Fig. 7. Definition of path function h .

transition time intervals do not overlap. Instead, the path from u^1 to u^2 is broken up into three sections: i) u^1 to s^1 , ii) s^1 to s^2 , and iii) s^2 to u^2 where s^1 and s^2 are straight line functions illustrated in Fig. 6 and defined precisely below.

Since u_i is monotonically increasing for u_i in the interval $[\epsilon, 1 - \epsilon]$, there are unique points t_1, t_2 such that: $u_i(t_1) = \epsilon$ and $u_i(t_2) = 1 - \epsilon$. The slope $(1 - 2\epsilon)/(t_2 - t_1)$ lies in the range $[C_2, C_1]$, thus the function $s^1 = (s_1^1, s_2^1)$ defined by

$$s_i^1(t) = \begin{cases} 0, & 0 \leq t < t_1 - \frac{\epsilon(t_2 - t_1)}{(1 - 2\epsilon)} \\ (t - t_1) \left(\frac{1 - 2\epsilon}{t_2 - t_1} \right) + \epsilon, & t_1 - \frac{\epsilon(t_2 - t_1)}{(1 - 2\epsilon)} < t < t_2 + \frac{\epsilon(t_2 - t_1)}{(1 - 2\epsilon)} \\ 1, & t \geq t_2 + \frac{\epsilon(t_2 - t_1)}{(1 - 2\epsilon)} \end{cases}$$

is an element of Γ .

The functions u^1 and s^1 can be shown to be [6] path connected with a path function $f(p): [1, 0] \rightarrow \Gamma$ defined by

$$f(p) \triangleq (1 - p)u^1 + ps^1.$$

Similarly it can be shown that s^2 and u^2 are path connected. All that remains is to show s^1 and s^2 are path connected. One cannot achieve this with the path function g : $g(p) \triangleq (1 - p)s^1 + ps^2$ since the condition of minimum slope may be violated along the path. Instead, the path function $h(p) = (h_1(p), h_2(p))$ can be defined [6] based on Fig. 7.

The slope of $h_i(p)$ between ϵ and $1 - \epsilon$ lies between the slopes of s_1^1 and s_2^1 between ϵ and $1 - \epsilon$. It is easy to verify that $h(p)$ constitutes a valid path connection. Thus, Γ is connected and there is a range of functions from Γ that produces metastable behavior.

This example illustrates the greater extent of applicability of the theorem to practical input functions with possibly complex waveform conditions.

VII. CONCLUSION

The notion of connectivity of the set of possible input functions, that includes two functions that drive the system to two stable states, is sufficient to imply metastable behavior can occur. The property of connectivity was seen to be much more applicable to practical inputs likely to be found in practice, compared to the set of functions having a bounded first derivative and necessarily including those functions with unbounded higher derivatives as presented in [4]. This correspondence has extended Marino's work [4] to apply in a much wider range of situations more closely matching "the real world."

REFERENCES

- [1] L. Kleeman and A. Cantoni, "Metastable behavior and digital system reliability," Dep. Elec. & Comput. Eng., Univ. Newcastle, Australia, Tech. Rep. EE8440, Sept. 1984; also to appear *IEEE Des. Test. of Comput.*
- [2] T. J. Chaney and C. E. Molnar, "Anomalous behavior of synchronizer and arbiter circuits," *IEEE Trans. Comput.*, vol. C-22, pp. 421-422, Apr. 1973.

- [3] M. Hurtado and D. L. Elliot, "Ambiguous behavior of logic bistable systems," in *Proc. 13th Ann. Allerton Conf. Circuit Theory*, Oct. 1975.
- [4] L. R. Marino, "General theory of metastable operation," *IEEE Trans. Comput.*, vol. C-30, pp. 107-115, Feb. 1981.
- [5] G. F. Simmons, *Introduction to Topology and Modern Analysis*. New York: McGraw-Hill, 1963.
- [6] L. Kleeman and A. Cantoni, "On the unavoidability of metastable behavior," Dep. Elec. & Comput. Eng., Univ. Newcastle, Australia, Tech. Rep. EE8516, Dec. 1984.

(λ, T) Complexity Measures for VLSI Computations in Constant Chip Area

HOWARD C. CARD, P. GLENN GULAK,
ROBERT D. MCLEOD, AND WERNER PRIES

Abstract—The computational complexity measures introduced here are motivated by the trend to higher VLSI integration levels (rather than increased chip area) to accomplish solutions to larger problem instances. It seems that the increase in the computational power of VLSI circuits can be mainly attributed to the reduction in the minimum feature size rather than to an increase in the chip area. In view of this, we present a constant area perspective and consider the discrete Fourier transform and related problems in a VLSI model that has λ and T as its resources. Advantages of the mesh algorithm over the shuffle-exchange algorithm in the computation time for the DFT are shown to arise from an upper bound on current density in the wires, which we suggest must be considered in any VLSI grid model.

Index Terms—Algorithms, $A - T - \lambda$ complexity, communication graphs, computational action, energy, and velocity, DFT, Fourier transform, grid models, mesh-connected computers, parallel computation, shuffle-exchange connected computers, VLSI.

I. INTRODUCTION

It is of importance for computer designers working at the architectural level to know the asymptotic behavior of VLSI algorithms which solve certain problem classes. A typical example is the N -element discrete Fourier transform, which is a focal point in digital signal processing. One wants to know how the silicon chip area A and computation time T grow with N in order to compare algorithms for this transform. The "best" VLSI circuit would have the least asymptotic complexity when AT^x is used as a resource measure.¹

Thompson [1], [2] has placed these discussions on a solid foundation by employing a "grid model" abstraction of a VLSI chip which contains only the information needed for the determination of A and T to within a constant factor. One way of looking at this formalism is: given a minimum feature size (λ), how do the chip area and computation time depend on N for a given problem?

Manuscript received April 30, 1985; revised February 3, 1986. This work was supported in part by the Natural Sciences and Engineering Council of Canada under Grant A1330, and by the Canadian Microelectronics Corporation.

The authors are with the Department of Electrical Engineering, University of Manitoba, Winnipeg, Man. R3T 2N2, Canada.

IEEE Log Number 8611263.

¹A more complete analysis of VLSI complexity would also have to determine the constant factors in the area and time complexities. This would allow proper discrimination between algorithms for a given N , but would require a protracted, and technology-dependent, calculation for each problem class.

We wish to develop a complementary perspective on the complexity of VLSI computations. During the past decade, the increase in the computational power of VLSI circuits can be attributed to the following. The minimum feature size has decreased and this has led to the increase in the packing density of transistors on a chip of a given area. Certainly, the chip size has also increased during the past two decades. However, notwithstanding the prospects of water-scale integration [3], we believe that this increase in chip size is a secondary factor [4]–[6]. Our justification for this perspective is that the reinterpretation of asymptotic results of the (constant λ) model brings out features such as the maximum allowed value of λ for a given algorithm which is a metric related to the ease of fabrication, reliability and yield. In addition, the present work generalizes the results obtained in the constant λ model to a situation in which both A and λ may be variables (general model). It is anticipated that the present formulation will be preferred in making at least some of the decisions regarding "best" VLSI algorithms.

Strictly, the complexity bounds discussed below could not be considered asymptotic if one takes the view that there is a feature size below which transistors cannot function. However, bearing in mind the technology independence of the model, we reject this view in favor of an ever decreasing λ . Both device principles and materials technologies are expected to change (for example, to molecular electronic devices [7], [8]) as λ continues to decrease below those for which conventional (silicon, MOS) devices are operable.

Thus, our point of view is that progressively larger problem instances of a particular VLSI computational problem find their solutions as the result of increasing densities of components on a VLSI chip of essentially constant area. Thus, the ability to handle problem instances of increasing N is primarily the result of the decrease in the minimum feature size (λ) in the components over time. This decrease in λ is driven by improved lithography for mask definition among other factors [9].

The distance metric λ is readily converted into a minimum wire pitch by employing the design rules of a given technology, which gives rise to the scale of the grid for a communication graph such as that advanced by Thompson [1]. The ratio of the wire pitch to λ remains roughly constant with changes in λ , so our perspective becomes one of a set of problem instances (one for each N) in which the grid becomes finer as N increases. The number of nodes (active devices or wire connections) and the number of edges (wire segments) increases with N in the same way as in the Thompson grid model for any problem but the absolute size of the communication graph in our case remains fixed.

II. CONSTANT CHIP AREA MODEL AND ITS IMPLICATIONS

We adopt the scaling theory of device and circuit performance, as described by Dennard *et al.* [10], a summary of which is shown in Table I.² As λ decreases all of the device dimensions decrease by the same factor, which we call α . Since all voltages and currents scale by the same factor, the electric fields and power dissipation per unit area remain constant. In addition to the increase in α^2 in the density of components (number per unit area), we are primarily concerned with the increase in switching speed by the factor α , and we adopt the normal capacitive model of wire delays [1], [11], [12]. For sufficiently small λ , the resistive effects of the wires also become important particularly with wire lengths that do not scale with the technology. This complication can usually be ignored, at least for the near future [11]. This increase in speed also applies to delays associated with driving large wire capacitances provided the lengths of the wires scale with λ . The effect of the reduced delay is that clock

² We are aware of the dangers of subscribing too closely to the scaling theory [10], [13]. In order to optimize transistor performance, the rules in Table I are routinely broken. There is also a tendency to keep field oxide thickness and power supply voltage constant with decreasing λ , for obvious reasons. There is no doubt however that with the decrease in λ comes a reduced drive capability and a faster device speed which promote the use of a scaled load capacitance. To first order, these time and capacitance metrics are proportional to λ .

TABLE I
SCALING THEORY OF CIRCUIT BEHAVIOR³ [10]

minimum feature size	λ	λ/α
device dimensions	L	L/α
voltages	V	V/α
currents	I	I/α
capacitances	C	C/α
switching times	T	T/α
switching energy	E	E/α^3
power	P	P/α^2
power density	D	D
temperature ³	T_0	T_0/α

α = scaling factor (dimensionless)

periods (which were $\Theta(1)$ in the constant λ model) are in the present case proportional to λ , and hence are N -dependent. This has to be included in determinations of asymptotic bounds on computation time T in the constant A model.

The area A in the constant λ model is generally considered to be a measure of cost, since it directly impacts the yield of correct chips [3], [13]. This area is replaced by $1/\lambda^2$ in the constant area model. While decreasing λ will certainly increase process complexity and hence cost, we do not wish to imply that $1/\lambda^2$ is the same cost measure as A was.

The assumptions in our model are similar to those made by Thompson [1] with respect to the geometric properties of the VLSI grid model. In particular, the treatment of I/O considerations (or rather the neglect of them) follows his work. That is, we assume convex circuits with when-determinate and where-determinate inputs. We further assume that all inputs appear simultaneously on chip (similarly for the outputs); we have a synchronous, semilective, unilocal computational system. A single level of metallization is assumed, which gives legitimate results, to within a constant factor [1], [12]. Near optimal results can be obtained even if the total number of metallization levels is some positive constant.

The assumptions of Thompson with respect to self-timed regions and delays in driving wires require some discussion when carried over to the constant A model. In particular, note that all time steps as determined in Thompson's work correspond to numbers of clock cycles. Therefore, there is a factor proportional to the minimum feature size λ to be included in the conversion from Thompson's model to our computation time T . Results from the constant λ model for computation time T correspond to T/λ in the present model. That is, T/λ clock cycles are sufficient to solve a problem. (Table II illustrates the metrics of the constant λ (Thompson) model, the general model in which both A and λ are variable, and the present constant A model.)

Theorem 1: Given a lower bound or upper bound on chip area A , computation time T , and a combined metric AT^{2x} ($0 \leq x \leq 1$) in the constant λ model, we can find the corresponding bounds on λ , T/λ , or T^x/λ^{1+x} for the constant A model by replacing A by $1/\lambda^2$ and T by T/λ . For the general model, replace A and T by A/λ^2 and T/λ , respectively.

Proof: The assumptions of the two models are the same since they refer to the properties of communication graphs. The number of primitive cells in the graph is in general A/λ^2 , which is equivalent to A in the constant λ model and is replaced by $1/\lambda^2$ in the constant A model. Upper and lower bounds in the constant λ model were determined by counting primitive cells; the minimum bounding rectangle was used for the upper bounds, whereas the wire area was used for the lower bound.

The computation time T in the constant λ model assumed a clock period $P = \Theta(1)$. In the constant A model this clock period is $\Theta(\lambda)$. The values of T in the constant λ model and T/λ in the constant A

³ In a rigorous scaling process.

TABLE II
METRICS OF GRID MODELS

	General model	Constant λ model	Constant A model
number of grid elements	A/λ^2	A	$1/\lambda^2$
number of clock cycles	T/λ	T	T/λ
normalized energy of computation	AT/λ^3	AT	T/λ^3
metric which is characteristic of problem (independent of bisection width)	AT^2/λ^4	AT^2	T^2/λ^4
general area-time- λ metric ($0 \leq x \leq 1$)	$AT^{2x}/\lambda^{2(1+x)}$	AT^{2x}	$T^{2x}/\lambda^{2(1+x)}$

model are therefore equal, since they represent the number of clock cycles required by the common communication graph. \square

In this correspondence, we do not intend to rederive the lower and upper bounds on A and T for the discrete Fourier transform (DFT) computation. The interested reader is referred to Thompson's Ph.D. dissertation for the basic assumptions of the grid model. [14], [15] present the derivation of optimal layouts (in the AT^2 sense) for the mesh and shuffle-exchange algorithms, as described in the constant λ interpretation of VLSI communication graphs. Here we simply reinterpret the results for the constant A model using Theorem 1, as a series of corollaries.

Although similar results to those below have been obtained for VLSI sorting, and the reinterpretation of those results is also straightforward, we confine our attention to results for minimum feature size λ and the computation time T in the VLSI solution of the N -element discrete Fourier transform. The results for the mesh and shuffle-exchange algorithms actually correspond to admissible VLSI communication graphs which implement the DFT using recirculation networks [1].

Corollary 1: For a problem with bisection flow $I(N)$, with N the input size parameter, $AT^2 = \Omega(\lambda^4 I^2(N))$.

Corollary 2: Any VLSI algorithm which implements the N -point DFT requires $\lambda = O(1/w)$ and $T/\lambda = \Omega(N \log N/w)$ where w is the minimum bisection width of the communication graph. Thus $T/\lambda^2 = \Omega(N \log N)$.

Corollary 3: The lower bounds on T^x/λ^{1+x} for Fourier transformation are given by $T^x/\lambda^{1+x} = \Omega(N^{(1+x)/2} \log^x N)$ for any x in the range $0 \leq x \leq 1$. In this case T is average time [1].

Corollary 4: The N -element DFT requires $\lambda = \Omega(1/N^{1/2} \log^{1/2} N)$ and $T/\lambda = O(N^{1/2} \log^{1/2} N)$ using N multiply-add cells in a mesh-connected recirculation network. Using a shuffle-exchange connection, $\lambda = \Omega(\log N/N)$ and $T/\lambda = O(\log^2 N)$ [14], [15].

Several other derived metrics are presented in Tables III and IV. Results for the constant λ model [1], [14], [15] appear in the central column of these Tables. Fig. 1 shows schematically the dependence of the normalized computational energy T/λ^3 and of the minimum feature size λ upon the problem size parameter N for Fourier transforms based upon the shuffle-exchange and mesh connections of the VLSI communication graph. We note that for a given N , the mesh algorithm can be realized with a larger λ than that required for the shuffle-exchange algorithm; this indicates an advantage of the mesh implementation. The normalized computational energy is also lower for the mesh connection. These relations are based upon Table IV, the traditional complexities, which ignore the constraints imposed by an upper bound on allowed current density in the wires. The current density considerations are discussed in Section III, and these result in still greater advantages of the mesh over the shuffle-exchange algorithm with respect to asymptotic computation time.

On the basis of the above discussion, optimal designs have been shown for $T = \Theta(\lambda(N \log N)^{1/2})$ in [14], and for $T \in [\Omega(\lambda \log^2 N), O(\lambda(N \log N)^{1/2})]$ in [16] when expressed in this general model. More recently [17], it has been shown that the lower bound on

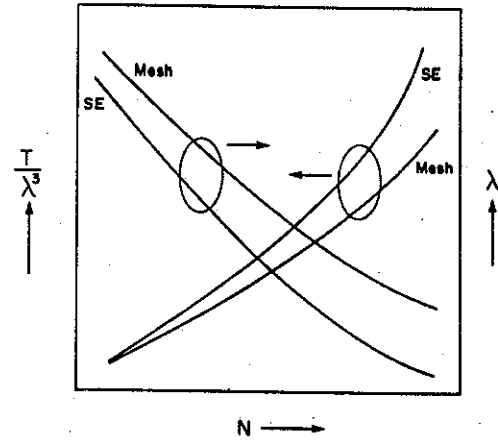


Fig. 1. Dependence of normalized computational energy T/λ^3 and feature size λ upon problem size parameter, for N -point Fourier transforms on mesh and shuffle-exchange graphs. Current density limitations are not included.

AT^2 can be achieved for any computation time $T \in [\Omega(\lambda \log N), O(\lambda(N \log N)^{1/2})]$, again translating [17] into the general model of this paper.

The area bound for the computation of the DFT has recently been shown [18] to satisfy $A = \Omega(\lambda^2 N \log N)$ which strengthens the bound on AT^{2x} to $AT^{2x} = \Omega((\lambda^2 N \log N)^{1+x})$. The original Thompson bound had been based on $A = \Omega(N)$ and was therefore weaker. For $x = 1/2$, we have a lower bound on energy, in the general model, of $AT = \Omega(\lambda^3(N \log N)^{3/2})$.

III. COMPUTATIONAL ENERGY AND CURRENT DENSITY

It is well known that power dissipation per unit area, which we call D , is upper bounded by a constant. Typical upper limits of D for silicon chips are 1 Wcm^{-2} with air cooling and 20 Wcm^{-2} with liquid coolants.

In most VLSI technologies, the power-delay product of the logic gates $D\tau$ is approximately constant over a wide range of D for a given λ . This means that, if we are interested in the effects of varying the power density (it has implicitly been assumed to be a constant in the expressions up to this point), then we must recognize that

$$T = k\tau f(N) = \theta(f(N)/D) \quad (1)$$

where k is a constant, τ is the primitive gate delay, and $f(N)$ represents the unknown dependence of T on the problem size parameter.

The energy required for the computation is $E = DAT$, but in view of (1), there is a $1/D$ dependence of T so that this energy is actually independent of D (within the assumption of the constant power-delay product). The computational energy is therefore given in the general model by

$$E = \lambda^3 \Omega(N^{3/2} \log N) \quad (2)$$

using AT from Table III.

The importance of power dissipation becomes apparent when the action of the computation is determined. The computational action is given by

$$C = ET = DAT^2 = \frac{\lambda^4}{D} \Omega(N^2 \log^2 N) \quad (3)$$

where the dependence of T upon D from (1) is written explicitly into the value of AT^2 from Table III. The concept of action is familiar from theoretical physics as the dimensions of uncertainty products

TABLE III
BOUNDS ON N -POINT DFT PROBLEM*

General model	Constant λ model	Constant A model
$A/\lambda^2 = \Omega(w^2)$	$A = \Omega(w^2)$	$1/\lambda^2 = \Omega(w^2)$ $\lambda = O(1/w)$
$T/\lambda = \Omega(N \log N/w)$	$T = \Omega(N \log N/w)$	$T/\lambda = \Omega(N \log N/w)$
$AT/\lambda^3 = \Omega(N^{3/2} \log N)$	$AT = \Omega(N^{3/2} \log N)$	$T/\lambda^3 = \Omega(N^{3/2} \log N)$
$AT^2/\lambda^4 = \Omega(N^2 \log^2 N)$	$AT^2 = \Omega(N^2 \log^2 N)$	$T/\lambda^2 = \Omega(N \log N)$
$AT^{2x}/\lambda^{2x+1} = \Omega(N^{1+x} \log^{2x} N)$	$AT^{2x} = \Omega(N^{1+x} \log^{2x} N)$	$T^x/\lambda^{1+x} = \Omega(N^{(1+x)/2} \log^x N)$
$A/\lambda^2 = \Omega(N)$	$A = \Omega(N)$	$\lambda = O(1/N^{1/2})$

* w = minimum bisection width of communication graph and N = problem size parameter.

in the Heisenberg uncertainty principle; these are the dimensions of Planck's constant. The lower bound on C is obtained for the maximum D which is $O(1)$ as discussed above. Thus

$$C = \frac{\lambda^4 \Omega(N^2 \log^2 N)}{O(1)} = \lambda^4 \Omega(N^2 \log^2 N). \quad (3')$$

The lower bound on AT^2 established by Thompson is therefore interpreted as the minimum value of the computational action arrived at under conditions of maximum power dissipation.

Also note that the parameter λ^3/T (or $1/AT$ in the constant λ model) is a measure of *computational throughput*, which by the above relation is upper bounded, by power dissipation, for any DFT algorithm. This computational throughput is the same measure as is used in the VHSIC program, in which case its units are *gate - Hz/mm²* [19].

In the mesh-connected algorithm of Preparata [14] for the DFT problem we can express the general model as

$$AT = \lambda^3 O(N^{3/2} \log^{3/2} N). \quad (4)$$

In the constant A model, $A = \Theta(1)$. The throughput exhibits a lower bound for the mesh connection of

$$\frac{\lambda^3}{T} = \Omega\left(\frac{1}{N^{3/2} \log^{3/2} N}\right). \quad (5)$$

In the shuffle-exchange algorithm of Leighton [15] for the DFT problem the throughput is given by

$$\frac{\lambda^3}{T} = \Omega\left(\frac{1}{N^2}\right). \quad (6)$$

The lower bound on λ^3/T is in this case well below the upper bound of Table III by a factor $N^{1/2}/\log N$. Thus the shuffle-exchange algorithm calls for much smaller values of λ than the mesh algorithm, in a system of fixed (maximum) power dissipation.

We now turn to a discussion of the limitations on current density in the wires, which we have recently discussed in the context of the constant λ perspective [20]. As the feature size λ decreases, the clock period decreases in proportion, according to the scaling theory. For wires of length L there are $\log(L)$ stages to drive these wires, each with a delay of one clock period [2]. The final stage charges or discharges the wire in a single clock period.⁴ Since the charge stored on the wire $Q = CV$ where C is the capacitance of the line and V is the logic swing, we note that Q is proportional to $(WL/d)V$ where W and L are the width and length of the wire being driven and d is the

⁴ This $\log L$ delay is a latency period in the algorithm which cannot be eliminated in a pipelined computation. This is because the wire lengths are in general unequal in various branches of the communication graph. The alternative to $\log L$ stages of delay is to stretch the clock period by a factor $\log L$ in a single stage [14].

height of the wire above the silicon substrate (the oxide thickness). As discussed in Section II, W , d , and V are each proportional to λ and they are reduced by the factor α when λ is reduced by this same factor. Thus Q is proportional to $L\lambda$. Since the clock period τ is proportional to λ we find that the current in the final driver stage is $I = Q/\tau$ and this is proportional to L . If the length of the wires employed in a particular algorithm scales with λ then I becomes proportional to λ as shown in Table I.

It is well known that there is an upper bound on the instantaneous current density J in any interconnect material used in VLSI [21]. Note that the cross-sectional area of these wires a varies asymptotically as λ^2 so that $J = I/a$ varies as $1/\lambda$ in the best case (I is proportional to λ). This imposes a serious limitation on the minimum λ if we insist that the clock period scales with λ . In fact, below some critical λ it becomes necessary to prevent further reductions in the clock period so that J does not increase beyond the allowed upper bound. These considerations arise from physical arguments that are not generally expressed in comparing computational complexities of algorithms. A similar timing model has been expressed by Chazelle and Monier [22].

We now consider again the mesh and shuffle-exchange graphs for the Fourier transform. The results [14], [15] for these VLSI algorithms, when taken over to the constant area model, indicate that the longest wires in the graphs do not scale with λ in the latter case. Rather we have that $L = \lambda$ and $L = \lambda(N/\log N)$ for the longest wires in the mesh and shuffle-exchange graphs, respectively. From the above discussion, the current density J in these wires varies as L/λ^2 if the clock period varies as λ . This implies that J increases as $1/\lambda$ and as $N/\lambda \log N \approx 1/\lambda^2$ for the mesh and shuffle-exchange cases, respectively. The limitation is much more severe for shuffle-exchange networks.

What does this (physical) upper bound on J imply for our algorithms? If we are interested in asymptotic (large N) behavior, we will certainly encounter this limit. Present λ values of approximately $1 \mu\text{m}$ are already sufficiently small to have to deal with this problem. The only solution is to allow the upper bound on J to establish the minimum allowed clock period. As N increases (λ decreases) the present discussion implies that the clock period must eventually saturate for the mesh and increase for the shuffle-exchange case as in Fig. 2, described below. With these considerations, the computation time comparisons of the earlier sections are no longer meaningful. The speed advantage of the shuffle-exchange algorithm is lost. Note that these restrictions on wire delays arise within the capacitive model, unlike the delays associated with the finite speed of light and the diffusion model of wire delay [1], [11]. The latter two effects are normally negligible; the present effect is not.

In particular J can be held constant as λ decreases with increasing N , by allowing the clock period to become constant or $\Theta(1)$ in the mesh and to increase as $\Theta(N/\log N)$ in the shuffle-exchange implementations. The number of clock cycles is $O(N^{1/2} \log^{1/2} N)$ for the mesh and $O(\log^2 N)$ for the shuffle-exchange. Since the computation time is the product of the clock period and the number of clock cycles, we have, for a system in which current density is upper bounded by a constant, the following asymptotic results.

Mesh:

$$\lambda = \Omega \left(\frac{1}{N^{1/2} \log^{1/2} N} \right) \quad (7)$$

$$T = O(N^{1/2} \log^{1/2} N). \quad (8)$$

Shuffle-Exchange:

$$\lambda = \Omega(\log N/N) \quad (9)$$

$$T = O(N \log N). \quad (10)$$

Thus the mesh algorithm proves to be superior to the shuffle-exchange algorithm for the Fourier transform, both in having a larger minimum required λ and a lower computation time in the asymptotic limit. These same conclusions may be reached in the constant λ model. The origin of these bounds are however made more clear by the use of the constant-area model.

Fig. 2 illustrates schematically the dependence of the clock period upon λ which is dictated by the upper bound on the current density in the wires. For large λ , the clock period is proportional to λ as a result of the scaling theory, for both mesh and shuffle-exchange graphs. Current density limits are encountered at a larger λ in the shuffle-exchange connection than in the mesh, as a consequence of the longer wires in the former case. Below this point, which implies a minimum in the clock period in Fig. 2, the increase in clock period for the shuffle-exchange with further decreases in λ is a consequence of maintaining the current density constant at its maximum allowed value.

IV. CLOSING REMARKS AND CONCLUSIONS

The implementation of the DFT using the mesh connection of the VLSI recirculation network is possible (for a given N) with a much larger λ than that required for the shuffle-exchange implementation (Table IV). The lower bound on λ for these algorithms means that λ need not be smaller than these values (for sufficiently large N).

The required λ is an important metric in the sense that it relates to the ease of fabrication and hence to the reliability of the system, as well as to many other cost factors. Mesh connections are clearly preferred over shuffle-exchange networks for sufficiently large N according to this metric.

In the absence of current density limitations, the shuffle-exchange connection has considerable speed advantage, as pointed out earlier [1], [2]. On the other hand, the speed advantage has to be considered together with the area advantage of the mesh connection (Table IV). If the (asymptotically) larger λ of the mesh is considered, then on a chip of given area more problems can be solved simultaneously by scaling the mesh to the same λ as that required for the shuffle-exchange case. The computation time also scales with λ so that the energy of computation is proportional to λ^3 (power dissipation as λ^2 by Table I). This motivates the *normalized energy metric* AT/λ^3 in the general model of Table II, which becomes T/λ^3 in the constant A model and AT in the constant λ model. This is the reciprocal of the computational *throughput*, which is λ^3/AT in the general model. For the present (constant A) model, $T/\lambda^3 = O(N^{3/2} \log^{3/2} N)$ for the mesh connection and $T/\lambda^3 = O(N^2)$ for the shuffle-exchange connection, which also supports the mesh connection as a better VLSI algorithm for the DFT. It may further be argued that T/λ^3 captures the testability of the algorithm (higher values of T/λ^3 are harder to test) since it represents the product of the number of potential nodes in the graph (number of primitive cells) and the number of clock cycles to compute the algorithm.

Finally, the speed advantage of the shuffle-exchange connection over the mesh connection is lost if one considers the physical limitations imposed by an upper bound on allowed current density in the wires. The net result of these considerations is that computation time for the shuffle-exchange algorithm is increased by N -dependent factors. Thus the computation time for the mesh algorithm becomes asymptotically faster.

The algorithms for the DFT described by Thompson are not expected to be optimal when current density limitations are included. One must in this case minimize the length of the longest wires in the communication graph. Leighton [15], [23] discusses the still open problem of finding shuffle-exchange layouts with minimax edge length. Leighton [15] had shown that this edge length is bounded

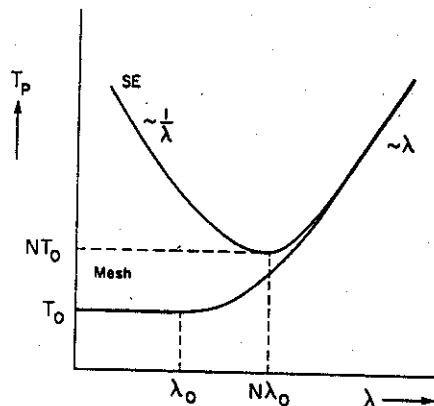


Fig. 2. Dependence of system clock period on minimum feature size λ resulting from upper bound on current density in the wires, for the Fourier transforms on mesh and shuffle-exchange graphs. For large λ , clock period is proportional to λ . For sufficiently small λ , clock period is $\Theta(1)$ and $\Theta(N/\log N)$ in the mesh and shuffle-exchange cases, respectively.

TABLE IV
PERFORMANCE OF ALGORITHMS FOR DFT

General model	Constant λ model	Constant A model
a) mesh connection		
$A/\lambda^2 = O(N \log N)$	$A = O(N \log N)$	$\lambda = \Omega(1/N^{1/2} \log^{1/2} N)$
$T/\lambda = O(N^{1/2} \log^{1/2} N)$	$T = O(N^{1/2} \log^{1/2} N)$	$T/\lambda = O(N^{1/2} \log^{1/2} N)$
b) shuffle-exchange connection		
$A/\lambda^2 = O(N^2/\log^2 N)$	$A = O(N^2/\log^2 N)$	$\lambda = \Omega(\log N/N)$
$T/\lambda = O(\log^2 N)$	$T = O(\log^2 N)$	$T/\lambda = O(\log^2 N)$

below at $\Omega(N/\log^2 N)$. If such a layout is found, it will outperform the shuffle-exchange graph considered here, and may represent the optimal shuffle-exchange layout in view of current limitations. It may be noted that recent developments have demonstrated optimal minimax edge length using other architectures: the mesh of shuffles [24] or the mesh of cube-connected cycles [25].

One may also regard λ/T , the reciprocal of the number of clock cycles required by an algorithm to reach a solution, as a figure of merit which we call the *computational velocity* of the algorithm. We have that λ/T is $\Omega(1/N \log N)$ for the mesh and $\Omega(1/N^2)$ for the shuffle-exchange graph, in view of the current density limitations.

We conclude that the constant-area (λ , T) model of VLSI complexity provides a useful complementary description to the customary constant- λ (A , T) model in describing performance of algorithms.

V. ACKNOWLEDGMENT

The authors would like to thank the referees for their comments and suggestions.

REFERENCES

- [1] C. D. Thompson, "A complexity theory for VLSI," Ph.D. dissertation, Dep. Comput. Sci., Carnegie-Mellon Univ., Pittsburgh, PA, Aug. 1980.
- [2] C. D. Thompson, "Fourier transforms in VLSI," *IEEE Trans. Comput.*, vol. C-32, pp. 1047-1057, Nov. 1983.
- [3] K. Hwang and F. A. Briggs, *Computer Architecture and Parallel Processing*. New York: McGraw-Hill, 1984, pp. 784-788.
- [4] R. N. Noyce, "Large-scale integration: What is yet to come?," *Science*, vol. 195, pp. 1102-1106, Mar. 1977.
- [5] R. W. Keyes, "The evolution of digital electronics towards VLSI," *IEEE J. Solid-State Circuits*, vol. SC-14, pp. 187-195, Apr. 1979.
- [6] D. J. McGreivy, "VLSI chip trends—Size, complexity, cost," in *VLSI Technologies*, D. J. McGreivy and K. A. Pickar, Eds. Washington, DC: IEEE Comput. Soc., 1982, pp. 31-39.
- [7] F. L. Carter Ed., *Molecular Electron Devices*. Marcel Dekker, 1982.
- [8] F. L. Carter, "From electroactive polymers to the molecular electronic device computer," in *VLSI Technologies*, D. J. McGreivy and K. A. Pickar, Eds. Washington, DC: IEEE Comput. Soc., 1982, pp. 328-342.
- [9] G. L. Varnell, P. L. Shah, and R. H. Havemann, "MOS and bipolar VLSI technologies using electron-beam lithography," *Proc. IEEE*, vol. 71, pp. 612-639, May 1983.

[10] R. H. Dennard, F. H. Gaenssien, H. N. Yu, V. L. Rideout, E. Bassous, and A. R. LeBlanc, "Design of ion-implanted MOSFETs with very small physical dimensions," *IEEE J. Solid-State Circuits*, vol. SC-9, pp. 256-268, Oct. 1984.

[11] G. Bilardi, M. Pracchi, and F. P. Preparata, "A critique of network speed in VLSI models of computation," *IEEE J. Solid-State Circuits*, vol. SC-17, pp. 696-702, Aug. 1982.

[12] J. D. Ullman, *Computational Aspects of VLSI*. Rockville, MD: Computer Science, 1984.

[13] C. A. Mead and L. Conway, *Introduction to VLSI Systems*. Reading, MA: Addison-Wesley, 1980.

[14] F. P. Preparata, "A mesh-connected area-time optimal VLSI multiplier of large integers," *IEEE Trans. Comput.*, vol. C-32, pp. 194-198, Feb. 1983.

[15] F. T. Leighton, "Complexity issues in VLSI: Optical layouts for the shuffle-exchange graph and other networks," Ph.D. dissertation, Dep. Math., Massachusetts Inst. Technol., Cambridge, MA, Sept. 1981.

[16] F. P. Preparata and J. Vuillemin, "Area-time optimal VLSI networks for computing integer multiplication and discrete Fourier transform," in *Proc. I.C.A.L.P.*, Haifa, Israel, July 1981, pp. 29-40.

[17] G. Bilardi and M. Scaffazadeh, "Optimal discrete Fourier transform in VLSI," in *VLSI: Algorithms and Architect.*, *Proc. of the Int. Workshop Parallel Comput. VLSI*, Amalfi, Italy, 1985, pp. 78-89.

[18] P. Duris, O. Sykora, C. D. Thompson, and G. Vrto, "On the area required for selection, discrete Fourier transformation and Walsh-Hadamard transformation," to be published.

[19] B. A. Bowen and W. R. Brown, *VLSI Systems Design for Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1982, pp. 280-281.

[20] H. C. Card, W. Pries, and R. D. McLeod, "Contributions to VLSI computational complexity theory from bounds on current density," *Integration*, vol. 4, pp. 175-183, 1986.

[21] R. W. Keyes, "Physical limits in digital electronics," *Proc. IEEE*, vol. 63, pp. 740-767, May 1975.

[22] B. Chazelle and L. Monier, "A model of computation for VLSI with related complexity results," *J. Ass. Comput. Mach.*, vol. 32, pp. 573-588, July 1985.

[23] D. Kleitman, F. T. Leighton, M. Lepley, and G. L. Miller, "An asymptotically optimal layout for the shuffle-exchange graph," *J. Comput. Syst. Sci.*, vol. 6, pp. 339-361, June 1983.

[24] A. Aggarwal, "On I/O placement in VLSI," in *Proc. 21st Ann. Allerton Conf. Commun., Contr., Comput.*, Monticello, IL, Oct. 1983, pp. 236-243.

[25] G. Bilardi and F. P. Preparata, "A minimum area VLSI network for $O(\log N)$ time sorting," *IEEE Trans. Comput.*, vol. C-34, pp. 336-343, Apr. 1985.

Systolic Arrays for Matrix Transpose and Other Reorderings

DIANNE P. O'LEARY

Abstract—In this correspondence, a systolic array is described for computing the transpose of an $n \times n$ matrix in time $3n - 1$ using n^2 switching processors and n^2 bit buffers. A one-dimensional implementation is also described. Arrays are also given to take a matrix in by rows and put it out by diagonals, and vice versa.

Index Terms—Matrix transpose, parallel computation, systolic array.

I. INTRODUCTION

The task of computing the transpose of a matrix arises in many matrix algorithms, for example, in computing congruence transformations $B = UAU^T$ and in computing a step of the QR algorithm for finding eigenvalues of a matrix.

Systolic arrays, introduced by Kung and Leiserson [4], are a very popular implement for parallel matrix computation. However, there seems to be no consensus on the best way to transpose a matrix for use in a systolic array algorithm. For example, Bojanczyk, Brent, and

Manuscript received April 5, 1985. This work was supported by the Air Force Office of Scientific Research under Grant AFOSR-82-0078.

The author is with the Department of Computer Science, University of Maryland; College Park, MD 20742.

IEEE Log Number 8611264.

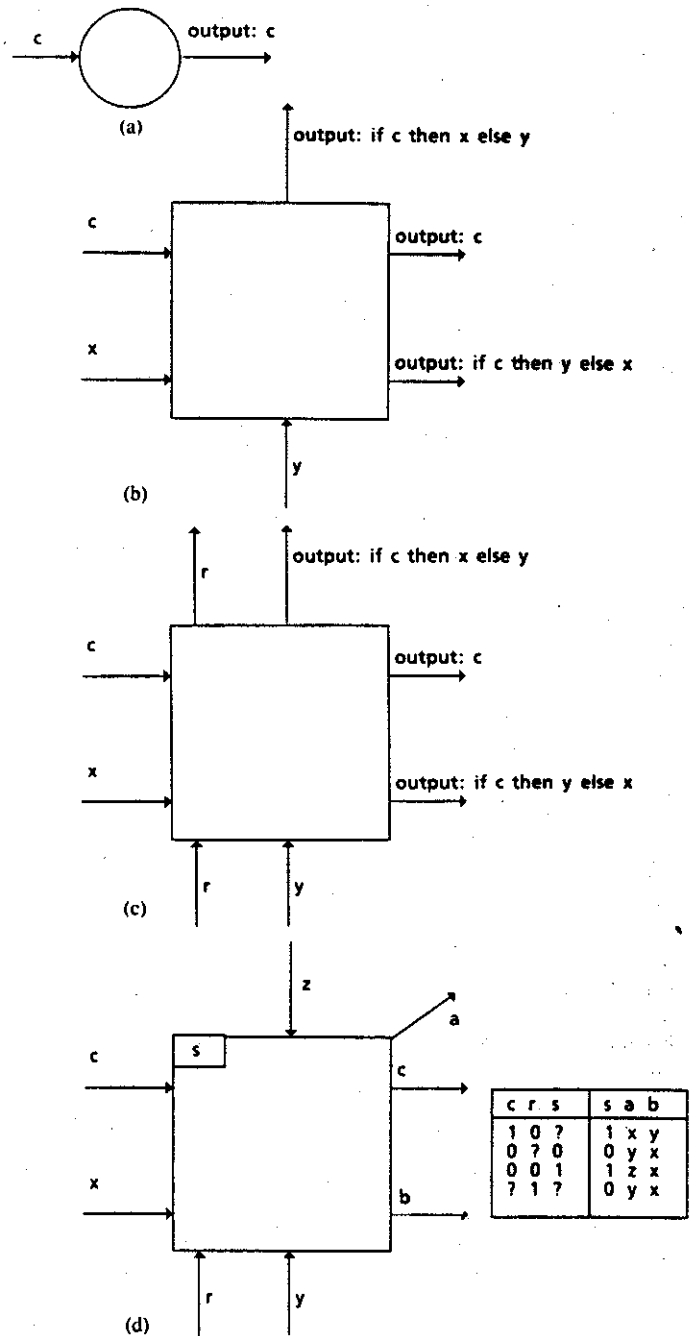


Fig. 1. Four systolic array elements. The outputs and new state definitions for the last one are indicated in the table.

Kung [2] suggest using "a buffer that supports fast two-dimensional addressing," and the speed of their algorithm depends critically on the speed of this hardware. Ullman [5] discusses a systolic algorithm of Atallah and Kosaraju [1] which transposes a matrix in place.

We present in this paper a rather simple systolic array which puts out the transpose of a matrix. The matrix is pumped in systolically, rather than being preloaded as in the above two methods. We also present two modifications of this array; one takes a matrix in by rows and puts it out by diagonals and the other reverses this process. A one-dimensional version of the matrix transpose array is also given. Alternate designs for the diagonal-to-row and row-to-diagonal conversions are given by Ipsen [3] who also presents an algorithm for transposition of a matrix in diagonal format.

To keep diagrams uncluttered, we adopt the convention that unlabeled inputs and outputs are either indeterminate or of no interest. The definitions of the four building blocks of the arrays are given in Fig. 1. The circles represent buffers which cause the control