AN ABSTRACT OF THE THESIS OF

Chee Lee for the degree of Master of Science in Electrical and Computer Engineering presented on May 30, 2003.
Title: A Resource Constrained Scheduling Scheme that Considers Resources Operating at Multiple Voltages and Register Assignment

Abstract approved: Redacted for Privacy

Wen-Tsong Shiue

Power and timing requirements are becoming more and more stringent as applications move from less mobile devices to more mobile ones. As such, it is important to optimize these applications as much as possible in order to provide the best solution that is low power and low latency. Although there are many different techniques to achieve a low power, low latency solution, this thesis focuses specifically on low power scheduling at the behavioral level where resource-constrained scheduling is the technique of choice since it directly considers the resource limitations of mobile devices. Conventional resource-constrained scheduling schemes are concerned with minimizing the latency or improving the speed of an algorithm—represented by a data flow graph (DFG)—given a limitation on resources. However, these conventional resource-constrained scheduling schemes are no longer applicable since power has grown to be a major issue, especially in mobile devices. Hence, the conventional resource-constrained scheduling schemes gave way to current resource-constrained scheduling schemes that utilize multiple voltages, which work to find a balance between speed and

power. These current multiple voltage schemes use various techniques to balance and meet the speed and power requirements. But while they do a good job of meeting these requirements, they fail to address a new issue that is beginning to surface—the number of memory registers needed. Therefore, to address this new arising issue, this paper presents a novel resource-constrained scheduling scheme that balances the speed, power, and register requirements. This algorithm is compared to both a conventional resource-constrained scheduling scheme and a current resource-constrained scheduling scheme with multiple voltages to show that it performs better in finding a scheduling solution. Benchmark results show that, on average, our algorithm has a better power savings while keeping the maximum number of registers needed and the latency low compared to conventional resource-constrained scheduling schemes and current resource-constrained scheduling schemes utilizing just multiple voltages.

A Resource-Constrained Scheduling Scheme that Considers Resources Operating at Multiple Voltages and Register Assignment

by
Chee Lee

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented May 30, 2003
Commencement June 2004

Master of Science thesis of Chee Lee presented on May 30, 2003.

APPROVED: Redacted for Privacy

Major Professor, representing Electrical and Computer Engineering

Redacted for Privacy

Director of the School of Electrical Engineering and Computer Science

Redacted for Privacy

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Redacted for Privacy

Chee Lee, Author

ACKNOWLEDGEMENTS

I would like to take this time to express my gratitude to my major advisor, Professor Wen-Tsong Shiue, for his role in my M.S. program. I have learned a lot from working with him. I would also like to thank the members of my committee, Professors Alexandre Tenca, Alexandre F.T. Yokochi, and Cetin Kaya Koc for taking the time to review my research.

Next, I am also grateful to all my peers who have helped me stay focused on my research. I am especially grateful to Weetit Wanalertlak for helping with the technical questions I had as well as helping with my teaching assistant duties when I was overloaded with deadlines.

I would also like to thank my very good friend Ange Ly for all he has done for me this year. I have learned many important lessons and have grown significantly as a person thanks to him. Not to mention, I am also a lot healthier since we have started playing soccer together. I am glad he is staying in Oregon for a few more years.

Thanks also go to my extend family—aunts, uncles, cousins, etc.—who have welcomed me with open arms when I needed to escape from the stress of my academic life. During my stress filled times, I could always count on them for a temporary escape while I relaxed and recharged. They allowed me to refocus and efficiently deal with my stressful situations.

And last, but not least, I am sincerely thankful for my parents , Giachue Lee and Mai Yang Lee, and my brother, Jai Lee, for all their support, encouragement, and patience. Without them, I could not have accomplished all that I have. They stood with me no matter what the situation was and have sacrificed a lot for me. For all that they have done, I am eternally grateful.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF TABLES (Continued)

A Resource-Constrained Scheduling Scheme that Considers Resources Operating at Multiple Voltages and Register Assignment

1. Introduction

The mobile devices market is expected to grow at approximately 20% each year for the next few years with sales exceeding 60 million units in 2008 [1]. As the market grows, more and more consumers are expected to use small, mobile devices such as PDAs and cell phones to perform multimedia tasks like browsing the Internet, taking and sharing pictures, and watching movies. However, the extent of which these tasks can be performed is extremely limited due to strict timing, area, and power requirements. Unlike larger devices such as desktop computer systems, mobile devices cannot support the requirements of complex applications. They are designed to be low power and have limited resources. Therefore, complex applications need to be optimized before they can be effectively used on mobile systems.

In particular, these complex applications need to be able to use less power while still maintaining an acceptable performance. Low power usage is desirable for the following reasons: 1) to increase battery lifetime; 2) to increase system reliability; and 3) to reduce packaging and cooling costs [2,3]. The battery lifetime is determined by the current consumption. If the power consumption is high, then the current consumption is also high, which may reduce the battery lifetime. High power consumption also translates to higher operating temperatures, which may lead to system/circuit failures. Also, with higher temperatures, cooling costs

increase, as more resources are needed to cool the system/circuit. This also leads to larger packaging and a larger overall size. In mobile devices, it is important to have long battery lifetimes, good reliability, and a small size. Hence, it is important to keep the power consumption low.

Power reduction can be achieved at many different levels, such as architecture, algorithm, behavioral, and transistor levels [4]. This thesis focuses on low power scheduling at the algorithm level. Although there are many different low power scheduling schemes, this thesis discusses a practical resource-constrained scheduling since it directly considers resource limitations making it more suitable to mobile devices.

This thesis presents a novel resource-constrained scheduling scheme utilizing multiple voltages and register assignment that performs better than current resource-constrained scheduling schemes for multi-dimensional signal processing. Unlike conventional resource-constrained scheduling schemes, which focus primarily on reducing latency, and current resource-constrained scheduling schemes with multiple voltages, which focus primarily on reducing power usage, our resource-constrained scheduling scheme works to find a balance between the latency and power usage by considering the register usage as well as multiple supply voltages. Latency is kept low by reducing the emphasis on using resources that operate at multiple voltages. These resources are still used, but they are used sparingly. Power usage is kept low by considering the data dependence and the

maximum number of registers needed. By exploiting data dependencies, the maximum number of registers needed can be reduced, which, in turn, helps to reduce the overall power usage. By balancing the latency and power, our resource-constrained scheduling scheme ends up performing, on average, better than either the conventional resource-constrained scheduling or current resource-constrained scheduling schemes utilizing just multiple voltages.

The rest of the thesis is organized as follows. The following section defines terms, equations, and constants used in our resource-constrained scheduling scheme as well as in the conventional resource-constrained scheduling and current resource-constrained scheduling with multiple voltages schemes. Section 3 discusses the background and related work in resource-constrained scheduling. Section 4 illustrates the significance or role that registers have in achieving low power. Section 5 discusses our resource constrained scheduling scheme. Section 6 presents the benchmark results. And section 7 concludes the thesis.

2. Preliminaries

This section defines terms, equations, etc. that are used in the typical resource-constrained scheduling scheme, our resource-constrained scheduling scheme, and our simulations.

2.1 Resource-Constrained Scheduling Definitions

The input to a resource-constrained scheduling scheme is a data flow graph (DFG) and a resource constraint. A resource constraint is a restriction or a limit imposed on the number of each type of resource such as an adder, multiplier, or shifter. A DFG is a directed acyclic graph whose nodes represent operations and edges represent dependencies between the operations. Each node also has information linked to it in order to help determine its priority.

The mobility of a node is defined as the difference between its as-late-as-possible (ALAP) schedule time and its as-soon-as-possible (ASAP) schedule time. Mobility determines which nodes are given priority when assigning to a certain resource. Nodes with high mobility are given priority when assigning to low-voltage resources. Nodes with low mobility are given priority when assigning to high-voltage resources. It is not used in the conventional resource-constrained scheduling scheme and applies primarily to resource-constrained scheduling with resources operating at multiple voltages.

The depth of a node is defined as the length of the path from the node to the sink in the DFG. Depth is directly linked to latency. In order to reduce the number of control cycles, nodes with higher depth are scheduled first. The conventional resource-constrained scheduling scheme as well as the scheme with multiple voltages makes use of the depth.

Resource-constrained scheduling schemes are primarily list-based. Therefore, there is a ready set associated with every control cycle. The ready set is defined as the set of nodes that could all be assigned at that particular control cycle if there were no resource constraints.

In addition to the items mentioned above, our resource-constrained scheduling scheme also takes a conflict flow graph (CFG) as an input. A CFG is similar to a DFG except that the edges represent conflicts between nodes. The CFG helps our algorithm address register concerns.

2.2 Delay and Power Models

The delay and power of the different functional units (adders, multipliers, and registers) operating at various voltages have been obtained from simulations using Mentor Graphic's *Design Architect* and *Accusim* in a Sun/Solaris unix environment. *Design Architect* is a schematic tool that allows the user to create circuits by placing components (resistors, capacitors, transistors, etc.) and wiring

them together. *Accusim* is a circuit simulation tool similar to spice except that it operates through a graphical user interface (GUI).

We constructed and simulated a 32-bit carry-ripple adder, a 32-bit carry-ripple multiplier, and a 32-bit register for three technologies (AMI 1.2, AMI 0.5, and TSMC 0.35) operating at 5V, 3.3V, 2.4V, 2.2V, 1.8V, 1.5V, 1.2V, and 1.0V. Figure 1 illustrates the typical 1-bit full adder that is used to construct our 32-bit carry-ripple adder in Figure 2 and our 32-bit carry-ripple multiplier in Figure 3. Figure 4 illustrates our 32-bit register.

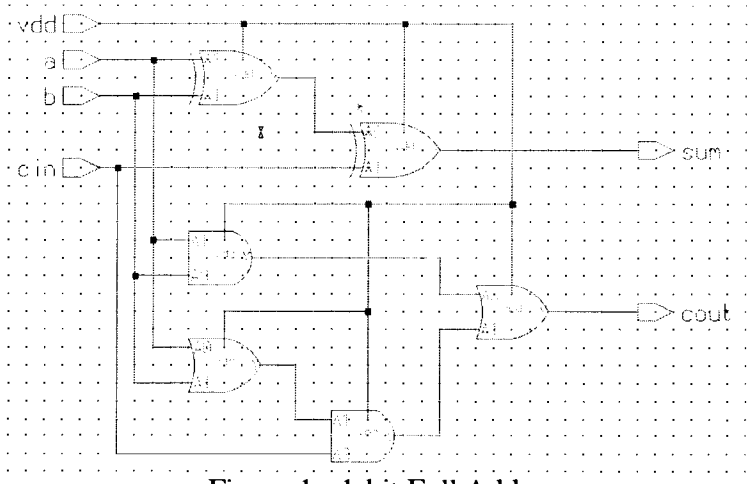


Figure 1. 1-bit Full Adder

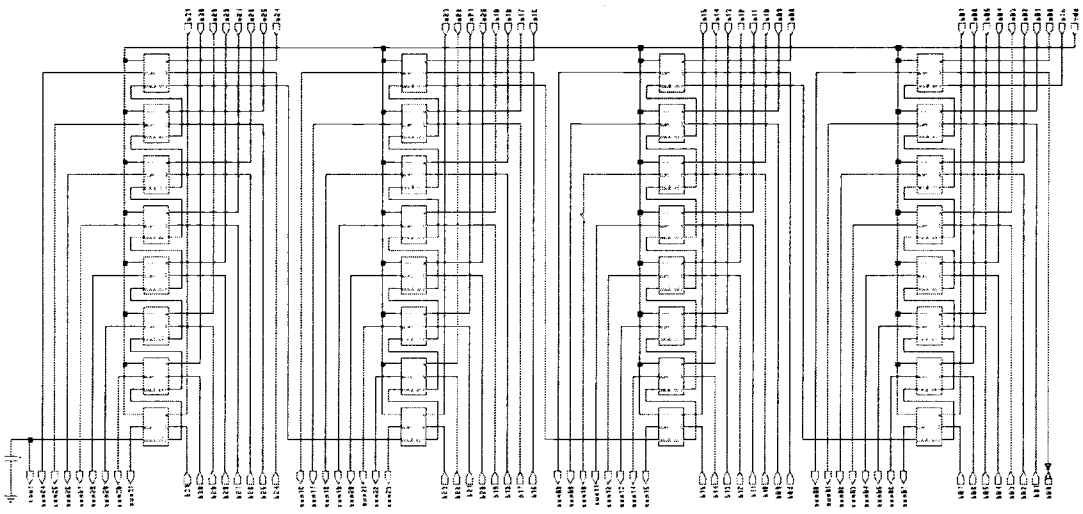


Figure 2. 32-bit Carry-Ripple Adder Schematic

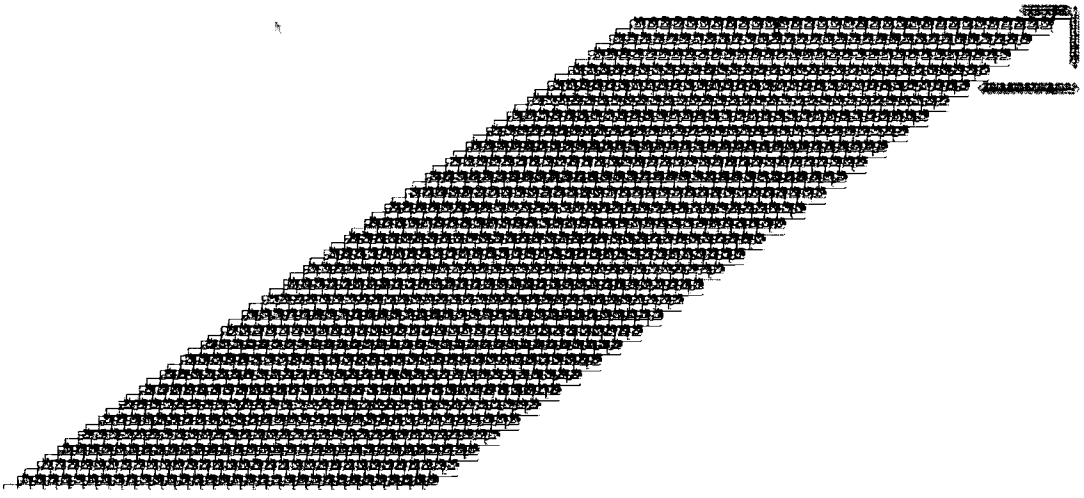


Figure 3. 32-bit Carry-Ripple Multiplier Schematic

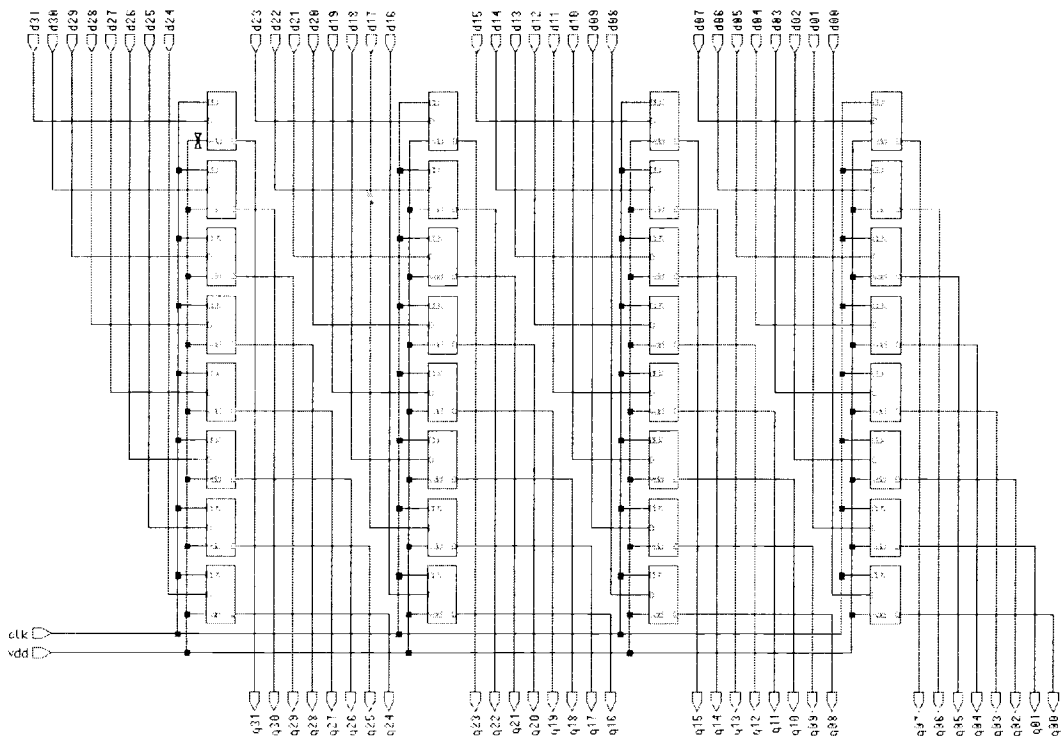


Figure 4. 32-bit Register Schematic

There are different adder and multiplier architectures. However, we chose the ones we did for simplicity and as a starting point. As our library grows, we will include different adder and multiplier architectures as well as other functional units. Only three technologies were used since we were only able to gain access to these technologies. Simulations were done at the specified voltages in order to address the national technology roadmap of semiconductor summarized in Table 1 from [5].

Table 1. National Technology Roadmap of Semiconductor

Year	1997	1999	2001	2003	2006	2009	2012
L (nm)	250	180	150	130	100	70	50
Vdd (V)	2.2	1.8	1.5	1.2	1	0.8	0.6

The delay is a measure of the time it takes for the output to see a change in the input. It is calculated based on the following equation from [6]: $(T_{PLH} + T_{PHL}) / 2$. T_{PLH} and T_{PHL} correspond to a low to high transition at the input and a high to low transition at the input, respectively. The power is determined using a power meter circuit similar to the one in [7] (see Figure 5).

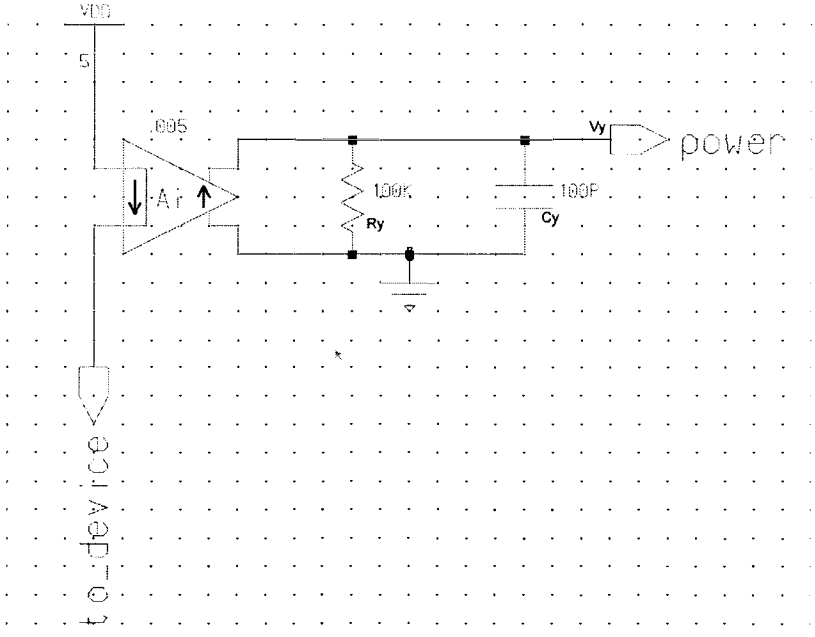


Figure 5. Power Meter Circuit for Simulation of Dynamic Power Dissipation

The meter measures the current drawn by the device and amplifies that current by a value $\beta = V_{DD}*C_y/T$, where V_{DD} is the supply voltage, C_y is a capacitance, and T is the period of the input signal. This current runs through R_y (a resistor) and C_y (a capacitor) and allows us to determine the power by taking measurements at V_y. Figure 6 and Figure 7 show the measurements used to determine the delay and power for the 32-bit carry-ripple adder operating at 5V.

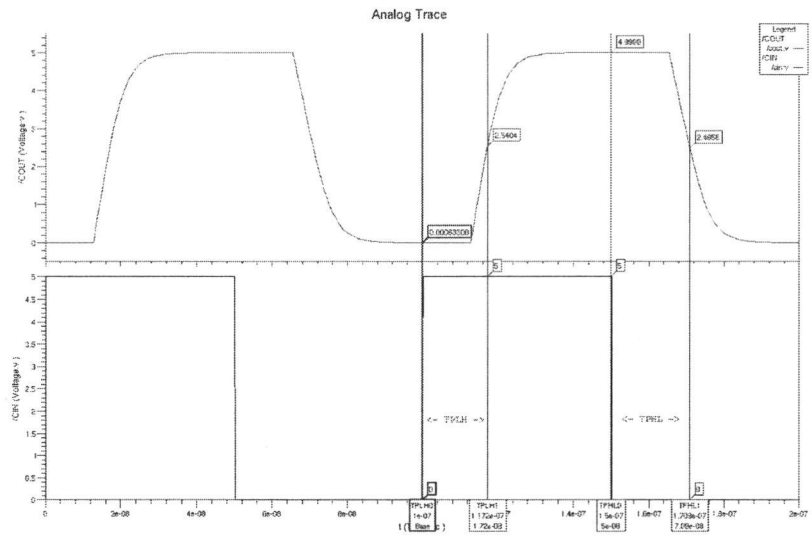


Figure 6. Delay Measurements for the 32-bit Carry-Ripple Adder

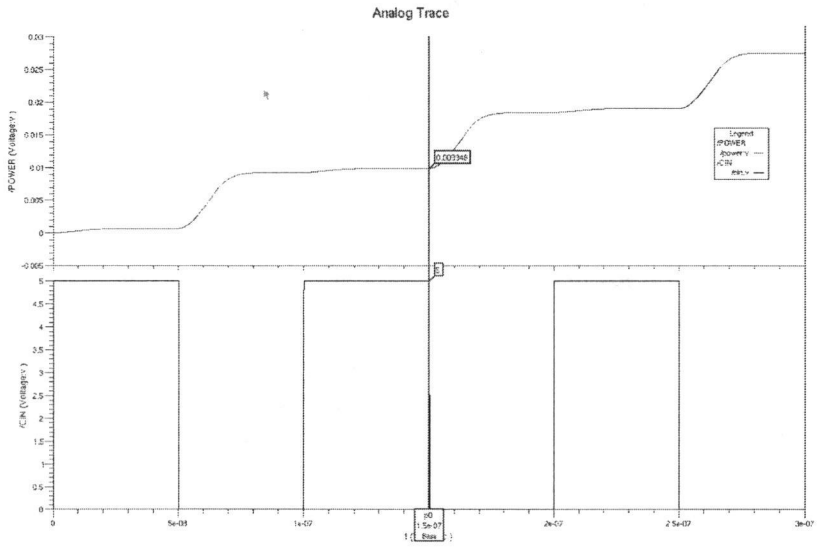


Figure 7. Power Measurements for the 32-bit Carry-Ripple Adder

Measurements for the 32-bit carry-ripple multiplier and 32-bit register are done exactly as with the 32-bit carry-ripple adder. Tables 2, 3, and 4 show the delay and power characteristics for the adder, multiplier, and register for each of the three different technologies and the different operating voltages.

Table 2. Delay and Power Characteristics for a 32-bit Carry-Ripple Adder

adder32	AMI 1.2 um		AMI 0.5 um		TSMC 0.35 um	
Voltages (V)	Delay (ns)	Power (uW)	Delay (ns)	Power (uW)	Delay (ns)	Power (uW)
5	35.09	10,692.00	19.03	9,946.00	13.51	9,335.60
3.3	51.73	4,457.55	31.16	4,246.00	16.50	3,984.48
2.4	78.45	2,272.50	38.38	2,213.40	21.49	2,068.21
2.2	90.31	1,878.33	43.94	1,846.70	23.66	1,835.20
1.8	132.75	1,151.84	64.67	1,081.86	31.20	986.25
1.5	210.60	439.92	104.82	419.17	43.71	402.57
1.2	494.98	137.91	263.88	129.28	78.81	121.41
1	1,983.55	17.49	1,117.20	16.37	170.64	15.54

Table 3. Delay and Power Characteristics for a 32-bit Carry-Ripple Multiplier

mult32	AMI 1.2 um		AMI 0.5 um		TSMC 0.35 um	
Voltages (V)	Delay (ns)	Power (uW)	Delay (ns)	Power (uW)	Delay (ns)	Power (uW)
5	101.46	32,562.00	52.57	30,290.00	33.22	28,431.00
3.3	149.56	13,575.27	86.10	12,930.96	40.57	12,134.50
2.4	226.81	6,920.78	106.04	6,740.79	52.85	6,298.60
2.2	261.09	5,720.37	121.42	5,624.02	58.19	5,588.98
1.8	383.80	3,507.87	178.70	3,294.73	76.72	3,003.56
1.5	608.88	1,339.75	289.65	1,276.56	107.47	1,226.01
1.2	1,431.06	419.99	729.18	393.72	193.79	369.75
1	5,734.75	53.28	3,087.15	49.86	419.59	47.32

Table 4. Delay and Power Characteristics for a 32-bit Register

reg32	AMI 1.2 um		AMI 0.5 um		TSMC 0.35 um	
Voltages (V)	Delay (ns)	Power (uW)	Delay (ns)	Power (uW)	Delay (ns)	Power (uW)
5	7.12	8,559.50	5.20	8,473.50	3.67	8,390.60
3.3	10.49	3,741.63	7.21	3,618.20	4.50	3,558.90
2.4	15.95	2,045.27	10.38	1,909.70	5.63	1,897.33
2.2	18.44	887.88	11.85	794.04	6.44	778.81
1.8	27.01	662.23	16.86	545.39	8.12	526.89
1.5	42.80	389.39	27.20	368.18	11.27	358.80
1.2	102.85	78.09	66.71	75.53	20.01	73.19
1	428.55	14.61	80.05	12.76	43.95	11.41

Notice that, in general, the 32-bit register is approximately four times faster than the 32-bit adder and that the 32-bit adder is approximately three times faster than the 32-bit multiplier (see Figure 8).

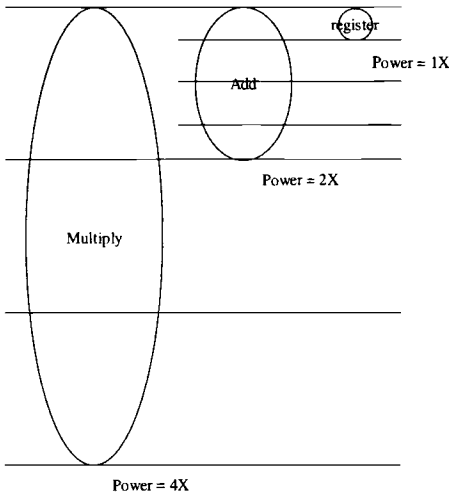


Figure 8. Functional Unit Latency Comparisons

This is an important factor in determining the latency of a DFG. A DFG with many multiply nodes will usually have a longer latency compared to a DFG with fewer multiply nodes. However, since a multiplier is approximately three times slower than an adder, we can complete three add operations during one multiplier operation, which will also impact the latency. As for the power, the register consumes the least, followed by the adder, and then the multiplier (see Figure 8). Thus, a DFG with many multiply nodes will also usually consume more power than a DFG with fewer multiply nodes. Two other relationships to note are: 1) as the voltage decreases, the delay increases but the power decreases; 2) as the technology becomes smaller, both the power and delay decrease. These relationships can be seen in more detail in Figure 9, 10, and 11.

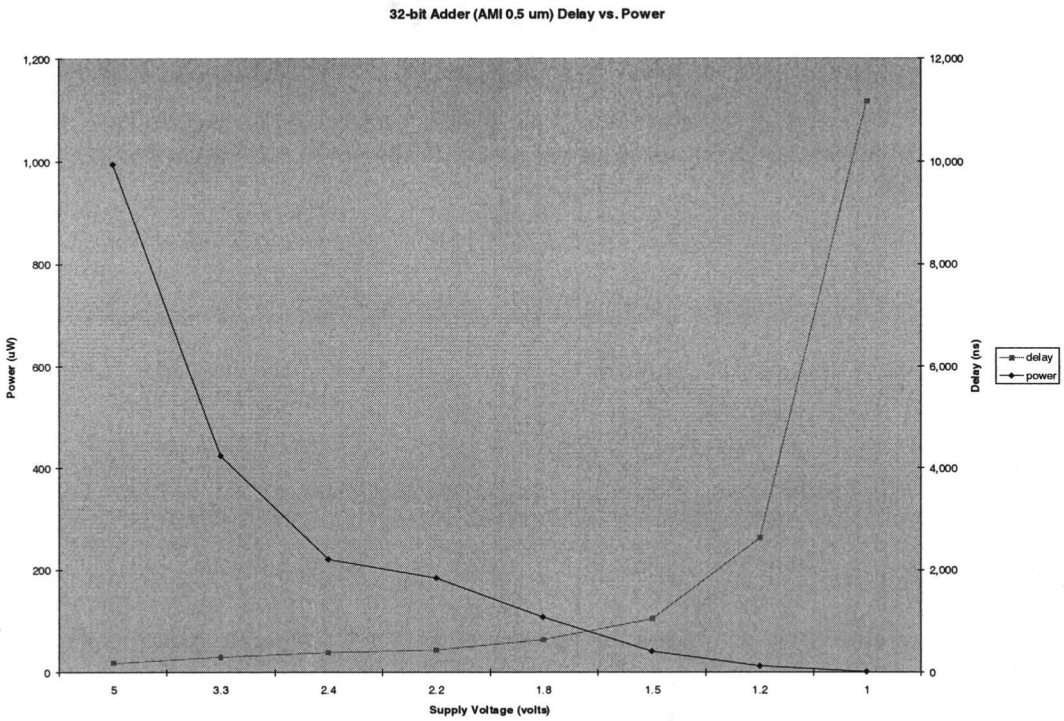


Figure 9. 32-bit Adder Delay vs. Power for AMI 0.5 um.

At higher voltages, the power usage is high while the delay is low. Therefore, if delay is the primary concern, then a higher voltage should be used. However, if power is the primary concern, then a lower voltage should be used. A complete balance can also be obtained. From Figure 9, if we choose the voltage where the power and delay lines cross, then we will have the best possible balance between the power usage and delay.

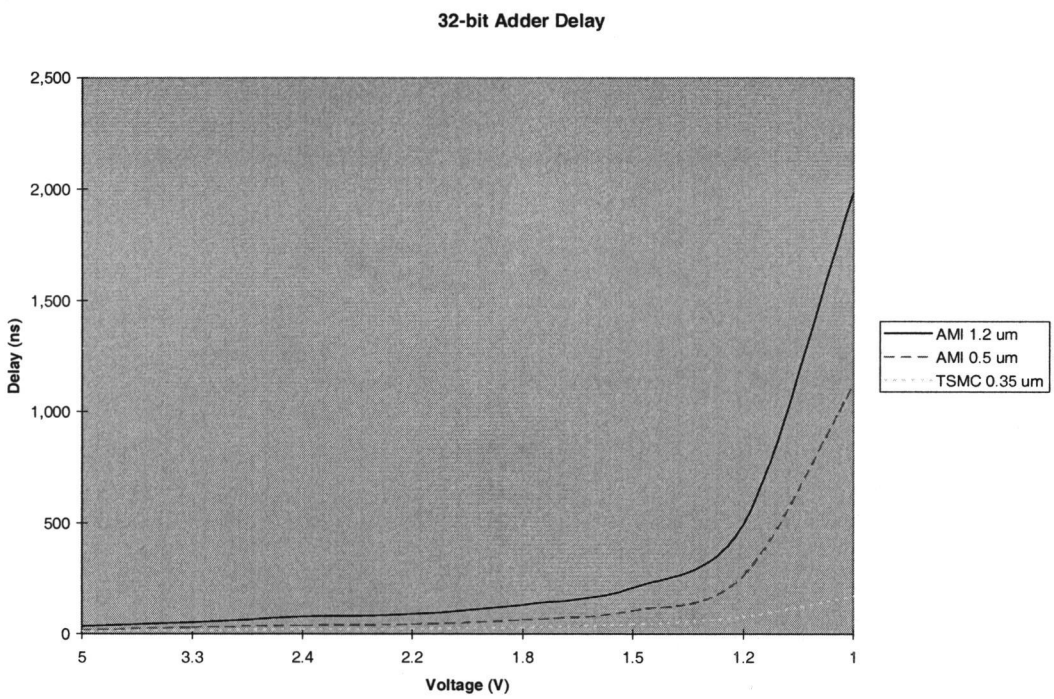


Figure 10. 32-bit Adder Delay for AMI 1.2 um, AMI 0.5 um, and TSMC 0.35 um.

The technology also plays an important role in the delay. AMI 1.2 um has the highest delay followed by AMI 0.5 um and then TSMC 0.35 um. However, it is not always necessary to choose the smallest technology. From Figure 10, all three technologies perform almost identically at the higher voltages. Therefore, if a high voltage is being used, then it may be more beneficial to use a bigger technology, which will cost less, over a smaller technology, which will cost more. However, if a low voltage is being used, then the technology is very important. At lower voltages, TSMC 0.35 um has the lowest delay followed by AMI 0.5 um and then AMI 1.2 um. Hence, if low delay and high voltage were the requirements,

then AMI 1.2 um would meet those requirements best considering performance and cost. However, if low delay and low voltage were the requirements, then TSMC 0.35 um would meet those requirements best considering performance and cost. AMI 0.5 um would give the middle of the line solution.

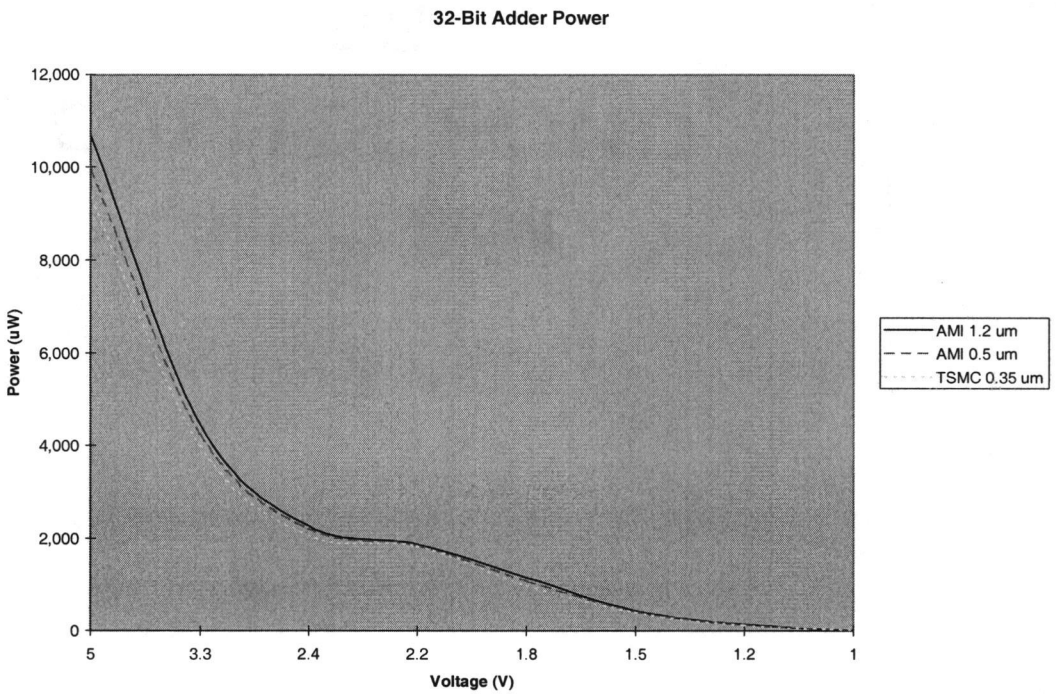


Figure 11. 32-bit Adder Power for AMI 1.2 um, AMI 0.5 um, and TSMC 0.35 um.

Like with delay, the technology also plays an important role in the power usage. AMI 1.2 um has the highest power usage followed by AMI 0.5 um and then TSMC 0.35 um. However, it is not always necessary to choose the smallest technology. From Figure 11, all three technologies perform almost identically at

the lower voltages. Therefore, if a low voltage is being used, then it may be more beneficial to use a bigger technology, which will cost less, over a smaller technology, which will cost more. However, if a high voltage is being used, then the technology is very important. At higher voltages, TSMC 0.35 um has the lowest power usage followed by AMI 0.5 um and then AMI 1.2 um. Hence, if low power and high voltage were the requirements, then TSMC 0.35 um would meet those requirements best considering the performance and cost. However, if low power and low voltage were the requirements, then AMI 1.2 um would meet those requirements best considering the performance and cost. AMI 0.5 um would give the middle of the line solution.

Overall, our library provides information for designers so that they may make a more informed choice about which technology to use. Then within each technology, we provide information that allows the designer to decide on which voltage. All this information allows the designer to make the best choice to meet their latency and power requirements.

The power characteristics for level shifters, which are needed to transfer data between resources operating at different voltages is derived from [8]. Table 5 summarizes the power characteristics. Delay characteristics are ignored because they are significantly smaller than the other functional unit delays.

Table 5. Power (uW) Characteristics for 32-bit Level Shifters

Vx to Vy	1.8	2.2	3.3	5
1.8	0	96	146	220
2.2	70	0	160	320
3.3	124	90	0	356
5	184	220	260	0

Although only the 32-bit adder was analyzed in detail in Figure 9, 10, and 11, the 32-bit multiplier and 32-bit register have the same relationships. The actual numbers for delay and power are different. However, the general trend and the relationships are similar to the 32-bit adder.

2.3 Simulation Environment

All our resource-constrained scheduling simulations are run on a Pentium 4, 1.7 GHz laptop with 512 MB of RAM running Microsoft Windows XP. Each of the resource-constrained scheduling schemes discussed in this thesis have been coded in Microsoft Visual C++ 6.0. All inputs are entered in a table or list format similar to the one shown in Figure 12b in section 3.1. For our resource-constrained scheduling simulations, we chose the delay and power characteristics for AMI 0.5 and the operating voltages of 5V, 3.3V, 2.2V, and 1.8V. This was an arbitrary choice. We could have easily used different characteristics. Implementing the resource-constrained scheduling schemes in C++ allows us to easily change the delay and power characteristics should the need arise. Also, we have assumed that a 32-bit shifter has the same characteristics as the 32-bit adder we built. Our

benchmarks are digital signal processing (DSP) kernels. We chose these benchmarks since they would be the most likely applications run on mobile devices where resource-constrained scheduling is most valuable. DSP refers to various techniques or algorithms for improving the accuracy and reliability of digital communications and work by clarifying, or standardizing, the levels of or states of a digital signal [9].

3. Related Work

Resource-constrained scheduling has been around for many years and simply refers to a scheduling scheme that tries to schedule an algorithm represented with a DFG given a limitation on the number of resources available. An operation in the DFG can only be scheduled if a resource exists and is unused, otherwise the operation must wait. Each operation may be given priority based on a number of different priority functions. The goal is to schedule the DFG, given resource constraints, such that the number of control cycles is minimum. Resource-constrained scheduling is a valuable technique for applications used on mobile devices since resource-constrained scheduling directly deals with the resource limitations of small, portable devices. Resource-constrained scheduling has the advantage of finding the best solution in terms of speed and power given the resource limitation. However, the best solution may have the disadvantage of not meeting the speed requirement (i.e. the latency is too long for practical purposes even though the power may be small and the resource requirements are met).

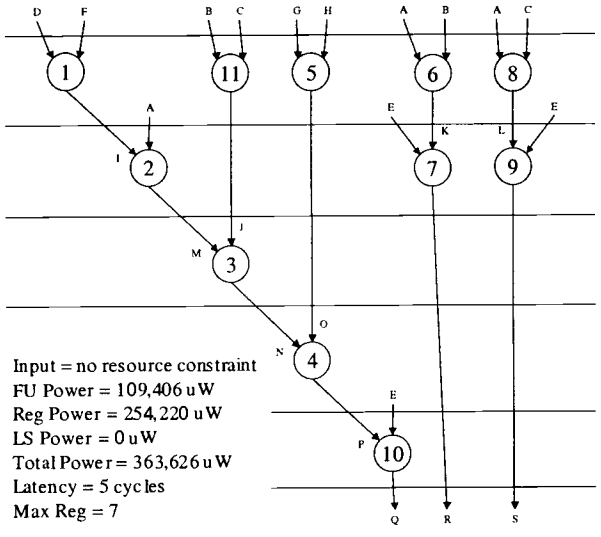
3.1 Conventional Resource-Constrained Scheduling

The conventional resource-constrained scheduling scheme, presented in [10] and [11], used a list scheduling approach that scheduled operations one control step at a time. For the current control step, a list of ready operations is constructed and then sorted according to a priority function. The operation with the highest

priority, usually the one on the longest path or with the highest depth, was scheduled first. The conventional resource-constrained scheduling scheme is as follows:

1) *From the DFG, construct a table listing the nodes, the time they are ready, and their depth.*
2) *For each cycle:*
   a. *Construct the ready set (collect all the nodes that are ready to be scheduled at this cycle into one set).*
   b. *Prioritize the ready set according to highest depth first.*
   c. *While resources are available, schedule nodes from the prioritized ready set.*

Figure 12 illustrates the conventional resource-constrained scheduling scheme for a simple example with a resource constraint of three adders all at 5V.



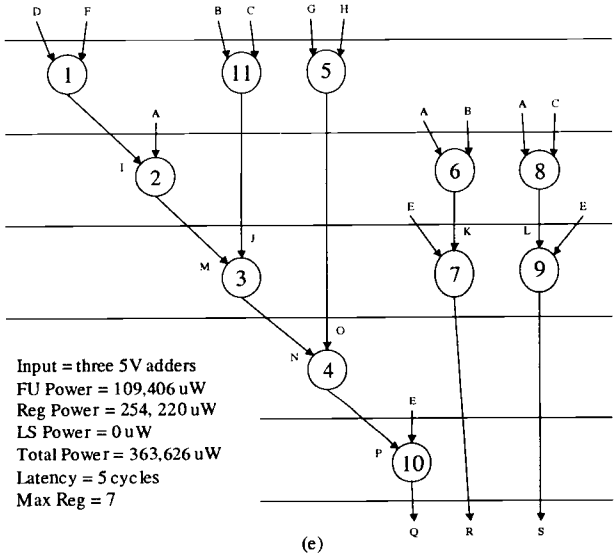
(a)

Node	ASAP	ALAP	Depth	Mobility
1	1	1	4	0
2	2	2	3	0
3	3	3	2	0
4	4	4	1	0
5	1	3	2	2
6	1	4	1	3
7	2	5	0	3
8	1	4	1	3
9	2	5	0	3
10	5	5	0	0
11	1	2	3	1

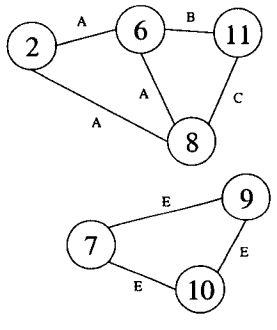
(b)

Control Cycle	Ready Set	5.0V	5.0V	5.0V
1	1 5 6 8 11	1	5	11
2	2 9 6 8 7	2	6	8
3	3 9 7	3	7	9
4	4	4	--	--
5	10	10	--	--

(c)



(e)



(d)

Figure 12. Conventional Resource-Constrained Scheduling Example. (a) Initial DFG. (b) ASAP, ALAP, depth, and mobility of each node. (c) Final Scheduling showing ready sets and scheduling at each cycle. (d) CFG. (e) Final DFG.

Figure 12 shows that the conventional resource-constrained scheduling scheme was able to maintain the lowest possible latency of five clock cycles given a

reduced number of resources to use. Instead of having five resources to use as in the no resource constraints case (see Figure 12a), there were only three resources available. Yet, the conventional scheme was able to schedule the nodes efficiently to keep the latency and power low.

Overall, the conventional resource-constrained scheduling scheme worked well in scheduling DFGs under resource limitations while keeping latency low. However, it did not address the power concerns in mobile devices very well. In Figure 12e, the total power was not reduced at all from what it was in the no resource constraint case, Figure 12a. Conventional resource-constrained scheduling schemes cannot deal with the technique of having resources operating at multiple voltages, which was developed to help with power issues. Hence, the conventional resource-constrained scheduling schemes gave way to current resource-constrained scheduling schemes that are capable of dealing with multiple voltages.

3.2 Resource-Constrained Scheduling with Multiple Voltages

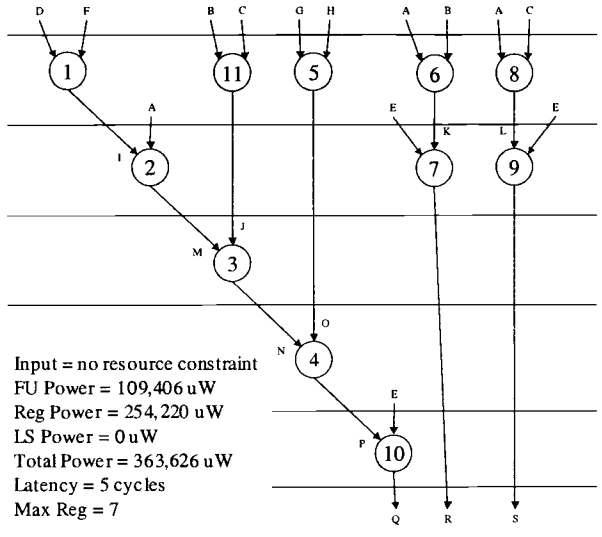
The need for resource-constrained scheduling schemes to address resources operating at multiple voltages led to the development of current resource-constrained scheduling schemes such as those in [11], [12] and [13], which use multiple voltages (5.0V, 3.3V, and 2.2V) and consider items such as switching and level shifters to achieve a low power, low latency scheduling of a DFG. In these newer schemes, the priority function is more complex allowing the resource-

constrained scheduling scheme to address both the power and latency issues simultaneously. However, there is a trade-off to consider. A resource operating at a lower voltage uses less power but takes longer to complete its operation (the latency increases), while a resource operating at a higher voltage completes sooner (the latency decrease) but uses more power. The scheduling schemes in [11], [12], and [13] realize this relationship and do their best to balance the conflicting requirements of reducing the latency and utilizing resources operating at reduced voltages. The general resource-constrained scheduling scheme with multiple voltages is as follows:

1) *From the DFG, determine the ASAP, ALAP, & mobility of each node.*
2) *For each cycle:*
   a. *For nodes in the ready set with mobility >= 2*
      i. *Compute priority.*
      ii. *Assign high priority nodes to available 2.2V resources.*
   b. *For nodes in the ready set with mobility >= 1*
      i. *Compute priority.*
      ii. *Assign high priority nodes to available 3.3V resources.*
   c. *For nodes in the ready set with mobility >= 0*
      i. *Compute priority.*
      ii. *Assign high priority nodes to available 5V resources.*
   d. *If resources remain, then for the nodes that have not been assigned:*
      i. *Compute priority.*
      ii. *Assign high priority nodes to available 2.2V resources, then 3.3V resources, and the 5V resources.*

The priority of a node for the scheduling schemes in [11], [12], and [13] is a function of its depth, mobility, switched capacitance, and level shifter. The depth and mobility are discussed in section 2. The switched capacitance refers to the total capacitance when the inputs of a functional unit are switching. Having a smaller capacitance will result in less power usage. Therefore, nodes with lower switched capacitance are given higher priority. The level shifters also consume power. Therefore, it is beneficial to reduce the total number of up and down level shifters. This is accomplished by assigning a voltage to the child that is the same voltage that has been assigned to its parents whenever possible.

Figure 13 illustrates the resource-constrained scheduling scheme with multiple voltages for a simple example with a resource constraint of three adders—one at 5V, one at 3.3V, and one at 2.2V.



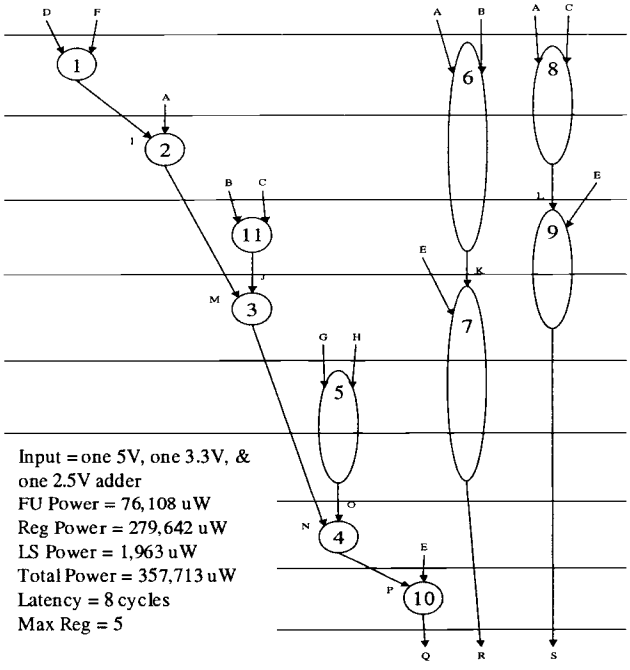
(a)

Node	ASAP	ALAP	Depth	Mobility
1	1	1	4	0
2	2	2	3	0
3	3	3	2	0
4	4	4	1	0
5	1	3	2	2
6	1	4	1	3
7	2	5	0	3
8	1	4	1	3
9	2	5	0	3
10	5	5	0	0
11	1	2	3	1

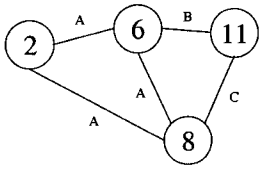
(b)

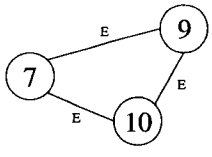
Control Cycle	Ready Set	5.0V	3.3V	2.2V
1	1 5 6 8 11	1	8	6
2	2 5 7 9 11	2	8'	6'
3	3 5 7 9 11	11	9	6'
4	3 5 7 4	3	9'	7
5	10 5 4	--	5	7'
6	10 4	--	5'	7'
7	10 4	4	--	--
8	10	10	--	--

(c)



(e)





(d)

Figure 13. Resource-Constrained Scheduling with Multiple Voltages Example. (a) Initial DFG. (b) ASAP, ALAP, depth, and mobility of each node. (c) Final Scheduling showing ready sets and scheduling at each cycle. (d) CFG. (e) Final DFG.

Figure 13 shows that the current resource-constrained scheduling scheme with multiple voltages was able to reduce the total power given a reduced number of resources to use. Instead of having five resources to use as in the no resource constraints case (see Figure 13a), there were only three resources available, each at a different voltage. These resources at different voltages allowed the total power to be reduced by 1.63%. The maximum number of registers needed also reduced going from seven to five. However, these power and register reductions came at the cost of latency. Instead of a latency of five like in the no resource-constraints case, the latency is now three greater at eight. This increase in latency may not be acceptable in some applications. Also, there is now an overhead associated with this solution. Voltage level shifters need to be used when using multiple voltages. In some cases, the added area and complexity that level shifters bring may out weigh the gains from using a resource-constrained scheduling scheme with multiple voltages. Figure 14 illustrates the overhead from voltage level shifters.

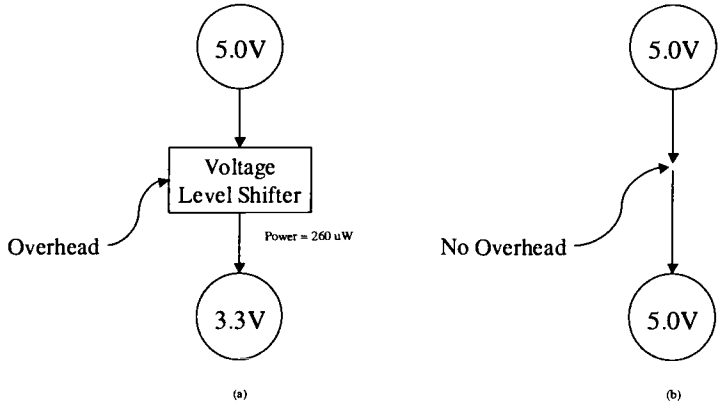


Figure 14. Multiple Voltages Level Shifter Overhead. (a) No Overhead. (b) Overhead

With more and more resources operating at different voltages, more and more voltage level shifters will be needed. If the area requirements are already tight, there may not be room for the level shifters. The power from the level shifters is not too much of a concern since they usually make up less than 1% of the total power.

Overall, resource-constrained scheduling schemes with multiple voltages perform well in finding an optimal solution that addresses the power issues at the slight cost of latency. Although they perform well in general, these schemes do not take into consideration a new surfacing issue—the number of registers needed. As the next section will illustrate, registers play an important role in power consumption and, thus, must be dealt with as well.

4. Register Significance

Memory systems are a vital part of any computer system and are needed by applications so that they can be run faster and perform better. Therefore, it is important to be aware of the contributions they make. In applications targeted for mobile devices, it is even more important due to the stricter power and timing requirements. Overall, there are many levels of memory. However, in this thesis, we focus only on registers.

A single 32-bit register operating at 5V consumes approximately 8,474 uW of power and has a delay of 5.2 ns. So as more and more memory is needed, the power and delay will increase. Although the power and delay of a single 32-bit register is small, they make up a large part of a circuit making the number of registers an important part in reducing latency and power. Figure 15 illustrates the contribution of registers in a circuit. We consider registers operating only at a single voltage since it is not very feasible to have registers operating at multiple voltages in a single device. The level shifter overhead would be larger than the benefit obtained from operating registers at multiple voltages.

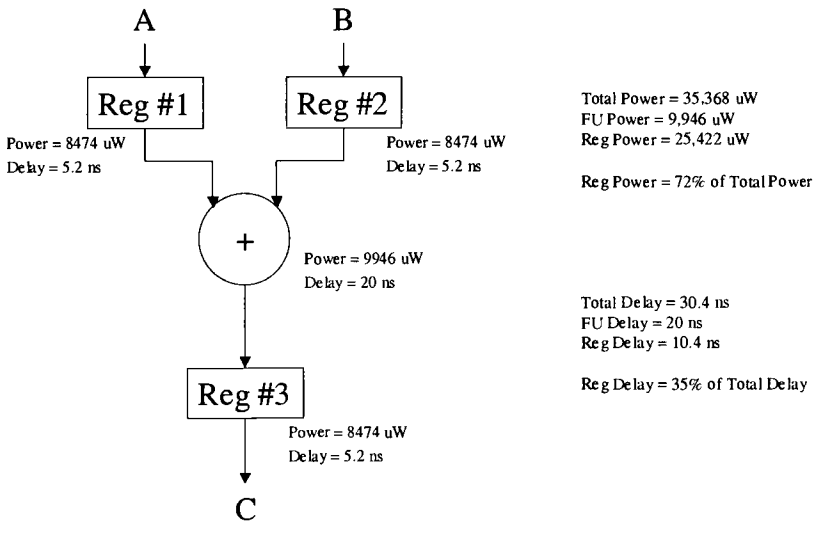


Figure 15. Register Contributions to Power and Delay

Even for a simple circuit with only one adder operating at 5V, the registers contribute approximately 72% of the total power and approximately 35% of the total delay. This can change drastically if the registers have a longer lifetime than one cycle (the registers have to hold the data for more than one cycle). Referring to Figure 15, if each of the registers has to hold its data for an extra cycle before it can be used, the power contribution jumps to approximately 84% while the delay remains unchanged. Hence, it is important to minimize the number of registers needed in order to efficiently reduce the power consumption. The delay plays less of a vital role. However, reducing the number of registers will also help reduce the latency. The register significance is also emphasized in [14]. Although [14] deals exclusively with latency and latency-constrained scheduling, the register principles are also directly applicable to resource-constrained scheduling.

5. Resource-Constrained Scheduling with Register Assignment

In addition to latency and power, the number of registers needed at each control cycle also need to be considered as illustrated in the previous section. While conventional and current resource-constrained scheduling schemes with multiple voltages do a good job of addressing the latency and power, they fail to address the registers needed at each cycle. Our resource-constrained scheduling scheme was developed to address the number of registers while still maintaining an acceptable balance between latency and power.

5.1 Our Scheduling Scheme and Example

We employ a simple technique to address the register problem. Recognizing the data dependency in the DFG, we use a CFG to help determine where to schedule nodes in order to reduce the number of registers needed. We note that if we schedule all the nodes in the CFG that share a conflict in the same cycle, then we can reduce the number of registers needed by the total number of conflicts or clique minus one.

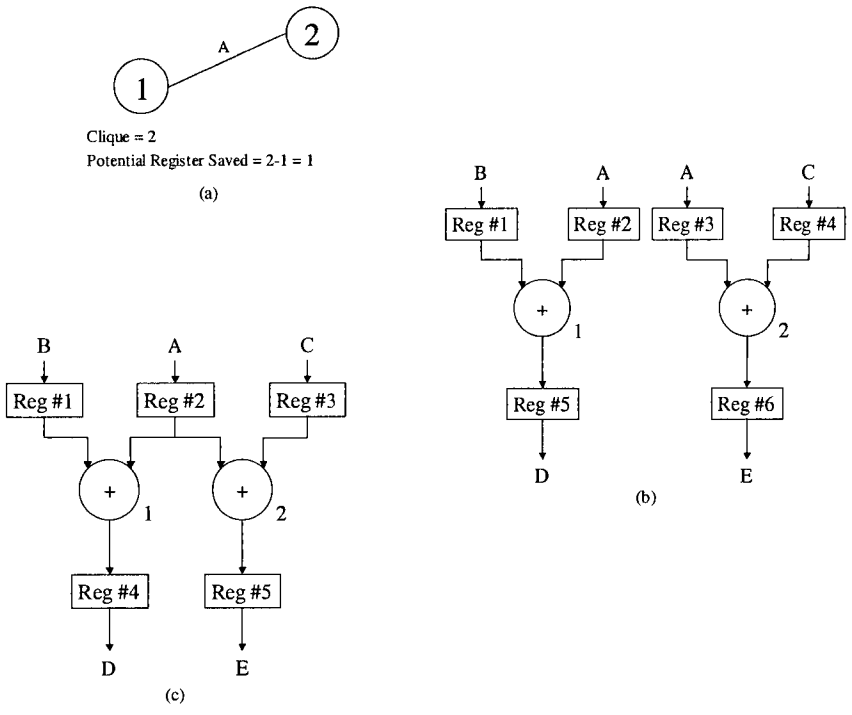


Figure 16. Register Savings from a CFG. (a) CFG. (b) DFG before utilizing CFG. (c) DFG after utilizing CFG.

For example, in Figure 16a, we see that node 1 and node 2 share an edge or have a conflict. If node 1 has an input of A and B and node 2 has an input of A and C, the CFG tells us that if we are able to schedule node 1 and node 2 in the same clock cycle, then we would be able to reduce the number of registers by one. The reason this is possible is because node 1 and node 2 can now use one register to store the value A instead of two registers (see Figure 16b and 16c).

To maintain a balance between latency and power, our scheme also uses multiple voltages (5V, 3.3V, 2.2V, and 1.8V) and a priority function that includes the depth and mobility of a node. Our heuristic, list-based resource-constrained scheduling scheme tries to balance the conflicting requirements of reducing the

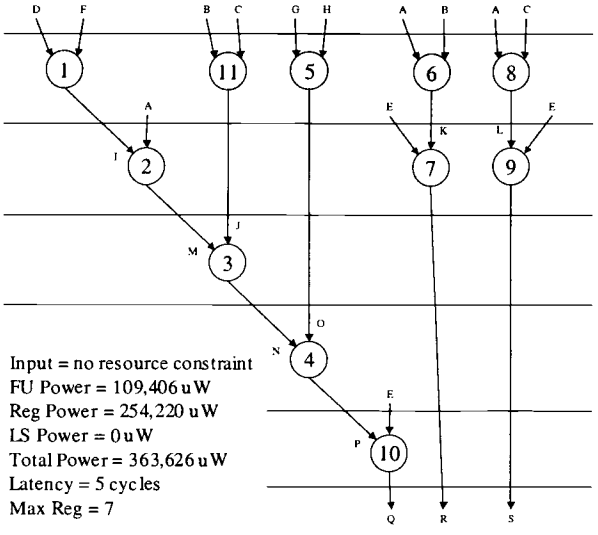
latency and utilizing resources operating at multiple voltages with the need to reduce the number of registers needed at each cycle and operates as follows:

1) *Determine the depth, mobility, ASAP, and ALAP of each node in the DFG.*
2) *Construct the ready set.*
3) *For the current cycle:*
   a. *For all nodes with a mobility of zero or less,*
      i. *Place all nodes with the same mobility into a single group and prioritize the nodes in each group according to highest depth first.*
      ii. *Prioritize the groups according to the lowest mobility first.*
      iii. *Schedule the prioritized nodes starting with the fastest available resource first and moving to the slowest resource.*
      iv. *Continue scheduling until all nodes with mobility less than zero have been scheduled or all resources have been used. If there are resources remaining, go to Step 3b. Otherwise go to Step 4.*
   b. *For all remaining nodes with a mobility greater than zero,*
      i. *Calculate the number of conflicts with each other.*
      ii. *Place all nodes with the same number of conflicts into a single group and prioritize the nodes in each group according to highest depth first.*
      iii. *Prioritize the groups according to the highest number of conflicts first.*
      iv. *Schedule the prioritized nodes starting with the fastest available resource first and moving to the slowest resource.*
      v. *Continue scheduling until all nodes with a conflict greater than zero have been scheduled or all resources have been used. If there are resources remaining, go to Step 3c. Otherwise go to Step 4.*
   c. *For all remaining nodes with no conflicts,*

   i. *Place all nodes with the same mobility into a single group and prioritize the nodes in each group according to highest depth first.*
   ii. *Prioritize the groups according to the lowest mobility first.*
   iii. *Schedule the prioritized nodes starting with the fastest available resource first and moving to the slowest resource.*
   iv. *Continue scheduling until all nodes have been scheduled or all resources have been used. Then go to Step 4.*

4) *Recalculate the mobility of each node, go to the next cycle, and repeat starting with Step 2.*

Our resource-constrained scheduling scheme schedules a DFG according to the following order of importance: latency, power determined by the number of registers, power determined from the operating voltages. Conventional schemes simply consider latency since power is determined by the resource constraint. Current multiple voltage schemes consider power determined from the operating voltages then latency. Figure 17 illustrates our resource-constrained scheduling scheme with a resource constraint of three adders—one at 5V, one at 3.3V, and one at 2.2V.



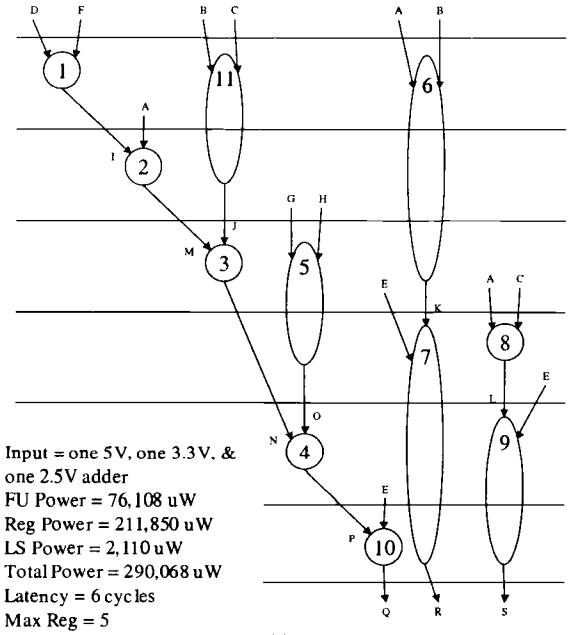
(a)

Node	ASAP	ALAP	Depth	Mobility
1	1	1	4	0
2	2	2	3	0
3	3	3	2	0
4	4	4	1	0
5	1	3	2	2
6	1	4	1	3
7	2	5	0	3
8	1	4	1	3
9	2	5	0	3
10	5	5	0	0
11	1	2	3	1

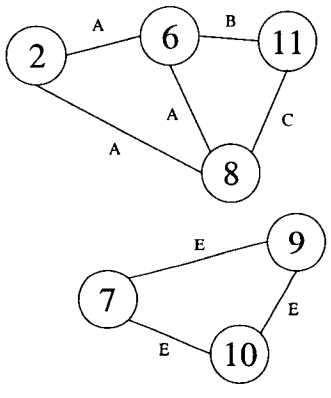
(b)

Control Cycle	Ready Set	5.0V	3.3V	2.2V
1	1 11 5 6 8	1	11	6
2	2 7 9 5 8	2	11'	6'
3	3 5 8	3	5	6'
4	4 7 9 8	8	5'	7
5	10 4 9	4	9	7'
6	10	10	9'	7'

(c)



(e)



(d)

Figure 17. Resource-Constrained Scheduling with Register Assignment Example. (a) Initial DFG. (b) ASAP, ALAP, depth, and mobility of each node. (c) Final Scheduling showing ready sets and scheduling at each cycle. (d) CFG. (e) Final DFG.

Figure 17 shows that our resource-constrained scheduling scheme with multiple voltages and register assignment was able to reduce the total power given a reduced number of resources to use. Instead of having five resources to use as in the no resource constraints case (see Figure 17a), there were only three resources available, each at a different voltage. These resources at different voltages allowed the total power to be reduced by approximately 20%. The maximum number of registers needed also reduced going from seven to five. This is similar to the example in Figure 13. However, the important item to note is that although the maximum number of registers needed is the same in the resource-constrained scheduling scheme with multiple voltages and our scheme, our scheme has a smaller total register power allowing it to achieve a 18% power reduction over the resource-constrained scheduling scheme with multiple voltages. This is possible since our scheme takes advantage of the CFG while the other two schemes do not.

Again though, these power and register reductions came at the cost of latency. Instead of a latency of five like in the no resource-constraints case, the latency is now one greater at six. This is still less than the resource-constrained scheduling scheme with multiple voltages case and is the lowest latency possible for this case. However, this increase in latency may still not be acceptable in some applications. And again, there is also a voltage level shifter overhead associated with this solution.

Overall, our resource-constrained scheduling scheme with multiple voltages and register assignment perform well in finding an optimal solution that addresses and balances the latency and power issues. Power is greatly reduced with only a slight increase in latency. If this slight increase is acceptable, then our scheduling scheme provides the best solution over the conventional resource-constrained scheduling scheme and the current resource constrained scheduling scheme with multiple voltages.

5.2. C++ Implementation

Our C++ implementation has three main parts: 1) the input file; 2) the scheduling scheme; and 3) the output file. The flow of our C++ implementation is shown in Figure 18.

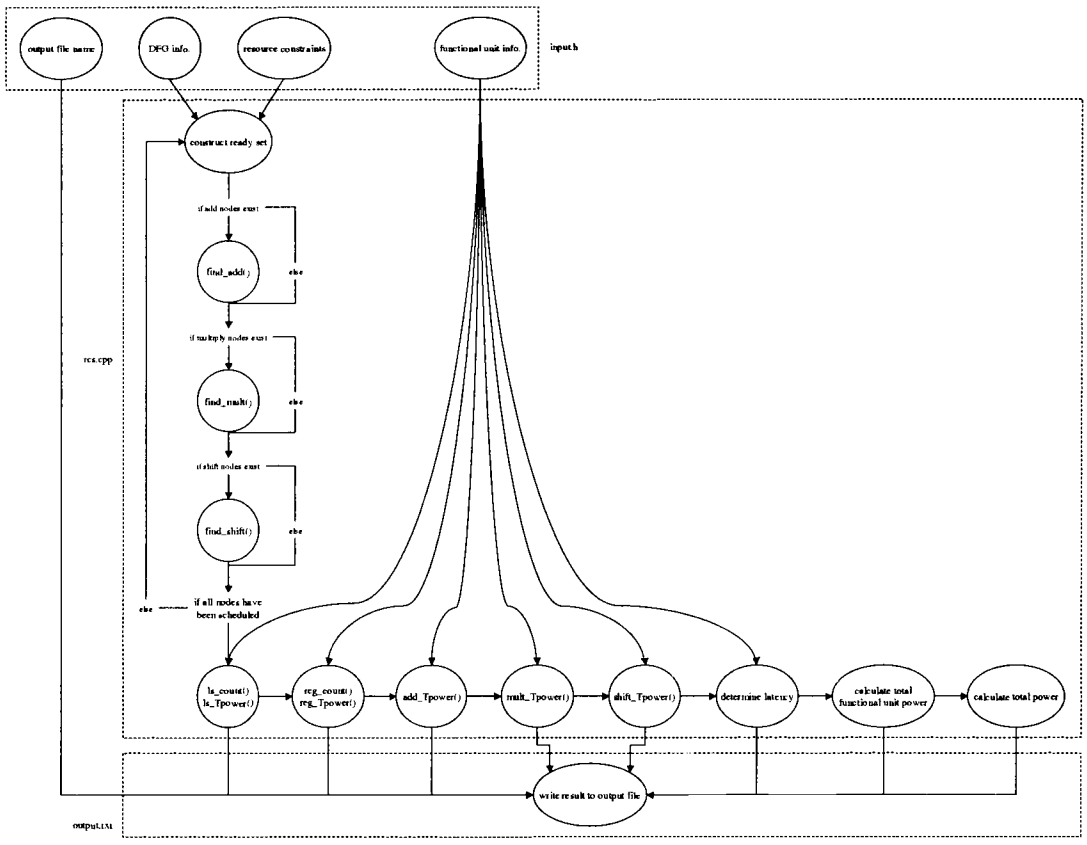


Figure 18. C++ Program Flow

The input file (input.h) is a C++ header file that includes information about the DFG, resource constraints, output file name, and delay and power of the functional units. The DFG is entered as a list of nodes where the format of each node is as follows: {Node Number, ASAP, ALAP, Depth, Mobility, Parent A, Parent B, Child, Node A, Node B, Operation}. The node number is the number of the current node. The ASAP, ALAP, depth, and mobility are defined in section 2. Parent A and B and child refer to the name of the input data values and output data value for the current node, respectively. Node A and B are the node numbers

providing the input data values to the current node. And operation is the type of operation of the current node. The output file name is simply the name the user specifies for the output file. The resource constraint is the maximum number of each type of resource that is available. The functional unit information is obtained from the delay and power characteristics in section 2.

The scheduling scheme (rcs.cpp) is the main portion of our implementation and performs the resource-constrained scheduling duties. It takes information from the input file and constructs the initial ready set, calls the appropriate scheduling functions, determines the power and register counts, and writes the results to the output file. The functions find_add(), find_mult(), and find_shift() select the best add, multiply, and shift node from the ready set and schedules them in the current cycle. The functions ls_count() and reg_count() find the number of level shifters and registers needed, respectively. The functions ls_Tpower(), reg_Tpower(), add_Tpower(), mult_Tpower(), and shift_Tpower() find the power usage of the level shifters, registers, adders, multipliers, and shifters, respectively. Total functional unit power refers to the power usage of all resources except level shifters and registers. The total power refers to the power usage of all resources: level shifters, registers, adders, multipliers, and shifters.

The output file is a simple text file containing the results, which include the ready set at each cycle, the scheduling at each cycle, the final scheduling, the latency, the power usage, and the maximum number of registers needed.

6. Benchmarks

In this section, we compare our resource-constrained scheduling scheme that utilizes multiple voltages and register assignment with conventional schemes and current multiple voltage schemes. The goal is to determine which scheme performs best in finding a scheduling solution that balances latency and power. We run only one simulation using a simple example to compare all three of the schemes since it has already been proven in [8], [11], [12], and [13] that multiple voltage schemes perform better than conventional schemes. This single simulation is used to verify that these results still remain true.

The remaining simulations compare only the current multiple voltage scheduling scheme from [12] and [13] and our scheduling scheme. We use five benchmarks: 1) a simple example; 2) a 2^{nd} order lattice filter; 3) a 5^{th} order elliptic wave filter; 4) an 8-point fast fourier transform (FFT); and 5) a fast discrete cosine transform (FDCT). With the exception of the simple example, these benchmarks are typical DSP kernels that could be used in mobile devices.

6.1 Simple Example

The initial DFG and final DFG for this benchmark for the three scheduling schemes can be seen in Figure 12a, 13a, and 17a and Figure 12e, 13e, and 17e. Table 6 summarizes the results for each of the three scheduling schemes. The conventional scheduling scheme has a resource constraint of three adders all

operating at 5V. The multiple voltages scheduling scheme and our scheduling scheme both have a resource constraint of three adders—one at 5V, one at 3.3V, and one at 2.2V.

Table 6. Simple Example Benchmark Comparisons

	No Constraints (See Figure 12a)	Conventional Scheme (See Figure 12e)	Multiple Voltage Scheme (See Figure 13e)	Our Scheme (See Figure 17e)
Power Reduction (%)	--	0.00%	1.63%	20.20%
Total Power (uW)	363,626	363,626	357,713	290,168
FU Power (uW)	109,406	109,406	76,108	76,108
Reg Power (uW)	254,220	254,220	279,642	211,950
LS Power (uW)	0	0	1,963	2,110
Latency (# cycles)	5	5	8	6
Registers Needed	7	7	5	5

From the results, we notice that the conventional scheme achieves the best latency followed by our scheme and then the multiple voltages scheme. However, the conventional scheme does not reduce the overall power usage at all. Hence, the conventional scheme does not do a good job in finding a scheduling solution that balances the latency and power issues, which is the goal. The multiple voltages scheme performs better than the conventional scheme since it is able to reduce the power usage while keeping a relatively low latency. It does this by taking advantage of functional units operating at multiple voltages.

Overall though, our scheduling scheme provides the best solution since it is able to reduce power more and keep the latency lower than both the conventional scheme and the multiple voltages scheme. We do this by not only taking advantage of functional units operating at multiple voltages, but also by reducing the number of registers needed at each cycle. Although both the multiple voltages scheme and

our scheme have the same register need of five, it is important to note that this is only the maximum register needed. Not all five registers are necessarily used at each cycle. This explains the difference between the "Reg Power" categories in the table. Notice that our scheme has the lowest total register power used since we specifically take into account the register contribution while the other two schemes do not. Remember, however, that our scheme and the scheme with multiple voltages have a level shifter overhead while the conventional scheme does not.

Table 7 and Table 8 summarize the results for more simulations comparing the multiple voltages scheme and our scheme, respectively. Optimization (Opt) 0 has no resource constraints. Opt 1 has five adders all operating at 5V. Opt 2 has five adders—two at 5V, one at 3.3V, one at 2.2V, and one at 1.8V. Opt 3 has three adders—one at 5V, one at 3.3V, and one at 2.2V. And Opt 4 has two adders—one at 5V and one at 3.3V.

Table 7. Simple Example Results for Resource-Constrained Scheduling with Multiple Voltages

	Opt 0	Opt 1	Opt 2	Opt 3	Opt 4	AVG
Power Reduction (%)	--	0.00%	19.79%	1.63%	-3.54%	4.47%
Total Power (uW)	363,626	363,626	291,660	357,713	376,490	347,372.17
FU Power (uW)	109,406	109,406	69,780	76,108	86,606	85,475.00
Reg Power (uW)	254,220	254,220	219,252	279,642	288,031	260,286.27
LS Power (uW)	0	0	2,628	1,963	1,853	1,610.90
Latency (# cycles)	5	5	8	8	8	7.25
Registers Needed	7	7	7	5	6	6.25
Opt 0: no resource constraints						
Opt 1: five adders all at 5.0V						
Opt 2: five adders - two at 5.0V, one at 3.3V, one at 2.2V, and one at 1.8V						
Opt 3: three adders - one at 5V, one at 3.3V, and one at 2.2V						
Opt 4: two adders - one at 5V and one at 3.3V						

Table 8. Simple Example Results for Resource-Constrained Scheduling with Multiple Voltages and Register Assignment

	Opt 0	Opt 1	Opt 2	Opt 3	Opt 4	AVG
Power Reduction (%)	--	0.00%	15.08%	20.23%	10.62%	11.48%
Total Power (uW)	363,626	363,626	308,779	290,068	325,005	321,869.50
FU Power (uW)	109,406	109,406	86,743	76,108	86,606	89,715.75
Reg Power (uW)	254,220	254,220	220,311	211,850	237,258	230,909.75
LS Power (uW)	0	0	1,725	2,110	1,141	1,244.00
Latency (# cycles)	5	5	6	6	8	6.25
Registers Needed	7	7	7	5	5	6.00
Opt 0: no resource constraints						
Opt 1: five adders all at 5.0V						
Opt 2: five adders - two at 5.0V, one at 3.3V, one at 2.2V, and one at 1.8V						
Opt 3: three adders - one at 5V, one at 3.3V, and one at 2.2V						
Opt 4: two adders - one at 5V and one at 3.3V						

Opt 1 is identical for both the resource-constrained scheduling scheme with multiple voltages and our scheduling scheme. The reason is due to the fact that we are not using multiple voltages, only limiting the number of resources available, which does not allow enough room for any reductions to occur.

In Opt 2, the resource-constrained scheduling scheme with multiple voltages performs almost 5% better than our scheduling scheme having a power reduction of 19.79% while ours has a power reduction of 15.08%. This occurs due to the differences in the total FU power. Our scheduling scheme has a higher total FU power at 86,743 uW compared to the multiple voltages scheme at 69,780 uW. The reason this occurs is because our scheme schedules the nodes according to the following priority: latency, power determined by the number of registers, and power determined from the operating/supply voltages. Our scheme utilizes the higher voltage functional units first in order to meet the latency goals causing it to

have a higher total FU power. The multiple voltages scheme starts with the lower functional units first allowing it to reduce more of the total FU power. This along with not being able to minimize the total register power—there is not much room for reduction—causes our scheme to have a total lower power reduction. On the other hand, our scheme has less level shifter overhead (1,725 uW compared to 2,628) and a lower latency (six compared to eight). Again, the reason is because our scheme puts less emphasis on the multiple voltages and balances the power and latency more than the multiple voltages scheme.

In Opt 3 and Opt 4, our scheme is able to reduce the power by 20.23% and 10.62%, respectively, compared to the multiple voltages scheme at 1.63% and negative 3.54%. The primary reason this occurs is due to the fact that our scheduling scheme is able to significantly reduce the total register power while the multiple voltages scheme is not. The latency is also less in our scheduling scheme. In Opt 4, the multiple voltages scheme actually uses more power than the no resource constraints case, Opt 0, because it is unable to reduce the total register power and because it has power contributions from the level shifters.

On average, our scheduling scheme performs better than the resource-constrained scheduling scheme with only multiple voltages. Our scheduling scheme has an average power reduction of 11.48% while the multiple voltages scheme has only an average power reduction of 4.47%. The latency in our scheme is also slightly better at seven compared to eight. In the cases where latency is a

higher priority than power, the solutions from our scheduling scheme would be the better choices. However, in the cases where power is a higher priority, for Opt 2, the resource-constrained scheduling scheme with multiple voltages solution would be the better choice while for the other optimizations, our scheduling scheme solutions would be the better choice. And finally, in the case where a balance is needed, our scheduling solutions would be the better choice since they are low power and low latency while the resource-constrained scheduling scheme with multiple voltages solutions are low power but not necessarily low latency.

6.2 Lattice Filter

Filters are typically used in sampling data. A lattice filter is only one of the many different types of filters used in sampling. The DFG of a 2^{nd} order lattice filter obtained from [15] and used for simulation is show in Figure 19.

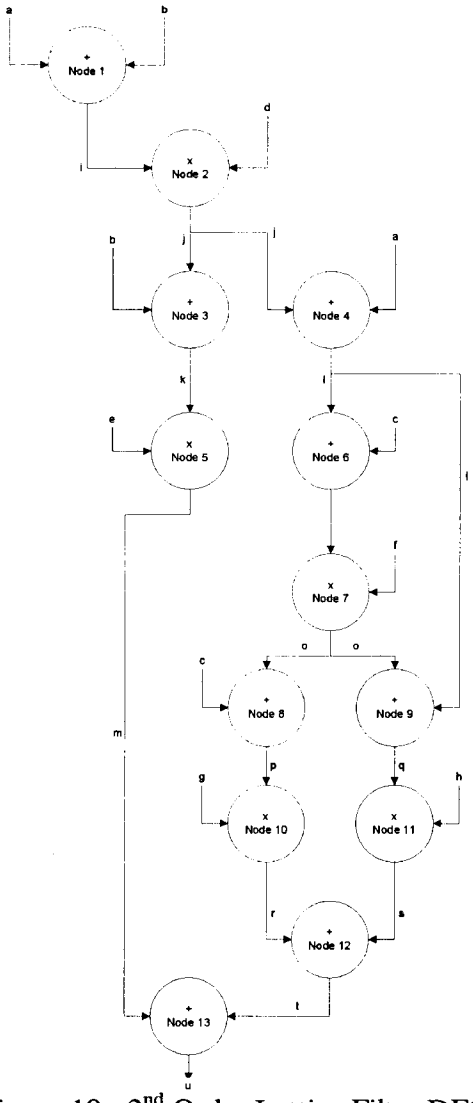


Figure 19. 2nd Order Lattice Filter DFG

Table 9 and Table 10 summarize the lattice filter simulations comparing the multiple voltages scheme and our scheme, respectively. Optimization (Opt) 0 has no resource constraints. Opt 1 has a four adders and four multipliers all operating at 5V. Opt 2 has four adders—one at 5V, one at 3.3V, one at 2.2V, and one at 1.8V—and four multipliers—one at 5V, one at 3.3V, one at 2.2V, and one at 1.8V. Opt 3

has two adders—one at 5V and one at 3.3V—and two multipliers—one at 5V and one at 3.3V. And Opt 4 has one adder and one multiplier both at 3.3V.

Table 9. Lattice Filter Results for Resource-Constrained Scheduling with Multiple Voltages

	Opt 0	Opt 1	Opt 2	Opt 3	Opt 4	AVG
Power Reduction (%)	--	0.00%	6.18%	6.59%	18.24%	7.75%
Total Power (uW)	442,856	442,856	415,493	413,673	362,076	408,524.51
FU Power (uW)	231,018	231,018	186,295	184,898	98,618	175,207.25
Reg Power (uW)	211,838	211,838	228,022	228,022	263,102	232,745.86
LS Power (uW)	0	0	1,176	754	356	571.40
Latency (# cycles)	15	15	21	20	40	24.00
Registers Needed	4	4	3	3	4	3.50
Opt 0: no resource constraints						
Opt 1: four adders and four multipliers all at 5.0V						
Opt 2: four adders and four multipliers - one of each at 5V, 3.3V, 2.2V, and 1.8V						
Opt 3: two adders and two multipliers - one of each at 5V and 3.3V						
Opt 4: one adder and one multiplier both at 3.3V						

Table 10. Lattice Filter Results for Resource-Constrained Scheduling with Multiple Voltages and Register Assignment

	Opt 0	Opt 1	Opt 2	Opt 3	Opt 4	AVG
Power Reduction (%)	--	0.00%	8.25%	8.25%	20.22%	9.18%
Total Power (uW)	442,856	442,856	406,329	406,329	353,293	402,201.73
FU Power (uW)	231,018	231,018	184,898	184,898	98,618	174,858.00
Reg Power (uW)	211,838	211,838	220,311	220,311	254,205	226,666.13
LS Power (uW)	0	0	1,120	1,120	470	677.60
Latency (# cycles)	15	15	23	23	40	25.25
Registers Needed	4	4	3	3	4	3.50
Opt 0: no resource constraints						
Opt 1: four adders and four multipliers all at 5.0V						
Opt 2: four adders and four multipliers - one of each at 5V, 3.3V, 2.2V, and 1.8V						
Opt 3: two adders and two multipliers - one of each at 5V and 3.3V						
Opt 4: one adder and one multiplier both at 3.3V						

In all the optimizations, our scheduling scheme provides the best low power solution at approximately a 2% power reduction over the resource-constrained

scheduling scheme with multiple voltages. Our scheduling scheme is able to achieve this by effectively reducing the total register power. However, the latency is slightly larger than the resource-constrained scheduling scheme with multiple voltages. This occurs due to the scheduling of node 5. In the multiple voltages scheme, node 5 and node 7 are ready to be scheduled at the same time. Therefore, node 5 is assigned to the lower voltage multiplier unit, allowing node 7 to be assigned to the higher voltage multiplier unit, which allows the latency to be lower than in our scheme. In our scheme, node 5 is ready to be scheduled before node 7. Therefore, node 5 is assigned to the higher voltage multiplier unit and node 7 has to be assigned to the lower voltage multiplier unit since the higher voltage multiplier unit is not free when node 7 is ready to be scheduled. This causes the latency in our scheme to be slightly higher than the latency in the multiple voltages scheme.

Overall, in this example, both schemes equally balance the latency and power issues. Our scheduling scheme provides solutions with lower power at 9.18% compared to 7.75% while the multiple voltages scheme provides solutions with lower latency at 24 cycles compared to 26 cycles. The scheme to choose in this case would depend on whether latency is more important or whether power is more important. If power is more important, our scheme is the better choice. If latency is more important, then the multiple voltages scheme is the better choice.

6.3 Elliptic Wave Filter

An elliptic wave filter is another type of filter used in sampling data. The DFG of a 5th order elliptic wave filter obtained from [16] and used for simulation is show in Figure 20.

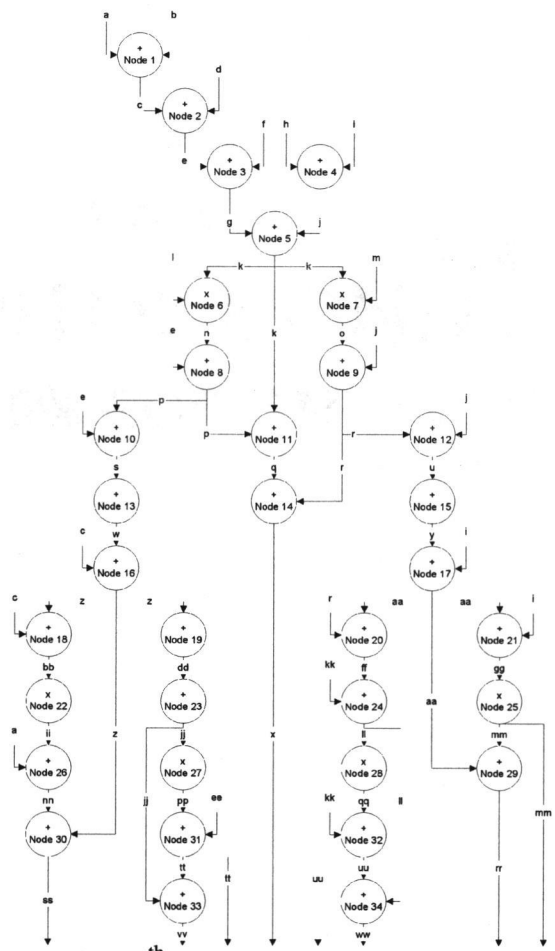


Figure 20. 5th Order Elliptic Wave Filter DFG

Table 11 and Table 12 summarize the elliptic wave filter simulations comparing the multiple voltages scheme and our scheme, respectively.

Optimization (Opt) 0 has no resource constraints. Opt 1 has a four adders and two multipliers all operating at 5V. Opt 2 has four adders—one at 5V, one at 3.3V, one at 2.2V, and one at 1.8V—and two multipliers—one at 5V and one at 3.3V. Opt 3 has two adders—one at 5V and one at 3.3V—and one multiplier at 5V. And Opt 4 has one adder and one multiplier both at 3.3V.

Table 11. Elliptic Wave Filter Results for Resource-Constrained Scheduling with Multiple Voltages

	Opt 0	Opt 1	Opt 2	Opt 3	Opt 4	AVG
Power Reduction (%)	--	0.00%	12.84%	3.12%	5.13%	5.27%
Total Power (uW)	1,077,114	1,077,114	938,781	1,043,484	1,021,806	1,020,296.45
FU Power (uW)	500,916	500,916	389,711	461,016	352,716	426,089.75
Reg Power (uW)	576,198	576,198	543,744	578,825	666,526	591,323.20
LS Power (uW)	0	0	5,326	3,643	2,565	2,883.50
Latency (# cycles)	20	20	30	29	57	34.00
Registers Needed	7	7	5	4	4	5.00
Opt 0: no resource constraints						
Opt 1: four adders and two multipliers all at 5V						
Opt 2: four adders and two multipliers - one of each at 5V and 3.3V, then one adder at 2.2V and one at 1.8V						
Opt 3: two adders - one at 5V, one at 3.3V; one multiplier at 5V						
Opt 4: one adder and one multiplier both at 3.3V						

Table 12. Elliptic Wave Filter Results for Resource-Constrained Scheduling with Multiple Voltages and Register Assignment

	Opt 0	Opt 1	Opt 2	Opt 3	Opt 4	AVG
Power Reduction (%)	--	0.00%	10.12%	3.26%	5.57%	4.74%
Total Power (uW)	1,077,114	1,077,114	968,077	1,042,014	1,017,071	1,026,069.10
FU Power (uW)	500,916	500,916	431,996	472,416	352,716	439,511.00
Reg Power (uW)	576,198	576,198	533,831	567,725	660,933	584,671.50
LS Power (uW)	0	0	2,250	1,874	3,422	1,886.60
Latency (# cycles)	20	20	26	29	55	32.50
Registers Needed	7	7	5	4	4	5.00
Opt 0: no resource constraints						
Opt 1: four adders and two multipliers all at 5V						
Opt 2: four adders and two multipliers - one of each at 5V and 3.3V, then one adder at 2.2V and one at 1.8V						
Opt 3: two adders - one at 5V, one at 3.3V; one multiplier at 5V						
Opt 4: one adder and one multiplier both at 3.3V						

The results for the elliptic wave filter provide the same insight as the results for the simple example. In all optimizations, the latency and total register power for our scheduling scheme are lower than for the resource-constrained scheduling scheme with multiple voltages. The only case where latency is not lower is in Opt 3 where both latency values are the same at 29 cycles. Our scheme also has a higher power reduction, but only in two of the four cases. In Opt 1, both scheduling schemes could achieve no power reduction for the same reasons as in Opt 1 for the simple example. In Opt 2, the multiple voltages scheme has a higher power reduction at 12.84% compared to our scheme at 10.12%. This occurs for the same reason as it does in the simple example. The savings from minimizing the total register power is not enough to offset the fact that our scheme has a higher total FU power since it utilizes FUs at higher voltages first. Our scheduling scheme performs better on two out of the three optimizations with power reductions. However, the multiple voltages scheme has a better average power reduction. This occurs since our scheme has a lower power reduction in Opt 2.

6.4 Fast Fourier Transform (FFT)

A fast fourier transform (FFT) is typically used in audio sampling or digital audio. The DFG of an 8-point FFT obtained from [17] and used for simulation is show in Figure 21.

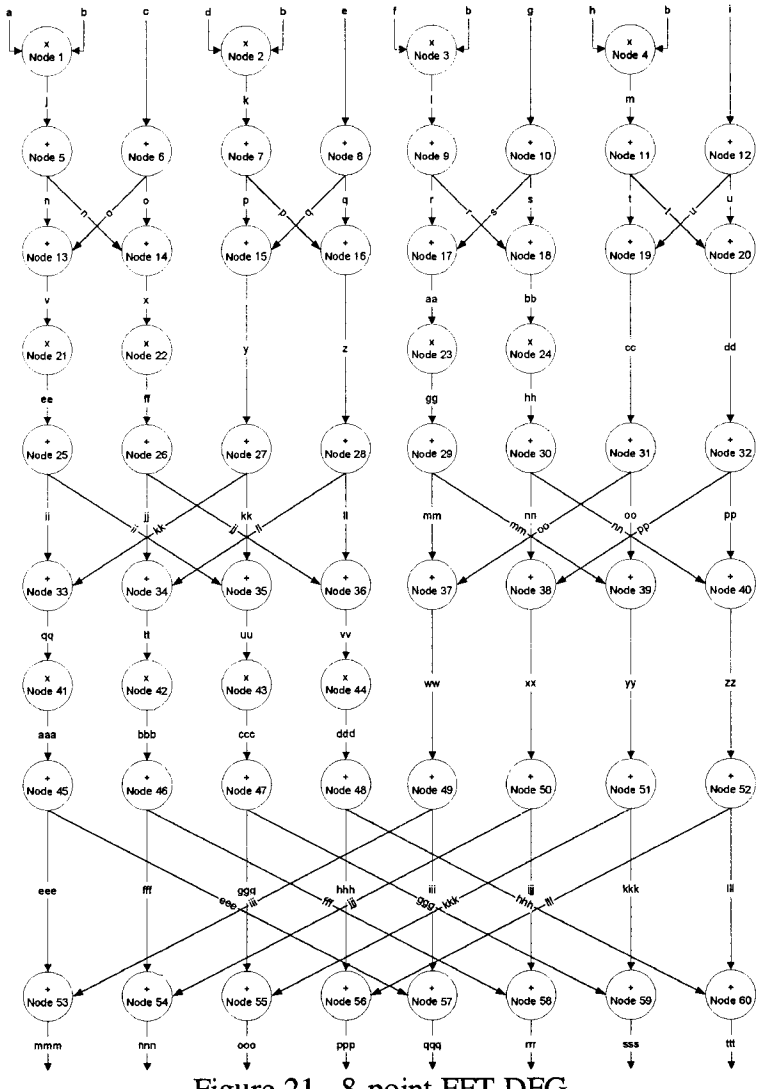


Figure 21. 8-point FFT DFG

Table 13 and Table 14 summarize the FFT simulations comparing the multiple voltages scheme and our scheme, respectively. Optimization (Opt) 0 has no resource constraints. Opt 1 has eight adders and four multipliers all operating at 5V. Opt 2 has eight adders—two at 5V, two at 3.3V, two at 2.2V, and two at 1.8V—and four multipliers—one at 5V, one at 3.3V, one at 2.2V, and one at 1.8V.

Opt 3 has four adders—one at 5V, one at 3.3V, one at 2.2V, and one at 1.8V—and two multipliers—one at 5V and one at 3.3V. And Opt 4 has two adders—one at 5V and one at 3.3V—and one multiplier at 3.3V.

Table 13. FFT Results for Resource-Constrained Scheduling with Multiple Voltages

	Opt 0	Opt 1	Opt 2	Opt 3	Opt 4	AVG
Power Reduction (%)	--	0.00%	13.42%	2.76%	-11.42%	1.19%
Total Power (uW)	1,544,189	1,544,189	1,336,906	1,501,644	1,720,583	1,525,830.48
FU Power (uW)	840,888	840,888	557,224	587,608	749,688	683,852.00
Reg Power (uW)	703,301	703,301	771,766	903,317	964,708	835,773.08
LS Power (uW)	0	0	7,916	10,718	6,187	6,205.40
Latency (# cycles)	15	15	35	30	39	29.75
Registers Needed	12	12	9	9	6	9.00
Opt 0: no resource constraints						
Opt 1: eight adders and four multipliers all at 5V						
Opt 2: eight adders - two of at 5V, two at 3.3V, two at 2.2V, and two at 1.8V; four multipliers - one at 5V, one at 3.3V, one at 2.2V, and one at 1.8V						
Opt 3: four adders - one at 5V, one at 3.3V, one at 2.2V, one at 1.8V; two multipliers - one at 5V and one at 3.3v						
Opt 4: two adders - one at 5V and one at 3.3V; one multiplier at 3.3V						

Table 14. FFT Results for Resource-Constrained Scheduling with Multiple Voltages and Register Assignment

	Opt 0	Opt 1	Opt 2	Opt 3	Opt 4	AVG
Power Reduction (%)	--	0.00%	12.19%	4.92%	-7.67%	2.36%
Total Power (uW)	1,544,189	1,544,189	1,356,008	1,468,209	1,662,660	1,507,766.23
FU Power (uW)	840,888	840,888	592,789	602,035	761,088	699,200.00
Reg Power (uW)	703,301	703,301	754,142	855,824	898,191	802,864.13
LS Power (uW)	0	0	9,077	10,350	3,381	5,702.10
Latency (# cycles)	15	15	27	28	39	27.25
Registers Needed	12	12	9	9	5	8.75
Opt 0: no resource constraints						
Opt 1: eight adders and four multipliers all at 5V						
Opt 2: eight adders - two of at 5V, two at 3.3V, two at 2.2V, and two at 1.8V; four multipliers - one at 5V, one at 3.3V, one at 2.2V, and one at 1.8V						
Opt 3: four adders - one at 5V, one at 3.3V, one at 2.2V, one at 1.8V; two multipliers - one at 5V and one at 3.3v						
Opt 4: two adders - one at 5V and one at 3.3V; one multiplier at 3.3V						

Again, our scheduling scheme solutions have a lower latency and a lower total register power for all the optimizations. However, it is only able to have a better power reduction in Opt 3 and Opt 4. Our scheduling scheme performs slightly worse in Opt 2, again, for the same reasons as in the elliptic wave filter and simple example simulations. The lower total register power was not enough to compensate for the higher total functional unit power. The negative values for Opt 4 tell us that neither of the two scheduling algorithms could find a scheduling solution that minimizes latency and reduces power for the resource constraint given. The multiple voltages scheme has a larger negative number in Opt 4 than our scheme stating that the multiple voltages scheme performed worse. In Opt 4, more power is used than in the no constraints case. The reason is due primarily to the increase in the total register power needed caused by having limited resources available. Overall, on average, our scheduling scheme performs better than the multiple voltages scheme in this simulation having a power reduction of 2.36% compared to 1.19% and a latency of 28 cycles compared to 30 cycles.

6.5 Discrete Cosine Transform (DCT)

The DCT is used to decompose an image into a set of waveforms by removing spatial redundancies. It is widely used in most image and motion compression. Although there are many different versions of the DCT, we focus on the fast DCT described in [18]. This fast DCT uses a fast transform in order to

reduce the overall number of calculations and improve performance over the conventional DCT. The fast DCT DFG is shown in Figure 22.

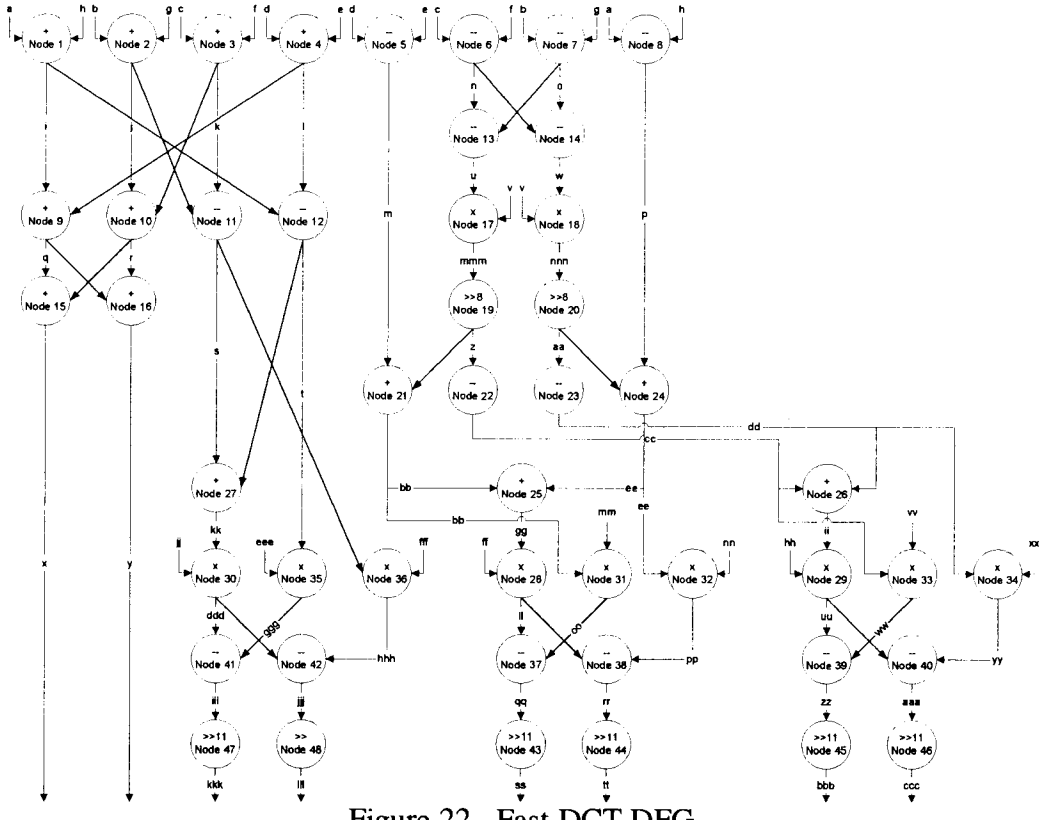


Figure 22. Fast DCT DFG

Table 15 and Table 16 summarize the fast DCT simulations comparing the multiple voltages scheme and our scheme, respectively. Optimization (Opt) 0 has no resource constraints. Opt 1 has eight adders, nine multipliers, and six shifters (assumed to have the same delay and power characteristics as adders) all operating at 5V. Opt 2 has eight adders—two at 5V, two at 3.3V, two at 2.2V, and two at 1.8V—nine multipliers—three at 5V, two at 3.3V, two at 2.2V, and two at 1.8V—

and six shifters—two at 5V, two at 3.3V, one at 2.2V, and one at 1.8V. Opt 3 has four adders—one at 5V, one at 3.3V, one at 2.2V, and one at 1.8V—five multipliers—two at 5V, one at 3.3V, one at 2.2V, and one at 1.8V—and three shifters—one at 5V, one at 3.3V, and one at 2.2V. And Opt 4 has two adders—one at 5V and one at 3.3V—three multipliers—one at 5V, one at 3.3V, and one at 2.2V—and two shifters—one at 5V and one at 3.3V.

Table 15. Fast DCT Results for Resource-Constrained Scheduling with Multiple Voltages

	Opt 0	Opt 1	Opt 2	Opt 3	Opt 4	AVG
Power Reduction (%)	--	0.00%	16.08%	6.25%	-2.78%	4.89%
Total Power (uW)	1,396,019	1,396,019	1,171,520	1,308,745	1,434,877	1,327,790.30
FU Power (uW)	701,192	701,192	487,560	484,396	525,680	549,707.00
Reg Power (uW)	694,827	694,827	675,296	815,617	903,317	772,264.20
LS Power (uW)	0	0	8,665	8,732	5,880	5,819.10
Latency (# cycles)	13	13	36	38	33	30.00
Registers Needed	10	10	8	8	8	8.50
Opt 0: no resource constraints						
Opt 1: eight adders, nine multipliers, six shifters all at 5V						
Opt 2: eight adders - two at 5V, two at 3.3V, two at 2.2V, and two at 1.8V; nine multipliers - three at 5V, two at 3.3V, two at 2.2V, and two at 1.8V; six shifters - two at 5V, two at 3.3V, one at 2.2V, and one at 1.8V						
Opt 3: four adders - one at 5V, one at 3.3V, one at 2.2V, and one at 1.8V; five multipliers - two at 5V, one at 3.3V, one at 2.2V, and one at 1.8V; three shifters - one at 5V, one at 3.3V, and one at 2.2V						
Opt 4: two adders - one at 5V and one at 3.3V; three multipliers - one at 5V, one at 3.3V, and one at 2.2V; two shifters - one at 5V and one at 3.3V						

Table 16. Fast DCT Results for Resource-Constrained Scheduling with Multiple Voltages and Register Assignment

	Opt 0	Opt 1	Opt 2	Opt 3	Opt 4	AVG
Power Reduction (%)	--	0.00%	14.44%	5.80%	-0.53%	4.93%
Total Power (uW)	1,396,019	1,396,019	1,194,502	1,315,068	1,403,400	1,327,247.13
FU Power (uW)	701,192	701,192	531,654	518,115	542,780	573,435.25
Reg Power (uW)	694,827	694,827	652,460	788,036	855,824	747,786.38
LS Power (uW)	0	0	10,388	8,918	4,796	6,025.50
Latency (# cycles)	13	13	22	28	29	23.00
Registers Needed	10	10	8	9	10	9.25
Opt 0: no resource constraints						
Opt 1: eight adders, nine multipliers, six shifters all at 5V						
Opt 2: eight adders - two at 5V, two at 3.3V, two at 2.2V, and two at 1.8V; nine multipliers - three at 5V, two at 3.3V, two at 2.2V, and two at 1.8V; six shifters - two at 5V, two at 3.3V, one at 2.2V, and one at 1.8V						
Opt 3: four adders - one at 5V, one at 3.3V, one at 2.2V, and one at 1.8V; five multipliers - two at 5V, one at 3.3V, one at 2.2V, and one at 1.8V; three shifters - one at 5V, one at 3.3V, and one at 2.2V						
Opt 4: two adders - one at 5V and one at 3.3V; three multipliers - one at 5V, one at 3.3V, and one at 2.2V; two shifters - one at 5V and one at 3.3V						

With the exception of Opt 4 where both schemes fail to find a good scheduling solution, the resource-constrained scheduling scheme with multiple voltages performs slightly better than our scheme in reducing power. For Opt 2 and Opt 3, the multiple voltages scheme is able to achieve power reductions of 16.08% and 6.25%, respectively, compared to our scheduling scheme, which was only capable of obtaining power reductions of 14.44% and 5,80%. Again, the reason for this is the same as in the FFT simulation for Opt 2, the elliptic wave filter simulation for Opt 2, and the simple example simulation for Opt 2. The total register power for our scheme is less than for the multiple voltages scheme. However, the gain from having a lower total register power was not enough to offset the higher total FU power in our scheduling scheme. Our scheme, however,

was able to achieve significantly lower latencies for all optimizations than the multiple voltages scheme.

On average, our scheme performed slightly better on reducing power at 4.93% compared to the multiple voltages scheme at 4.89%. Our scheme also performed significantly better on reducing latency with an average of 23 cycles compared to the multiple voltages scheme with an average of 30 cycles. The only drawback is that our scheme requires one more register than the multiple voltages scheme at ten registers needed compared to nine. If ten registers were not available, then our scheduling scheme solution would not be an option.

6.6 Optimal Scheduling Scheme

Table 17 and 18 summarizes the average performance of the resource-constrained scheduling scheme with multiple voltages and our resource-constrained scheduling scheme with multiple voltages and register assignment for all the simulations.

Table 17. Average Performance of Resource-Constrained Scheduling with Multiple Voltages

	Simple	Lattice Filter	Elliptic Wave Filter	FFT	Fast DCT	AVG
Power Reduction (%)	4.47%	7.75%	5.27%	1.19%	4.89%	4.71%
Total Power (uW)	347,372	408,525	1,020,296	1,525,830	1,327,790	925,963
FU Power (uW)	85,475	175,207	426,090	683,852	549,707	384,066
Reg Power (uW)	260,286	232,746	591,323	835,773	772,264	538,479
LS Power (uW)	1,611	571	2,884	6,205	5,819	3,418
Latency (# cycles)	7	24	34	30	30	25
Registers Needed	6	4	5	9	9	7

Table 18. Average Performance of Resource-Constrained Scheduling with Multiple Voltages and Register Assignment

	Simple	Lattice Filter	Elliptic Wave Filter	FFT	Fast DCT	AVG
Power Reduction (%)	11.48%	9.18%	4.74%	2.36%	4.93%	6.54%
Total Power (uW)	321,870	402,202	1,026,069	1,507,766	1,327,247	917,031
FU Power (uW)	89,716	174,858	439,511	699,200	573,435	395,344
Reg Power (uW)	230,910	226,666	584,672	802,864	747,786	518,580
LS Power (uW)	1,244	678	1,887	5,702	6,026	3,107
Latency (# cycles)	6	25	33	27	23	23
Registers Needed	6	4	5	9	9	7

From the results, we see that our scheduling scheme, on average, performs better in reducing power and latency than the resource-constrained scheduling scheme using just multiple voltages. Our scheme has an average power reduction of 6.54% while the multiple voltages scheme has an average power reduction of 4.71%. Our scheduling scheme latency is also better with an average of 23 cycles compared to 25 cycles. Our scheme achieves a scheduling solution that optimally balances the latency and power allowing it to perform better than the multiple voltages scheme. Looking at each of the simulations separately, this is also generally true. With the exception of the elliptic wave filter simulation, our scheduling scheme provides solutions that are better than the solutions provided by the multiple voltages scheme.

The only time our scheduling scheme performs slightly worse than the multiple voltages scheme is when the gains from reducing power from the registers is too small to negate the power used by the functional units. During this case, our scheme still has a lower latency. But the power reduction is less than with the multiple voltages scheme. When this case occurs, the best choice depends on the

actual latency and power requirements. For example, in the elliptic wave filter simulation, if the power reduction requirement is 5.0% and the latency is 34 cycles, then the solution from the multiple voltages scheme is the better choice since our scheme is only capable of a 4.74% power reduction even though our scheme only has a latency of 33 cycles while the multiple voltage has a latency of 34. However, if the power reduction requirement is only 4% and the latency is 35 cycles, then the choice depends on the users preference between power and latency. If power were more important, then the multiple voltages scheme would be the better choice. However, if latency were more important, then our scheduling scheme solution would be the better choice. Generally, though, a faster device that is still low power is preferred over a slow, low power device.

It is important to note that since our scheduling scheme focuses on minimizing latency first, then the power from the registers, and finally the power from the functional units, our scheme will always have a slightly larger functional unit power usage than the multiple voltages scheme. Also, as the latency increases (which occurs from using functional units at multiple voltages or decreasing the number of available functional units), the power from the registers increase, and the power from the functional units decrease. Hence, in order to obtain a good scheduling solution, it is important to balance the latency with the power from the registers and the power from the functional units. From the benchmark results, our scheduling scheme balances these relationships best.

7. Conclusion

From our simulations, we show that, when performing resource-constrained scheduling, it is important not only to consider the functional units or the number of resources, but it is also important to consider the memory as well. We also illustrate the relationships and trade-offs between the number of available resources, the latency, the power from registers, and the power from functional units. In order to find an optimal solution, these relationships and trade-offs need to be delicately balanced. Our resource-constrained scheduling scheme that utilizes multiple voltages and register assignment is capable of finding the most optimal solution by balancing the latency, power from registers, and power from functional units. On average, it performs better than conventional resource-constrained scheduling schemes and current resource-constrained scheduling schemes that utilize only multiple voltages. Our benchmark results show that our resource-constrained scheduling scheme is capable of a 6.5% power reduction improvement over conventional resource-constrained scheduling algorithms and a 1.83% power reduction improvement over resource-constrained scheduling schemes utilizing only multiple voltages while still maintaining a low latency and a lower latency than with resource-constrained scheduling schemes utilizing only multiple voltages. Hence, our scheduling scheme would be a viable and valuable technique used to optimize applications or algorithms intended for mobile devices or other low power, low latency devices.

Our future plan is to continue building our cell libraries to include more than just a 32-bit carry-ripple adder, a 32-bit carry-ripple multiplier, and a 32-bit register. We would also like to include more technologies than the three that we currently have access to. This would make our scheduling algorithm even more valuable since modern technology uses a vast array of functional units and a number of different process technologies. We are also looking at applying our resource-constrained techniques to other scheduling algorithms such as latency-constrained scheduling in order to develop more optimal schemes in those areas as well. Currently, a complete latency-constrained scheduling solution that deals with latency, power from registers, and power from functional units does not exist. Latency-constrained schemes, like the ones in [14], [19], and [20], either deal with the latency, power from registers, and power from functional units separately or fail to address one of them at all. Hence, a solution that deals with all three simultaneously would be useful.

Bibliography

[1] "Report: Worldwide Handheld Shipments Decline in 2Q; Pocket PC to Surpass Palm PDAs in 2004", *MobileVillage*, September 12, 2002, http://www.mobilevillage.com/news/2002.09.12/PDAmarket.htm (October 9, 2002).

[2] H.S. Yun and J. Kim Yun, "Power-aware modulo scheduling for high-performance vliw processors," *Proc. of the ISPLED 2001*, pp. 40-45.

[3] Rabaey D. Singh and others, "Power conscious cad tools and methodologies: A perspective," *Proceedings of the IEEE 83,* April 4, 1995, pp. 570-594.

[4] R.S. Martin and Knight, "Optimizing power in asic behavioral synthesis," *IEEE Design & Test of Computers 13*, pp. 58-70.

[5] Semiconductor Industry Association (SIA), "The National Technology Roadmap for Semiconductors: Technology Needs," 1997, http://www.macs.ece.mcgill.ca/~roberts/ROBERTS/COURSES/SUPPORT/SIAroadmap97.pdf (October 12, 2002).

[6] Sedra and Smith, "*Microelectronic Ciruits, 4th Edition*," Oxford University Press, Oxford, New York, 1998.

[7] Sung-Mo Kang and Yusuf Leblebici, "*CMOS Digital Integrated Circuits Analysis and Design, Third Edition,*" McGraw-Hill, New York, 2003.

[8] J. Chang and M. Pedram, "Energy Minimization Using Multiple Supply Voltages," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 5, December 1997.

[9] "DSP: A What is Definition?" *Whatis.com*, August 11, 2002, http://whatis.techtarget.com/definition/ (December 1,2002).

[10] Robert A. Walker and Raul Camposano, "*A Survey of High-Level Synthesis Systems.*" Kluwer Academic Publishers, Boston, 1991.

[11] Raje Salil and M. Sarrafzadeh, "Scheduling with multiple voltages," *Integration the VLSI journal,* 1997, pp. 37-59.

[12] Wen-Tsong Shiue and Chaitali Chakrabarti, "Low-Power Scheduling with Resources Operating at Multiple Voltages," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 6, June 2000, pp. 536-543.

[13] Ali Manzak and Chaitali Chakrabarti, "A Low Power Scheduling Scheme with Resources Operating at Multiple Voltages," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, no. 1, February 2002.

[14] Jennifer Wong, Seapahn Megerian, and Miodrag Potkonjak, "Forward-Looking Objective Functions: Concept & Applications In High Level Synthesis," *DAC 2002*, June 10-14, 2002, New Orleans, Louisiana.

[15] Martin Gotschlich and Bernhard Wess, "Automatic Generation of Constrained Expression Trees for Global Optimized DSP Assembly Code," *Proceedings of the ICSPAT*, vol. 1, October 1996, pp. 732-736.

[16] Radoslaw Szymanek, "Introduction to Constraint Programming Lab Instruction," *Department of Computer Science, Lund University*, http://www.cs.lth.se/home/Krzysztof_Kuchcinski/CLP/info/instruction/instruction003.html, (January 19, 2003).

[17] Dan Nyborg, "The Fast Fourier Transform and its use in the Spectral Analysis of Digital Audio," *Concordia University*, April 22, 1998, http://physics.concordia.ca/~grob/298/fft2_paper.html, (February 19, 2003).

[18] Mohammed Ghanbari, "Video Coding: An Introduction to Standard Codecs (Iee Telecommunications Series, No 42)," *Inspec/IEE*, December, 1999.

[19] Itai M. Pines, "Force Directed Scheduling for Behavioral Synthesis," *UCLA*, October 22, 1998, http://www.janet.ucla.edu/~mbs/courses/ee298-2/1998f/lectures/SP1_itai.ppt, (Februrary 12, 2003).

[20] Wim F.J. Verhaegh and others, "Improved Force-Directed Scheduling in High-Throughput Digital Signal Processing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 14, no. 8, August 1995, pp. 945-960.

APPENDICES

Appendix A My Publications

Chee Lee and Wen-Tsong Shiue, "Consideration of Control System and Memory Contributions in Practical Resource-Constrained Scheduling for Low Power," accepted for presentation at 13th International Workshop on Power and Timing Modeling, Optimization and Simulation, Torino, Italy, September 10-12, 2003.

Chee Lee and Wen-Tsong Shiue, "Control System and Memory Contributions in Resource-Constrained Scheduling for Resources Operating at Multiple Voltages," accepted for presentation at the 7th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, FL, July 27-30, 2003.

Weetit Wanalertlak, Chee Lee, and Wen-Tsong Shiue, "Cool Hardware and Fast Software Solutions for Future Video Codec Applications," accepted for presentation at the 7th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, FL, July 27-30, 2003.

Weetit Wanalertlak, Chee Lee, and Wen-Tsong Shiue, "High Performance Low Power Video Compression Techniques", published in IEEE 45th Midwest Symposium on Circuits and Systems, Tulsa, Oklahoma, June 2002. (MWSCAS 2002)

Appendix B Biographical Sketch

Chee Lee received his B.S. (magna cum laude) from Oregon State University in June 2001 where he majored in Computer Engineering and minored in Computer Science. He was a Presidential Scholar and a member of the Multiple Engineering Cooperative Program (MECOP). During the spring and summer of 2000, he worked at IBM NUMA-Q in Beaverton, Oregon providing software maintenance and development for test engineering. Then in the summer and fall of 2001, he worked at Tektronix in Beaverton, Oregon developing an interface board for the hardware group. His research interests include VLSI architectures and algorithms and low power design. He currently has four publications (see Appendix A) and is working on more. In his spare time, he enjoys basketball and soccer and learning about other cultures. He is a member of the Tau Beta Pi, Eta Kappa Nu, and Golden Key Honor Societies as well as H.M.O.N.G and the Hmong Association of Oregon where he helps serve the community.