Q denote the set of all sequences that appear in a given system of equations, and we partition Q into three disjoint sets, namely, Q_p; the set of input sequences, Q_o; the set of output sequences, and $Q_r = Q - \{Q_p \cup Q_o\}$. Here, an input sequence (output sequence) is a sequence that does not appear on the left side (right side) of any equation in the system. Accordingly, a solution to the given system of sequence equations is defined as a set of formulas, involving only well defined sequence operators, that explicitly describe the sequences in Q_o in terms of those in Q_p. Here, a well defined operator is understood to mean any operator whose image can be obtained from its operands using a deterministic algorithm.

Let q_p, q_o and q_r be the cardinalities of the sets Q_p, Q_o and Q_r, respectively. We enumerate the sequences in Q by integers $j=1, \cdots, q_p+q_o+q_r$, such that for any sequence $\xi_j \in Q$,

$$\begin{array}{ll} \xi_j \in Q_o & \text{if } j \le q_o \\ \xi_j \in Q_r & \text{if } q_o < j \le q_o+q_r \\ \xi_j \in Q_p & \text{if } q_o+q_r < j \le q_o+q_r+q_p \end{array}$$

The structure of the system of equations can then be described in terms of a dependency matrix A which is a square matrix of order $q_o+q_r+q_p$ defined by $a_{i,j}$ = 1, if ξ_j appears on the right side of the equation describing ξ_i, and $a_{i,j}$ = 0, otherwise. For example, consider the following two systems of sequence equations:

System S :

$$\begin{aligned} \xi_1 &= \Gamma_1(\xi_3, \xi_6) \\ \xi_2 &= \Gamma_2(\xi_4, \xi_5) \\ \xi_3 &= \Gamma_3(\xi_4, \xi_6) \\ \xi_4 &= \Gamma_4(\xi_5, \xi_7) \\ \xi_5 &= \Gamma_5(\xi_6, \xi_7) \end{aligned}$$

System $\bar{S}$: This is the same as system S except that the last equation is replaced by

$$\xi_5 = \Gamma_6(\xi_3, \xi_6, \xi_7)$$

Here Γ_i, $i=1, \cdots, 6$ are well defined sequence operators. In both systems, we have $Q_p = \{\xi_6, \xi_7\}$, $Q_o = \{\xi_1, \xi_2\}$ and $Q_r = \{\xi_3, \xi_4, \xi_5\}$ and the dependency matrices are

$$A = \begin{bmatrix} 0&0&1&0&0&1&0 \\ 0&0&0&1&1&0&0 \\ 0&0&0&1&0&1&0 \\ 0&0&0&0&1&0&1 \\ 0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \end{bmatrix} \qquad \bar{A} = \begin{bmatrix} 0&0&1&0&0&1&0 \\ 0&0&0&1&1&0&0 \\ 0&0&0&1&0&1&0 \\ 0&0&0&0&1&0&1 \\ 0&0&1&0&0&1&1 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \end{bmatrix}$$

for S and $\bar{S}$, respectively.

Clearly, any dependency matrix A can be partitioned into the following form:

$$A = \begin{bmatrix} O & A_{o,r} & A_{o,p} \\ O & A_{r,r} & A_{r,p} \\ O & O & O \end{bmatrix}$$

where the dimensions of the sub-matrices $A_{o,r}$, $A_{o,p}$, $A_{r,r}$ and $A_{r,p}$ are $q_o \times q_r$, $q_o \times q_p$, $q_r \times q_r$ and $q_r \times q_p$, respectively, and each O denotes a zero sub-matrix of the appropriate dimension. If $A_{r,r}$ is a strictly lower or strictly upper triangular, then by back substitution, we can express the sequences in Q_o in terms of those in Q_p. For example, for the system of equations S we obtain

$$\begin{aligned} \xi_4 &= \Gamma_4(\ \Gamma_5(\xi_6,\xi_7)\ ,\ \xi_7) = \Lambda_4(\xi_6,\xi_7) \\ \xi_3 &= \Gamma_3(\ \Lambda_4(\xi_6,\xi_7)\ ,\ \xi_6) = \Lambda_3(\xi_6,\xi_7) \end{aligned}$$

which leads to

$$\begin{aligned} \xi_1 &= \Gamma_1(\ \Lambda_3(\xi_6,\xi_7)\ ,\ \xi_6) = \Lambda_1(\xi_6,\xi_7) \\ \xi_2 &= \Gamma_2(\ \Lambda_4(\xi_6,\xi_7)\ ,\ \Lambda_5(\xi_6,\xi_7)) = \Lambda_2(\xi_6,\xi_7) \end{aligned}$$

where the operators Λ_i, i=1,...,4 are defined in terms of the well defined operators Γ_i, and hence are themselves well defined.

It should be noted that the structure of the matrix $A_{r,r}$ depends primarily on the numbering of the sequences in Q_r. However, if the system of equations does contain any direct or indirect recursion, then for any numbering of the sequences in Q_r, the matrix $A_{r,r}$ cannot be strictly upper or lower triangular, and hence, the simple back substitution scheme cannot be carried to completion. For example, in the system of equations $\bar{S}$, we cannot express the sequences ξ_3 and ξ_4 in terms of ξ_6 and ξ_7 unless we have a method for solving coupled equations of the form

$$\xi_4 = \bar{\Lambda}_4(\xi_3,\xi_6,\xi_7) \qquad \text{and} \qquad \xi_3 = \bar{\Lambda}_3(\xi_4,\xi_6) \tag{1}$$

where $\bar{\Lambda}_3$ and $\bar{\Lambda}_4$ are well defined operators.

Yet, in the special case when the operators $\bar{\Lambda}_3$ and $\bar{\Lambda}_4$ are causal operators, it is possible to calculate the sequences ξ_3 and ξ_4 for any given specific sequences ξ_6 and ξ_7. In other words, the equations (1) have always a solution. The inability to express this solution analytically is due to the absence of the time dimension from sequence equations. This motivates the introduction of the iteration operator.

3. The Iteration operator.

It can be easily shown that the solution of any coupled system of equations may be obtained if we have a means for solving recursive equations of the form

$$\zeta = \Gamma(\zeta, \xi_1, \cdots, \xi_n) \tag{2}$$

and obtaining ζ in terms of $\xi_1, \cdots, \xi_n$. Here, Γ is some sequence operator.

In general, the solution of (2) may not be well defined. However, if Γ is causal, then we may prove the following

Theorem 1: Given a causal operator $\Gamma: [\bar{R}_\delta]^{n+1} \to \bar{R}_\delta$, the solution ζ of equation (2) is well defined.

Proof: Consider the following procedure for the computation of ζ:

ALG1 1) Let $\alpha_0 = \delta^*$, the don't care sequence defined by $\delta^*(t) = \delta$ for any t.

2) FOR $k = 1, 2, \cdots$ DO

2.1) Compute the sequence α_k as follows

$$\alpha_k(t) = \begin{cases} \alpha_{k-1}(t) & t < k \\ [\Gamma(\alpha_{k-1}, \xi_1, \cdots, \xi_n)](t) & t = k \\ \delta & t > k \end{cases}$$

2.2) Set $\zeta(k) = \alpha_k(k)$.

In order to prove that the sequence ζ computed by ALG1 satisfies (2) we define the step operators $S_k : \bar{R}_\delta \to \bar{R}_\delta$ for $k = 0, 1, 2, \cdots$ by

$$[S_k \, \zeta](t) = \begin{cases} \zeta(t) & \text{if } t \le k \\ \delta & t > k \end{cases}$$

With this, it is directly seen that, for any t, $\alpha_k(t) = [S_k\zeta](t)$ and hence that $\alpha_k = S_k\zeta$. From ALG1, we then have

$$\zeta(t) = \alpha_t(t) = [\Gamma(S_{t-1}\,\zeta,\xi_1, \cdots \xi_n)](t) \tag{3}$$

However, the definition of causality implies that $\zeta(t)$ may depend only on any element $[S_{t-1}\,\zeta](\tau)$ with $\tau < t$; that is, we may replace $S_{t-1}\,\zeta$ in (3) by ζ. This gives

$$\zeta(t) = [\Gamma(\zeta,\xi_1, \cdots ,\xi_n)](t)$$

and proves that ζ computed by ALG1 indeed satisfies the equation (2). □

Theorem 1 proves the existence of a solution to recursive causal equations and gives a procedure for its computation. Next, we provide a suitable notation for expressing this solution.

Definition: Let $\Gamma: [\bar{R}_\delta]^{n+1} \rightarrow \bar{R}_\delta$ be a given causal operator. The iteration operator I_{η_r} applied to the image sequence $\Gamma(\eta_1, \cdots ,\eta_{n+1})$ with respect to any of the arguments η_r, $1 \le r \le n+1$ shall be defined by

$$\zeta = I_{\eta_r}\,\Gamma(\eta_1, \cdots ,\eta_r, \cdots ,\eta_{n+1})$$

where for any t

$$\zeta(t) = [\Gamma(\eta_1, \cdots ,\zeta, \cdots ,\eta_{n+1})](t) \qquad \square$$

Using a procedure similar to the one given in the proof of Theorem 1, we can show that the image sequence ζ in the above definition is well defined. Note that η_r in the combined operator $I_{\eta_r}\Gamma: [\bar{R}_\delta]^n \rightarrow \bar{R}_\delta$ specifies the argument of Γ to which the recursion is applied. In other words, the arguments of $I_{\eta_r}\Gamma$ are only $\eta_1, \cdots, \eta_{r-1},\eta_{r+1}, \cdots ,\eta_{n+1}$.

Theorem 2: For any causal operator $\Gamma: [\bar{R}_\delta]^{n+1} \rightarrow \bar{R}_\delta$, the solution of the recursive equation (2) is given by

$$\zeta = I_\eta\,\Gamma(\eta,\xi_1, \cdots ,\xi_n)$$

Proof: Follows directly from the definition of the iteration operator. □

Given a system of equations that models a particular network, equations of the form (2) may appear in the solution process only if the network contains some feed back loops. In this case, Γ results from the combination of the operators describing the function of each cell in the loop. If the network is systolic, then each cell is modeled by a causal operator and hence Γ is causal. On the other hand, if the network is self timed, then it may be shown [9] that deadlock will definitely occur if the network contains a loop in which each cell is modeled by a weakly causal operator. In other words, loops in deadlock free self timed networks should contain at least one cell that is modeled by a causal operator. Hence, for deadlock free networks, the combined operator Γ is guaranteed to be causal.

Theorem 2 provides a means for expressing the solution of recursive causal equations. Its application to the verification of systolic networks, however, depends on our ability to manipulate expressions that combine the iteration operator and other sequence operators. The following theorem provides the basis for such a manipulation.

Theorem 3: If $\Lambda\colon [\bar{R}_\delta]^{n+1} \to \bar{R}_\delta$ is a causal sequence operator, and $\Phi\colon \bar{R}_\delta \to \bar{R}_\delta$ is any sequence operator with the property that

$$\Lambda(\Phi\zeta, \Phi\xi_1, \cdots, \Phi\xi_n) = \Phi\, \Gamma(\zeta, \xi_1, \cdots, \xi_n) \tag{4}$$

where Γ may or may not be identical to Λ, then

$$I_\eta\, \Lambda(\eta, \Phi\xi_1, \cdots, \Phi\xi_n) = \Phi\, I_\eta\, \Gamma(\eta, \xi_1, \cdots, \xi_n) \tag{5}$$

Proof: Let $\gamma = I_\eta\, \Gamma(\eta, \xi_1, \cdots, \xi_n)$. By Theorem 2, we know that γ also satisfies $\gamma = \Gamma(\gamma, \xi_1, \cdots \xi_n)$. From the hypothesis (4) we have

$$\Phi\, \gamma = \Phi\, \Gamma(\gamma, \xi_1, \cdots, \xi_n) = \Lambda(\Phi\gamma, \Phi\xi_1, \cdots, \Phi\xi_n)$$

which by Theorem 2 has the solution

$$\Phi\,\gamma = I_\eta\,\Lambda(\eta,\Phi\xi_1,\cdots,\Phi\xi_n)$$

Evidently, this reduces to (5). □

We next give an example that illustrates the application of the iteration operator to the verification of computational networks with feed back loops.

4. Verification of networks with feed back loops - An example.

Consider the back substitution network shown in Figure 1. This network may compute the solution vector x of the equation $L\,x = y$, where y is an n-dimensional vector and L is an $n \times n$ unit lower triangular matrix with band width $k+1$. That is

$$x_t = y_t - \sum_{q=1}^{\min\{k,t-1\}} l_{t,t-q}\ x_{t-q} \qquad t=1,\cdots,n \tag{6}$$

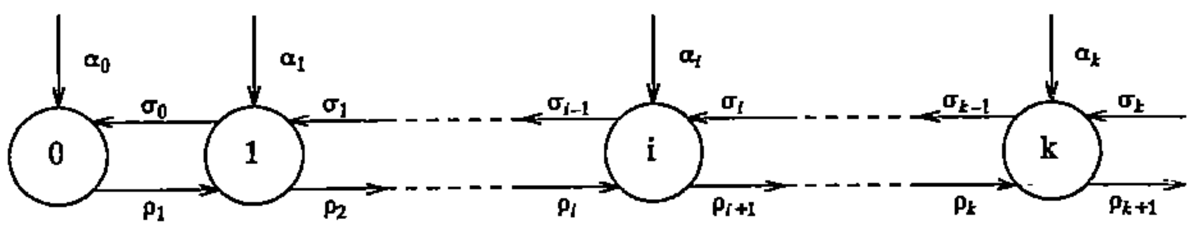

Figure 1 - A back substitution network

Given the labels shown in Figure 1, and following the model of [8], we may describe the operation of the network by the equations

$$\sigma_{i-1} = \Omega_0\,[\,\sigma_i + \rho_i * \alpha_i\,] \qquad i=1,\ldots,k \tag{7.a}$$
$$\rho_1 = \Omega_0\,[\,\alpha_0 - \sigma_0\,] \tag{7.b}$$
$$\rho_{i+1} = \Omega_0\,\rho_i \qquad i=1,\cdots,k \tag{7.c}$$

For proper operation, the input to the network should be specified by

$$\sigma_k = \iota \tag{8.a}$$
$$\alpha_i = \Omega_0^i\,\Theta\,\lambda_i \qquad i=0,\cdots,k \tag{8.b}$$

where $\iota(t) = 0$ for any t, $\lambda_0(t) = y_t$ for any $t \le n$, and $\lambda_i = l_{t+i,t}$ for $i = 1,\ldots,k$ and any $t \le n-i$. Here, the spread operator Θ inserts a δ element between any two successive elements of its operand.

Note that the network contains some feed back loop which creates a mutual dependence between σ_0 and ρ_1. In [8], it was shown that this type of networks may be verified if we assume some knowledge about the form of the output sequences. Here, we will apply the iteration operator to obtained the network I/O from which we may easily obtain the output for the specific given input.

By straight forward manipulation it may be shown that equations (7) reduce to

$$\rho_{k+1} = \Omega_0^k \, \rho_1 \tag{9.a}$$

$$\rho_1 = \Omega_0 \, [\, \alpha_0 - \sum_{j=1}^{k} \Omega_0^j \, [\alpha_j * \Omega_0^{j-1} \, \rho_1 \,] + \Omega_0^k \, \sigma_k \,] \tag{9.b}$$

Application of Theorem 2 to (9.b) and substitution of the result in (9.a) gives the following I/O description of the network

$$\rho_{k+1} = \Omega_0^k \, I_\eta \, [\, \Omega_0 \, [\, \alpha_0 - \sum_{j=1}^{k} \Omega_0^j \, [\alpha_j * \Omega_0^{j-1} \, \eta \,] + \Omega_0^k \, \sigma_k \,] \,]$$

Now, substitution of the input sequences (8) and application of Theorem 3 to factor $\Omega_0 \, \Theta$ from the resulting expression give

$$\rho_{k+1} = \Omega_0^{k+1} \, \Theta \, \xi$$

where

$$\xi = I_\eta \, [\, \lambda_0 - \sum_{j=1}^{k} \Omega_0^j \, [\lambda_j * \eta \,] \,].$$

That is

$$\xi(t) = [\, \lambda_0 \,](t) - \sum_{j=1}^{k} [\, \Omega_0^j \, [\lambda_j * \xi] \,](t).$$

But from the definition of the shift operator Ω_0

$$[\Omega_0^j \, (\lambda_j * \xi) \,](t) = \begin{cases} \lambda_j(t-j) \, \xi(t-j) & \text{if } t > j \\ 0 & \text{if } t \le j \end{cases}$$

Hence

$$\xi(t) = y_t - \sum_{j=1}^{\max\{k,t-1\}} l_{t,t-j} \;\; \xi(t-j), \qquad t=1, \cdots, n$$

which proves that $\xi(t) = x_t$.

5. Conclusion

The existence of correctness proofs in the Chen/Mead functional model [1] for concurrent systems is a direct consequence of the fixed point theory [5]. In this paper, we presented a variation of this theory that is applicable to sequence equations thus establishing the existence of analytical proofs for computational networks in the sequence model of [8].

References

1. M. C. Chen and C. A. Mead, "Formal Specification of Concurrent Systems," *USC Workshop on VLSI and Modern Signal Processing*, Nov. 1982.
2. O. Ibarra, M. Palis, and S. Kim, "Designing Systolic Algorithms Using Sequential Machines," *Proc. of the 25th. FOCS Symposium* , pp. 46-55, 1984.
3. H. T. Kung and W. T. Lin, "An Algebra for VLSI Algorithm Design," *Proc. Conf. Elliptic Problem Solvers, Monterey, CA.*, 1983.
4. M. Lam and J. Mostow, "A Transformational Model of VLSI Systolic Design," *Computer*, vol. 18, no. 2, pp. 42-52, Feb. 1985.
5. J. Lassez, V. Nguyen, and E. Sonenberg, "Fixed Point Theorems and Semantics: A Folk Tale," *Information Processing Letters*, vol. 14, no. 3, pp. 112-116.
6. G. Li and B. Wah, "The Design of Optimal Systolic Arrays," *IEEE Trans. on Computers*, vol. C-34, no. 1, pp. 66-77, Jan 1985.
7. R. G. Melhem, "Formal Analysis of a Systolic System for Finite Element Stiffness Matrices," Technical report ICMA-83-56, The University of Pittsburgh, May 1983. (Also, Journal of Computer and System Sciences, in press)
8. R. G. Melhem and W. C. Rheinboldt, "A Mathematical Model for the Verification of Systolic Networks," *SIAM J. on Computing.*, vol. 13, no. 3, pp. 541-565, Aug. 1984.

9. R. G. Melhem, "Verification of a Class of Deadlock-Free, Self Timed, Computational Arrays," Report ICMA-84-76, The University of Pittsburgh, Sept 1984.

10. R. G. Melhem, "A Language for the Simulation of Systolic Architectures," *Proc. of the 12th International Symposium on Computer Architecture - Boston - Massachusetts*, June 1985.

11. D. J. Moldovan, "On the Analysis and Synthesis of VLSI Algorithms," *IEEE Trans. on Computers*, vol. C-31, no. 11, pp. 1121-1126, Nov. 1982.