High Speed Adder Design using BiCMOS SiGe Technology

by

Paul Belemjian

A Thesis Submitted to the Graduate Faculty of Rensselaer Polytechnic Institute in Partial Fulfillment of the Requirements for the degree of

DOCTOR OF PHILOSOPHY

Major Subject: Computer Engineering

Approved by the Examining Committee:

______________________________
John McDonald, Thesis Adviser

______________________________
Khaled Salama, Member

______________________________
Mona Hella, Member

______________________________
Christopher Carothers, Member

Rensselaer Polytechnic Institute
Troy, New York
July 21, 2008
(For Graduation August, 2008)
CONTENTS

High Speed Adder Design using BiCMOS SiGe Technology .................................................. i
CONTENTS ................................................................................................................................ ii
LIST OF TABLES ........................................................................................................................ iv
LIST OF FIGURES ...................................................................................................................... v
ACKNOWLEDGMENTS ............................................................................................................ x
ACKNOWLEDGMENTS ............................................................................................................ x
ABSTRACT ................................................................................................................................ xi

1. Introduction ..................................................................................................................... 1
   1.1 Motivation ................................................................................................................. 1
   1.2 History ...................................................................................................................... 3
   1.3 Parallel vs Sequential ............................................................................................. 4

2. SiGe Bipolar Design ....................................................................................................... 9
   2.1 SiGe HBT .................................................................................................................. 9
   2.2 CML and ECL Design .......................................................................................... 10

3. General Adder Design ..................................................................................................... 19
   3.1 Parallel Prefixes ...................................................................................................... 19
   3.2 Effect of different fan-ins ...................................................................................... 21
   3.3 Pseudo-Carry Look-Ahead (PCLA) ...................................................................... 26
   3.4 Generalized pseudo-carry equations .................................................................... 29
   3.5 Look-ahead gate ..................................................................................................... 30

4. 7HP Adder Design ............................................................................................................ 35
   4.1 Comparison of 7HP Designs .................................................................................. 35
   4.2 7HP Carry Chain .................................................................................................... 42
   4.3 Design of ALU Carry Chain for 7HP Full ALU ....................................................... 47
   4.4 Design of ALU unit for 7HP Full ALU ................................................................. 56
5. 8HP Adder Design .................................................................................................................. 60
   5.1 8HP Adder Carry Chain Design ...................................................................................... 60
   5.2 8HP Adder Carry Chain Results ...................................................................................... 75
   5.3 Extracted 8HP Work ......................................................................................................... 79
   5.4 8HP Adder Carry Chain Redesign ................................................................................... 80
   5.5 8HP MEASURED RESULTS ....................................................................................... 82
   5.6 Further Optimization of 8HP circuits to maximize speed .............................................. 85
5. 32 BIT ADDER STRUCTURES .............................................................................................. 88
   6.1 8HP power savings design ............................................................................................. 90
6. 7HP Test Chip ........................................................................................................................ 92
   7.1 Adder Only Test ............................................................................................................. 97
   7.2 Register File Only Test ................................................................................................. 98
   7.3 Adder and Register File Test ........................................................................................ 100
   7.4 Voltage Droop Analysis ............................................................................................... 101
7. 3D Design ................................................................................................................................ 104
8. Discussion and Conclusion ................................................................................................... 109
REFERENCES .............................................................................................................................. 110
APPENDIX A: SiGe HBT Microprocessor Core Test Vehicle .................................................. 111
APPENDIX B: Binary Addition Structures with SiGe HBT Digital Circuits ......................... 126
**LIST OF TABLES**

Table 1: Comparison of design strategies ........................................................................ 17
Table 2: Comparison of CML/ECL and ECL designs using 7HP .................................... 18
Table 3: Fan-in comparison .......................................................................................... 22
Table 4: Comparison of different fan-ins ....................................................................... 25
Table 5: ALU Functions ............................................................................................... 57
Table 6: Comparison of power saving design changes ................................................. 88
Table 7: Reducing current usage in non-critical paths to reduce power ....................... 89
Table 8: Comparison of 8HP power saving designs .................................................... 91
Table 9: Results of Adder only test ............................................................................. 97
Table 10: Results off Register File only test ............................................................... 98
Table 11: Results of Adder and Register File test ....................................................... 100
Table 12: Fan-in power and size comparison of 7HP design ......................................... 148
Table 13: Wire delay percentages in various designs ................................................... 148
Table 14: Power comparison of 8HP designs ............................................................... 148
LIST OF FIGURES

Figure 1: Performance cap of parallel code ................................................................. 5
Figure 2: Processor utilization ..................................................................................... 6
Figure 3: Comparison of parallel performance ............................................................ 7
Figure 4: CML circuit ................................................................................................. 12
Figure 5: ECL Circuit ................................................................................................. 13
Figure 6: CML/ECL Circuit ....................................................................................... 14
Figure 7: CML Circuit with dotted AND/OR .............................................................. 15
Figure 8: ECL Circuit with dotted AND/OR ............................................................... 16
Figure 9: CML/ECL Circuit with dotted AND/OR ....................................................... 17
Figure 10: Simulations of different fan-ins ................................................................. 23
Figure 11: Breakdown of delay for different fan-ins .................................................... 24
Figure 12: Combining sum generation with two-way look-ahead gate utilizing dotted emitter/collector ................................................................. 33
Figure 13: Latched sum output in a single gate ........................................................... 34
Figure 14: Comparison of three version of etree4 cell .............................................. 36
Figure 15: Schematic of hstart cell ............................................................................ 37
Figure 16: Different layouts of hstart cell ................................................................. 38
Figure 17: Schematic of etree4 cell .......................................................................... 39
Figure 18: Different layouts of the etree4 cell ........................................................... 40
Figure 19: Schematic of etree2sum cell .................................................................... 41
Figure 20: Different layouts of the etree2sum cell ...................................................... 41
Figure 21: Schematic simulation of adder carry chain .............................................. 43
Figure 22: Layout of adder carry chain .................................................................... 44
Figure 23: Extracte simulation of adder carry chain ................................................ 45
Figure 24: Photograph of chip the adder carry chain was fabricated on .................... 46
Figure 25: Measured adder carry chain waveform ................................................... 47
Figure 26: Layout of buffer in full adder .................................................................... 49
Figure 27: Layout of hstart2 and istart2 cells ............................................................ 50
Figure 28: Layout of hstart2 and istart2 row ............................................................. 50
Figure 88: Latched sum output in a single gate............................................................. 142
Figure 89: Layout of the 7HP adder test structure......................................................... 142
Figure 90: Microphotograph of 7HP adder carry chain test structure ......................... 143
Figure 91: Layout of the 8HP adder test structure......................................................... 143
Figure 92: Microphotograph of adder carry chain test chip .......................................... 144
Figure 93: Schematic simulation of the 8HP adder test structure (waveform is the adder test signal that is divided in frequency by eight) ................................................... 144
Figure 94: Extracted simulation of the 8HP adder test structure (waveform is the adder test signal that is divided in frequency by eight) ................................................... 145
Figure 95: Measured waveform of the 8HP adder test chip (waveform is the adder test signal that is divided in frequency by eight) ................................................... 146
Figure 96: Layout comparison of no DT sharing design to DT shared design .............. 147
Figure 97: Simulated waveform of DT shared 8HP circuit (waveform is the adder test signal that is divided in frequency by two) ................................................... 147
Figure 98: Schematic of non-differential ECL cell ....................................................... 148
ACKNOWLEDGMENTS

This research was sponsored in part by DARPA/MARCO iFRC, DARPA/IBM
HPCS and the DARPA/SPAWAR/IBM 3DI programs.

Thank you to Professor McDonald for all of his hard work in securing this
contracts and providing very consistent support over the course of the project.

I would also like to thank Sun for providing a scholarship under which some of
this research was completed.

This work could not have been completed without the help of all of the people in
the FRISC lab: Okan Erdogan, Michael Chu, Philip Jacob, Aamir Zia, and Jin-Woo
Kim.

Finally, thank you to my parents and fiancée who supported me throughout.
ABSTRACT

For decades, CMOS has been the technology of choice for building fast and efficient computers. As CMOS transistor sizes continued to shrink, computers with ever-faster speeds continued appearing regularly. However, advances in clock rate have come less readily recently. This has initiated a trend towards multiple core microprocessors operating at slower clock rates. Hence, there is an opportunity to explore other technologies that might be used to build future computers with faster clock rates. Since, adders are essential component of the data path in any computer; we investigate the design of fast adders using SiGe Heterojunction Bipolar Transistors, which are currently the highest speed devices in silicon based circuit manufacturing.

The design and testing of these high speed 32-bit adders using IBM’s 7HP and 8HP SiGe technology constitute the bulk of the work described in the thesis. Designs running at speeds up to 17 GHz have been recorded in 7HP, while speeds up to 26.7 GHz have been observed in 8HP circuits. With improvements in layout and temperature it is predicted that an 8HP 32-bit adder would be able to run at 32 GHz. This adder, in addition to a register file designed in the same technology would provide the basis for a CPU core running at speeds vastly superior to that of CMOS. Lower speed designs with significant power reduction are also discussed.
1. Introduction

This thesis is primarily about the construction of high speed adder circuitry using SiGe BiCMOS technology. The motivation behind this investigation is that an adder is a very basic building block of CPU designs, and would be a limiting factor in how fast a CPU could perform. Speeds for the adder are demonstrated at 16 GHz and 26 GHz.

The bulk of the design work discussed within was done in IBM’s 7HP SiGe technology. Three separate designs were done in 7HP, two being adder carry-chain designs and one being a full ALU design. The design of the microprocessor test chip done in 7HP is also discussed, but further analysis of it is reserved till further testing is done on chips only recently returned from fabrication.

Several circuits designed in 8HP are also discussed, from the early design that used full DT sharing, to designs that suffered from changes in design rules, all the way to the latest redesign which improved the speed of the circuit to 26.7 GHz. Included in this are extracted simulations of all the various designs, to show how wiring delay affects the circuit.

1.1 Motivation

For decades, CMOS has been the technology of choice for building fast and efficient computers. As CMOS transistor sizes continued to shrink, computers with ever-faster speeds continued appearing regularly. Now, CMOS device scaling has apparently reached a fundamental limit due to wire resistance. At 65 nm, approximately half the microprocessor clock cycle is due to wire parasitic effects. Once this happens, together with power dissipation due to leakage, advances in clock rate come less readily. This
has initiated a trend towards multiple core CMOS microprocessors operating at slower clock rates. Hence, there is an opportunity to explore other technologies that might be used to build future computers with faster clock rates. Since, adders are essential component of the data path in any computer; we investigate the design of fast adders using SiGe Heterojunction Bipolar Transistors, which are currently the highest speed devices in silicon based circuit manufacturing.

Producing a high speed single core processor is the eventual goal of this project. Such a processor would have many advantages over a multiple-core approach. Programming for performance on a single core is a well understand process, while programming on many cores is a problem that has not yet been solved. There are also computational tasks which parallelize very poorly, where a single high clock rate processor would suit them very well.

Such a high speed processor needs certain components that can operate very fast. The two main components are the ALU and the register file. The critical path within an ALU is the carry-chain used for addition. Developing this carry-chain represents the bulk of the work in this thesis.

An attractive application of such high-speed adders is in the message-packing computers that form the backbone of high-productivity computing systems (HPCS). It is well known that a big chunk of network latency in HPCS comes from the dedicated CPUs that are used for message packing and unpacking during transfer of data packets between multiple nodes. However, such high-speed message packing computers will inevitably feel the impact of memory wall due to slow access to off-chip memory with limited bus-width. A possible solution to the memory-wall problem in these message-
packing computers can be using a vertically integrated (three-dimensional) processor-memory stack, thus providing a very wide data bus, with low interconnect latency, between processor and memory. If such a 3D processor-memory stack of very fast SiGe HBT CPU and vertically integrated 3D memory is employed as a message-packing computer at every node of HPCS, it could drastically reduce the network latency problem in the HPCS.

1.2 History

Early adder research led to carry-skip [1] and carry-select circuits [2], both of which have the advantage of a nearly bit-slice arrangement in the physical layout [3]. Full examination of parallelization led to block carry look-ahead [4]. Although requiring a large area for its speed increases, carry look-ahead often attracts the interest of bipolar designers who strive for the fastest possible circuits [5], and that of BiCMOS designers as well [6]. Other work has focused on adders incorporated in other circuits, such as multipliers, which present an uneven input profile to an adder [7].
1.3 Parallel vs Sequential

The debate between parallel and sequential computing has been around in the computing world nearly since its inception. In a 1967 AFIPS paper Amdahl presented a figure that would become known as Amdahl’s Figure of Merit:

\[ FOM_1 = \frac{S + P}{S + P/n} \]

In this equation \( S \) represents the amount of cycles that are inherently sequential and cannot be parallelized, \( P \) represents the amount of cycles that can be parallelized and \( n \) is the number of parallel processing nodes. The interesting part comes when one considers how much extra performance can be obtained with various values of \( S \) and \( P \). There is a hard cap on the performance gains directly tied to the value of \( S \). This hard cap is \( 1/S \), which means that no matter how many parallel processing nodes you throw at the problem you will never see performance better than this. All you will do is get closer and closer to the cap, never passing it. Figure 1 shows this cap for value of \( S \) between 1% and 10%.
What is interesting is that even if only 1% of your entire code is sequential, it still limits the speed-up to a factor of 100. As can be seen, the speed up gets progressively worse as more and more sequential code is added. A good point to look at is the 5% sequential code point. At this point the hard cap on performance is a factor of 20. Figure 2 shows the percentage of the processing power that is utilized by a number of nodes ranging from 1 to 1024.
What this shows is that you are utilizing less and less of your total processing power as you add nodes, simply trying to reach the hard cap on performance. The figure also shows the amount of the cap that you are utilizing. While you are using less nodes then the cap, these two numbers are the same, but as the number of processors grows they start to separate, and it takes a large number of nodes to even reach the theoretical cap.

For many years the computer industry was more concerned about chasing higher speeds for individual processors then it was about creating parallel collections of nodes. This chase seems to have come to an end in CMOS, with the focus switching to including more and more cores on a single chip, while keeping the speed of each chip relatively the same. With such a decision computer designers fall into this limit on paralyzing based on their code base.
To avoid this trap, it is necessary to continue increasing the clock rate of a single processor. As CMOS seems to have hit a limit on its single processor speeds, it falls to some other process to continue this advance in clock rate. This thesis focuses on using Silicon Germanium HBT circuits to push the processing speed of a single processor high enough to have more processing power than a multi-core design.

A single 32 GHz processor in SiGe is reasonable to imagine based on the work done in this thesis on optimizing an adder carry-chain, as well as work done in other areas in SiGe, such as register file design. Such a processor would have a decided advantage over the multi-core approach, particularly if the amount of sequential code is more than 1% of the total code base. Figure 3 shows a comparison of a single 32 GHz processor to a collection of 1 GHz cores.

![Figure 3: Comparison of parallel performance](image)

In perfectly parallelizable code, it would take a minimum of 32 1 GHz nodes to equal the speed of a single 32 GHz processor. As the figure shows, the actual number of
nodes needed is more than 32, and if the amount of sequential code is large enough no
number of slower processors will ever outperform the single faster processor. Only in
the cases of 1% and 2% sequential code do the multiple nodes outperform the single
processor, at 32 and 64 cores respectively. The 3% case will eventually level out at an
effective speed of 33 GHz, but at 1024 nodes it still hasn’t reached the goal of 32 GHz.
The other two cases shown, 4% and 5%, will never even reach the goal, leveling out at 25
GHz and 20 GHz respectively.

All of this effort should not be taken to mean that multiple cores are a bad thing
just that scaling up the number of cores imposes certain fundamental limits on the
amount of performance you can gain. As long as the number of cores is kept under the
hard cap on performance gains, the advantages of multiple cores can be great. As such,
the one other interesting data point shown in the figure is what happens if a small
number of multiple cores at 32 GHz are used. As the figure shows, using just two high
speed cores has a three times greater speed than 1024 cores at the same percentage of
sequential code. Four such cores would also have a greater effective speed than the cap
for 1% sequential code.
2. SiGe Bipolar Design

2.1 SiGe HBT

Silicon Germanium (SiGe) Heterojunction Bipolar Transistors (HBT) were chosen as the technology to demonstrate a high speed ALU. Historically, bipolar devices produce a circuit with 2.5 times the speed of the comparable CMOS circuits, assuming the same level of photolithography is used in both technologies. The IBM SiGe BiCMOS processes were chosen in particular. In conjunction with their high speed they offer relatively good yields compared to other bipolar processes and good current driving capability. They do not include new low-k dielectric materials, but do use Cu for a minimized interconnection related delay. The yield of the process enables the demonstration of more complex circuits, which may be incorporated in a microprocessor.

SiGe HBTs are bipolar transistors grown on a silicon substrate that exhibit speed advantages for various reasons. The transistors are actually not heterojunction bipolar transistors but rather base graded transistors. The alloy grading present in the base results in a valence band offset between the SiGe and the silicon interface. The valance band offset helps to confine holes in the SiGe layer, thus reducing the reverse injection of holes from the emitter region into the base region during the forward bias of the transistor. The presence of Ge in the base layer also contributes to speed improvements by straining the Si layer; the increased atomic spacing allows for higher electron mobility. The net result is that SiGe HBTs offer a lower barrier to electron injection into the base than a similarly configured conventional bipolar junction transistor. Some of
this can be traded for lower base resistance. The built in field resulting from Ge grading also enhances speed.

The SiGe HBTs speed, yield, linear characteristics and high gain make it very suitable for many applications. The current applications of SiGe are mainly in the telecommunications industries where high frequency analog and mixed signal circuits are used. As a consequence, contemporary SiGe technology has been somewhat optimized for analog applications. By grading Ge concentration in the base fully across the neutral base; HBT transistors have a high Early voltage with a relatively high cut-off frequency (fT).

The original IBM SiGe HBTs were designed for mainframe computers. Therefore initially they were optimized for that application. The digital profile placed the grade across the highest base doped region to achieve the highest fT. As a result SiGe HBTs have been capable of achieving fT in excess of 100 GHz since 1993. Breakdown voltage tends to limit fT, but numerous device and process improvements have permitted the SiGe HBTs to circumvent some of the implications of the Johnson relation.

The contemporary SiGe HBT satisfies a blend of analog and digital requirements, but its readily available speed advantages made it a prime choice for this study.

2.2 CML and ECL Design

CML and ECL are two different but similar types of circuit design. CML stands for Current Mode Logic, while ECL stands for Emitter Coupled Logic. A combination of the two is used throughout the design of the adder. CML was mainly used along critical
paths where high speed switching was most required. ECL was mainly used along non-
critical paths where higher fan-in was required.

An understanding of different circuit design styles and how they impact the
design of the adder is important to know. The two main design styles are Current Mode
Logic (CML) and Emitter Coupled Logic (ECL). How these styles work separately and
together with regards to the basic structures that make up the adder will be explored. In
addition, designs using dotted AND/OR collectors are shown. For each style a small and
large common circuit found in the adder is evaluated. The logical equation for the small
circuit is:

\[ H = H_1 + I_1H_0 \]

The logical equation for the large circuit is:

\[ H = H_3 + I_3H_2 + I_3I_2H_1 + I_3I_2I_1H_0. \]

The first design strategy that will be looked at is CML. CML offers the
advantage of smaller voltage swings, as well as faster switching time for a critical signal.
Figure 4 shows the CML implementation of the simple circuit.
This shows a tree depth of 4, although it does not take any signals on the first level of the tree. Each signal comes in as a differential pair, which means that the voltage swing per signal need only be 250 mV. Adding more signals, such as with the large circuit, causes the depth of the tree to grow by one for each signal added. As the tree depth increases so does the power supply voltage which results in high power consumption.

The second design strategy that will be looked at is ECL. ECL offers the advantage of higher fan in, which reduces the depth of the trees. Figure 5 shows the ECL implementation of the simple circuit.
The tree depth here is 3, and it also does not take any signals on the first level of the tree. Each signal comes in as a single ended signal, which means it has a voltage swing of 400 mV. Adding more signals does not necessarily increase the tree depth. Instead, each additional term that is added adds another level to the tree. This makes the large circuits trees much shorter in pure ECL versus pure CML.

It is possible to combine CML and ECL designs into the same current tree, to try and reap the benefits of both. Figure 6 shows the small circuit designed using both CML and ECL. The critical signals are placed at the top of the tree using ECL, as they have the fastest switching times there. The non-critical signals are added lower in the tree using CML.
The tree depth here is again 3, just like the pure ECL design. Similarly, as additional terms are added, the tree depth increase by one for each term added. This makes it very comparable to the pure ECL design. This was the design strategy used by a previous student in his thesis.

While these design strategies worked well enough in the past, something different was still needed to optimize the performance of the adder. The adder circuits have critical and non-critical signals in them, and their placement in the trees can have a large effect. Ideally the critical signals would be placed at the top of the tree, where switching is fastest, while the non-critical signals would be placed lower in the tree. As well, it would be best if the critical signals used CML while the non-critical signals used ECL. The first condition is met by the combined CML/ECL circuit above, but the second is not. In order to fulfill the second criteria another strategy must be used.
This alternate strategy is to use a dotted AND/OR. In these circuits, some of the logic is moved into the emitter followers. This strategy can be used with CML, ECL or a combination of both. Shown in Figure 7 is the CML implementation with dotted AND/OR.

![Figure 7: CML Circuit with dotted AND/OR](image)

The tree depth here is 3, and the first level is not used for signals. The tree depth here grows differently than in previous examples. The tree depth is set by the term with the most variables. In the large circuit the largest term has 4 variables, which would set the tree depth at 5. Adding additional terms does add additional power usage, as each new term comprises a new current tree. This design offers all critical signals located at the top of the tree, as well as being in CML. However, the fact that the depth of the tree increases with larger terms means the power usage will still be high.
It is also possible to combine an ECL circuit with the dotted AND/OR design as well. Figure 8 shows the small circuit using ECL and dotted AND/OR.

![Figure 8: ECL Circuit with dotted AND/OR](image)

The tree depth for this circuit is only 2, and unlike all of the other designs, it is fixed at 2, no matter how many additional signals and terms are added. This design offers the minimum tree depth and voltage supply of any of the designs, which makes it good for power consumption. The critical signals are on the top level, although they are in ECL which slows them down some.

The final circuit design is to combine CML and ECL along with the dotted AND/OR. This circuit combination is shown in Figure 9. It looks very similar to the pure CML with dotted AND/OR design, but the difference is that only the top level is CML, while the bottom level is ECL.
The tree depth for this circuit is constant just like the previous one, but this one is a tree depth of 3 instead of 2. Additional terms and signals do not add to the tree depth. This circuit design exhibits all of properties that we were looking for in the adder. The critical path signals are located at the top of the tree and in CML. The non-critical path signals are located one level lower in the tree in ECL, which keeps the depth of the tree constant. This is the design strategy that was decided upon and is used as the basis of all adder designs in this thesis.

Table 1 is a comparison of the six different design strategies for the large circuit with regard to tree depth, number of current trees per circuit, power supply voltage, power usage and finally the number of devices used in each circuit.

<table>
<thead>
<tr>
<th></th>
<th>CML</th>
<th>ECL</th>
<th>CML/ECL</th>
<th>CML+dot</th>
<th>ECL+dot</th>
<th>CML/ECL + dot</th>
</tr>
</thead>
</table>

Figure 9: CML/ECL Circuit with dotted AND/OR
This work shows that the two most desirable designs to examine further are the ECL with dotting and the combined CML and ECL with dotting. If the goal was to simply minimize power consumption, it would be easy to pick ECL with dotting as being the optimal design, but the main goal is actually to maximize speed. As such it is important to look at the difference in speeds between these two designs. Table 2 shows a comparison of speed and power between these two designs, using the 7HP design kit.

<table>
<thead>
<tr>
<th></th>
<th>Delay (ps)</th>
<th>Speed (GHz)</th>
<th>Change</th>
<th>Power (W)</th>
<th>Change</th>
<th>Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>CML/ECL</td>
<td>53.35</td>
<td>18.7</td>
<td>100%</td>
<td>3.1</td>
<td>100%</td>
<td>1731</td>
</tr>
<tr>
<td>ECL</td>
<td>65.7</td>
<td>15.2</td>
<td>81%</td>
<td>2.2</td>
<td>70%</td>
<td>1379</td>
</tr>
</tbody>
</table>

This data shows that the ECL design only performs at 81% of the speed to the combined CML/ECL design, making it undesirable when attempting to maximize the speed of the circuit. However, the savings in power and devices make it an ideal candidate for discussion later when power saving designs will be discussed.
3. General Adder Design

3.1 Parallel Prefixes

Arithmetic carries belong to the set of functions called “prefix problems”. This kind of function generates a series of results where each term depends on the previous term, i.e. each result is the “prefix” for the next. Efficient solutions to prefix problems depend on generating in parallel certain sub-sections of prefixes and then combining them to produce the complete results. Prefixes are one of the core ideas of parallel computation.

Consider a series of terms on some associative operator “•”, e.g. \(x_0, x_0 \cdot x_1, x_0 \cdot x_1 \cdot x_2, \ldots\). If we look at each term of the series, \(F_n=x_0 \cdot \ldots \cdot x_{n-1} \cdot x_n\), it is clear that due to the associativity of our operator each term may be rewritten as a recurrence \(F_n=F_{n-1} \cdot x_n\). In other words, each term of the series is generated by applying our associative operator to a new variable and a prefix that turns out to be the previous term in the series. The set of problems that such a construction applies to are referred collectively as “prefix problems”. Prefix problems form the theoretical basis of several practical computational circuits, notably among them carry trees for addition. The idea of prefix circuits was first introduced as part of a fast binary adder.

If the series is generated purely by means of direct application of the recurrence relation, it is apparent that the time to generate the output of a prefix problem will grow linearly with the size of the input. To reduce this time, methods of parallel prefix generation are required.

A serial prefix circuit with \(n\) inputs can be shown by inspection to require \(n-1\) operations and require time \(n-1\) to complete. It can be shown for any prefix circuit that
the lower bound of the sum of the time and the size of the circuit is $2n-2$. This suggests that increasing the number of operations could be used to reduce the depth of the circuit. In VLSI circuits, the ever-increasing integration causes this tradeoff to be easy to make, as well as desirable.

Since the operator used to build the prefixes is associative (by definition), the prefix circuit could be built up by binary division. The inputs are divided into a lower-order and a higher-order half, and the prefixes for each half computed. The results from the lower-order half are then applied to the partial prefixes for the higher-order half to produce the complete prefixes. This division would be applied recursively to produce the prefixes for each half. The size of an $n$ input circuit would be twice the size of an $n/2$ circuit plus the operations to produce the complete prefixes for the higher-order half, while the depth would be one more than the depth for $n/2$ inputs. Removing the recurrence means that the size is $(n/2)\log n$ and the depth is $\log n$.

It can be shown that addition is a type of prefix operation. Specifically, it is the carry generation that constitutes a system of prefixes. The sum at each bit position is a function of the operand bits at that position and the carry-out out from the preceding position. Carry-out of position $n$ depends on the operands at position $n$ and the carry from the proceeding position $n-1$, which can be expressed in terms of the recurrence relation for a prefix operation given above. The basic series computation of prefixes is the equivalent of ripple carry. If we consider the delay of a gate that computes “•” and the area that the circuit occupies as our basic units, generating prefixes in series over $n$ terms will take $n-1$ time units but occupies a space of $n-1$. At the other extreme, each prefix could be computed independently in constant time at great expense in area (and
with highly impractical fan-in and fan-out requirements). For a large set of these carry structures, the relationship between depth and size (in terms of processing nodes) is so strong that it is possible to expand or contract the prefix graph with a non-heuristic algorithm to pass from one structure to another.

### 3.2 Effect of different fan-ins

An area that has not been explored previously in this thesis is what the exact effect of increasing the fan-in of the carry look-ahead gates is. The fan-in that is used throughout the circuits presented herein has been 4. Work done by a previous student in an earlier technology used a fan-in of 3. A more traditional implantation of carry look-ahead gates would use a fan-in of 2. How changing the fan-in from 2 to 3 to 4 effects speed, power and size is discussed.

The basic reason for increasing the fan-in is to reduce the total number of gate delays that are needed for the circuit. All three fan-in values would share a common beginning and end stage, so it comes down to what is done between the two to differentiate the different values of fan-in. The number of gate delays for each fan-in value is given in Table 3. The table shows how the number of gate delays decrease as the fan-in increases. Notice that for a fan-in of 4, there remains only 2 gate delays between the beginning and ending stages. To reduce this number to only 1 stage, the fan-in would have to jump to 16, something that would be impractical. Also given is the expected slow down from increasing the number of gate delays. These are just simple assumptions, assuming as they do that each gate delay contributes the same amount to the overall delay. In reality each gate delay will differ a little from each other, with more
complicated circuits, such as are found in the higher fan-in design, having a larger amount of delay.

Table 3: Fan-in comparison

<table>
<thead>
<tr>
<th>Fan-in</th>
<th>Gate Delays</th>
<th>Expected Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fan-in 4</td>
<td>4</td>
<td>100%</td>
</tr>
<tr>
<td>Fan-in 3</td>
<td>5</td>
<td>80%</td>
</tr>
<tr>
<td>Fan-in 2</td>
<td>6</td>
<td>67%</td>
</tr>
</tbody>
</table>

With the basic comparison done and showing that using a higher fan-in gate for the carry-look ahead should be better, a more thorough analysis should be done to confirm that it will indeed produce an increase in speed. Schematics for a full adder with fan-in 4 already existed, and these already included all of the cells necessary to create both the fan-in 2 and fan-in 3 top level schematics. Figure 10: Simulations of different fan-ins

Figure 10 shows the simulation results from these three circuits. The times shown in the figure are twice the total time through the adder, and do not have any extracted parasitics included in them. The wire parasitics will scale as the number of gate delays increases, so a fair speed comparison can be obtained without them.
As the figure shows, increasing the fan-in does in fact increase the speed of the circuit. The speed gain is good, although it does not meet the optimistic estimates from earlier. The 3 fan-in design was 86% of the speed of the 4 fan-in design, while the 2 fan-in design was 76% of the speed of the 4 fan-in design. Why these differ can be determined by breaking the delays down even further to see how each gate delay contributes to the overall delay. Figure 11 shows a breakdown of each gate delay for the three different fan-in values.
As the table/figure shows, the delays for the first and last stage of the carry-chain did not change between the three different fan-in values. What does change is the time it takes for the in-between stages. The time per stage does increase as the fan-in increases, but the increase is not enough to offset the decrease in the number of stages. This is the reason that decreasing the number of stages resulted in an increase in speed.

While speed is the desired quantity to maximize, it is always important to keep in mind how the changes impact the power consumption and size of the design. As such, Table 4 shows a comparison of speed, power and size for the three fan-in values. The power and size numbers are based on the critical paths of the adder. The non-critical
paths remain the same for all three values, and are thus left off for the purposes of comparison.

**Table 4: Comparison of different fan-ins**

<table>
<thead>
<tr>
<th></th>
<th>Delay (ps)</th>
<th>Difference</th>
<th>Power (W)</th>
<th>Difference</th>
<th>Devices</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 fan-in</td>
<td>36.25</td>
<td>100%</td>
<td>1.96</td>
<td>100%</td>
<td>1745</td>
<td>100%</td>
</tr>
<tr>
<td>3 fan-in</td>
<td>42.3</td>
<td>86%</td>
<td>2.32</td>
<td>118%</td>
<td>1819</td>
<td>104%</td>
</tr>
<tr>
<td>2 fan-in</td>
<td>48.2</td>
<td>75%</td>
<td>2.61</td>
<td>133%</td>
<td>1960</td>
<td>112%</td>
</tr>
</tbody>
</table>

The interesting result here is that using gates with a higher fan-in to maximize speed actually results in a reduction in both size and power compared to the lower fan-in designs. The natural conclusion to all of these investigations is that based on the technology that is being used, using a higher fan-in gate is the correct design decision.
3.3 Pseudo-Carry Look-Ahead (PCLA)

The pseudo-carry is based on carry look-ahead addition. Carry look-ahead addition, sometimes referred to as "parallel addition", creates a carry for a bit by combining signals generated at each preceding bit. This can be performed by wide fan-in gates or by a tree-like structure where the look-ahead combines larger and larger blocks. The pseudo-carry improves on that by distinguishing limiting and non-limiting delay paths, and moving delay from the limiting paths to the non-limiting paths.

In the context of carry look-ahead, the computation of a carry out can be reformulated in term of the propagate ($P_n$) and generate ($G_n$) signals:

$$P_n = A_n + B_n$$
$$G_n = A_n B_n$$

$$C_n = A_n B_n + (A_n + B_n) C_{n-1} = G_n + P_n C_{n-1}.$$

The recursion on the $C_{n-1}$ term can be expanded:

$$C_n = G_n + P_n G_{n-1} + P_n P_{n-1} G_{n-2} + \ldots + P_n \ldots P_0 C_{in}.$$

If in fact $P_n$ is computed with an inclusive or, it is clear that the following identities exists:

$$G_n = G_n P_n,$$

and

$$P_n = G_n + P_n.$$

Thus we can factor the $P_n$ out of the equation for $C_n$:

$$C_n = P_n (G_n + G_{n-1} + P_{n-1} G_{n-2} + \ldots).$$
The remainder of the terms is known as the "pseudo-carry", $H_n$. The $P_n$ term can then moved into the equation for $S_n$:

$$S_n = A_n \oplus B_n \oplus C_{n-1}$$

$$S_n = A_n \oplus B_n \oplus (P_{n-1}H_{n-1}).$$

The $A_n \oplus B_n$ term, identified as $T_n$, can be computed in parallel with the $H_{n-1}$ term as well as the $P_{n-1}$ term resulting in the expression

$$S_n = T_n \oplus (P_{n-1}H_{n-1})$$

The question remains, however, what has been gained by this manipulation? The key is the reduction in fan-in from $C_n$ to $H_n$ for a given block size to cover. Specifically in the case of the current-steering logic gates being considered, increasing complexity severely decreases the available fan in for a single gate. Furthermore, by inspection, neither $T_n$ nor $P_{n-1}$, since both are simple Boolean functions, will exceed the delay for $H_{n-1}$. Thus delay on a critical path has been traded for delay on non-critical paths.

The generalized equations for group pseudo-carry $H^n_0$ can be derived in the same manner as those for group generates $G^n_0$. Given the expression for carry in terms of propagates and generates

$$C_n = G_n + P_nG_{n-1} + P_nP_{n-1}G_{n-2} + \ldots + P_n \ldots P_0C_{in}$$

we can factor those lesser terms into group generates and propagates of the form $G_{\text{group}} + P_{\text{group}}C_{\text{group}}$ by the following rules:

1. Each term of $G_{\text{group}}$ will contain exactly one subgroup generate.

2. Each term of $G_{\text{group}}$ will also contain every higher ordinal subgroup-propagate than the subgroup-generate for that term. The term with the highest ordinal subgroup will thus contain no subgroup propagates.
3. \( P_{\text{group}} \) will contain only subgroup propagates, corresponding one for one with the subgroup generates in \( G_{\text{group}} \).

Signals that look-ahead over groups are represented by the symbol \( T^n_{i} \), where:

- \( T \) is the type of look-ahead signal,
- \( n \) is the width of the group in bits,
- \( i \) is starting bit position of the group.

The signals for a single bit represented by \( T_i \) map to \( T^1_i \) in the group look-ahead notation.

As an example of group look-ahead, consider the four-bit carry

\[
C_3 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_{\text{in}}.
\]

Such a carry can be rewritten in terms of two-bit group look-ahead signals:

\[
C_3 = G^2_2 + P^2_2G^2_0 + P^2_2P^2_0C_{\text{in}},
\]

where

\[
G^2_2 = G^1_3 + P^1_3G^1_2,
\]

\[
G^2_0 = G^1_1 + P^1_1G^1_0,
\]

\[
P^2_2 = P^1_3P^1_2,
\]

and

\[
P^2_0 = P^1_1P^1_0.
\]

In a similar manner group pseudo-carries can be made by adding a rule to factor out a \( P^1_n \) term

\[
C_3 = P^1_3(H^2_2 + I^2_1H^2_0 + I^2_1I^1_0C_{\text{in}}),
\]

where

\[
H^2_2 = G^1_3 + G^1_2.
\]
One can compute higher-level $H^n_i$ terms from those of sub-blocks in the same manner as for carry select. Note that due to the factoring which creates the pseudo-carry, the $I^n_i$ terms are one bit position behind the corresponding $P^n_i$ terms.

### 3.4 Generalized pseudo-carry equations

In general, the pseudo-carry look-ahead tree is expanded by:

\[
H^{n+m}_i = H^m_{i+n} + I^m_{i+n-1} H^n_i,
\]

or

\[
H^2_i = G^1_{i+1} + G^1_i
\]

and in the case of a three-input look-ahead gate

\[
H^{n+m+p}_i = H^p_{i+n+m} + I^p_{i+n+m-1} H^m_{i+n} + I^m_{i+n+m-1} I^m_{i+n-1} H^n_i.
\]

The carry-out of a 32-bit adder can thus be generated from the 32-bit pseudo-carry tree using four sets of group pseudo-carries such as these:

1. \(H^2_{30} = G^1_{31} + G^1_{30}, H^2_{28} = G^1_{29} + G^1_{28}, H^2_{26} = G^1_{27} + G^1_{26},\)
   \(H^2_{24} = G^1_{25} + G^1_{24}, H^2_{22} = G^1_{23} + G^1_{22}, H^2_{20} = G^1_{21} + G^1_{20},\)
   \(H^2_{18} = G^1_{19} + G^1_{18}, H^2_{16} = G^1_{17} + G^1_{16}, H^2_{14} = G^1_{15} + G^1_{14},\)
   \(H^2_{12} = G^1_{13} + G^1_{12}, H^2_{10} = G^1_{11} + G^1_{10}, H^2_{8} = G^1_{9} + G^1_{8},\)
   \(H^2_{6} = G^1_{7} + G^1_{6}, H^2_{4} = G^1_{5} + G^1_{4}, H^2_{2} = G^1_{3} + G^1_{2},\)
   \(H^2_{0} = G^1_{1} + G^1_{0}\)
2. \( H^6_{24} = H^2_{28} + I^2_{27} H^2_{26} + I^2_{27} I^2_{25} H^2_{24}, \)
   \( H^6_{18} = H^2_{22} + I^2_{21} H^2_{20} + I^2_{21} I^2_{19} H^2_{18}, \)
   \( H^6_{12} = H^2_{16} + I^2_{15} H^2_{14} + I^2_{15} I^2_{13} H^2_{12}, \)
   \( H^6_{6} = H^2_{10} + I^2_{9} H^2_{8} + I^2_{9} I^2_{7} H^2_{6}, \)  \( H^6_{0} = H^2_{4} + I^2_{3} H^2_{2} + I^2_{3} I^2_{1} H^2_{0} \)

3. \( H^{14}_{18} = H^2_{30} + I^2_{29} H^6_{24} + I^2_{29} I^6_{23} H^6_{18}, \)
   \( H^{18}_{0} = H^6_{12} + I^6_{11} H^6_{6} + I^6_{11} I^6_{10} H^6_{0} \)

4. \( H^{32}_{0} = H^{14}_{18} + I^{14}_{17} H^{18}_{0} \)

3.5 Look-ahead gate

In developing a look-ahead style carry tree there is a particular function that is so ubiquitous that developing a specialized circuit to generate it is called for. In the look-ahead tree, a generate is asserted for a block of bits when: a generate is asserted for the highest-order bit; a propagate is asserted for the highest-order bit and a generate is asserted for the second-highest-order bit, etc. It is easy to see that for an N-bit block the look-ahead function will require \(2N-1\) inputs, and thus \(2N-1\) levels in a series-gated differential ECL/CML gate. The height of such a current tree quickly hits the limit of reasonable supply levels, thus placing a lower bound on a B-bit carry tree of \(\log N B\). Three series-gated levels allows only two bits of look-ahead, so that a 32-bit carry tree would require five gate delays.

If four bits of look-ahead could be computed in a single gate, a 32-bit carry tree would require only 3 gate delays. Fully differential inputs won’t allow this without a larger supply voltage, but by mixing single-ended signals with differential ones and
suitably rearranging the Boolean function being generated it is possible to make a four-bit look-ahead gate.

The four-way look-ahead gates that allow group pseudo-carries of the form
\[
H_{i+n+m+p+1} = H_{i+n+m+p+1}^q + I_{i+n+m+p+1}^p + I_{i+n+m+p-1}^p + I_{i+n+m-1}^m + I_{i+n-1}^m + I_{i+n}^n
\]
to be built up from four subgroup pseudo-carries in a single gate have already been discussed. There is also an equivalent two-way gate of the form \( H_{i+n+m} = H_{i+n} + I_{i+n-1}^m \).

A look-ahead tree arrangement similar to the Han and Carlson variant of the Kogge-Stone tree is used, which generates only the odd positions. This would increase the depth of the circuit by one in the Han and Carlson case, however the pseudo-carry factorization in the sum-generation circuit allows both even and odd sums to be generated from only odd position carries.

The first layer of the carry tree, driven directly by the operand bits, cannot utilize the four-way look-ahead gate due to the number of inputs and will cover groups of two bits, not three. This is where the smallest group pseudo-carries are generated from the operands. The expression given above
\[
H_i^2 = G_{i+1}^1 + G_i^1
\]
can be reduced to be directly expressed in the operand bits
\[
H_i^2 = A_{i+1} B_{i+1} + A_i B_i.
\]

The next two layers of the carry tree are built up with the two-, three- and four-way look-ahead gates. Where the first layer created groups of two bits, the next two layers using four-way look-ahead created groups of eight and thirty-two bits respectively.
The depth of the tree can be calculated as $1 + \log_4(32/2)$ or 3 gates. This tree only produces the pseudo-carries however. One additional gate would be needed to produce the final sum computation and latching resulting in a five gate deep adder.

A circuit using multiple current trees with dotted-or and dotted-and outputs is used to implement this design. The differential outputs of a number of current trees are logically OR-tied by dotting collectors of the inverting output and dotting emitters on the non-inverting output. Any single-ended switching is moved to the non-critical I inputs, if it is necessary at all. When the gate is expanded to handle more inputs, it gets wider but not taller, allowing it to operate on the same supply voltage. The possibility of an excessive swing exists on the inverting output when multiple trees pull down, but this is clamped at the output by emitter-dotting with a normal logic "0" voltage.

The carry-out of a 32-bit adder can thus be generated from the 32-bit pseudo-carry tree using three sets of group pseudo-carries such as these:

1. $H^2_{30} = G^1_{31} + G^1_{30}$, $H^2_{28} = G^1_{29} + G^1_{28}$, $H^2_{26} = G^1_{27} + G^1_{26}$,
   $H^2_{24} = G^1_{25} + G^1_{24}$, $H^2_{22} = G^1_{23} + G^1_{22}$, $H^2_{20} = G^1_{21} + G^1_{20}$,
   $H^2_{18} = G^1_{19} + G^1_{18}$, $H^2_{16} = G^1_{17} + G^1_{16}$, $H^2_{14} = G^1_{15} + G^1_{14}$,
   $H^2_{12} = G^1_{13} + G^1_{12}$, $H^2_{10} = G^1_{11} + G^1_{10}$, $H^2_{8} = G^1_{9} + G^1_{8}$,
   $H^2_{6} = G^1_{7} + G^1_{6}$, $H^2_{4} = G^1_{5} + G^1_{4}$, $H^2_{2} = G^1_{3} + G^1_{2}$,
   $H^2_{0} = G^1_{1} + G^1_{0}$

2. $H^8_{24} = H^2_{30} + I^2_{29}H^2_{28} + I^2_{29}I^2_{27}H^2_{26} + I^2_{29}I^2_{27}I^2_{25}H^2_{24}$,
   $H^8_{16} = H^2_{22} + I^2_{21}H^2_{20} + I^2_{21}I^2_{19}H^2_{18} + I^2_{21}I^2_{19}I^2_{17}H^2_{16}$,
   $H^8_{8} = H^2_{14} + I^2_{13}H^2_{12} + I^2_{13}I^2_{11}H^2_{10} + I^2_{13}I^2_{11}I^2_{9}H^2_{8}$,
   $H^8_{0} = H^2_{6} + I^2_{5}H^2_{4} + I^2_{5}I^2_{3}H^2_{2} + I^2_{5}I^2_{3}I^2_{1}H^2_{0}$
3. \[ H_{20}^{12} = H_{24}^8 + I_{17}^8 H_{16}^8 + I_{17}^8 I_{17}^8 H_{16}^8 + I_{17}^8 I_{17}^8 I_{17}^8 H_8^0 \]

With the addition of a current tree to produce the final sums, the total dept of a 32-bit adder would be 4 current trees. Figure 12 shows a combined sum and look ahead gate, which allow generation of every sum from a carry-tree that generates pseudo-carries for only the even terms. This reduces the area and power necessary for carry generation without impacting overall delay.

![Figure 12: Combining sum generation with two-way look-ahead gate utilizing dotted emitter/collector](image)

Furthermore, a latch is be incorporated into the sum gate for only a modest time penalty, which is much less than an additional gate delay if that is the desired location for a pipeline latch. To properly latch a dotted emitter/collector gate, it is necessary to drive the keeper current switch from the outputs of the emitter-followers, not the inputs as is possible with single current tree gates (Figure 13).
Figure 13: Latched sum output in a single gate
4. 7HP Adder Design

4.1 Comparison of 7HP Designs

There were three 7HP circuit designs done. The first two designs were adder carry chains, while the third was a full ALU. The carry chain circuits were fabricated and tested separately, while the full ALU was included in CPU core test chips and was only simulated.

The very first 7HP adder carry chain was done with an early version of the kit that had a very limited selection of resister models to choose from. The available resister, K1RES, was a very large resistor that could not be placed close to another resistor, thus the design itself was very large. Figure 14 shows a comparison of a cell from the three different designs and shows how they get progressively more compact. A version of this circuit was designed and fabricated, but not tested extensively as newer kits were released and better circuits could be designed. The second 7HP chip design is the one that will be discussed in the most detail, and was the foundation for all later designs.
The adder carry chain is a simple structure that can be built upon to create a fully working adder. The carry chain has two thirty-two bit long differential inputs, as well as a differential carry-in input. The chain itself only has one differential output, which is the sum for the most significant bit. For testing purposes this signal is output as a divided by two and a divided by sixteen signal.

The carry chain is broken up into four stages. Each stage will be explained in general, and some layouts for each stage will be shown as well. Two sets of layout for each stage will be presented, one from the adder carry chain design, and one form the full adder design. The purpose of this is to show that noticeable size reductions in the basic cells were made. These size reductions were mainly the result of two major design changes. The first design change was using vertically shared deep trenches. The second design change was the use of NMOS devices to control the current through the trees.
instead of a resistor. Each change helped greatly, but the use of the NMOS devices was more helpful in reducing the size of the circuit than the shared deep trench.

The first stage in the carry chain is made up of cells that take as inputs two bits from each of the two inputs to the adder. There are sixteen copies of this cell in the adder carry chain, covering all thirty-two bits of input. Each of these cells outputs a single differential signal, thereby generating sixteen signals for use in the next stage of the carry chain. This stage is used to generate the first piece of the carry. Figure 15 is the schematic of this cell.

![Schematic of hstart cell](image)

Figure 15: Schematic of hstart cell

Figure 16 is the layout for the hstart cell. The cell on the left is from the adder carry chain design, and the cell on the right is from the full adder design. Both cells are included in the same layout so that the difference in size can be noted.
Figure 16: Different layouts of hstart cell

The second stage of the adder carry chain is made up of cells that take four inputs each from the previous stage, and are called the etree4 cells. There are four copies of this cell, covering all sixteen outputs from the previous stage. Each of these cells outputs a single differential signal, generating four signals for use in the third stage. This stage is used to generate pieces of the carry. Figure 17 is the schematic of this cell.
Figure 18 compares two different layouts for the etree4 cell. The cell on the left is from the adder carry chain design, and the cell on the right is from the full adder design. Both cells are included in the same layout so that the difference in size can be noted.
The third stage is made up of the same cell that is used in stage two. It takes as inputs the four signals generated in stage two. It outputs a single differential signal that is used by the final stage. This stage produces the final carry value.

The fourth and final stage is made up of one cell, called the etree2sum cell. It takes the carry generated in the previous stage and generates the sum for the circuit. This sum is then fed back to the beginning of the circuit as the carry-in, thus making a ring oscillator. Figure 19 is the schematic of the etree2sum cell.
Figure 20 compares different layouts for the etree2sum cell. The cell on the left is from the adder carry chain design, and the cell on the right is from the full adder design. Both cells are included in the same layout so that the difference in size can be noted.
The full adder uses the same basic design as the carry chain and contains the same number of stages. The main difference is that in the full adder there are more circuits in each stage doing calculations in parallel.

4.2 7HP Carry Chain

The carry chain for the adder was designed and fabricated. This accomplished two things; setting a minimum time for signals to pass through the adder and providing a comparison between measured and simulated results. The output of the carry chain was tied back to the input to create a ring oscillator for ease of testing.

The carry chain was initially simulated directly from the schematic. Figure 8 shows the waveform generated from this simulation. This signal, along with later signals in this section, are all four times the time through the adder. Figure 21 shows that it takes 39ps for a signal to traverse the entire carry chain.
A layout was made for the adder and an extracted simulation with parasitic resistors and capacitors was done to provide an idea of how the manufactured circuit would perform. Figure 22 shows the layout and Figure 23 shows the waveform from this simulation. The time through the carry chain was 56ps, meaning that the wiring contributed 17ps to the total delay.
Figure 22: Layout of adder carry chain
Figure 23: Extract simulation of adder carry chain

The carry chain was fabricated along and then tested. Figure 24 shows a picture of the finished chip and Figure 25 shows the measurement of the carry chain. The time through the adder was 57.5ps, only a 1.5ps difference from the extracted simulation.
Figure 24: Photograph of chip the adder carry chain was fabricated on
The difference between the extracted simulation and the measured waveform was only three percent, giving great confidence in the ability to accurately predict the performance of a fabricated circuit based on extracted simulations with parasitics.

4.3 Design of ALU Carry Chain for 7HP Full ALU

A full ALU was created using the basic design used in the adder carry chain circuitry. This full ALU was used in the first and second CPU core test chips that were assembled. The ALU accepted two 32-bit numbers as input, and produced a 32-bit
result. It provided a wide range of both arithmetic and logical functions. It also incorporated a latch into the final sum calculation.

The full ALU can be broken down into two major areas of design. The first area of design was the carry tree, which had been the focus of all the previous designs, as it represents the critical path through the adder. The second area is the ALU circuitry, which provides almost all of the additional functionality of the ALU, but is not on the critical path through the circuit,

The numbers entering the adder get sent to both areas of the circuit. The numbers go directly to the carry chain, but are passed through a buffer before going to the ALU. The buffers are incorporated in the front of the design to allow the signals to be passed directly through while at the same time buffering them up for the ALU section. This was done to prevent excess loading on the signals coming into the adder, as each bit can drive up to 5 signals in the ALU circuitry. Figure 26 is the layout for the buffer.
The first stage of the carry chain is the hstart2 and istart2 cells. The hstart2 cells take 2 bits of input from each number entering the adder, as do the istart2 cells. The difference is that the hstart2 cells start with the 0th bit while the istart2 cells start from the 1st bit. There are 16 copies of the hstart2 cell and 15 copies of the istart2 cell. Since each neighboring hstart2 and istart2 cell share one bit of input in common it was easy to combine the hstart2 and istart2 cells together to decrease wire lengths further in the design. Figure 27 shows the hstart2 and istart2 cells combined together. Figure 28 shows the entire row of hstart2 and istart2 cells.
The hstart and istart cells feed the next set of cells, which are made up out of buffers, etree2, etree3, etree4 and AND cells. There are four of each type of these cells in the next layer. The basic structure is a block that contains one buffer, one etree2, one etree3 and one etree4. There are four of these blocks that make up this stage. The signals from the previous stage are broken up such that hstart 0-3 feeds the first block,
hstart 4-7 feeds the second block and so forth. The istart signals also go to these blocks, as well as the AND blocks. Figure 29 shows the combined buffer, etree2/3/4 block, and Figure 30 shows the entire row of cells.

![Figure 29: Layout of etree4, etree3, etree2 and buffer](image)

![Figure 30: Layout of etree4, etree3, etree2, and buffer row](image)

The next stage is fed completely by the previous stage. It is made up of buffers, etree2, etree3 and etree4 cells. There are four of each of the cells in this stage, although the configuration is different than the previous stage. In this stage, the configuration is 4
buffers, then 4 etree2s, then 4 etree3s, and finally 4 etree4s. The etree4s from the previous stage provide most of the inputs for this stage, while the other cells from the previous stage usually only provide one input. Figure 31 shows the layout of the entire row.

![Figure 31: Layout of second etree4, etree3, etree2, and buffer row](image)

The final stage takes input from the previous stage of the carry chain, as well as from the ALU circuitry. There are 3 basic cells that make up this stage, the sumlatch, the etree2sumlatch and a normal latch. The sumlatch and etree2sumlatch contain the first half of a master/slave latch, as well as additional circuitry, while the latch is the second half of the master/slave latch. Figure 32 and Figure 34 show the schematics of the sumlatch and etree2sumlatch cells, and Figure 33 and Figure 35 are the layout of those cells.

![Figure 32: Schematic of sumlatch cell](image)
Figure 33: Layout of sumlatch cell

Figure 34: Schematic of etree2sumlatch cell
There are 16 copies of the sumlatch cell and 16 copies of the etree2sumlatch cell in this stage. Each sumlatch can be paired up with a corresponding etree2sumlatch as they share similar inputs. Figure 36 shows the layout of these two cells combined together.
The size and number of these cells is much greater than the previous stages, so it requires two rows to be used to minimize the wire length of the critical path signals going to this stage. A third row is used for the final latch stage. Figure 37 shows layout of the three rows.
4.4 Design of ALU unit for 7HP Full ALU

The ALU circuitry is made up of 2 stages. The first stage is made up of P, G, X and Y cells while the second stage is made up of T cells. The G cell is an AND cell and creates the generate term. The P cell is a more complicated gate, being a basic OR gate ANDed with a control bit to determine mode of operation. The X and Y cells are used to create the majority of different operations the ALU is capable of. The T cell is simply an XOR of the X and Y cells.

Since these cells are not on the critical path they are split up, with bits 0-15 being on the one side of the carry chain circuitry and bits 16-31 being on the other side of the carry chain circuitry. Each side is also stacked up in 8 rows to keep them close to the rest of the circuitry. Figure 38 shows the ALU circuitry on the left and right sides.
The ALU has 6 main control bits. The M control bit determines whether an arithmetic or logical operation will take place within the adder. A logical operation does not use the carry chain circuitry while an arithmetic operation does. There are 4 select bits that are used to select one of sixteen different operations. The last control bit is the carry-in, which is only used in arithmetic operations. There are 16 logical operations, representing every combination of 2 values. There are 16 basic arithmetic operations, but the result will change based on what you set the carry-in value to. Table 5 details all the different operations that the ALU can perform.

Table 5: ALU Functions

<table>
<thead>
<tr>
<th>SELECTION</th>
<th>M = H; LOGIC</th>
<th>M = L; ARITHMETIC OPERATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S3 S2 S1 S0</td>
<td>Cn = L (no carry)</td>
<td>Cn = H (with carry)</td>
</tr>
<tr>
<td>L L L L</td>
<td>F = !A</td>
<td>F = A MINUS 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F = A</td>
</tr>
</tbody>
</table>
It should be noted that subtraction operations require an inversion in the B input. This inversion is not handled in the adder, but instead is handled outside of the adder by the previous stage in the pipeline. This allows for faster calculations within the adder.

The assembled adder is shown in Figure 39.
Basic functionality for the adder was tested by setting one input to 1 and adding it to the other output, with the result being latched and then fed back in. This tests the basic speed and functionality of the adder.
5. 8HP Adder Design

5.1 8HP Adder Carry Chain Design

The adder carry chain is composed of four stages and is turned into a ring oscillator for the purpose of testing the speed through it. It has two external outputs that can be measured, a divide by four and a divide by sixteen of the sum signal. Figure 40 shows the schematic of the circuit.

![Figure 40: Schematic of 8HP adder carry chain](image)

The carry chain is made up of three types of cells, named hstart2, etree4, and xor2. The first stage is hstart2, the second and third stages are etree4 and the last stage is xor2. Figure 41, Figure 42, and Figure 43 show the schematics for them in order, to show the basic design and device count for each cell.

![Figure 41: Schematic of 8HP hstart2 cell](image)
The initial design used 1µm devices and used 1mA of current per tree. This size and current were selected because they were comparable to the 7HP design, and a comparison of speed up using the same current would be possible. Figure 44 shows a schematic simulation of the 8HP circuit. The simulation shows a time of 250ps, which is eight times the time through the adder, for a single cycle time delay of 31.25ps.
The $f_T$ at this size and current is around 190 GHz in 8HP. In 7HP the devices we used at 1mA had an $f_T$ of around 120 GHz. There was around a 50% improvement in $f_T$, most of which is reflected in the decreased time in the adder.

8HP offers a higher $f_T$ with larger device sizes, so another design was made that used 3µm devices using 3mA per tree. This produced a simulation with a time of 204ps, when divided by eight yields a time of 25.5ps through the adder. The $f_T$ at this size and current was around 200 GHz, which was a 60% improvement over 7HP. This improvement was reflected in the decreased time through the adder. Figure 45 shows the simulation of the 3µm circuit.
After the schematic simulations were completed layout of the 1µm and 3µm designs began. Figure 46, Figure 47, and Figure 48 show a comparison of sizes between the 1µm and 3µm designs for the three cells used in the carry chain. The cell on the left in the figures is the 1µm cell, and the cell on the right is the 3µm cell.
Figure 46: Layout comparison of 1u and 3u hstart2 cell
Figure 47: Layout comparison of 1u and 3u etree4 cell
Figure 48: Layout comparison of 1u and 3u xor2 cell

After the layout for each cell was completed they were assembled into a full chip. Figure 49 and Figure 50 show layouts of the complete 1µm and 3µm designs.

Figure 49: Complete 8HP 1um adder carry chain layout
These designs were put together in a single chip with a set of pads to enable testing. Two copies of the 1µm design were placed in the chip, and one copy of the 3µm design was used. Figure 51 shows the layout of the completed chip.

It was at this point that a problem was encountered. A design rule had been added to the 8HP design kit after its initial release that radically affected the layout strategies used previously. This design rule limited the amount of deep trench, or DT, in a 25µm by 25µm area to only 20% of that area. This was a huge problem as clever design strategies had been used to pack as many devices together as possible, to decrease wire lengths. In some of the cells used in the adder carry chain the percentage of DT was as high as 40-
50%. Obtaining a waiver for this was impossible as the high DT area would have destroyed MOS devices all over the wafer.

As stated, this rule did not exist in early versions of the 8HP kit, or in the 7HP kits used for earlier versions of the circuit. If this design rule violation had been caught early in the design process it would have been easier to correct, but this was not the case. Doing a normal DRC of an individual cell that violated this rule did not raise any errors, so catching it that way was impossible. The only check that catches this error is a check run after the chip ring is in place, which is something that is not usually run until the chip as a whole is considered finished.

This problem was discovered on the day of the tape out deadline, so drastic measures needed to be taken to correct the problem as soon as possible. The quickest solution was to expand the basic cells out such that they no longer violated the design rule. During this expansion it was discovered that having two columns of 1µm devices was the most that could fit in the 25µm by 25µm area. This meant that having a column of 2µm emitter followers along with a column of 1µm devices would violate the 20% rule. A decision was made to simply reduce the emitter follower sizes down to 1µm to fix this problem quickly.

Figure 52, Figure 53, and Figure 54 are comparisons of the new and old hstart2, etree4 and xor2 cells layouts. Notice the border around the new cells, which was put there to make the minimum spacing between cells easier to handle.
Figure 52: Comparison of old and new hstart2 cell

Figure 53: Comparison of old and new etree4 cell
These changes left a lot of empty space within the new cells. With the time constraints on finishing the chip it was decided to simply leave them empty. If there had been more time the resistors at the top of the circuit and the NFET’s at the bottom of the circuit could have possibly been moved into that space.

The new cells were then put together into a new layout. Figure 55 shows a comparison of the two completed circuits. The old version is on top and the new version is on the bottom.
The completed design was then put in a chip ring, along with another copy of it. The 3µm design was abandoned because of the time constraints. Figure 56 shows the final layout of the new completed chip.

Although the 3µm design was shelved for the tape out, some work was done on it to check the feasibility of doing the design with devices that size. The main problem that was found was that fewer 3µm devices can fit in a 25µm by 25µm area than 1µm devices due to a new maximum density design rule imposed to insure yields in the wafer fabrication process. The rule requires that the density of the DTI features in any 25µm x
25μm region not exceed 20% of the area. This means that the cells need to be spread out even more. Figure 57 shows a comparison of the old and the new 3μm etree4 cell.

![Figure 57: Comparison of fold and new 3um etree4 cell](image)

Overall, this new design rule had a large effect on design strategies for new 8HP circuits. It will make wires longer, which will impact performance negatively as wire delays already comprise over a third of the total delay in a circuit.

This rule alone has reduced the performance gains expected from more advanced SiGe processes with higher f\textsubscript{T} parameters (9HP, 10HP) since wiring delays become the dominant speed limitations with faster devices. Overall performance gains depend on both shrinking interconnect length and reduced device delay. Large DTI features and relatively widely separated devices will derail progress unless new solutions can be found. One extremely promising development involves IBM’s own research into SiGe BiCMOS fabrication on SOI wafers, as announced in the summer of 2003 in Toulouse, France. The main advantage of the SOI fabrication process is that the very large ~1μm DTIs which are required to be many microns deep are replaced with much smaller STIs which only need to be a few hundred Angstroms to less than a micron deep. Although
the commercialization of this process is a few years away, it returns to the design process the required device shrinkage to obtain the performance boosts expected by the advanced SiGe HBT technologies.

After the initial 8HP run, another layout was done of the design, to clean up some of the problems caused by the last minute changes brought about by the DT20 rule. The goal of the cell redesign was to spread the devices out within the cell, as well as allowing cells to be more easily placed next to each other. In general, the cells use the same amount of area, because they both aim to use at much DT in the area given as possible. The proportions of the cells are different, with the new cells being taller and the old cells being wider. When placed in the final circuit the width of the cells is more important as they are stacked side by side, and a smaller width means shorter long wires. Figure 58, Figure 59, and Figure 60 show a comparison of the hstart2, etree4 and xor2 cells.

![Figure 58: Comparison of 8HP hstart2 cell](image)
Figure 59: Comparison of 8HP etree4 cell
5.2 8HP Adder Carry Chain Results

The 8HP chip was designed and finally fabricated to determine what kind of speeds could be achieved with this technology. The circuit implemented the entire critical path for the sum of the most significant bits. The sum was then tied back to the input producing a ring oscillator. There are two outputs to the chip, a divide-by-four and a divide-by-sixteen of the sum signal. The period of the signal also needs to be divided in half as the period is the time through the adder twice. Figure 61 shows a picture of the layout, while Figure 62 shows a microphotograph of the fabricated chip.
Figure 63 shows the schematic simulations of the chip. The simulations of the circuit showed a time of 31ps through the adder, which corresponds to a 32 GHz speed. This compares to a simulated value of 20 GHz in 7HP. 8HP is not an exact doubling of fT from 7HP, so this speed up is around what is expected. These simulations do not take into account wire parasitics.
Figure 63: Schematic simulation of 8HP Adder Carry Chain

Figure 64 shows the measured results of the chip. Measurements of the circuit showed a time of 62ps through the adder, which corresponds to a 16 GHz speed. This compares to a measured value of 16 GHz in 7HP. One reason for this poor performance in relation to 7HP is a greatly increased chip size in 8HP. Comparing similar designs in 7HP and 8HP, the 8HP chip is around 1.6 times larger than the 7HP design.
The reason for this increased size is an inability in 8HP to pack deep trench (DT) close together because of a new design rule that limits the local amount of DT to 20%. A version of this circuit was designed prior to knowing about this rule. The chip that was actually fabricated and measured was 2.4 times larger than the chip without the DT20 rule.
5.3 Extracted 8HP Work

With the introduction of new 8HP kits extracted parasitic simulations were capable of being done. These simulations were done on several different versions of the 8HP circuits. The first of these was the initial 8HP design that used full DT sharing and did not have the DT 20 rule in place. This design showed a time of 30.2 ps through the adder, corresponding to a speed of 33.1 GHz. An extraction of the finished design showed a time of 44.5 ps through the adder, which corresponds to a 22.5 GHz speed. Figure 65 is a pie chart with the breakdown of delays.

![Figure 65: Timing of original 8HP design](image)

The second set of extracted simulations were done on the circuit with the DT 20 rule. This circuit showed a simulated time through the adder of 30.9 ps, which corresponds to a speed of 32.4 GHz. The individual cells were extracted and showed a time of 42.7 ps, corresponding to 23.4 GHz. The extraction of the final circuit showed a time of 47.7 ps, corresponding to 20.9 GHz. Figure 66 is a pie chart showing the breakdown of the delays.
5.4 8HP Adder Carry Chain Redesign

The first two version of the 8HP adder carry chain test circuits showed a disappointing lack of speed, and thus a redesign of the circuit was undertaken. The first change was to update the circuit using the latest 8HP kit. In doing so, the peak fT current for the 1 micron devices increased from around 1 mA to 1.4 mA. In previous versions of the kit the 1u device was appreciably slower then the 3u device, but in the latest version they were much closer together.

The other changes were to undo the alterations that had been made to the design to alleviate some of the problems caused by the DT 20 rule. The first alteration that was undone was to go back to the combined CML and ECL design that had been used in the 7HP designs. This moved the critical path signal to the CML part of the circuit, which uses smaller voltage swings, and allows for faster switching times.
The second alteration was to increase the size of the emitter followers. In the old design they had been reduced to 1 micron devices, but for the new design they were increased to 2 micron devices. This provided better driving capability.

The schematic simulations for the new design showed a time of 25.1 ps through the adder, which corresponds to a time of 39.8 GHz. The individual cells were then extracted and showed a time of 29.7 ps through the adder, which corresponds to a time of 33.6 GHz. Finally, the entire design was extracted and simulated, showing a time of 34.1 ps through the adder, which corresponds to a time of 29.3 GHz. Figure 67 is a pie chart that shows how each part contributes to the final speed.

![Figure 67: Timing of redesigned 8HP design](image)

Additional simulations were done on this design at a temperature of 100 C to see how high temperatures affected the speed. The schematic simulations showed a time of 27.9 ps through the adder, which corresponds to a time of 35.9 GHz. The cell extractions showed a time of 32.5 ps, corresponding to 30.8 GHz. The extraction of the final circuit showed a time of 38.1 ps through the adder, corresponding to 26.2 GHz. This shows a slow down over 10% at high temperatures. Figure 68 is a pie chart showing these times.
5.5 8HP MEASURED RESULTS

The redesigned 8HP chip was sent out to fabrication and was tested upon its return. It was included alongside another student’s circuit to both save space, as well as create a larger die to ease the difficulty of testing a small die. Shown below in Figure 69 is the layout for this design. The size of the design was 207 microns by 461 microns.

Figure 69: Layout of redesigned 8HP adder

The design was first simulated with no parasitics extracted. Figure 70 shows the waveform for this simulation, which is eight times the time through the adder. The
A simulation with extracted parasitics was then done at room temperature. Figure 70 shows the waveform for this simulation, which is eight times the time through the adder. The simulation shows a time of 277 ps which is 34 ps through the adder. This corresponds to a speed of 29.3 GHz.
Once the chip was fabricated it was tested to determine how close the simulated results predicted actual performance. The measured results are shown in Figure 71 and show a time of 300 ps which is 37.5 ps through the adder. This corresponds to a speed of 26.7 GHz. This differs from the extracted simulations by 3.4 ps, or 10%.
Simulations at higher temperature show that an increase in temperature leads to a decrease in speed. A simulation done at 85 degrees Celsius shows a time of 300 ps which is 37.5 ps through the adder, and is shown in Figure ?. This matches exactly with the measured results, which should allow for more accurate simulations in the future.

5.6 Further Optimization of 8HP circuits to maximize speed

There are further steps that can be undertaken to increase the speed of an 8HP circuit even further. The first method for doing so is to reduce the operating temperature of the circuit. The 26.7 GHz number corresponds to a extracted simulation done at 85 C, and my decreasing the temperature the simulated speed can be increased. Figure 72
shows a graph of simulated speed in GHz versus temperature. Simply cooling the circuit down to 0 C would result in a speed of 29.7 GHz, a speed increase of over 10%.

![Graph of Speed vs Temperature](image)

**Figure 72: Speed vs Temperature**

The other method for improving the speed is to make the design smaller, thus decreasing the wire delay in the circuit. As has been discussed previously, the 8HP design is substantially larger than the 7HP design because of the DT20 rule. This rule is to be removed in a future version of the design kit, which would allow for smaller designs to be fabricated. Until then, the designs can still be laid out and simulated to predict the speed up that would occur. Figure 73 shows a comparison of the etree4 cells with and without the DT20 rule. The design with the DT20 rule uses more than 3 times the area of the design without the DT rule. Simulations done at 85 C of this circuit show a speed through the adder of 30 GHz an improvement of 12% over the previous design.
These two methods can be combined for even further speed up. When the smaller design is simulated at 0 C, the resulting speed is 33 GHz, an improvement of 23% over the base design.
6. 32 BIT ADDER STRUCTURES

The previous designs all dealt with the carry chain within the adder, which is the critical path. The question still remains how this translates into an actual 32-bit adder, both in terms of speed obtained and power used. These numbers are all computed using the 7HP designs. Table 6 shows a breakdown of different design decisions in building the final adder and their impact on speed and power. The baseline design is what is presented earlier, in which a combination of CML and ECL circuit design is used, as well as the Ling Pseudo-Carry.

Table 6: Comparison of power saving design changes

<table>
<thead>
<tr>
<th></th>
<th>Delay (ps)</th>
<th>Speed (GHz)</th>
<th>Change</th>
<th>Power (W)</th>
<th>Change</th>
<th>Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>53.35</td>
<td>18.7</td>
<td>100%</td>
<td>3.1</td>
<td>100%</td>
<td>1731</td>
</tr>
<tr>
<td>ECL</td>
<td>65.7</td>
<td>15.2</td>
<td>81%</td>
<td>2.2</td>
<td>70%</td>
<td>1379</td>
</tr>
<tr>
<td>No PCLA</td>
<td>54.05</td>
<td>18.5</td>
<td>99%</td>
<td>2.5</td>
<td>81%</td>
<td>1587</td>
</tr>
<tr>
<td>ECL/No PCLA</td>
<td>67</td>
<td>14.9</td>
<td>80%</td>
<td>1.8</td>
<td>58%</td>
<td>1299</td>
</tr>
</tbody>
</table>

The other three designs show what would happen if two changes were made to the overall design. The first change is converting all the trees from a combination of CML and ECL to simply ECL gates. The main benefit of this is that it allows for a smaller power supply, which results in a lower power design. The second change is to use standard carry-look ahead instead of the Ling Pseudo-Carry. The effect of this is two fold. Firstly, the Pseudo-Carry removes complexity from the first stage of the adder,
which translates to one fewer tree per hstart2 cell. It then adds complexity into the final stage, adding an extra tree into the sum and etree2sum cells. To understand how this affects the power of the circuit, one must look at the number of each of these cells in the circuit. It turns out that there are sixteen copies of the hstart2 cell, while there are sixteen copies each of the sum and etree2sum cells. Thus this change ends up removing sixteen current trees from the design. The second change occurs by a simplification in the non-critical path circuitry. It is possible to generate fewer terms in the non-critical path without the pseudo-carry, which also leads to a power savings. The pseudo-carry circuitry requires an XOR and OR for every bit, as well as an AND for every other bit, for a total of 80 circuits. The normal carry-look ahead circuit allows the XOR gates to be used in place of the OR circuits, which reduces the total number to only 48.

Further power reduction can be obtained by reducing the current along the non-critical path. In the baseline design, the non-critical path consumes more than 38% of the total power. It is possible to reduce the current along this path to lower the total power. Table 7 presents the power numbers that would result from reducing the current in the non-critical path to one-half and to one-quarter. Simulations show that a one-half reduction in current results in 1.4x slowdown, a one-quarter reduction in a 2.1x slowdown. As the non-critical path uses one simple gate as opposed to two complex gates in the critical path, a slowdown of two times would be appropriate to match the delay of the critical path.

<table>
<thead>
<tr>
<th>Base Power (W)</th>
<th>½ current (W)</th>
<th>¼ Current (W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>89</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7: Reducing current usage in non-critical paths to reduce power
### 6.1 8HP power savings design

The 8HP design kit offers an opportunity to both increase the speed from 7HP while also decreasing the overall power usage. The highest speed device in 8HP is the 1 micron device, which is what was used to craft the highest speed adder carry-chain structure. Smaller devices that use less current at max fT are available though, so by creating a circuit based on these devices a circuit with a lower power consumption can be made. The smaller device is 0.52 microns and has a peak current of 0.74 mA.

Some basic decisions on how to build this adder needed to be made first, based on the work done in 7HP on how different circuit designs affect the power consumption. The first thing that can be seen is that using a normal carry look ahead design versus pseudo-carry look-ahead design saves 20% on the power, with only a very minimal decrease in speed. The second thing to do is use lower currents on the non-critical path. Building a complete 32-bit adder using devices this size yields an adder that uses 1.6 W of power, 60% of the power of the highest speed design. This design also runs at 24.8 GHz, which is 93% of the speed of the highest speed design. This shows that a large savings in power can be gained easily by a small decrease in speed.

Further savings in power can still be accomplished through two different methods, but both will reduce the speed of the circuit even more. The first method is to

<table>
<thead>
<tr>
<th></th>
<th>3.1</th>
<th>2.4</th>
<th>2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No PCLA</td>
<td>2.5</td>
<td>2.1</td>
<td>2.0</td>
</tr>
<tr>
<td>ECL / No PCLA</td>
<td>1.8</td>
<td>1.5</td>
<td>1.4</td>
</tr>
</tbody>
</table>

The highest speed device in 8HP is the 1 micron device, which is what was used to craft the highest speed adder carry-chain structure. Smaller devices that use less current at max fT are available though, so by creating a circuit based on these devices a circuit with a lower power consumption can be made. The smaller device is 0.52 microns and has a peak current of 0.74 mA.
from a mixed CML/ECL cell design to simply an ECL cell design. This change does not change the current through the trees, but does reduce the power supply voltage, resulting in a lowering the total power consumption.

The second method is to decrease the current for each cell in the adder, not simply the non-critical path. This will reduce the speed of the circuit, but it will also reduce the power consumption. Table 8 shows how using an ECL design and halving the current through each tree causes a decrease in both power and in speed. These results show that it is possible to get large power savings with only a small loss in speed.

Table 8: Comparison of 8HP power saving designs

<table>
<thead>
<tr>
<th></th>
<th>Speed (GHz)</th>
<th>Percentage</th>
<th>Power (W)</th>
<th>Percentage</th>
<th>Power Delay Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>High power</td>
<td>26.7</td>
<td>100%</td>
<td>2.91</td>
<td>100%</td>
<td>109</td>
</tr>
<tr>
<td>Low power</td>
<td>24.8</td>
<td>93%</td>
<td>1.55</td>
<td>53%</td>
<td>63</td>
</tr>
<tr>
<td>Low power (ECL)</td>
<td>21.7</td>
<td>81%</td>
<td>1.05</td>
<td>36%</td>
<td>48</td>
</tr>
<tr>
<td>Half power</td>
<td>19.4</td>
<td>73%</td>
<td>0.86</td>
<td>29%</td>
<td>44</td>
</tr>
</tbody>
</table>
7. 7HP Test Chip

The previous work done with the adder carry chain, and even the full adder, was the first step in building a microprocessor core, along with work done by a fellow student on a high speed register file. A microprocessor test design using the full adder circuitry was made in 7HP and fabricated. The block diagram of the fabricated microprocessor test design is shown in Figure 74, its layout in Figure 75 and a microphotograph of the chip in Figure 76.

![Diagram of 7HP CPU core control and datapath](image)

Figure 74: The flow diagram of the control and datapath on the 7HP CPU core
Figure 75: Layout of the 7HP processor core, with modules identified and labeled
There have been 2 different versions of the CPU core chip, both done in 7HP. The first chip had some design errors in it that prevented it from functioning. The second chip was a simplified design that allowed for testing of the ALU and register file separately.

The test chip has a pipeline of 5 stages. The first two stages are contained within the register file. The first is used for address decoding while the second is used for reading and writing. The third pipeline stage is made up of a 2:1 MUX and a latch. The
MUX is used to modify the data path to test different pieces. The fourth pipeline contains the ALU. The fifth pipeline is made up of a 2:1 MUX and a latch.

The chip has several configuration bits that can be set to change how it functions and what outputs are displayed. The simplest configuration bits are those that determine which register in the register file is used in the other tests. There are 3 bits for this, corresponding to the 8 registers.

One control signal is used to determine whether an internal VCO or external clock is used to clock the rest of the circuit.

Two control signals determine what output is displayed. The output comes from the latch following the adder and preceding the register file. The control signals determine whether the 0th, the 1st, the 15th or the 31st bit of output is displayed. This allows us to find out if the adder is in fact adding correctly.

Two more control signals are used to determine the data path for the chip, although only three modes of testing are used for the test chip. The first mode of test lets the adder be tested separately from the register file. The output of the adder is routed back to one of the inputs of the adder. The other input of the adder is set as a static value of 1. This allows the value to be continually incremented, resulting in a periodic waveform on the output that can be measured. Changing which bits are viewed on the output will show that the adder is working as intended.

The second test mode allows the register file to be tested separately from the adder. The output of the register file is fed back to the input inverted through a latch. This causes the value in the register file to be continually inverted, producing a periodic
waveform on the output. The configuration bits can be set up so that each register in the
register file can be tested separately, to find out if they all work. Four different bits in
each register can be checked by using the different output bits.

The third and final mode of testing combines the register file and adder together in
the data path. A value is read from the register file and sent through a MUX and a latch
to the ALU. The ALU adds 1 to the value and then passes the result to another MUX
and latch. The output of the latch is then sent back to the register file to be written. The
new value of the register is then read and the process continues. With the other testing
bits available, each register in the register file is able to be tested. The output bits can
also be configured to ensure that addition is taking place.

Testing of the test chip

There were issues involved with testing the test chip. Some portions of the chip
worked, while others did not.

One component that did not work correctly was the divide-by-sixteen clock
output. The circuit was designed to take either the on board clock or the external clock
and output it sixteen times slower. The purpose of this was to find out the speed the
design was running at, as well as to provide a triggering signal for the other outputs. It
did so by using four master-slave latches in serial. This circuit did not behave as
expected, and nothing appeared on the output. Simulations of the circuit do not show
this behavior. This problem has also been observed in other divide-by circuits designed
around the same time. Newer designs have shown that adding a buffer in between each
master-slave latch will alleviate this problem and allow for correct operation of the
divide-by circuit. The failure of this circuit complicates further testing of the chip, but
does not preclude it. It particularly makes using the onboard VCO difficult because there is no external gauge of how fast it is running.

The problems with the clock output meant that using the external clock was the best option for testing the remainder of the chip. As stated previously, the chip contains both an adder and a register file, along with data path elements to test them separately and together. The results of each test will be described here.

7.1 Adder Only Test

The test of just the adder and one multiplexer was the first test. The results are summarized in Table 9.

<table>
<thead>
<tr>
<th>Clock Rate (MHz)</th>
<th>Output Frequency (MHz)</th>
<th>Pipeline Stages</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>166</td>
<td>1.5</td>
</tr>
<tr>
<td>1000</td>
<td>333</td>
<td>1.5</td>
</tr>
<tr>
<td>1400</td>
<td>466</td>
<td>1.5</td>
</tr>
<tr>
<td>2000</td>
<td>500</td>
<td>2</td>
</tr>
<tr>
<td>2500</td>
<td>626</td>
<td>2</td>
</tr>
<tr>
<td>3000</td>
<td>750</td>
<td>2</td>
</tr>
<tr>
<td>3500</td>
<td>880</td>
<td>2</td>
</tr>
<tr>
<td>4000</td>
<td>800</td>
<td>2.5</td>
</tr>
<tr>
<td>5000</td>
<td>820</td>
<td>3</td>
</tr>
</tbody>
</table>

The correct number of pipeline stages for this test is 2, which is seen for speeds from 2 to 3.5 GHz. Simulations of this design all show 2 pipeline stages, from 0.5 GHz up to 16 GHz. For speeds less than 2 GHz, the chip is actually working faster then
predicted and planned. The most likely reason for this is that one of the pipeline stages is being skipped in only one direction. This conjecture is not born out from the simulations, as they all show 2 pipeline stages for those speeds. These simulations are done without wire parasitics included, as the chip size is too large to simulate fully with them.

For speeds greater then 3.5 GHz, the chip begins to operate slower then expected. Even if the speed of the external clock is increased up to 16 GHz, the output frequency stays constantly in the 800-900 MHz range. This is very different then the simulated results, which show the adder working at speeds up to 16 GHz without problem.

When the circuit is switched from the external clock to the internal clock, the frequency of the output stays in the same 800-900 MHz range. This fact makes it impossible to tell what the exact frequency the internal VCO is operating at, as it is only tunable over a 12-16 GHz range.

The result of this test is that the adder component can be said to be working at speed up to 3.5 GHz, well below the desired goal of 16 GHz.

### 7.2 Register File Only Test

The second component tested was the register file alone. The results from this test are summarized in Table 10.

<table>
<thead>
<tr>
<th>Clock Rate (MHz)</th>
<th>Output Frequency (MHz)</th>
<th>Pipeline Stages</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>83</td>
<td>3</td>
</tr>
<tr>
<td>1000</td>
<td>166</td>
<td>3</td>
</tr>
<tr>
<td>1500</td>
<td>250</td>
<td>3</td>
</tr>
<tr>
<td>Speed</td>
<td>Output Frequency</td>
<td>Pipeline Stages</td>
</tr>
<tr>
<td>--------</td>
<td>------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>2000</td>
<td>333</td>
<td>3</td>
</tr>
<tr>
<td>2500</td>
<td>415</td>
<td>3</td>
</tr>
<tr>
<td>3000</td>
<td>500</td>
<td>3</td>
</tr>
<tr>
<td>3500</td>
<td>580</td>
<td>3</td>
</tr>
<tr>
<td>4000</td>
<td>665</td>
<td>3</td>
</tr>
<tr>
<td>4500</td>
<td>750</td>
<td>3</td>
</tr>
<tr>
<td>5000</td>
<td>820</td>
<td>3</td>
</tr>
<tr>
<td>6500</td>
<td>720</td>
<td>4.5</td>
</tr>
<tr>
<td>7000</td>
<td>700</td>
<td>5</td>
</tr>
</tbody>
</table>

The correct number of pipeline stages for the register file is 3, which is seen for speeds up to 5 GHz. These results match the simulations done up to 5 GHz. For speeds greater than 5 GHz, the simulations again predict 3 pipeline stages, but these are not born out by the measurements. For speeds greater than 5 GHz up to 16 GHz, the measured output frequency stays constantly in the 700-800 MHz range. The conclusion that can be drawn from these results is that the register file component of the test chip works at speeds up to 5 GHz.

The register file included in the test chip was very similar to previous registers that have been tested at speeds up to ?? GHz. The only difference was the use of NMOS devices at the bottom of current trees to set the current through the trees, as opposed to a bipolar current mirror. These current trees have been used successfully in newer design in 8HP, so their inclusion is not thought to be a factor in the reduced speed. There is a separate chip designed by another student that uses this same register file, but it has not been tested as of yet.
Switching from the external clock to the internal clock again shows an output frequency between 700-800 MHz. No appreciable difference is seen within the tuning range of the VCO.

7.3 Adder and Register File Test

The final test contained both the adder and register file together in series. The results from this test are summarized in Table 11.

<table>
<thead>
<tr>
<th>Clock Rate (MHz)</th>
<th>Output Frequency (MHz)</th>
<th>Pipeline Stages</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>55</td>
<td>4.5</td>
</tr>
<tr>
<td>1000</td>
<td>110</td>
<td>4.5</td>
</tr>
<tr>
<td>1500</td>
<td>166</td>
<td>4.5</td>
</tr>
<tr>
<td>2000</td>
<td>200</td>
<td>5</td>
</tr>
<tr>
<td>2500</td>
<td>250</td>
<td>5</td>
</tr>
<tr>
<td>3000</td>
<td>300</td>
<td>5</td>
</tr>
<tr>
<td>3500</td>
<td>316</td>
<td>5.5</td>
</tr>
<tr>
<td>4000</td>
<td>360</td>
<td>5.5</td>
</tr>
<tr>
<td>4500</td>
<td>405</td>
<td>5.5</td>
</tr>
<tr>
<td>5000</td>
<td>380</td>
<td>6.5</td>
</tr>
<tr>
<td>6000</td>
<td>380</td>
<td>7.5</td>
</tr>
</tbody>
</table>

The correct number of pipeline stages is 5. This number of pipelines is seen from 2 GHz to 3 GHz only. For speeds less than 2 GHz, the same issue as the adder alone is seen, where one of the pipeline stages is skipped in one direction only. Simulations of this chip show 5 pipeline stages, up to speeds off 16 GHz. Measured results with a clock rate over 3 GHz vary in speed between 300 and 400 MHz. The
results here show that when combined together the maximum speed of this design is 3 GHz.

7.4 Voltage Droop Analysis

A tool named Voltage Storm became available for use after the test chip had been made. This tool allowed for the analysis of voltage droop in a circuit. Figure 77 shows the color key, Figure 78 shows the voltage droop on the ground wires and Figure 79 shows the voltage droop on the power wires.

![Table of Voltage Droop Analysis](image)

Figure 77: Color key of voltage droop
Figure 78: Voltage droop of ground wires

Figure 79: Voltage droop of power wires
In general, areas that are purple, blue and green are within an acceptable range of voltage drop, while those in yellow, orange and red indicate areas where voltage drop should be addressed. In the test chip, these problem areas occur in two areas: within the register file, and within the clock buffering circuitry. In future designs these areas can be fixed with the addition of more connection to power.

These voltage drops, particularly in the clock buffer, are a possible cause for the poor performance seen when measuring the chip. However, extracted resistor simulations run at higher speeds do not show any problem.
8. 3D Design

The majority of the 3D work was done very early on, during which 5HP was the main kit that was being used. The idea was to bond 2 wafers face to face and then thin the backside of one of the top wafer. More wafers could then be bonded onto the top wafer and thinned as well, stacking up several wafers if it was needed. Vias would be etched down from the top and filled to connect the various wafers.

One of the major design decisions that we were concerned about was what type of via to use to connect the wafers. The two basic styles of vias were bridge and column vias. In the bridge via, two separate wafers would be connected together on the top level using 2 vias. Each wafer would have a metal landing area that would be used to stop the etching process, as well as connecting to the via. The main advantage of this style was simplicity, since both wafers landing pads would look the same. The disadvantage was more metallization on the top level was needed, taking up space and adding another processing step.

The other type of via was the column via. In this case the landing pads for each wafer were situated above each other when stacked. The landing pad on the top chip had a cullet in the center, which allowed the etching to continue through to the bottom wafer. After etching, the via would be filled, connecting the two wafers together. The advantage here was no extra metallization on the top level, and a shorter connection between the wafers. The disadvantage was more precise alignment was needed to connect the wafers.

To determine which type should be used a test structure featuring both sets of vias was created. As well as having the two different types of vias the test structure also
contained different via sizes, to determine what a good size to use for them was. In addition, different lengths of via chains between test pads were used, to see how reliably the via connections could be made.

The test chain structure was laid out in Cadence using 2-3 levels of metal and 1 level of contact. To save on the cost of photo-lithography plates the metallization for both the top and bottom wafer were done on the same chip. This was done by doing a standard layout on the left half of the chip, using 2 levels of metal to represent the metallization on both wafers. Once that layout was finished, the metal layer on the top chip was copied and mirrored onto the right half of the chip. It was then changed to the metal for the bottom chip. This process allowed for only 1 version of wafers to be made, which could act as both the top or the bottom wafer in a wafer stack. While in a commercial chip this would be wasting a lot of space, in a test chip it works out very well.

The major benefit of designing the adder in 3D is the reduction of between each stage. If each pair of bits is placed on alternating layers the wire length can be halved. This arrangement can be seen in Figure 80.

![Figure 80: 3D adder arrangement](image-url)

105
This reduction in wire length will lead to a corresponding reduction in delay through the adder. Figure 81 shows how 2 level and 4 level 3D would effect the adder time for several different technologies.

![Diagram showing wire delay in 3D](image)

**Figure 81: Comparison of wire delay in 3D**

An adder using this 3D scheme was made using the 5HP technology, for a 2 layer process. Although it was fabricated, it was never tested as the wafer bonding process was not fully developed. Figure 82 shows the layout for this design.
Figure 82: 3D adder layout in 5HP
9. Discussion and Conclusion

The adder is one of the key components of a CPU, and is one of the limiting factors in CPU speed. By using Silicon Germanium bipolar technology a 32-bit adder can be made that will run at 26 GHz, with even faster speeds possible. In combination with a register file and a data path a more complete CPU will be designed in the future.

This thesis describes in detail how the carry chain of the adder can best be constructed using a combination of CML and ECL designs. These designs are optimized for maximum performance. Additional work is done expanding the carry-chain into a full 32-bit adder. Ways to greatly decrease the power consumption with only modest decreases in speed are also shown.

Future efforts will entail further layout improvements, along with cooling, to increase adder speeds up to the desired 32 GHz. With the release of new design kits, a design decision will also need to be reached whether to further increase the speed of the adder, or to decrease the power usage while maintaining the same speed.
REFERENCES


APPENDIX A: SiGe HBT Microprocessor Core Test Vehicle

Paul M. Belemjian, Okan Erdogan, Student Member, IEEE, Russell P. Kraft, John F. McDonald, Senior Member, IEEE

SiGe HBT Microprocessor Core Test Vehicle

Abstract—A major impediment to the continuation of Moore’s Law in the years to come is the performance of interconnections in integrated circuits at high frequencies. Microprocessors are using a greater portion of their clock cycle charging and discharging interconnections. Silicon-Germanium (SiGe) Heterojunction Bipolar Transistors (HBT) provide a fast track technology for the exploration of the effect of interconnections on high-speed computer design. Industry has pursued Low-k dielectrics to decrease wire capacitance. Cu metallization has been used to reduce wire resistance which becomes important as the wire dimensions are scaled down. These are not the only issues for high frequency interconnections. Some other high frequency issues include coupling, transmission line propagation, skin effects, dielectric and substrate loss. These phenomena cause signal attenuation, noise and dispersion in addition to delay. In the limit of zero device delay, interconnection delay will remain in addition to these problems. Wire shortening has been possible using more layers of interconnections, but this approach may be reaching its limit. An unconventional approach, three-dimensional integration, attempts to shorten wiring through increased circuit component placement flexibility. The approach considered here for 3D integration uses wafer-to-wafer aligning and bonding, wafer thinning and deep, high-aspect ratio Cu vias formation. This provides an intimate interconnection between CPU components, and extremely wide path to memory that would be infeasible in conventional or Multi-Chip Module packaging. This combination of SiGe HBT BiCMOS and 3D chip stack technologies enables small computing engines in the 16 to 32 GHz range.

Index Terms—Bipolar digital integrated circuits, Current Mode Logic, Emitter Coupled Logic, Heterojunction bipolar transistors, High-speed integrated circuits, Full wafer 3D integration

INTRODUCTION

A significant challenge in the design of high speed SiGe circuits is interconnection delay. Wire parasitics, primarily in the form of resistance, capacitance, and inductance, are important considerations to optimize circuit performance. Historically the industry used low-k dielectrics to reduce wire capacitances. Copper (Cu) metallization has been used to reduce wire resistance. However there are limits to the improvement in performance of the integrated circuits using Cu/low-k [8]. A different approach to reducing interconnection parasitics is three dimensional (3D) integration. [9, 10, 11]

A 3D microprocessor test vehicle could demonstrate the device speed advantages of a SiGe design through the mitigation of interconnection parasitics. The processor test vehicle discussed here consists of an adder, register files, and a finite state machine working in tandem at speeds of 8 GHz in IBM’s 5HP

Manuscript received July 31, 2004. This work has been sponsored through an IBM subcontract supported by the DARPA IBM HPCS program; contract number HBCH30390004 and the DARPA IBM 3DI program, contract number: N66001-04-C-8032. Additional support was provided by the DARPA/MARCO Focus Research Center on Interconnections with matching NY State funds. Additional fellowship support from SUN Corporation is gratefully acknowledged.

P. M. Belemjian is with the Electrical Computer and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, NY 12180 USA (phone: 518-276-2513; fax: 518-276-8761; e-mail: belemp@rpi.edu).

O. Erdogan is with the Electrical Computer and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, NY 12180 USA (e-mail: erdogo@rpi.edu).

R. P. Kraft is with the Electrical Computer and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, NY 12180 USA (e-mail: kraftr2@rpi.edu).

J. F. McDonald is with the Electrical Computer and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, NY 12180 USA (e-mail: mcdonald@unix.cie.rpi.edu).
SiGe technology, speeds of 16 GHz in IBM’s 7HP SiGe technology and higher in IBM’s 8HP SiGe technology. These processes have HBTs that have maximum cut-off frequencies (fT) of approximately 47 GHz in 5HP, 120 GHz in 7HP and 210 GHz in 8HP. Future technologies may have a fT as high as 375 GHz [5] which would allow microprocessor speeds in excess of 32 GHz. To obtain the full benefit of these faster devices, it would be necessary to use 3D integration to reduce the impact of interconnections.

SiGe HBT

Silicon Germanium (SiGe) Heterojunction Bipolar Transistors (HBT) were chosen as the technology to demonstrate a high speed microprocessor core that would also benefit from the interconnection advantages of 3D. Historically, bipolar devices produce a circuit with 2.5 times the speed of the comparable CMOS circuits, assuming the same level of photolithography is used in both technologies [13]. The IBM SiGe BiCMOS processes were chosen in particular. In conjunction with their high speed they offer relatively good yields compared to other bipolar processes and good current driving capability. They do not include new low-k dielectric materials, but do use Cu for a minimized interconnection related delay. These processes provide a means of demonstrating the approximate performance enhancement attainable from 3D interconnections. The yield of the process enables us to demonstrate more complex circuits, which may be incorporated in a microprocessor. This extra complexity allows the 3D processor test vehicle to contain interconnection lengths which approximate some of the critical paths in more realistic processor cores. The interconnection limitations of a full size computational device are significantly greater and would lend themselves even more to 3D integration.

SiGe HBTs are bipolar transistors grown on a silicon substrate that exhibit speed advantages for various reasons. The transistors are actually not heterojunction bipolar transistors but rather base graded transistors [14]. The alloy grading present in the base results in a valence band offset between the SiGe and the silicon interface. The valance band offset helps to confine holes in the SiGe layer, thus reducing the reverse injection of holes from the emitter region into the base region during the forward bias of the transistor [15]. The presence of Ge in the base layer also contributes to speed improvements by straining the Si layer; the increased atomic spacing allows for higher electron mobility. The net result is that SiGe HBTs offer a lower barrier to electron injection into the base than a similarly configured conventional bipolar junction transistor. Some of this can be traded for lower base resistance. The built in field resulting from Ge grading also enhances speed.

SiGe HBTs speed, yield, linear characteristics and high gain make it very suitable for many applications. The current applications of SiGe are mainly in the telecommunications industries where high frequency analog and mixed signal circuits are used. As a consequence, contemporary SiGe technology has been somewhat optimized for analog applications. By grading Ge concentration in the base fully across the neutral base; HBT transistors have a high Early voltage with a relatively high cut-off frequency (fT).

The original IBM SiGe HBTs were designed for mainframe computers. Therefore initially they were optimized for that application. The digital profile placed the grade across the highest base doped region to achieve the highest fT [16]. As a result SiGe HBTs have been capable of achieving fT in excess of 100 GHz since 1993 [17]. Breakdown voltage tends to limit fT, but numerous device and process improvements have permitted the SiGe HBTs to circumvent some of the implications of the Johnson relation.

The contemporary SiGe HBT satisfies a blend of analog and digital requirements, but its readily available speed advantages made it a prime choice for our study. One cautionary note, however, is that bipolar device depends more on vertical rather than horizontal scaling for improved speed. This makes extrapolation of the results for some future advanced CMOS less direct.

Current Mode Logic (CML)

CML is a logic family with significant speed advantage. In the design of larger non-differential high frequency circuits, principally CMOS, switching of all the devices in the circuit causes simultaneous switching noise [18]. This noise is a result of the current flowing through the circuit from the power...
supply to ground changing abruptly. Any inductance in the power and ground rails creates a back electromotive force which appears throughout the system as noise. One approach to noise reduction in CMOS is to determine the worst-case noise scenario when the greatest current change occur, and providing that amount of noise margin in the circuit. However, the problem’s solution space has been found to be exponentially as complex as the inputs to the problem [18]. Increasing the number of power and ground connections to the chip is another technique. A bypass capacitor in parallel with the power rails also minimizes the switching noise.

However, at very high speed these approaches become problematic because the current changes too abruptly. Instead CML employs a current steering approach to represent a logic state rather than current switching. Noise mitigation by an order of magnitude is possible in CML [19]. The full differential signaling in CML further enhances the noise immunity of the circuit.

It is also possible to combine traditional CML circuits with other logic families such as Emitter Coupled Logic (ECL). Such designs take advantage of some of the benefits of CML while incorporating the higher fan-in that ECL allows. An example of this is shown in Figure 1. The circuit on the left shows a standard CML circuit while the circuit on the right shows a combined CML and ECL circuit. This compromises the full-differential input but decreases the height of the tree and reduces the total number of devices used in the tree.

![Fig. 1. Comparison of CML and combined CML/ECL circuit.](image)

### SiGe Test Vehicle

The processor core is a 32 bit system built around the ALU and the register files. The processor core follows the Harvard Architecture that stores instructions and data in separate memories to improve performance and allows single cycle execution. The processor core employs Reduced Instruction Set Computer (RISC) principles with instructions to test the datapath while limiting test and control complexity. Test patterns are generated on chip to avoid the cost of high frequency off-chip testing equipment. The instruction repertoire involves only a few arithmetic operations and branch operations.

The processor core does not include the hardware required to do higher level cache loads and stores, as these are not part of the present study. Instead two L0 cache files are used. Many of the advanced features found in modern processors, such as a scheduler, or memory management are part of a continuing study.

In order to verify the feasibility of a 16 GHz processor core using IBM’s 7HP SiGe process, emphasis has been placed on the critical timing paths of the ALU and the register file, since these are often the limiting factors in processor speed. It has been reported in the literature [20] that a 32 bit CMOS processor core achieved a clock frequency of 5 GHz with the register file as the limiting factor.

The processor has four main stages; the instruction fetch stage, the instruction decode stage, the execution stage and the write stage as shown in Table 1. These stages of the processor have different levels of pipelining. The instruction fetch stage takes place in three pipeline stages, due to pipelining present in the register file. In the two pipeline stages used in the register file, the first stage is used for decoding and the second stage is used for memory access. The decode and execution stages take place in
one pipeline stage each, and the write stage takes three pipeline stages. Consequently the pipelined processor has a total of eight stages.

<table>
<thead>
<tr>
<th>Table I</th>
<th>PIPELINE STRUCTURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instruction Fetch</td>
<td>Decode</td>
</tr>
<tr>
<td>ALU operation</td>
<td>Inst. Fetch</td>
</tr>
<tr>
<td>Branch operation</td>
<td>Inst. Fetch</td>
</tr>
</tbody>
</table>

This compares favorably to the sixteen integer stages found in the IBM’s PowerPC 970 or the twenty integer stages in the Intel’s Pentium 4. These processors aren't directly comparable due to the advanced features found in both 970 and the Pentium 4, but we believe it highlights the possibility of including a fast 16 GHz execution core of a similar design in these processors, if not in other applications.

**Controller**

The role of the controller is two-fold. It is present to allow for datapath multiplexing at clock speed, and it allows us to reduce the required pad count to control the chip effectively. The design limits the number of stages in the critical path, because the controller needs to be able to recognize a particular previous state and inputs at the chip’s clock rate of approximately 16 GHz in a 7HP. Using the 7HP process, a buffer, the minimum size unloaded circuit, has a mean rise time of 14 ps. Ideally this would allow for five stages of logic per pipe, but this assessment is unrealistic due to loading effects. Additional margin is required to compensate for process variations in chemical mechanical polish steps (which affect parasitic capacitances), and temperature variations. As a result, the controller was designed with a stage budget of three CML stages. This was achieved by pseudo-pipelining the states, and the use of non-differential CML gates, which have much greater fan-in. The pseudo-pipelining of the controller was feasible since the state progression isn’t random. It is actually quite predictable since the main goal of the controller is to set the processor test vehicles components in a predefined state so as to place a specific test vector on the datapath for at-speed verification.

**32-Bit Carry Look-Ahead Adder**

The 32-bit adder is designed using a modified carry look-ahead design. The design allows for the adder to achieve a performance level at which interconnection limitations in high frequency technologies become apparent. As discussed in section III, the adder employs combination of CML and ECL logic trees. The CML parts of the trees are used for signals along the critical path, while the ECL parts are used for less critical signals that need higher fan-in. The carry chain of the adder is made up of three logic blocks, and one final sum logic block. The improvement achievable from 3D integration increases significantly as faster technologies with smaller minimum features size are utilized. Wire delay becomes an increasingly large amount of the total delay, as shown later in Figures 15 and 16.

Figure 2 shows a circuit utilized in the adder that illustrates this mix of CML and ECL. The critical path signals use the CML part of the trees to have a smaller voltage swing, while the non-critical path signals use the ECL part of the trees for their higher fan in. The circuit also uses a wired-and structure to create a more complicated logic function in a lower tree height to reduce power dissipation.

**The Cache**

Each L0 cache is a static random access memory (SRAM) register file array with sense amplifiers. The size of this 2D memory is dictated by the requirement for read and write access time to match the clock
rate. The memory has been split into two blocks separated by support circuitry to reduce word line loading. A 3D solution however permits stacking of SRAM blocks sharing common support circuitry to increase the size of each L0 memory.

The shared support circuits can be distributed on different levels for further layout compaction. Similar comments can be applied to bit line shortening, but it was not needed for this test core demonstration. The 3D process allows the use of a variety of previously impractical memory layout configurations to minimize these interconnection delays.

The 3D process would permit us to double the capacity of memory banks when placed one on top of the other. The prime advantage of this configuration is more memory may be integrated into a limited chip area, and the total length of the buses interconnecting the various banks is reduced.

**Implementation**

A good understanding of the relationship between simulated designs and the fabricated chip must be established to accurately predict the performance of a 3D chip through simulation. In consequence, the 2D implementations of the main components were first designed and fabricated in the 5HP process. With the availability of 7HP fabrication the main components were redone with the new technology. The microprocessor test core using these components in 7HP has been designed and fabricated. The block diagram of the fabricated microprocessor test design is shown in Figure 3, its layout in Figure 4 and a microphotograph of the chip in Figure 5.
Adder Carry Chain Ring Oscillator Test

One of the key tests to validate the 16 GHz speed is to isolate sections of the core such as the adder carry chain and test them separately. The output of the carry chain can be tied back to the input to create a ring oscillator for ease of testing.

The carry chain was initially simulated directly from the schematic. Figure 6 shows the waveform generated from this simulation. This signal, and all later signals, are four times the time through the adder. This simulation shows that it takes 39ps for a signal to traverse the entire carry chain.

An extracted simulation with parasitic resistors and capacitors was done to provide an idea of how the manufactured circuit would perform. Figure 7 shows the waveform from this simulation. The time through the carry chain was 56ps, meaning that the wiring contributed 17ps to the total delay.

Figure 8 shows the measurement of the carry chain. The time through the adder was 57.5ps, only a 1.5ps difference from the extracted simulation.
The difference between the extracted simulation and the measured waveform was only three percent, giving us great confidence in our ability to accurately predict the performance of a fabricated circuit based on extracted simulations with parasitics.

**Register File**

The L0 memory cell structure used in the CPU core was that of a register file which is also isolated for testing. Figure 9 shows simulation of the register file operating at 16 GHz. During this simulation first the addresses activate, then writing is enabled to store data to the registers. Upon storing the information the data output is observed as a periodic signal as the read addresses are incremented continuously. For a period of time in the middle of the simulation the addresses are stopped to observe the effects on the output signals. This simulation includes parasitic effects on the wires along the critical path.
CPU Core

Figure 10 shows the CPU core running through a simple instruction, incrementing a register by 1. Data flows from the register file through the adder, and back to the register file.
Fig. 10. Register file, adder and data path simulation results operating at 16 GHz.

After the write enable signal goes high, the output for the register file changes two cycles later. It then takes one cycle for the data to get the input of the adder. The adder takes one cycle to add one to the number and produce the result on its output. One cycle later the data is at the input of the register file, waiting for the write enable signal so that it can be stored. The figure shows both halves of the differential signals for the data, so the crossing point shows when they have switched. The simulation shows that the whole CPU is operating at nearly the desired speed.

**Three Dimensional Integration**

The 3D platform on which a 3D processor test vehicle will be built is based on 3D interconnection using wafer stacking. There are other approaches to 3D integration such as die/chip level stacking. These are primarily based on perimeter interconnection, limiting their contribution to interconnection improvements [3]. Although other approaches exist for 3D integration, the wafer stacking method is chosen due to its ability to support multi-level 3D stacking and its compatibility with most standard wafer processing. Circuit level 3D integration is made possible by advances in wafer bonding and alignment, in conjunction with high aspect ratio etching and filling.

The 3D process uses an additional back-end-of-line (BEOL) process. A conceptual cross section of a processed wafer is presented in Figure 11. The first step in the process is to ensure that the wafers are planarized. Wafer planarity is essential since the vias connecting the two wafers need to be of uniform length. Therefore a uniform distance between wafers is required. A small degree of unevenness is acceptable as long as it is comparatively small to the 3D via length. The wafers are subsequently aligned and bonded. The alignment accuracy of the wafer directly affects the smallest allowable pitch of vertical vias. Large vertical vias would adversely affect the compactness of a design and increase the length of the 2D interconnections present in a design. A large via pitch would make it impractical to integrate circuits.
using a high via density. A two sigma wafer-to-wafer alignment of less than one micron is possible, and a four micron 3D via size has been used. The wafer bonding is accomplished by gluing the two wafers together face to face. The glue used currently is a dielectric polymer: Benzocyclobutene (BCB). BCB is thermally stable up to 350°C [10], which is too low for soldering using industry standard materials. The back of the top via in the stack would then be thinned. This thinning reduces the length of the vertical vias and ensures that the interconnection gains of the 3D integration are not lost due to the processing. Once thinned the vias are created using high aspect ratio etching and then filled. The via etch process creates holes through the wafer stack. These holes are then filled with the metal. The last step in the process is passivating the top of the wafer stack.

The added complexity of the 3D process poses thermal, reliability and yield issues, but these issues can be managed. Thermal management may be achieved using approximately 10% of the chip area for inter-wafer aluminum thermal vias [21]. These vias allow heat from the bottom wafer to be drawn to the top of the wafer stack to the heat sink and heat spreading area.

The yield of the 3D process is limited by two main factors: the 3D bonding process and the yield of the constituent the wafers. The 3D process’s yield is still under investigation, but it is expected to achieve reasonable yields since many of the processing steps involved in 3D integration involve mature technologies. Therefore the yield of the conventional wafer processing affects the final yield, since two good reticle sites need to be matched to produce a good 3D chip stack. Assuming a perfect yield on the 3D BEOL process, the overall yield would be the product of two conventional wafer processing yields. This effect is mitigated because the area of the circuitry on each layer is somewhat smaller in 3D than the total area of the 2D circuit which would improve the yield of the conventional process. Redundancy and repair can be used to improve the wafer die yield in certain cases such as for memory chip stacking.

3D integration would also help microprocessors access memory on and off chip faster and easier. One advantage would be the shortening of wires needed to connect the memory and the microprocessor by using 3D vias instead of long wires. Another advantage comes from the ability to have many 3D vias connecting two wafers. Normal chip-to-chip communication is often limited by the number of pins available. 3D interconnections can do away with this pin requirement, allowing for much more parallelism when accessing memory. Figure 12 shows a conceptual four-wafer stack with a microprocessor core on one wafer and memory on the other three.
3D Integration Decisions

A major design choice in the implementation of the 3D circuit was the choice between micro and macro integration. A micro-integration approach makes extensive use of vertical vias and takes full advantage of 3D path optimization continuously. A macro-integration approach involves the design of each level optimized only in 2D macros, assembling them in 3D with limited vertical vias. Figure 13 is an example of a micro-integration approach to the adder carry-chain. The adder carry-chain is divided evenly between the two wafers. The top half of the circuit is meant to be used on the bottom wafer, while the bottom half of the circuit is meant to be used on the top wafer. Figure 14 shows how separating the design into two levels shortens the longest wire. This type and amount of wire shortening was used when making the estimates about performance improvements in 3D, as seen in Figures 15 and 16.
Fig. 13. 3D adder carry-chain.

Fig. 15. Comparison of estimated wire and device contributions to 32-bit adder time in IBM’s 5HP, 7HP, and 8HP SiGe processes, using conventional 2D, 2 level 3D, and 4 level 3D approaches.
Fig. 16. Comparison of estimated wire and device contributions to 32bit register file time in IBM’s 5HP, 7HP, and 8HP SiGe processes, using conventional 2D and 2 level 3D approaches.

The limited size and scope of the test vehicle permits 16 GHz clock operation in 2D. The pipelining of the design permitted confinement of the major intercircuit wire effects to the stage dedicated to datapath control, where it could be handled readily. But 3D integration has been found to permit roughly 20% faster operation. Rent’s rule [9, 11] was used to quantitatively understand the issues involved, but the task of realizing this improvement falls to the designer through the use of circuit layout and simulation for verification.

The choice of micro-integration introduces other design considerations and issues. Micro-integration greatly increases via density of the circuit; consequently the size of the chip might be increased due to the size of the vias. Currently 3D integration research has demonstrated via sizes around 1-2 µm. Micro-integration may decrease the yield due to the high via density. The effect is dependent on the maturity of the 3D process. Micro-integration may increase the local thermal budget as a result of a high density of similar high-speed circuits. Limited application scope is another drawback of micro-integration since certain types of circuits do not benefit from 3D integration.

A conventional adder and register file [22] using IBM’s 5HP SiGe technology has been demonstrated at frequencies of 5 GHz. In IBM’s 7HP SiGe technology an adder is shown above at frequencies of 16 GHZ with parasitics, and a register file is shown at frequencies of 16 GHz with parasitics. These measured results are next extended with layout and simulation of device and wire delays to evaluate 3D. Performance for adder modules using different SiGe technologies and different 2D and 3D approaches is shown in Figure 15. Estimates for the register file are shown in Figure 16. These estimates for 3D wire delay are based on the 2D delay and estimated wire shortening as shown in Figure 14.

10. Conclusion

In the limit of extremely short device delay, microprocessor performance becomes interconnection limited. SiGe HBT technology enables the development of microprocessor core circuitry suitable for exploration of these interconnection limitations and evaluate methods for mitigating them. One of these approaches is 3D integration which can provide a way to shorten interconnection lengths. Using IBM’s 7HP SiGe HBT, 16 GHz operation of simple RISC core components has been demonstrated in 2D with perhaps another 20% improvement possible in 4 layer 3D by wire shortening. More advanced SiGe HBT processes such as 8HP offer still more speed but 3D stacking will become more important to harvest this speed.

ACKNOWLEDGMENT

This work has been sponsored through an IBM subcontract supported by the DARPA IBM HPCS program; and the DARPA IBM 3DI program. Additional support was provided by the DARPA/MARCO Focus Research Center on Interconnections with matching NY State funds. Additional fellowship support from SUN Corporation is gratefully acknowledged. This work relied upon QuickCap from Magma Design.
Automation for high accuracy interconnect modeling analysis.

REFERENCES


Paul M. Belemjian was born in Albany, NY in 1977. He received a B.S.C.E from Rensselaer Polytechnic Institute, Troy, NY in 1999. Currently he is pursuing his PhD degree in computer and systems engineering at Rensselaer Polytechnic Institute. His research interests include adder design using high-speed bipolar circuits.

Okan Erdogan (S’02) was born in Sivas, Turkey in 1977. He received a B.S.E.E. from Hacettepe University, Ankara, Turkey in 1999, and M.S. in Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY in 2002. Currently he is pursuing his PhD degree in computer and systems engineering in Rensselaer Polytechnic Institute, NY. His research interests include high-speed bipolar digital circuit design.

Russell P. Kraft received a B.S.E.E. degree in 1976, an M.S.E.E. degree in 1978, and a Ph.D. in 1983, all from Rensselaer Polytechnic Institute, Troy, New York. His dissertation was on the optimization of phased ultrasonic and radar arrays for steering by nonlinearly adjusting the element gains and phases. In 1986, he joined the Center for Manufacturing Productivity at Rensselaer Polytechnic Institute. The research group was folded into the Center for Integrated Electronics where he is now as a sr. staff scientist and adjunct faculty member. His research interests include electronic manufacturing, robotics, metrology, inspection, and fast digital IC design and his teaching responsibilities include control and communication laboratory applications, embedded microprocessor systems design and linear circuits. Previous to this, he was a Sr. Controls Engineer at Mechanical Technology Inc., Latham, NY where he developed vision based inspection and gauging systems and real-time imbedded combustion controls for automotive and generator Stirling engines. Dr. Kraft is a memberEta Kappa Nu, Tau Beta Pi, Sigma Xi and the Machine Vision Association of SME. He is a technical reviewer for the IEEE Transactions on Components, Packaging, and Manufacturing Technology - Part C. He has two patents in the computer-vision area for non-contact gauging and is co-author of several publications in high-speed digital design with SiGe HBTs, vision inspection, phased array design, homomorphic signal processing and control system design.

John F. McDonald (M’65–SM’01) was born in 1942 in Narberth, PA. He received his B.S.E.E. in 1963 from the Massachusetts Institute of Technology (M.I.T.), Cambridge, MA, and the M.Eng. and PhD degree in engineering and applied science from Yale University, New Haven, CT, in 1965 and 1969, respectively. He served as a Member of the Technical Staff at Bell Labs in 1964. He was an Instructor at Yale in 1969, and Assistant Professor the following year. In 1974 he joined the faculty at Rensselaer Polytechnic Institute in the Department of Electrical, Computer and Systems engineering as Associate Professor. He was made Full Professor in 1985. His background includes a wide range of topics including Communication Theory and DSP, Computer Hardware Design, Flooded Electron and Ion Beam Systems, HF and MCM packaging, GaAs/AlGaAs HBT RISC processor design, HBT technology, and VLSI Design and Design Automation. His current interests are concentrated on SiGe HBT BiCMOS circuit design and VLSI
interconnection technology. He has coauthored 255 refereed articles roughly one third of which are archival journal articles and has been awarded 10 patents.
APPENDIX B: Binary Addition Structures with SiGe HBT Digital Circuits

Paul M. Belemjian, Matthew W. Ernest, Okan Erdogan, Aamir Zia, Philip Jacob, Michael Chu, Russell P. Kraft, and John F. McDonald.

Binary Addition Structures with SiGe HBT Digital Circuits
Abstract--Adder structures utilizing silicon-germanium heterojunction bipolar transistor (HBT) digital circuits are proposed for use in multi-gigahertz digital applications requiring high-speed integer arithmetic. A 4-gate deep structure for 32-bit addition using a 200 GHz \( f_T \) process experimentally verified a 37.5 ps delay or 26.7 GHz speed at a power of 2.9 watts and a power-delay product of 109 ps-W. Results are extrapolated out to a full 32-bit adder to show power and size. Additionally with certain recent changes in design rules, simulations show that 32 GHz is achievable at slightly above room temperature.

Index Terms-- adders, bipolar integrated circuits, carry look-ahead, CML, ECL, HBT, SiGe.

INTRODUCTION

Introduction
For decades the computer industry has been driven by ever-faster computers appearing regularly. Silicon Germanium Heterojunction Bipolar Transistors (HBTs) are currently the highest speed devices in silicon based circuit manufacturing, and offer an opportunity to continue building computers with faster clock rates. Devices with a cut off frequency \( f_T \) in excess of 375 GHz have been reported [24] and THz \( f_T \) performance is predicted in the future [25]. Figure 83 shows a plot of \( f_T \) performance across four generations of IBM SiGe HBT technology. The HBT derives its speed primarily by vertical scaling of its base region. Power savings derive from lateral scaling. As a result HBT and FET scaling proceed somewhat differently. The low on-resistance of the HBT reduces the need for wire scaling. As a result HBT circuit wire
resistance can scale differently from FET technology creating unique, unexpected opportunities for higher speed.

Digital circuits designed for high-speed operation with SiGe HBT devices require current steering operation to avoid simultaneous switching noise. Such circuits might be designed in full-differential Current Mode Logic (CML) or non-differential Emitter Coupled Logic (ECL)[27][28]. In this paper, we explore a combination of these two styles of current steering tree design to minimize the longest path delay through a 32-bit adder.

**High-speed addition**

As adders are essential component of the data path in any computer we investigate the design of fast adders. An attractive application of such high-speed adders is in the message-packing computers that form the backbone of high-productivity computing systems (HPCS). It is well known that a significant amount of network latency in HPCS comes from the dedicated CPUs that are used for message packing and unpacking during transfer of data packets between multiple nodes. However, such high-speed message packing computers will inevitably feel the impact of memory wall due to slow access to off-chip memory with limited bus-width. A possible solution to the memory-wall problem in these message-packing computers can be using a vertically integrated (three-dimensional) processor-memory stack [29], thus providing a very wide data bus with low interconnect latency, between processor and memory. If such a 3D processor-memory stack of very fast SiGe HBT CPU and vertically integrated 3D memory is employed as a message-packing computer at every node of HPCS, it could drastically reduce the network latency problem.

Adders are not restricted to general-purpose processors. Digital signal processors and
network processors also need to perform arithmetic in their operations [30]. As a simple circuit and a building block for popular systems, binary adders are strong candidates for special-case optimization. The simplicity of the operation leads to the tractability of the problem of delay optimization. The ubiquity of the operation allows the savings due to optimization to be reaped multiple times. Examination of the logic of addition exposes underlying parallelism. Exploiting this parallelism speeds up the delay by doing work in different areas at the same time when possible. Beyond basic logic, wiring parasitics exact their penalties. Custom layout of circuit features results in the optimum performance for the specific conditions by the parallel logic by minimizing these wiring effects. Through adjustment in areas such as gate topology, input symmetry, and re-ordering of functions, delay can be moved off the critical paths that define the overall delay of the adder and onto other, shorter paths.

Early adder research led to carry-skip [1] and carry-select [2] circuits, both of which have the advantage of a nearly bit-slice arrangement in the physical layout [3]. Full examination of parallelization led to block carry look-ahead [4]. Although requiring a large area for its speed increases, carry look-ahead often attracts the interest of bipolar designers who strive for the fastest possible circuits [5], and that of BiCMOS designers as well [6]. Other work has focused on adders incorporated in other circuits, such as multipliers, which present an uneven input profile to an adder [7].

Bipolar SiGe adders offer a high-speed alternative to standard CMOS circuits. An 8.3 GHz 32-bit adder was demonstrated using 90 nm CMOS circuits [38]. An adder in IBM’s 0.18-µm 7HP SiGe technology is referenced here at 17 GHz, more then twice the speed of the CMOS circuit. Also presented is a circuit designed using IBM’s 0.13-µm
8HP SiGe technology at 26.7 GHz, more than three times the speed of CMOS. The power delay product of the 8HP circuit ranges from 2 to 4 times that of CMOS. Future generations of the SiGe technologies promise even more speed and power gains.

**SiGe Technology**

A Silicon Germanium (SiGe) Heterojunction Bipolar Transistors (HBT) process was chosen as the technology to demonstrate a high-speed adder carry chain. The IBM SiGe BiCMOS processes were chosen in particular. In conjunction with their high speed, they offer relatively good yields compared to other bipolar processes and good current driving capability. They do not include new low-k dielectric materials, but do use Cu for a minimized interconnection related delay. The yield of the process enables us to demonstrate more complex circuits, which may be incorporated in a microprocessor [39].

The original IBM SiGe HBTs were designed for mainframe computers. Therefore, initially they were optimized for that application. The digital profile placed a gradient across the highest base doped region to achieve the highest $f_T$ [40]. As a result, SiGe HBTs have been capable of achieving $f_T$ in excess of 100 GHz since 1993 [41]. Breakdown voltage tends to limit $f_T$, but numerous device and process improvements have permitted the SiGe HBTs to circumvent some of the implications of the Johnson relation. The contemporary SiGe HBT satisfies a blend of analog and digital requirements, but its readily available speed advantages made it a prime choice for our study.
Adder Design

The basic carry-look ahead structure is well known. A look-ahead tree arrangement similar to the Han and Carlson variant of the Kogge-Stone tree is used, which generates only the odd positions [42]. This would increase the depth of the circuit by one in the Han and Carlson case, however the wider fan-in in the sum-generation circuit allows both even and odd sums to be generated from only odd position carries. Early versions of the design used the Ling Pseudo-Carry[43][44], but it was removed from later designs due to power concerns.

The general approach to carry-look ahead generation focuses on a fan-in of two at each stage of the structure. A larger fan-in is possible and such a change provides a reduction of the number of gate delays within the circuit. A fan-in of 3 decreases the number of gate delays by 1, while a fan-in of 4 decreases the number of gate delays by 2. Increasing the fan-in of a gate does increase the delay of that gate, but this increase is more than offset by the reduction in total gate delays. Figure 84 shows how the total delay and the individual gate delays change as you go from a fan-in of 2 to 3 to 4. The times given are in picoseconds, and are based on the 7HP implementation of the circuit.

While speed is the desired quantity to maximize, it is always important to keep in mind how the changes impact the power consumption and size of the design. As such, Table 12 shows a comparison of speed, power and size for the three fan-in values. The power and size numbers are based on the critical paths of the adder, using the 7HP design. While the values will change for the 8HP design, the percentage of change will remain the same. The non-critical paths remain the same for all three values, and are thus left off for the purposes of comparison.
The interesting result here is that using gates with a higher fan-in to maximize speed actually results in a reduction in both size and power compared to the lower fan-in designs. The natural conclusion to all of these investigations is that based on the technology that is being used, using a higher fan-in gate is the correct design decision.

**Short-and-Wide Very High Fan-in Look-Ahead**

**Dotted Emitter/Collector Gates**

The circuit shown in Figure 3 using multiple current trees with dotted-or and dotted-and outputs allows for the increase of fan-in from 2 to 4. The differential outputs of a number of current trees may be logically OR-tied by dotting collectors of the inverting output and dotting emitters on the non-inverting output. Any single-ended switching is moved to the non-critical inputs, if it is necessary at all. If the gate is expanded to handle more inputs, it gets wider but not taller in current tree structure, allowing it to operate on the same supply voltage. The possibility of an excessive swing exists on the inverting output when multiple trees pull down, but this is clamped at the output by emitter-dotting with a normal logic ”0” voltage. The size of this gate is limited foremost by the total collector capacitance on the inverting output, but using a cascoded device in a common-base configuration can alleviate this.

**Reduced-Depth Carry Tree**

The wider look-ahead gates allow implementation of shallower carry trees to be generated. The carry-out of a 32-bit adder can thus be generated from the 32-bit carry tree using three sets of group carries such as these, where G is the generate, P is the propagate, H is the partial carry, and I is the partial propagate:
1. \[ H_{20} = G_{18}^{1} + P_{18}^{1} G_{18}^{1}, H_{22} = G_{20}^{1} + P_{20}^{1} G_{20}^{1}, H_{24} = G_{22}^{1} + P_{22}^{1} G_{22}^{1}, \]
   \[ H_{26} = G_{24}^{1} + P_{24}^{1} G_{24}^{1}, H_{28} = G_{26}^{1} + P_{26}^{1} G_{26}^{1}, H_{30} = G_{28}^{1} + P_{28}^{1} G_{28}^{1}. \]
   
2. \[ H_{18}^{8} = H_{30}^{8} + I_{29}^{8} H_{28}^{2} + I_{29}^{8} I_{27}^{8} H_{26}^{2} + I_{29}^{8} I_{27}^{8} I_{25}^{8} H_{24}^{2}, \]
   \[ H_{16}^{8} = H_{22}^{8} + I_{21}^{8} H_{20}^{2} + I_{21}^{8} I_{19}^{8} H_{18}^{2} + I_{21}^{8} I_{19}^{8} I_{17}^{8} H_{16}^{2}, \]
   \[ H_{8}^{8} = H_{14}^{8} + I_{13}^{8} H_{12}^{2} + I_{13}^{8} I_{11}^{8} H_{10}^{2} + I_{13}^{8} I_{11}^{8} I_{9}^{8} H_{8}^{2}, \]
   \[ H_{0}^{8} = H_{6}^{8} + I_{5}^{8} H_{4}^{2} + I_{5}^{8} I_{3}^{8} H_{2}^{2} + I_{5}^{8} I_{3}^{8} I_{1}^{8} H_{0}^{2}, I_{n}^{8} = I_{n}^{8} I_{n-2}^{8} I_{n-4}^{8} I_{n-6}^{8}, \]
   
3. \[ H_{0}^{32} = H_{24}^{32} + I_{17}^{32} H_{16}^{2} + I_{17}^{32} I_{9}^{32} H_{8}^{2} + I_{17}^{32} I_{9}^{32} I_{4}^{32} H_{0}^{2}. \]

The prefix graph for the carry with four-way look-ahead is shown in Figure 4. With the addition of a cell to produce the final sums, the total gate delay of a 32-bit adder would be 4 gates. Figure 87 shows a combined sum and look-ahead gate, which allow generation of every sum from a carry-tree that generates carries for only the even terms. This reduces the area and power necessary for carry generation without impacting overall delay.

Furthermore, a latch can be incorporated into the sum gate for only a modest time penalty, which is much less than an additional gate delay if that is the desired location for a pipeline latch. To properly latch a dotted emitter/collector gate, it is necessary to drive the keeper current switch from the outputs of the emitter-followers, not the inputs as is possible with a single current tree gates. This latched sum gate is shown in Figure 88.

**Adder Carry Tree Test Structures**

The basic test structures was first designed and fabricated in the IBM 7HP SiGe
HBT process, and provided the basis for the analysis done here. Results for this design were reported in [39]. The layout of this test structures is shown in Figure 89, and the microphotograph is shown in Figure 90.

**0.13 μm 8HP adder carry-chain structures**

A test structure using the IBM 8HP SiGe HBT process with the short-and-wide design scheme was also designed and fabricated. It was the same basic design as the 7HP design, simply implemented in a newer and faster processing technology. The layout of this test structure is shown in Figure 91 and the microphotograph of the chip is shown in Figure 92.

Simulation of the 8HP circuit without parasitic extraction yielded a time of 25.1 ps through the carry chain and is shown in Figure 93. This is 55% faster than the 7HP circuit. The simulation with parasitic extraction predicts a time of 34.1 ps, which corresponds to a clock speed of 29.3 GHz and is shown in Figure 94.

The test chip for 8HP was fabricated and tested to determine how close the simulated results predicted actual performance. The measured result of the 8HP test chip was found to be 37.5 ps, which corresponds to a clock speed of 26.7 GHz and is show in Figure 95. This is a difference of 3.4 ps, or 10%. Extracted simulations at higher temperatures show that an increase in temperature leads to a decrease in speed. For instance, at 85 ºC a speed of 26.7 GHz is predicted.

One of the initial disadvantages of working with 8HP relative to 7HP was a restriction on deep trench isolation layer area density percentage in early design kits. This limited the ability to share DTI walls to compress layout. Recent design kits removed this restriction, permitting more compact designs which reduce the wire delay.
Figure 96 compares the layouts of a cell from each design to show the large difference in size. Table 13 shows the amount of wire delay in the different versions of the design. Simulations show that a new compact design would run 10% faster than the previously measured design, with an expected speed of 30 GHz at 85 °C. If this design is cooled to 30 °C, simulations predict that it will run at 32 GHz as shown in Figure 97.

32-bit Adder Structures

The earlier designs dealt with the carry chain within the adder, which is the critical path. The question still remains how this translates into an actual 32-bit adder, both in terms of speed obtained and power used. The fastest device in 8HP technology is the 1 µm device, which is used to craft the highest speed adder carry-chain structure. Smaller devices that use less current at max $f_T$ are available in 8HP technology, which provides a way to create lower power versions of the adder by trading speed with power. The smaller device is 0.52 µm and has a peak current of 0.74 mA, at the maximum operation frequency, which is half the current of the larger device. In addition, the power consumption of the non-critical path can be reduced to lower the power even more.

Building a complete 32-bit adder using the larger 1 µm devices yields a design that runs at 26.7 GHz, consumes 2.91 W of power and has a power delay product of 109 ps-W. The design using the smaller devices on the other hand runs at 24.8 GHz, consumes only 1.55 W of power and has a power delay product of 63 ps-W. This shows that a large savings in power can be gained easily by a small decrease in speed.
Further savings in power can still be accomplished through two different methods, but both will reduce the speed of the circuit even more. The first method is to go from a mixed differential and non-differential CML/ECL cell design to just a non-differential ECL cell design. This change does not change the current through the trees, but does reduce the power supply voltage, resulting in lowering the total power consumption. A modified circuit with only non-differential ECL is shown in Figure 98.

The second method is to decrease the current for each cell in the adder, not simply the non-critical path cells. This will reduce the speed of the circuit, but it will also reduce the power consumption. Table 14 shows these four designs in comparison to each other. These results show that it is possible to trade speed for better power consumption.

If DTI sharing and cooling are included the speed of the design increases while the power remains the same. This will cause the power delay product to change. In the case of the high power design this reduces it to 88 ps-W, while in the low power case it reduces it to 52 ps-W.

**Conclusion**

This paper presented a 32-bit adder carry-chain circuit using SiGe bipolar technology that ran at 26.7 GHz with a power consumption of 2.91 W. Performance of future designs was predicted up to 32 GHz within the same technology. In addition, a power saving design was introduced that provided 93% of the highest achievable speed while only using 53% of the power.

**Acknowledgment**

The authors would like to thank the DARPA/MARCO iFRC, DARPA/IBM HPCS and
the DARPA/SPAWAR/IBM 3DI programs. They would also like to thank INTEL for providing an INTEL fellowship.

REFERENCES


Paul M. Belemjian was born in Albany, NY in 1977. He received a B.S.C.E from Rensselaer Polytechnic Institute, Troy, NY in 1999. Currently he is pursuing his PhD degree in computer and systems engineering at Rensselaer Polytechnic Institute. His research interests include adder design using high-speed bipolar circuits.

John F. McDonald (M’65–SM’01) was born in 1942 in Narberth, PA. He received his B.S.E.E. in 1963 from the Massachusetts Institute of Technology (M.I.T.), Cambridge, MA, and the M.Eng. and PhD degree in engineering and applied science from Yale University, New Haven, CT, in 1965 and 1969, respectively. He served as a Member of the Technical Staff at Bell Labs in 1964. He was an Instructor at Yale in 1969, and Assistant Professor the following year. In 1974 he joined the faculty at Rensselaer Polytechnic Institute in the Department of Electrical, Computer and Systems engineering as Associate Professor. He was made Full Professor in 1985. His background includes a wide range of topics including Communication Theory and DSP, Computer Hardware Design, Focused Electron and Ion Beam Systems, HF and MCM packaging, GaAs/AlGaAs HBT RISC processor design, HBT technology, and VLSI Design and Design Automation. His current interests are concentrated on SiGe HBT BiCMOS circuit design and VLSI interconnection technology. He has coauthored 255 refereed articles roughly one third of which are archival journal articles and has been awarded 10 patents.
Russell P. Kraft received a B.S.E.E. degree in 1976, an M.S.E.E. degree in 1978, and a Ph.D. in 1983, all from Rensselaer Polytechnic Institute, Troy, New York. His dissertation was on the optimization of phased ultrasonic and radar arrays for steering by nonlinearly adjusting the element gains and phases. In 1986, he joined the Center for Manufacturing Productivity at Rensselaer Polytechnic Institute. The research group was folded into the Center for Integrated Electronics where he is now as a sr. staff scientist and adjunct faculty member. His research interests include electronic manufacturing, robotics, metrology, inspection, and fast digital IC design and his teaching responsibilities include control and communication laboratory applications, embedded microprocessor systems design and linear circuits. Previous to this, he was a Sr. Controls Engineer at Mechanical Technology Inc., Latham, NY where he developed vision based inspection and gauging systems and real-time imbedded combustion controls for automotive and generator Stirling engines. Dr. Kraft is a member Eta Kappa Nu, Tau Beta Pi, Sigma Xi and the Machine Vision Association of SME. He is a technical reviewer for the IEEE Transactions on Components, Packaging, and Manufacturing Technology - Part C. He has two patents in the computer-vision area for non-contact gauging and is co-author of several publications in high-speed digital design with SiGe HBTs, vision inspection, phased array design, homomorphic signal processing and control system design.

Okan Erdogan (S’02, M’08) was born in Sivas, Turkey in 1977. He received a B.S.E.E. from Hacettepe University, Ankara, Turkey in 1999, M.S. and Ph.D. in Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY in 2002 and 2006 respectively. Currently he is a research associate in electrical, computer and systems engineering department in Rensselaer Polytechnic Institute, NY. His research interests include high-speed circuit design.

Matthew Ernest (S’95-M’03) received the B.S in Computer and Systems Engineering and the M.S and Ph.D. in Electrical Engineering from Rensselaer Polytechnic Institute, Troy, NY in 1994, 1996 and 2002 respectively. His Ph.D. research focused on high-speed arithmetic with bipolar circuits. Since 2003, he has been a Component Design Engineer in design proliferation for Intel Corp., Hillsboro, OR, working on ALU and register file circuits and statistical analysis of circuits with device variation.
Michael Chu received the B.S. and M.S. degrees in Electrical Engineering from Rensselaer Polytechnic Institute (RPI) in 2001 and 2006, respectively. He is currently working toward the PhD degree at RPI. His research interests include high speed analog-to-digital conversion, low phase-noise oscillators, and RF circuit design.

Philip Jacob is a PhD candidate in the Department of Electrical, Computer and Systems Engineering at Rensselaer Polytechnic Institute. His research interests include 3D processor memory stack architecture study and high speed digital circuit design. Jacob has a MS in Electrical Engineering from Rensselaer Polytechnic Institute and a BTech in Electronics and Communication engineering from the University of Kerala, India. He is a Student member of the IEEE.

Aamir Zia is a PhD candidate in the Department of Electrical, Computer and Systems Engineering at Rensselaer Polytechnic Institute, Troy, NY. His research interests include 3D IC technology, SOI and SiGe BiCMOS integrated circuit design. He has a BS from University of Engineering and Technology, Lahore, Pakistan and an MS from the University of Oklahoma, Norman, OK, both in Electrical Engineering.
Figure 83: $f_T$ performance across four generations of IBM SiGe HBT (note horizontal axis is current density)[26]

Figure 84: Fan-in speed comparison of 7HP design
Figure 85: Schematic of dotted and/or implantation of the three-way look-ahead function

Figure 86: Prefix graph for 4-way look ahead

Figure 87: Combining sum generation with two-way look-ahead gate utilizing dotted emitter/collector
Figure 88: Latched sum output in a single gate

Figure 89: Layout of the 7HP adder test structure
Figure 90: Microphotograph of 7HP adder carry chain test structure

Figure 91: Layout of the 8HP adder test structure
Figure 92: Microphotograph of adder carry chain test chip

Figure 93: Schematic simulation of the 8HP adder test structure (waveform is the adder test signal that is divided in frequency by eight)
Figure 94: Extracted simulation of the 8HP adder test structure (waveform is the adder test signal that is divided in frequency by eight)
Figure 95: Measured waveform of the 8HP adder test chip (waveform is the adder test signal that is divided in frequency by eight)
Figure 96: Layout comparison of no DT sharing design to DT shared design

Figure 97: Simulated waveform of DT shared 8HP circuit (waveform is the adder test signal that is divided in frequency by two)
Figure 98: Schematic of non-differential ECL cell

Table 12: Fan-in power and size comparison of 7HP design

<table>
<thead>
<tr>
<th>Fan-in</th>
<th>Delay (ps)</th>
<th>Difference</th>
<th>Power (W)</th>
<th>Difference</th>
<th>Devices</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 fan-in</td>
<td>36.25</td>
<td>100%</td>
<td>1.96</td>
<td>100%</td>
<td>1745</td>
<td>100%</td>
</tr>
<tr>
<td>3 fan-in</td>
<td>42.3</td>
<td>86%</td>
<td>2.32</td>
<td>118%</td>
<td>1819</td>
<td>104%</td>
</tr>
<tr>
<td>2 fan-in</td>
<td>48.2</td>
<td>75%</td>
<td>2.61</td>
<td>133%</td>
<td>1960</td>
<td>112%</td>
</tr>
</tbody>
</table>

Table 13: Wire delay percentages in various designs

<table>
<thead>
<tr>
<th>Design</th>
<th>Simulated (ps)</th>
<th>Extracted (ps)</th>
<th>Wire Delay (ps)</th>
<th>Percentage of Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>7HP Design</td>
<td>39</td>
<td>56</td>
<td>17</td>
<td>30%</td>
</tr>
<tr>
<td>Base 8HP Design</td>
<td>28.5</td>
<td>37.5</td>
<td>9</td>
<td>24%</td>
</tr>
<tr>
<td>Compact 8HP Design</td>
<td>28.6</td>
<td>33.3</td>
<td>4.7</td>
<td>14%</td>
</tr>
</tbody>
</table>

Table 14: Power comparison of 8HP designs

<table>
<thead>
<tr>
<th>Design</th>
<th>Speed (GHz)</th>
<th>Percentage</th>
<th>Power (W)</th>
<th>Percentage</th>
<th>Power Delay Product (ps-W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>High power</td>
<td>26.7</td>
<td>100%</td>
<td>2.91</td>
<td>100%</td>
<td>109</td>
</tr>
<tr>
<td>Low power</td>
<td>24.8</td>
<td>93%</td>
<td>1.55</td>
<td>53%</td>
<td>63</td>
</tr>
<tr>
<td>Low power (ECL)</td>
<td>21.7</td>
<td>81%</td>
<td>1.05</td>
<td>36%</td>
<td>48</td>
</tr>
<tr>
<td>Half power</td>
<td>19.4</td>
<td>73%</td>
<td>0.86</td>
<td>29%</td>
<td>44</td>
</tr>
</tbody>
</table>