SUB-NANOSECOND ARITHMETIC

Michael J. Flynn, Giovanni De Micheli,
Robert Dutton, Bruce Wooley, and Fabian Pease

Technical Report: CSL-TR-90-428

November 1986

This research was supported by the National Science Foundation,
under contract MIP 88-22961.

SUB-NANOSECOND ARITHMETIC

by
Michael Flynn, Giovanni De Micheli, Robert Dutton,
Bruce Wooley, and Fabian Pease

Technical Report CSL-TR-90-428
May 1990

Computer Systems Laboratory
Departments of Electrical Engineering and Computer Science
Stanford University
Stanford, California 943054055

Abstract

The SNAP (Stanford Nanosecond Arithmetic Processor) project is targeted at realizing an arithmetic processor with performance approximately an order of magnitude faster than currently available technology. The realization of SNAP is predicated on an interdisciplinary approach and effort spanning research in algorithms, data representation, CAD, circuits and devices, and packaging. SNAP is visualized as an arithmetic coprocessor implemented on an active substrate containing several chips, each of which realize a particular arithmetic function.

Key Words and Phrases: Subnanosecond arithmetic, floating point, active substrate, high speed interconnect, microchannel cooling, CORDIC, wave pipelining, microscopic contacts, di/dt noise, arithmetic algorithms, BiCMOS, computer generated layout.

Contents

1 Introduction

The SNAP project is targeted at realizing an arithmetic processor with performance approximately an order of magnitude faster than currently available technology. The realization of SNAP is predicated on an interdisciplinary approach and effort spanning research in algorithms, data representation, CAD, circuits and devices, and packaging. SNAP as visualized represents an arithmetic coprocessor implemented on an active substrate containing several chips, each of which realize a particular arithmetic function (Figure 1). The target performance for each of the arithmetic functions is:

Integer add	0.5-1 ns
Floating point add	1.5-3 ns
Floating point multiply	3-5 ns
Divide and square root	10–15 ns

The active substrate is an important aspect of the project, as it allows the various functions to be separately partitioned and realized relatively independent one from another. It also realizes the extremely important (high integrity) transmission lines and chip to chip contacts, which are essential to 1-nanosecond pulse propagation.

The faculty project team consists of:

M. J. Flynn	— SNAP architecture
	— Algorithms, pipelining, clocking
G. De Micheli	— Computer aided design
R. Dutton	— Devices/technology
B. Wooley	— Circuits/technology
F. Pease	— Systems integration

2 Status

Despite some fabrication difficulties, overall progress has been commendable. In this section we summarize the various accomplishments and problems relating to various aspects of the project. Later in the report there is a somewhat more detailed progress summary.

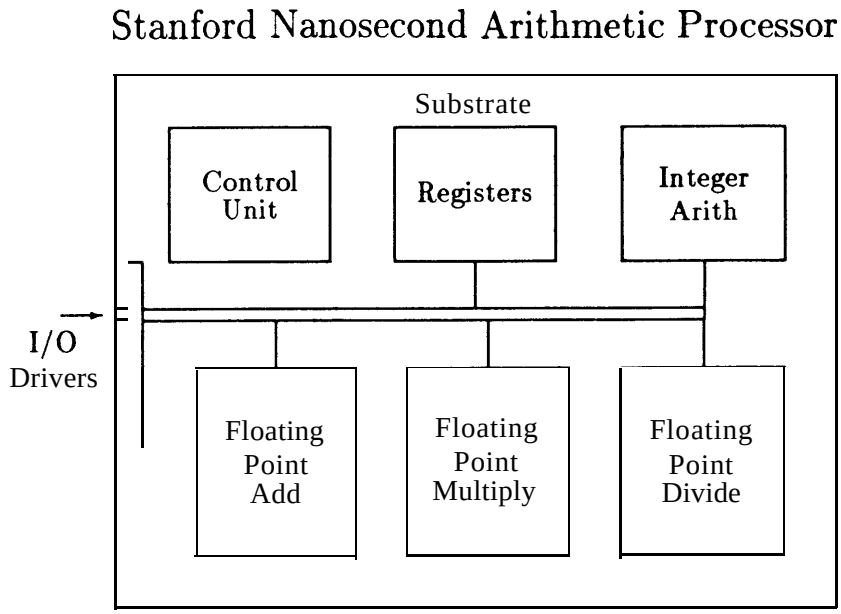

Figure **1:** Stanford nanosecond arithmetic processor (SNAP).

2.1 Circuits and Packaging

With a goal of one nanosecond pulse rate across an active substrate, it is clear that aspects of the package become severe limitations. In earlier work Pease [1] developed techniques to significantly improve the cooling density of an active substrate (Figure 2). Densities of 100 watts/sq. cm can be cooled by liquid cooling with laminar flow through microchannels in the base of the substrate. Active devices mounted on the substrate are then thermally connected via a thermally conductive liquid which bonds the active chip to the substrate by surface tension. In this project, research is focused on interchip connection and associated switching and transient noise. We are currently in the process of evaluating high density interconnects made using laser-induced planarization and micro-spring contacts between the chip and the substrate. We believe that the micro-contact and its associated noise and signal transmission characteristics will ultimately determine the performance limits of the SNAP processor. Signals cannot reliably be propagated with rise times less than the switching transients that are caused in the power and ground lines.

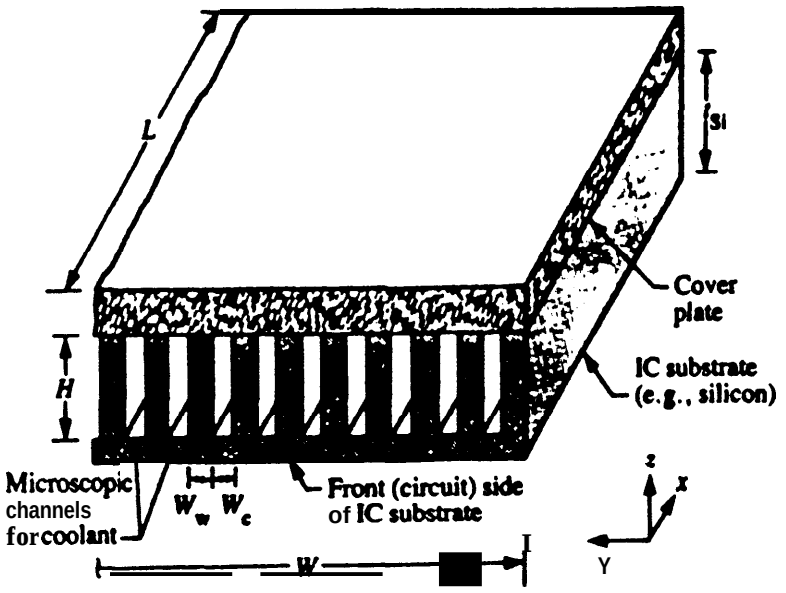

Figure 2: Schematic view of a microscopic heat sink incorporated into an integrated circuit chip. For a 1 x 1 cm^2 silicon chip, the optimal dimensions are approximately $W_w = W_c = 57\ \mu m$ and $Z = 350\ \mu m$. The specific heat conductance is more than 10 $W/K\text{-}cm^2$ (from [1], reproduced with permission).

In the packaging area we are currently devoting a significant amount of effort to understanding the design and characterization of microcontacts so as to develop a high integrity chip to substrate connection. Associated with this is the development of a low loss, high speed transmission line and associated active devices that can be used on the substrate to implement driver and receiver interfaces and hence the pipeline data paths for the overall SNAP.

We currently envision that the circuitry on the substrate will be primarily current-switching bipolar circuits, while the functional units are generally expected to be realized in either bipolar or BiCMOS circuit technologies, depending on density requirements.

2.2 Functional Units and Algorithms

In this area we have continued our work on addition and multiplication, and have submitted an ambitious floating point multiplier design for fabrication. Because of the advanced nature of the technology used in the fabrication process, production has been extremely slow and so far parts are unavailable. On a simulated basis, the bipolar floating point multiplier design should have total execution delay under 10 ns. In the event that the current fabrication fails, we expect to recycle much of the elements of the design with an alternate fabrication process.

In the area of addition, we have been able to follow up our earlier success of bipolar addition with an improved algorithm applicable to CMOS addition. With this algorithm, it is expected that 32-bit addition could be accomplished in 1.5 nsec using 0.5 micro CMOS.

Another area of important promising developments is the techniques being developed for wave pipelining. The goal of this project is to move signals from chip to chip at or under a 1 nsec rate, and to move operands through at least some of the functional units at approximately the same rate.

It is clearly physically impossible to do this unless one has a method of relying on the intrinsic storage of the transmission and processing media. The limits on cycle time (δt) are:

$$\text{At} = P_{\max} - P_{\min} + t_g$$

where $P_{\max}$ is the maximum delay through a particular section and $P_{\min}$ *is* the minimum delay through that section. t_g is set up and hold times related

to register latching. The goal of wave pipeline techniques is to **pad out the** minimum paths with CAD **assistance so that extremely high clock rates can** be achieved across transmission **media and within processing units. The** CAD tools in this area seem quite robust and may **be used over the next** year to validate much of both the **theory and the tool development effort.**

The purpose of SNAP is not only to realize basic high speed arithmetic operations, but to integrate these operations into representative higher level mathematical functions that can be self contained on **the SNAP** processor. Thus, SNAP becomes its own floating point processor with the ability to handle some basic signal processing and matrix manipulation operations. The more operations kept internal to the SNAP, the faster these operations can be performed. Thus, various CORDIC matrix arithmetic and root finding operations are being studied to better understand the type of support they would require on a SNAP processor.

2.2.1 High Level Functions

To support these studies work on mathematical and software tools, including symbolic computations, was carried out. Two complementary approaches were pursued to improve high level function performance: the first speeds up a given set of hardware primitives and the matching arithmetic routines, the second attempts to find better primitives, hence jointly optimizing primitives and arithmetic routines. The first approach serves as a baseline to explore incremental improvements. The second approach attempts to achieve a more global optimization; it is in part motivated by a "top down" functional **point** of view, with an eye towards compiler support.

A number of applications **were pursued with several objectives in mind:** to study the process of matching/mapping algorithms, architectures, and arithmetic, to derive "natural" primitives, and to obtain statistical characterizations. Arithmetic studies included adders, multipliers, and symbolic computations (CORDIC algorithms, CMOS adders [Quach90b]). **Device,** circuit, and architecture modeling was a second set of topics. **Among the** main tools to evaluate the performance of the various alternatives are **statistical** and analytic models for such parameters as speed, area, and power [6].

Higher level functions were also explored with the concept of Information Preserving Transformations (IPT), e.g., CORDIC functions, and exchange

gates. This is related to the problem of writing programs or translating them to 'natural' primitives. A number of these primitives have been studied with semi-automated algebraic and symbolic tools. General synthesis procedures have been developed for the algebraic level, logic level, and Finite State Machines (FSM).

2.3 Computer-Aided Design

The CAD effort in the SNAP project has been crucial in several areas. In earlier efforts we have developed techniques for "vendor-independent design," that is, the creation of design files that can be moved from process to process within a single vendor house, or a design from vendor to vendor without a great deal of redesign. When a particular process proves intractable or infeasible (as perhaps in the case of our floating multiply design mentioned earlier), the tools allow us to retarget the design to alternate processors and fabrication clines.

CAD is of pivotal importance to our study of wave pipelining and wave pipelining techniques. We use a two phase technique to achieve wave pipelining. The first phase involves a rough tuning of the paths so that the minimum paths have delay within the same general range as the maximum delay. The second phase refines the match so that a minimum clock time can be achieved. Wave pipelining has illustrated the limitation of much of our current circuit technology. With a conventionally clocked circuit, the only delay that is important is the maximum delay. In a wave pipelined circuit, the skew between maximum and minimum delay is of vital importance. A circuit that has a asymmetric rise and fall delay are much less useful than circuits with balanced rise and fall delay. Typically CMOS circuits have a large delay skew and do not at this time appear useful or promising for wave pipelining. However, one active area of research is to find or design BiCMOS circuit realizations that achieve a better control over delay skew.

In another area, CAD has provided a significant insight into the arrangement of counters for the reduction of partial products in multiplication. These tools allow a rapid generation of the partial product reduction trees based on various assumptions regarding the structure of the counter, that is, used as the basis for partial product reduction.

3 Progress Summary

3.1 Circuits and Packaging

3.1.1 General Approach (B . Wooley)

In the architecture currently being examined for the implementation of arithmetic processors with execution times on the order of one nanosecond or less, the achievable performance seems likely to be governed by the delays associated with fetching operands from, and returning results to, a register file. Our initial research into circuits and packaging has therefore focussed on these operations.

For execution times below 10 nanoseconds, the delays associated with interchip interconnections become an increasingly severe limitation. Advanced passive packaging technologies attack this problem through scaling of the interconnections. However, there typically remain severe impedance mismatches, and therefore delay penalties, at the chip-substrate boundaries. In this program considerable effort is being devoted to the exploration of an active substrate approach to system integration. With this approach, processor and memory chips are attached to a silicon substrate containing both low-loss, high-speed transmission lines and active devices that can be used to implement driver and receiver interfaces as well as pipelined data paths.

To demonstrate the feasibility of the active substrate approach, we have begun the design of a prototype system that combines a functional processor element with a register file. Custom interface circuits are being implemented at the register-to-substrate boundary, but not at the processor-to-substrate interface. In this way, it is possible to examine the relative benefits of active and passive packaging approaches within a single demonstration vehicle.

The technology for the active substrate combines a BiCMOS integrated circuit technology with special interchip interconnection metallization that provides lower loss transmission lines than conventional VLSI metallization. To achieve the highest possible speeds within the active substrate, low voltage swings must be maintained throughout the critical paths. Current-switching bipolar circuits are considered essential to this purpose. However, as in the processor elements themselves, it may be necessary to extensively pipeline the interchip data flow. The availability of MOS circuits within

the substrate, for both dynamic and static storage elements, addresses this need.

Important design parameters for the register **file** itself include the number of registers (at least 16), the register width (64 or 80 bits), and the number of independent read, write and read/write ports. All of these parameters depend intimately on the instruction set architecture and the pipelining strategy adopted for the overall system.

As part of the active substrate demonstration vehicle, we are examining approaches to reducing the access time of the register file. In particular, we are investigating the use of CMOS Storage Emitter Access (CSEA) circuit techniques in this application. We have already used these techniques to achieve access times of less than **4** nanoseconds in both a two-port static memory and a translation lookaside buffer that combines a content addressable memory with a static RAM.

3.1.2 Microcontacts and Noise (R. Dutton and F. Pease)

System performance is limited by our ability to distribute signals and power rapidly and reliably among the IC's comprising the system. Increasing the density of the chip-to-chip interconnect system is clearly highly desirable since, for a given system complexity, the length of each interconnect will be proportionately shortened and delay time reduced. **A** variety of novel micromechanical approaches have been reported (both from Stanford and elsewhere) that have the promise of ameliorating this problem. Such approaches include high density interconnects made using laser induced planarization, micro-spring contacts between chip and substrate and the use of silicon substrates to allow good heat sinking, high resolution fabrication of interconnect structures and the possibility of incorporating active circuitry into the substrate and so transfer the communication function from the chip to the substrate. One serious and poorly understood factor is the noise (by which we also mean internally generated interference) that can arise from a number of sources. **Two** that we are initially investigating are:

1. Microscopic pressure contacts between chip and substrate.
2. Switching transients that cause rapid current fluctuations in inductive power and ground lines ('Delta-I noise').

Pressure contacts have two applications in the engineering of high-speed complete systems. The first is to allow replacement of chips or multi-chip modules both during system assembly and during maintenance and trouble-shooting. The second is for the high speed testing of chips prior to dicing and packaging. At present contact pads consume a significant fraction of the total area of high-speed die. Miniaturization of the contacts is clearly desirable along with the continued miniaturization of the rest of the circuitry. However, smaller pressure contacts can lead to increased noise and resistance. Our research is aimed at studying the electrical and mechanical behavior of such contacts down to the atomic level using such revolutionary new tools such as the scanning tunneling microscope (STM) and the atomic force microscope (AFM) and combinations thereof.

To study noise in microscopic contacts we have set up a scanning tunneling microscope (STM) along with a spectrum analyzer to characterize noise spectra of the STM current under a variety of conditions and materials. In this way the contact can be modeled as an array of STM's operating under this variety of conditions; the variable parameters include tip and target material, applied voltage, current and separation.

Our preliminary experiments indicated in general the '1/f' (inverse frequency) spectral character of the noise (Figure 3) but the STM proved unsuitable for this work because the current per se is used to control the tip-to-target separation. Our newly designed apparatus is essentially a combined atomic force microscope (AFM) and STM (Figure 4).

Since the only feedback-controlled variable in a standard STM is the tunneling current, this instrument is unsuitable for directly observing physical contacts where conduction is not exponentially dependent on tip-sample separation. When an STM tip touches the surface, the contact pressure is an unknown function of current through the interface. The recently developed atomic force microscope (AFM) is more suited to making a controllable contact. The AFM uses a measurement of cantilever bending to produce an accurately controlled force between a tip and a surface, and is capable of atomic resolution microscopy. The absolute position of the tip and substrate may be measured independently, allowing the tip-surface penetration depth to be measured as well.

A modification of the standard imaging AFM is proposed that uses a known and controllable force on a conducting tip to form a well-defined contact area on the substrate. With optical sensing of cantilever deflection, the contact

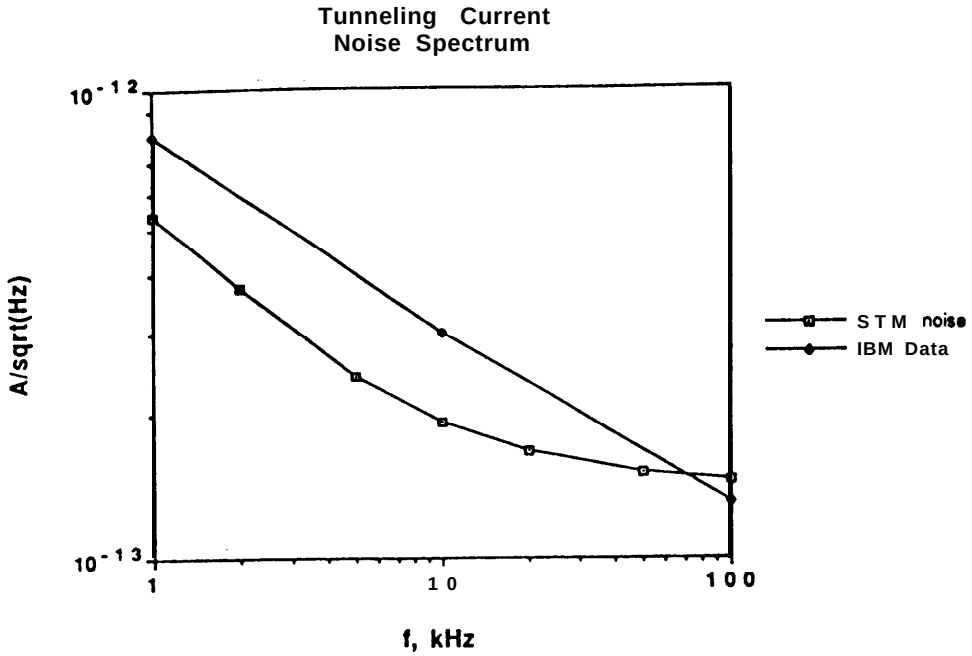

Fig. 3 This STM noise spectrum was taken with a Pt-Ir tip tunneling to a graphite substrate. Measured 1/f noise corresponds to that observed on graphite with similar equipment by D. W. Abraham et. al., Appl. Phys. Lett. 53 (16) 1503 (1988)

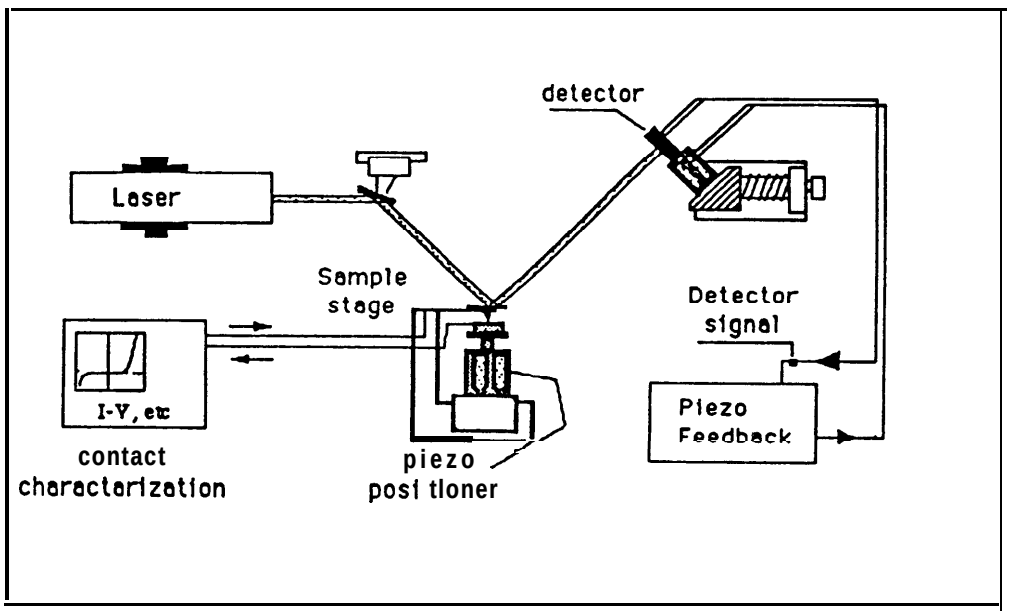

Fig. 4. Proposed apparatus to evaluate contact noise over a wide variety of conditions. The system is essentially a combination of a scanning tunneling and an atomic force microscope.

is electrically isolated from force and position sensing circuits. This contact could then be characterized electrically as a function of applied pressure, contact area, and tip and substrate materials. A review of the existing AFM literature has shown that such a modified force microscope is very promising for making accurate contact measurements on a nanometer scale. The AFM has already been applied to measuring mechanical properties of materials on a nanometer scale [2], making possible much more sensitive measurements than are available from traditional microindenters. With very sharp tips (etched tungsten tips for use in STM have been reported [3] with end radii of 7.5 ± 3 nm, and atomically sharp tips have been achieved with ion milling), contact pressure is limited only by tip and substrate hardness. With a tungsten tip and a nickel substrate, for instance, pressures of 105 bar may be achieved. In this limit, contact area is determined by applied force. Since AFMs have demonstrated atomic resolution, contact dimensions down to a single bridging atom may be possible.

The delta-I noise problem is being addressed initially by building up SPICE-based models of representative off-chip driver/transmission-line/on-chip receiver circuits. Preliminary results (Figure 5) indicate the severity of the voltage excursions on power and ground lines unless special precautions are taken. One encouraging finding is that the increase in voltage excursion resulting from increased numbers of simultaneously-switched gates is less than the straight-line prediction of elementary models (Figure 6).

3.2 Functional Units and Algorithms

3.2.1 Addition

32-bit high-speed CMOS adder (N. Quach)

We completed the design of a 32-bit high-speed adder. This CMOS adder uses a modified Ling approach [4] developed here at Stanford and has an addition time of 4 gate delays. Organizationally, the adder is divided into three 9-bit blocks and a 6-bit block. Each block is further divided into groups. A group size of 3 is used in the adder; hence, a 9-bit block has three groups and a 6-bit block has two groups. The local sum logic uses the conditional sum algorithm. The global carry lookahead circuitry uses a modified Ling approach.

Table 1 compares the adder (called SNAP adder in the table) with other

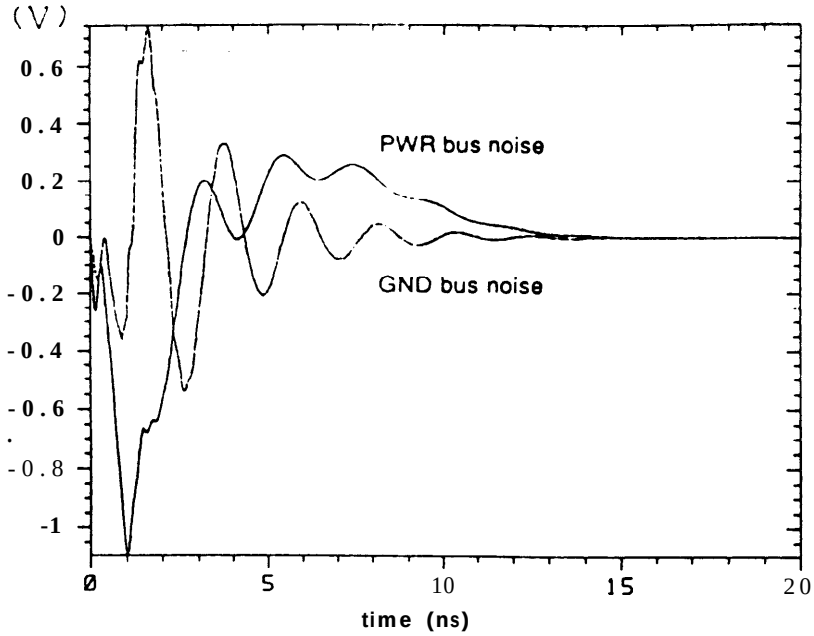

Fig.5 Power and ground bus noise for N=16 using a 10 pF decoupling capacitor.

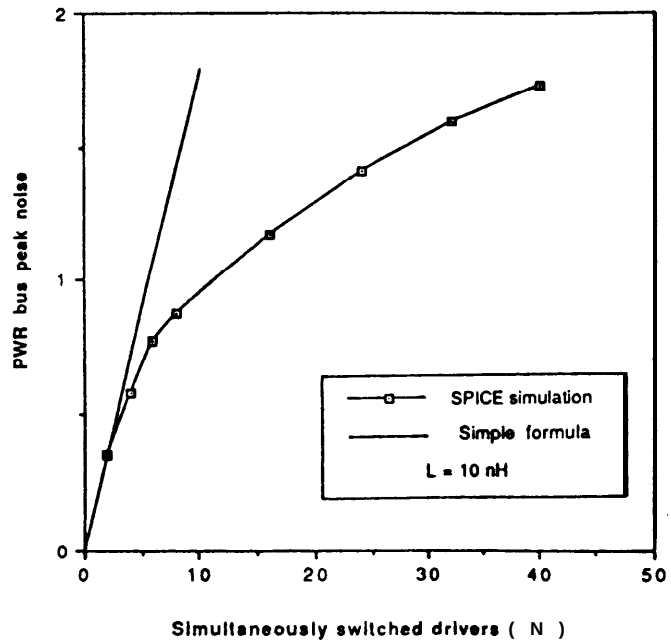

Fig. 6 Power bus peak noise vs. number of simultaneously switched drivers **(N).**
For large N the noise amplitude is less than the straight- line prediction.

Table 1: Comparison of present adder with others in terms of gate delay and number of serial transistors.

Adders	Gate Delay from c_{in} to S_{32}	Number of Serial Transistors from c_{in} to S_{32}
CLA	6	21
Condition-Sum Adder	5	16
Carry-Select Adder	5	18
MODL Adder	5	18
SNAP Adder	4	14

adders proposed in the literature in terms of gate delays and serial transistors in the critical path. The adder has the fewest number of gate delays and serial transistors. The Multiple-Output Domino Logic (MODL) adder reported by researchers at AT&T uses dynamic (precharge) techniques to speed up the adder. Our adder is a fully static one.

The number of serial transistors is the total number of transistors in the critical path of an adder that a signal must travel. Because an important factor determining the speed of a CMOS logic gate is the number of transistors in series from the output node to the power or ground node, reducing this number therefore speeds up the adder. Admittedly, the measure is relatively crude, but it does provide a quick way to evaluate the potential performance of an algorithm.

A 55 bit adder for floating multiply (G. Bewick)

A high speed 55 bit adder has been submitted for fabrication to Signetics/PRLS. This adder is suitable for use in an IEEE double precision floating point adder or multiplier. All logic for rounding, normalization, and exponent handling are included in this submission. Like the 32 bit adder, this adder was designed using bipolar ECL circuitry. There is a maximum of four levels of logic from any adder input to any adder output. Performance is expected to be around 5 nsec, from input pin to output pin, which makes it suitable for a 10 nsec double precision floating multiply.

Despite submission in March 1989, we are still waiting for complete fabrication. Due to internal difficulties (since the process is experimental), the outlook for fully functioning chips is not good. We hope to be able to recycle

much of the design to a different technology and with other chip vendors and also use it in the floating multiplier (see below).

11-bit CMOS exponent adder (N. Quach)

We completed the design and layout of an U-bit **CMOS** exponent adder for use in a BiCMOS 64 x 64 bit multiplier developed with Gary Bewick, and Paul Song in cooperation with *Signetics.* The adder features a low addition time and uses a modified Ling algorithm. Some new circuit ideas were also tested in this adder. The adder is now being fabricated by *Signetics* at Sunnyvale.

3.2.2 Algorithms (G. Bewick)

Significant progress has been made in the area of multiplication algorithms. An algorithm has been developed which has significant advantages over previously published or implemented algorithms. The algorithm is a variation of a 3-bit Booth algorithm, but does not require a time consuming full length carry propagate addition to compute the 3-times multiple which is normally required. The full length carry propagate add is replaced by a series of small length additions. These small adds (which occur in parallel) are significantly faster than the larger add. Overall, the algorithm should be faster than a conventional 2-bit Booth multiplication, using 20% less hardware.

Fast and efficient methods for implementing the small adds required by the new multiplication scheme are crucial. Since the additions are actually computing multiples (such as three times a number). significant hardware and delay savings are possible.

Floating point multiplication involves some kind of rounding of the final result. Proper IEEE rounding requires the generation of a "sticky" bit. A new scheme for efficiently generating the "sticky" bit has been developed. This scheme is particularly efficient when used with the multiplication scheme presented above. In fact, the extra overhead for computing the "sticky" bit is on the order of 2-3 gates, when used in conjunction with the new multiply scheme.

VLSI **area modeling** (N. Quach)

This work is concerned with the extension and verification of a chip area

model developed by Mulder [5]. The area model allows an area-based comparison of various on-chip buffers (e.g., cache vs. register file). With a few extensions, the area model is found to give less than **10%** error when verified against real caches and register files. Studying the effectiveness of various cache organization in reducing memory traffic as a function of area, rather than storage capacity, led to a significantly different set of tradeoffs.

3.2.3 Wave Pipelining (D. Wong)

Wave pipelining is a design met hod that can boost the pipeline rate of a system without using additional registers. In ordinary pipelined systems, there is one "wave" of data between register stages. When a new set of values is clocked into one set of registers, the values are allowed to propagate to the next set of registers before the first set is clocked again.

In contrast, wave pipelining is the use of multiple coherent "waves" of data between storage elements. This is achieved by clocking the system faster than the propagation delay between registers. In this method, the data values at the first set of registers are changed before the old data values have propagated to the next set of registers. The capacitance in the combinational logic circuit is being used to store values for pipelining.

Ideally, if all paths from input to output have equal delay, then the circuit's clock frequency is limited by rise/fall times, clock skew, and set-up and hold times of the storage elements. In practice, due to the above limits and variations in fabrication, clock frequency can be increased by a factor of 2 to 3 using the best available design methods.

An analysis of circuit technologies shows that CML and super-buffered ECL are well suited for designing circuits with uniform delay. Regular ECL is not as good, and static CMOS is substantially worse.

In this past year, we have completed a first prototype of the CAD algorithms and tools for rough and fine tuning circuits. These programs take an ordinary combinational logic design and balance the delays of all paths to a user-specified value. The circuit can then be wave-pipelined by applying new inputs at a higher rate than the propagation delay through the logic. **The** tuning programs also globally minimize the power consumption in the design for the given delay using a linear program.

3.2.4 **High Level Functions** (M. Morf)

The accomplishments during the reporting period include work on mathematical and software tools (including symbolic computation) and two complementary approaches to improve high **level function performance: the first** aims at speeding up a given set of hardware primitives and the matching arithmetic routines, the second is based on finding better primitives, hence jointly optimizing primitives and arithmetic routines.

The first approach serves as a baseline to explore incremental improvements. The second approach attempts to achieve a more global optimization; it is in part motivated by a "top down" functional point of view, with an eye towards compiler support.

Speeding up the execution of scientific or signal processing (DSP) programs under power and chip area constraints involves speeding up frequently used primitives (e.g., Floating Add, Mult, Div), reducing communication delays and decision bottlenecks, and speeding **up arithmetic routines. The con**straints make speeding up a process of matching and balancing hardware and software.

One of the main tools to evaluate the performance of the various alternatives are analytic models for such parameters as speed, area, and power. Statistics extracted from various hardware and software designs are used to obtain families of models from simple models to study trends, to more sophisticated models for accurate performance predictions. Examples of analytic device models that have better than 10% accuracy can be obtained for delay and power estimates.

In summary: multiple, independently optimized functional units lead to high speed primitives; however, the cost is increased power and chip area. Jointly optimized functional units tend **to obtain their speed with more** tightly coupled primitives; this allocates **resources such** as power and area to critical paths and relaxes the use of resources in less critical areas. Joint optimization typically equalizes the speed of different paths (a desirable property for wave pipelining for instance).

We are comparing a SNAP design based **on** independently optimized primitives, and a CORDIC function based design. **A classical** CORDIC based design-even if pipelined-has a high latency, hence **a balanced** reduction in the number of iterations per stage, while increasing the complexity of each stage (increasing total area) is necessary in order to reduce latency. This

leads to various hybrid schemes that we are exploring now.

The search for High Level Function primitives:

The joint optimization approach is explored from a higher level function point of view, with an eye towards compiler support; again the focus is to speed up program execution. The notion here is to map applications onto high level functions and to implement or decompose these functions into lower level "natural" primitive functions, taking advantage of algebraic and other structural properties of the high level functions. Statistical information of these functions (at all levels) can be used as one of the guidelines to judge the "naturalness" of a primitive.

The evaluation of sets of primitives involves transformation between such sets of primitives, in particular information preserving transformations (IPT's) between these sets, or alternatively transformations between programs that are using these primitives. Primitive IPT's are themselves good candidates for 'natural' primitives, due to the fact that any function can be imbedded in an information preserving function/transformation (isometries).

An interesting set of ALU functions, IPT's, are the modify accumulate type instructions in C-like notation: `+=`, `-=`, `*=`, `/=`, and `^=`, sign, and the less classical true-three-in/out: X-gate or conditional swap (intel486), and rotations/CORDIC functions (sin, cos, arctan, sqrt, `+^2`, `-^2`).

A number of applications are being pursued with several objectives in mind: to study the process of matching/mapping algorithms, architectures and arithmetic, to derive 'natural' primitives, and to obtain statistical characterizations.

Matrix arithmetic and root finding is an important and instructive area for studying the process of matching/mapping algorithms and architectures. Our past work has demonstrated that the family of (matrix) functions computed by CORDIC algorithms is a very well matched set of high level function primitives that could be termed 'natural' primitives. These multi-input-output primitives naturally also display parallelism and pipelining opportunities in matrix arithmetic.

Arithmetic-Adders, Multipliers, and symbolic computations: N. Quach's work on CMOS adders provides another basis to explore our tools to search for proper primitives. Our transformation approaches give an indication what primitive functions (e.g., exchange-gates) are prevalent.

Device, Circuit and Architecture Modeling: High performance VLSI designs require accurate models. Our symbolic and algebraic tools both improve designs and enable the process of defining proper primitives. An ideal example was the optimization of a master cell for a BiCMOS sea-of-gates [6]. Another recent success is a new method for identifying exponential parameters with a resolution down to at least 2 micro Volts; this accuracy would allow to decide between competing semiconductor theories to explain device behavior.

Writing programs or translating them to 'natural' primitives: We have explored a number of these primitives with semi-automated algebraic and symbolic tools. Some tools are translators to different representations; they can be used also as an indicator of the 'naturalness' of certain primitives. In addition, algebraic, logic and Finite State Machine (FSM) synthesis procedures have been developed and are in the process of being tested and compared with other design tools.

3.3 CAD (G. De Micheli)

The work on CAD-related techniques for the SNAP project has addressed mainly two topics in the past grant period: algorithms and implementations for designing high-speed wave pipelined circuits and algorithms for designing optimal Wallace trees for multipliers under technology constraints.

The research on wave pipelining was motivated by the desire of showing that wave pipelined arithmetic units, and, for example, multipliers, can run two to three times faster on a given technology by using an ad hoc design methodology. In particular, wave pipelined designs should satisfy the property that all paths from any input to any output have the same delay. In practice, the frequency of operations of wave pipelined designs is still limited by the clock skew and the timing characteristics of the registers. Therefore, the design objective is to make the mismatch of the I/O path delays small compared to these two quantities.

We have investigated two techniques for balancing the I/O path delays in a circuit, that are called fine and rough tuning respectively. The former takes advantage of the possibility of tuning gate delays in ECL/CML technology by adjusting their power setting. We have formally shown that there exists a class of circuits such that the delay balancing problem can be cast as a linear program and solved efficiently. We have implemented an interface

between a circuit specification in a standard **netlist** format and a linear program solver. We have shown that adequate balancing can be achieved, even though computing time grows over-linearly with the size of the circuits. Results are presented in the enclosed paper.

The rough tuning approach for balanced circuit design is motivated by the fact that not all circuits are fine tunable. Indeed, there are cases in which it is necessary to insert active tunable delay elements to add delay to short paths. This is necessary because the power setting of any ECL/CML gate has a fixed range. Algorithms for rough tuning have been investigated. In particular, the I/O path delay balancing problem can be cast as a loop balancing problem. We have shown, that it is sufficient to balance a spanning set of loops in the circuit and that an appropriate spanning set can be derived by adding links to a maximal-length spanning tree. Based on this approach, a feasible balancing can be computed in linear time. In addition, the corresponding set of additional padding elements can be made minimal by means of a repadding technique that shifts around the inserted elements. While such a technique can be solved by matching in polynomial time, an efficient solution can be obtained by casting the repadding problem in a linear program. We have implemented the rough-tuning technique using the latter approach and we have tested it successfully on several circuits. Results are presented in the enclosed paper.

The multiplier design project has the goal of achieving the fastest implementation under the constraints of a given technology. For this purpose, we have explored first the design of components of a multiplier, called counters, in BiCMOS technology. A Wallace tree using different types of counters, and different counters in isolation, has been designed and fabricated in an experimental BiCMOS technology. We plan on measuring and characterizing the components as soon as the chip comes back from fabrication. With the basic components characterized in terms of area and delay, we have questioned the problem of building a Wallace tree structure that best exploits the given technology. For this reason, we have developed algorithms and a program that generates a Wallace tree structure that uses a variety of basic components and that aims at minimizing the overall area under stringent timing constraints. The results have been surprising. Once counter and interconnect delays have been taken into account, the Wallace tree constructed by the algorithm uses a blend of different counters, in contrast to the usual approach where carry-save adders are used all over.

4 List of Reports and Papers

The following is a list of reports and papers that have been produced so far under this contract:

1. J. Mulder, N. Quach, M. J. Flynn, *An Area-utility Model for On-chip Memories and its Application,* Stanford Technical Report No. CSL-TR-90-413, Feb. 1990.

2. J. Mulder, N. Quach, M. J. Flynn, "An Area Model for On-chip Memories and Its Application," submit ted for review to the *IEEE Journal of Solid-State Circuits.*

3. Nhon T. Quach and Michael J. Flynn, *High-Speed Addition in CMOS,* Stanford Technical Report No. CSL-TR-90-415, Feb. 1990.

4. Nhon T. Quach and Michael J. Flynn, "High-Speed Addition in CMOS," submitted for review to the *IEEE Transactions on Computers.*

5. Geert Rosseel, *Bipolar Transistor* for *BiCMOS Circuits,* Ph.D. thesis, Stanford University. IC Lab Technical Report, March, 1990.

6. D. Wong, G. De Micheli, and M. Flynn, "Designing High-Performance Digital Circuits Using Wave Pipelining," *VLSI* '69, August 1989, Munich, West Germany.

   This article describes the concept of wave pipelining, its application to different circuit technologies, and the fine tuning algorithms.

7. D. Wong, G. De Micheli, and M. Flynn, "Inserting Active Delay Elements to Achieve Wave Pipelining," *ICCAD* '89, November 1989, Santa Clara, CA. (Extended version published as Stanford Technical Report CSL-TR-89-386.)

   This article provides an overview of wave pipelining and describes the rough tuning algorithms in detail.

5 References

[1] D. B. Tuckerman and R. F. W. Pease, "High Performance Heat Sinking for VLSI," *IEEE Electron Dev. Lett.* EDL-2, 126-129 (1981).

[2] "Measuring the nanomechanical properties and surface forces of materials using an AFM," N. A. Burnham, R. J. Colton, *J. Vac. Sci. Technol.* A **7 (4)** 2906 (1989).

[3] "STM on rough surfaces: Tip-shape-limited resolution," G. Reiss, J. Vancea, H. Wittmann, J. Zweck, H. Hoffmann, *J. Appl. Phys. 67 (3)* 1156 (1990).

[4] H. Ling, "High speed binary adder," *IBM Journal of Research and Development,* Vol. 25, No. 2 and 3 (1981).

[5] J. M. Mulder, "Tradeoffs in Data-Buffer and Processor-Architecture Design," Ph.D. thesis (Stanford Technical Report No. CSL-TR-87-345) Dec. 1987.

[6] A. El Gamal, J. L. Kouloheris, D. How, M. Morf, "BiNMOS: A Basic Cell for BiCMOS Sea-of-Gates," *IEEE 1989 Custom Integrated Circuits Conference,* San Diego, May 1989, pp. 8.3.1-4.