Area-Time Complexity for VLSI

C. D. Thompson
Carnegie-Mellon University
Pittsburgh, Pennsylvania

Abstract

The complexity of the Discrete Fourier Transform (DFT) is studied with respect to a new model of computation appropriate to VLSI technology. This model focuses on two key parameters, the amount of silicon area and time required to implement a DFT on a single chip. Lower bounds on area (A) and time (T) are related to the number of points (N) in the DFT: $AT^2 > N^2/16$. This inequality holds for any chip design based on any algorithm, and is nearly tight when $T = \Theta(N^{1/2})$ or $T = \Theta(\log N)$.

1. Introduction

The theory of computation is valid over a synthetic domain: its formal models have relevance only if they correspond to possible computational systems. Technological changes can affect the realm of possiblilty. In this light, it would be surprising if the "VLSI revolution" did not spawn new theoretical models. This paper is an attempt to show that interesting complexity results are available through the use of a "VLSI model of computation".

Two parameters are of overriding interest in a VLSI design, its speed and its size. Speed can be handled with familiar complexity tools, that is, measuring time by counting elementary operations. Size in the VLSI world is best expressed as the total area of silicon used. This is quite a different metric from a count of "active elements", "gates", or "registers". It may be the case that most of the chip is devoted to connections between such active elements. A complexity theory for VLSI must thus concern itself with the layout of active elements in the plane, along with their interconnections.

This research is supported in part by the National Science Foundation under Grant MCS 75-222-55 and the Office of Naval Research under Contract N00014-76-C-0370, NR 044-422. C. D. Thompson is an NSF fellow.

The VLSI model: area and time.

There is a natural unit of area for VLSI. Manufacturing and physical limitations give rise to a "minimum feature width", λ. This is the width of the narrowest wire, and λ^2 is approximately the area of the smallest transistor. The 64K RAM currently available has an area of about $10^5\lambda^2$. Chips of 10^7 or $10^8\ \lambda^2$ may be possible [Mead 78].

The choice of a unit of time is slightly more problematical. Here, following [Mead 78], it will be taken as the length of time that it takes a signal to propagate along a wire, or on-chip interconnection. This propagation time can be made independent of the length of the wire, by fitting larger drivers to longer wires. Larger drivers of course occupy more area, but need never take more than 10% of the area of the wire they drive ($1\lambda^2$ for a wire of length 10λ, $10^4\lambda^2$ for a $10^5\lambda$ wire). By fudging λ upwards by 5%, the area of the driver is thus absorbed into the area of its wire.

A full exposition of the VLSI model is deferred to Section 2.

The DFT.

The computational problem studied in this paper is the Discrete Fourier Transform (DFT). The DFT is defined over any commutative ring, but only finite rings will be considered here. Elements of infinite rings have no fixed-length representation, leading to grave computational difficulties. Approximate methods are beyond the scope of this paper.

A satisfactory ring does exist for VLSI, the integers modulo m. If $m = 2^k-1$, ordinary fixed-point arithmetic on k bit words will produce exact answers. An N-point DFT can be performed in this ring if N divides $p-1$ for each prime p dividing m [Bonneau 73].

Formally, the DFT is a matrix-vector multiplication, $A\tilde{x} = \tilde{y}$. The input vector is $\tilde{x}$, the output vector is $\tilde{y}$, and A is an N by N matrix of constants,

$$A[i,j] = w^{ij}$$

The constant w must be a principal Nth root of unity. That is, it must satisfy

$$w \neq 1,$$
$$w^N = 1, \text{ and}$$
$$\sum_{0\le j<N} w^{jp} = 0, \text{ for } 1\le p<N$$

This definition and a fuller explanation of the DFT may be found in [Aho 74].

Proof strategy, results.

The strategy of this paper is to focus on the cost of communications in a parallel system. Little consideration is given to the silicon area or time taken to perform arithmetics on "local" data. Instead, the area-time analysis is performed on the area of the connecting wires and the time taken by the on-chip communication of intermediate results. This emphasis will be justified in Sections 4 and 5, when the area and time bounds are derived.

Each possible chip design will be analyzed in terms of the pattern of interconnections it provides between arithmetic units. A well-connected pattern occupies a lot of area, but a poorly-connected network takes a long time to compute a DFT.

One important factor ignored in this paper is the difficulty of performing off-chip communication. If the pinout and gating speed of the chip is fixed, there is clearly a maximum rate at which chip I/O can occur. The time taken to perform an N-point DFT on data initially off the chip is thus $\Omega(N)$, a trivial complexity result. For this reason, the computational problem studied here is the DFT of on-chip data, under the assumption that the timing constants are such that off-chip communication is not critical. An analysis of the achievable performance of multi-chip systems for the computation of the DFT remains as an interesting open problem.

The complexion of the area-time tradeoff in the computation of the DFT may be expressed in two ways. Following [Mead 78], a minimum value may be found for some particular cost function, such as the product of area with time. Alternatively, one may seek a function of area and time that describes the performance of many "good" designs.. The result of this paper is expressed in both of these ways.

For cost functions of the form AT^x with $0 \le x \le 2$, any chip that performs an N point DFT costs at least $\Omega(N^{1+x/2})$. This minimum is nearly achieved on chips whose arithmetic units are connected in a mesh-type pattern.

The relation $AT^2 > N^2/16$ bounds the performance of any chip of area A that computes an N point DFT in time T. At least two designs come close to this limit: those with either a perfect shuffle or a mesh-type interconnection pattern.

Other approaches.

Area and time considerations have been studied previously. This paper's model is based on the assumptions of [Mead 78], who found an optimal value for the area-time product in VLSI memory chips.

Two studies have been made of the area requirements of interconnection patterns. A random graph on N nodes was found to require $\Theta(N^2)$ area, using a model suggested by the wiring of a printed circuit board [Sutherland 73]. The problem of embedding bipartite graphs in the plane was explored by [Cutler 78].

The theory of cellular automata [von Neumann 66] can be used to elucidate some aspects of the area-time tradeoff. However, cellular automata have only nearest-neighbor connections, making them an inconvenient vehicle for the study of the effect of different interconnection pattterns on area and time.

The existence of a space-time tradeoff in a computation of the DFT has been demonstrated [Savage 77]. Unfortunately, it is unclear what connection can be made between the space needed to store intermediate results and the silicon area needed to implement a slower DFT.

Outline.

Section 2 details the VLSI model of computation. In Section 3, a graph-theoretic quantity is defined that will be used to derive lower bounds on area (in Section 4) and time (in Section 5) for chips that perform DFTs. Section 6 concludes with the main result, that $AT^2 > N^2/16$ for any chip with area A that computes an N point DFT in time T.

2. The VLSI Model

The useful area of any VLSI layout can be assigned to one of two functions, either processing or communication. In a loose sense, "processing silicon" combines or stores information while "communicating silicon" transmits it unchanged.

PEs, Wires.

The loci where processing occurs are called Processing Elements, or PEs. The loci of communication are called interconnections, or wires.

Words.

The basic chunk of information considered in this paper is a word. Words are in one-to-one correspondence with elements of the finite commutative ring over which the DFT is defined. To avoid unedifying detail, the word length (in bits) is treated as a constant in this paper.

Wires, units of length and time.

A wire has unit width and transmits a word from one end to the other in unit time. If the transmission is performed bit-serially, the unit of time is proportional to the word length in bits. If the transmission is word-parallel, the unit of length is proportional to word length.

PEs.

A PE contains at most one word of storage. If larger PEs are envisioned, they must be decomposed into word-sized PEs with connecting wires.

A PE may use words from any number of connecting wires to update its own word in any way, but it may take only a constant amount of time to do so. The functions performed by a PE are thus in $R^k \times R$, where R is the ring used to define the DFT.

A PE may output words onto any number of connecting wires, but may only output its own word or any of the words it received in the last time unit. There is thus no bandwidth limitation on PEs: they may act as many-to-many switches.

There are constants a and t such that no PE occupies more than a units of area nor takes more than t units of time to perform an update on its word.

Nexi.

Wires deliver words to and from a nexus associated with each PE. There is exactly one PE per nexus. Communication between a nexus and its PE is free, costing no area or time.

Each nexus is square in aspect, with side d if d wires connect to it. This ensures that there is more than enough edge length on the nexus to accomodate all connecting wires.

The square shape does entail a large area charge for a nexus of large degree, but in this case its associated PE could be very powerful. It would be permissible, for example, for a PE with degree N to act as a "big switch", permuting N words at a time. Charging $\Theta(N^2)$ area allows enough room for a cross-point; fancier switches with greater delay but less area may be built from small crosspoints. (If a PE with degree N is not a "big switch", it should be decomposed into smaller PEs with lesser degrees. For example, a fan-out of N can be acheived with a tree of $\Theta(N)$ constant degree PEs for a total of $\Theta(N)$ area and $\Theta(\log N)$ delay.)

Input, Output PEs.

An input PE initially contains one of the N elements of the vector that is to undergo Fourier transformation. There are N input PEs.

An output PE will eventually contain one of the N elements of the result of the DFT. The N output PEs are not necessarily distinct from the input PEs.

Wire layout.

Wires are laid out on a grid with unit spacing. Restricting wires to run along grid lines assures that unit width is available for each line if the grid is physically realized with two layers of silicon. One layer is devoted to the "x" direction, one to the "y".

Wires may bend at grid corners. This corresponds to a connection between the two layers of silicon.

At grid corners, wires may cross at right angles with no effect on each other's signals or timing. This corresponds to insulating the two layers of silicon from each other.

Wires may not connect or fan-out at grid corners! A PE is required at the confluence of wires, to avoid non-constant fan-out without area or time cost.

Justification of the model.

The VLSI model of computation is justified if its area and time charges are appropriate, and if it allows for all possible designs.

Area charges can be simply stated: wires have unit area per unit length, and nexi have area equal to the square of their degree. The area of PEs is largely ignored, leading to weaker but valid lower bound results. Nexus area is the only arguable charge, and it is unimportant in the usual case of constant fan-out at each PE. In this case, nexus area is only a constant multiple of wire area.

Time charges are even simpler than area charges. Wires transmit one word per unit time. This is consistent with the common VLSI design strategy of matching drivers to their wires. (On-chip propagation speed is limited by wire capacitance, not the speed of light.) PE delays are ignored, which of course cannot invalidate the lower bound results of this paper.

The VLSI model of computation applies to all possible designs, according to the following correspondence scheme. Any chip that performs a DFT must have at least N words of storage for the input vector. The elements that store these values are the N input PEs. Similarly, N output PEs may be identified, even though they may not be distinct from the input PEs. Next, one may identify the wires on the chip by noting the silicon used to carry information from input PEs to other registers on the chip. Finally, the nexi are the switching or gating elements that connect wires to storage elements (PEs) or to other wires.

There is one critical assumption built into the VLSI model, that the information about the N input words is initially localized. That is, each word is stored in a compact region (a PE) of the chip. This assumption is necessary to ensure that the DFT involves some computation, for otherwise one would consider the output words a legitimate initial encoding of the input vector. A similar argument can be made for requiring each of the output words to be stored in its own PE. Localization of the input and output PEs ensures that their nexi are also localized, so that there is indeed exactly one nexus for each PE.

The choice of the word as the basic unit of information is also defensible. Recall that wires of unit width transmit one word in unit time. Wires (and PEs) of smaller capacity are clearly conceivable, and should have fractional width or fractional delay to be true to VLSI implementation costs (for bit-parallel or bit-serial transmission, respectively). The introduction of such fractional capacities would only obscure the results of this paper, not invalidate them. None of the proofs depend upon the integral nature of the degree of a nexus or of the information capacity of any wire.

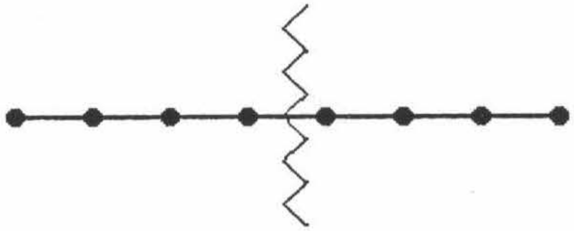
The description of the VLSI model of computation is now complete. It will be seen in the sequel that it is the pattern of interconnections, not the "programming" of PEs, that limits the speed or magnifies the area of a chip.

Interconnection patterns will be analyzed with the aid of the graph-theoretic definitions developed in the next section.

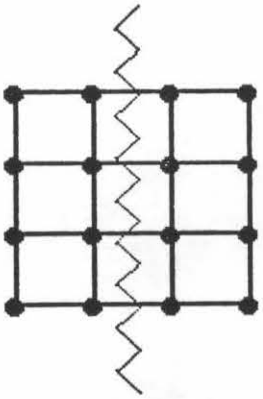
3. Minimal Bisection Width

The minimal bisection width of a graph is, informally, the number of cuts needed to slice it in half. In other words, it is the smallest number of edges whose removal disconnects one half of the vertices from the other.

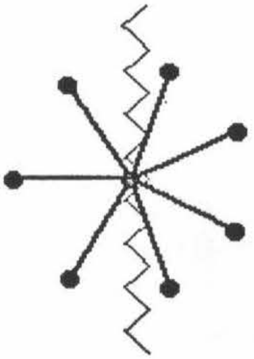
For example, the minimal bisection width of a linear graph of N nodes is 1:



The minimal bisection width of a mesh of N nodes is $N^{1/2}$:



The minimal bisection width of a star of *N* nodes is *N*/2:



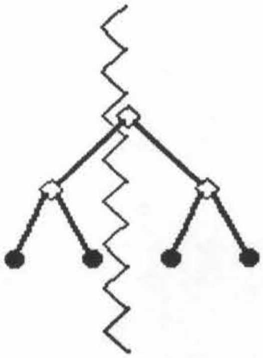
The minimal bisection width of a graph with an odd number of nodes is defined by relaxing the bisection requirement slightly. A bisection of a graph of 2*N*+1 nodes splits it into two disconnected subgraphs of *N* and *N*+1 nodes.

Formally, the minimal bisection width of a graph G = (V,E) is w iff w is the minimum integer satisfying the property that there is a bisection $V = H_1 \cup H_2$ with $H_1 \cap H_2 = \varphi$, $|H_1| \leq |H_2| \leq |H_1|+1$, and $|\{(u,v)\epsilon E: u\epsilon H_1, v\epsilon H_2\}| = w$.

Minimal bisection width of a subgraph.

The minimal bisection width of a subgraph is defined in the following fashion for each subset $U \subseteq V$ of the vertices of a graph $G = (V,E)$. The minimal bisection width of U in G is the smallest number of edges in E whose removal disconnects $H_1 \subseteq U$ from $H_2 \subseteq U$, where $|H_1| \leq |H_2| \leq |H_1|+1$, $H_1 \cup H_2 = U$, and $H_1 \cap H_2 = \varphi$.

The minimal bisection width of the leaves in a binary tree is 1:



In general, it is difficult to compute minimal bisection widths: the problem is

NP-complete, in fact [Garey 74]. Fortunately, it is enough to know that every graph has a set of edges that realizes its minimal bisection width.

The following sections will derive bounds on area and time for any VLSI design. A graph will be associated with each design, defining a minimal bisection width, w. Lower bounds of $w^2/4$ and $N/(2w)$ will be found on area and time respectively. Thus $AT^2 > N^2/16$.

4. Area

The total area occupied by a VLSI design is the sum of the areas of its PEs, wires, and nexi. A lower bound on wire and nexus area is derived in this section. Inclusion of PE area can at most affect the result by a constant factor, since there is one nexus per PE, and PEs have area bounded by a constant.

Associate with each VLSI design the following graph, G. Each nexus is a vertex. Each wire connecting two nexi is an edge between corresponding vertices. Denote by I the subset of vertices that are nexi of the N "input PEs".

Let w be the minimal bisection width of I in G.

Theorem 1: The area occupied by the wires and nexi of a VLSI design that corresponds to a graph of width w is greater than $w^2/4$.

Proof.

Orient the VLSI design in a Cartesian space. Consider lines of the form $x=a$. Each will separate the nexi of the N input PEs into three subsets: those to the left, split by, or to the right of the line. Denote these sets by L, S, and R. Find a value of a for which $|L| + |S| \geq N/2$ and $|R| + |S| \geq N/2$ (such an a exists by monotonicity). If $|S| = 0$, the line $x=a$ would define a cutset of I in G, hence it would cut at least w edges. Otherwise, split S into two subsets, S_1 and S_2, such that $|L| + |S_1| = |R| + |S_2| = N/2$. Remembering that a nexus of degree d occupies a square of side d, each nexus in S_1 may be considered to lie wholly to the left of the line $x=a$ (the wires connecting to the nexus on the right side of $x=a$ may be "continued" right up to the $x=a$ border). So the bisecting line still cuts at least w wires.

The line $x=a$ is said to account for the w square units of area of wire and nexus that lie within 1/2 unit distance of it.

Next, consider "zig-zags" of the form $\{x=a-1$ for $y \geq b$, $x=a+1$ for $y \leq b$, and $y=b$ for $a-1 \leq x \leq a+1\}$. Using the a determined above, vary b to obtain another bisection of the input nexi. Only two wires may cross the horizontal ($y=b$) segment, so that its vertical sections must cross at least w-2 wires.

This zig-zag accounts for the w-2 square units of wire and nexus that lie within 1/2 unit of its vertical sections.

In all, $\lfloor w/2 \rfloor$ zig-zags may be drawn, each of the form $\{x=a-k$ for $y \geq b$, $x=a+k$ for $y \leq b$, and $y=b$ for $a-k \leq x \leq a+k\}$. Each accounts for w-2k area.

The total area of wire and nexus is thus

$$\sum_{0 \leq k \leq \lfloor w/2 \rfloor} w-2k \;>\; w^2/4.$$

5. Time

The speed of a VLSI design may be limited either by the time taken by arithmetic operations or by the time taken to get intermediate results to the proper place. It is the latter that generally places the stricter limit on large VLSI designs for the DFT, by the following argument.

As noted in section 2, a PE can take at most t units of time to update its word. The information used to perform this update must have taken at least one unit of time to reach the PE. Thus, within at most a factor of t, it is the routing of intermediate results rather than the arithmetics performed on them, that limits the timing of VLSI designs.

The following theorem places a lower bound on the time required to compute a DFT. Its proof is based on a consideration of the amount of information that must be transmitted during the course of any computation of a DFT.

> **Theorem 2:** Associate a graph with each VLSI design as in Section 4. At least N/(2w) time is required to compute an N point DFT on a VLSI design that corresponds to a graph of width w.

Proof sketch.

The theorem is a consequence of the following property of the DFT. View the DFT as a matrix-vector multiplication

$$\tilde{y} = A\tilde{x}$$

Bisect the N elements of $\tilde{y}$ into two equal-sized subsets $\tilde{y}_1$ and $\tilde{y}_2$. Partition the elements of $\tilde{x}$ into *any* two disjoint subsets $\tilde{x}_1$ and $\tilde{x}_2$. This decomposes the DFT problem into two subproblems of the form

$$\tilde{y}_1 = A_{11}\tilde{x}_1 + A_{12}\tilde{x}_2$$
$$\tilde{y}_2 = A_{21}\tilde{x}_1 + A_{12}\tilde{x}_2$$

It is possible to show that

$$\mathrm{Rank}(A_{12}) + \mathrm{Rank}(A_{21}) \geq N/2 \qquad (1)$$

This follows (non-trivially) from an argument on the number of zeros of a polynomial of degree $N/2$. If k of the elements of $\tilde{x}_1$ are among the first $N/2$ elements of $\tilde{x}$, then $\mathrm{Rank}(A_{12}) \geq k$. In this case, $\mathrm{Rank}(A_{21}) \geq N/2 - k$.

This property of the DFT holds for any bisection of the elements of $\tilde{y}$. In particular, it holds for the bisection of the output PEs that realizes w, the minimum bisection width.

Choose $\tilde{x}_1$ to be the set of input PEs that are included with $\tilde{y}_1$ in the bisection of the output PEs. The computation of $\tilde{y}_1$ will require k words of information about $\tilde{x}_2$, if $\mathrm{Rank}(A_{12}) = k$. A similar argument holds for $\tilde{y}_2$, so that $N/2$ words of information must pass over the w wires that separate $\tilde{x}_1$ from $\tilde{x}_2$.

It takes at least $N/(2w)$ time to pass $N/2$ words over w wires, hence the theorem.

6. Conclusion

Theorems 1 and 2 can be immediately combined to give the main result of the paper.

> **Theorem 3:** If a VLSI design with area A computes an N point DFT in time T, then $AT^2 > N^2/16$.

This theorem is nearly tight for at least two cases. Using a mesh-type interconnection pattern, $O(N)$ space and $O(N^{1/2})$ time is sufficient to compute the DFT. Using a perfect shuffle interconnection, $O(N^2/\log^{1/2}N)$ area and $O(\log N)$ time is sufficient [Pease 68, Thompson 80].

The following theorem is of interest if a particular cost function is to be minimized.

Theorem 4: For any VLSI design with area A and time T, and for any non-negative $x \le 2$, $AT^x = \Omega(N^{1+x/2})$.

Proof.

The area of any VLSI design with N input PEs must be $\Omega(N)$ since each PE has a nexus of at least unit area. By Theorems 1 and 2,

$$AT^x = \Omega(N + w^2/4) * (N/2w)^x$$

Without loss of generality, let $w = N^{1/2+\epsilon}$. Then

$$AT^x = \Omega(N^{1+x/2+\epsilon x} + N^{1+x/2+\epsilon(2-x)})$$

Since $0 \le x \le 2$, the second term increases with ϵ while the first term decreases with ϵ. Clearly, $\epsilon=0$ achieves the minimum value, hence the theorem. ■

From the proof of Theorem 4, it is clear that the optimal design has $w=\Theta(N^{1/2})$, which corresponds to a mesh-type interconnection pattern.

A similar analysis may be performed for other problems, including matrix multiplication, Gaussian elimination, transitive closure, sorting, and permutation [Thompson 80].

Acknowledgements.

James B. Saxe formulated and proved equation 1. Leo J. Guibas and the XEROX Palo Alto Research Center gave guidance and generous support during the early phases of the research that led to this paper.

Mere acknowledgement can hardly discharge my debts to my thesis advisor, H. T. Kung. No one else could have given better hints at crucial moments, and I despair of ever finding a more selfless researcher.

References

[Aho 74] Aho, Alfred V., Hopcroft, John E. and Ullman, Jeffrey D.
The Design and Analysis of Computer Algorithms.
Addison-Wesley, 1974.

[Bonneau 73] Bonneau, Richard J.
A Class of Finite Computation Structures Supporting the Fast Fourier Transform.
Technical Report 31, Massachusetts Institute of Technology Project MAC, March 1973.

[Cutler 78] Cutler, M. and Shiloach, Y.
Permutation Layout.
Networks 8:253-278, 1978.

[Garey 74] Garey, M. R., Johnson, D. S., and Stockmeyer, L.
Some Simplified Polynomial Complete Problems
Proc. 6th Annual ACM Symposium on Theory of Computing, SIGACT, pages 47-63, 1974.

[Mead 78] Mead, Carver and Rem, Martin.
Cost and Performance of VLSI Computing Structures.
Technical Report 1584, California Institute of Technology Computer Science Department, 1978.

[Pease 68] Pease, Marshall C.
An Adaptation of the Fast Fourier Transform for Parallel Processing.
Journal of the ACM 15(2):252-264, April 1968.

[Savage 77] Savage, J. E. and Swami, Sowmitri.
Space-Time Tradeoffs in the FFT Algorithm.
Technical Report TRCS-31, Brown institution, August 1977.

[Sutherland 73] Sutherland, Ivan E. and Oestreicher, Donald.
How Big Should a Printed Circuit Board Be?.
IEEETC C-22:537-542, May 1973.

[Thompson 80] Thompson, C. D.
A Complexity Theory for VLSI.
PhD thesis, Carnegie-Mellon University Computer Science Department, 1980.

[von Neumann 66] von Neumann, J.
The Theory of Self-Reproducing Automata.
Univ. of Illinois, Urbana, Ill., 1966.