The Energy Complexity of Register Files *

V. Zyuban and P. Kogge

Computer Science & Eng. Department, University of Notre Dame, IN 46556, USA

Abstract

Register files (RF) represent a substantial portion of the energy budget in modern processors, and are growing rapidly with the trend towards wider instruction issue. The actual access energy costs depend greatly on the register file circuitry used. This paper compares various RF circuitry techniques for their energy efficiencies, as a function of architectural parameters such as the number of registers and the number of ports. The Port Priority Selection technique was found to be the most energy efficient. The dependence of register file access energy upon technology scaling is also studied. However, as this paper shows, it appears that none of these will be enough to prevent centralized register files from becoming the dominant power component of next-generation superscalar computers, and alternative methods for inter-instruction communication need to be developed. Split register file architecture is analyzed as a possible alternative.

Introduction

Current microprocessor design has a tendency towards wider instruction issue and increasingly complex out-of-order execution. This leads to growth of the on-chip hardware, and, consequently, an increase in dissipated power. In [8] the authors have described and analyzed those portions of a microarchitecture where complexity grows with increasing instruction-level parallelism. Among them are: register rename logic, wakeup logic, selection logic, data bypass logic, register files, caches and instruction fetch logic. These structures are usually centered on multiported memory macros whose storage size and the number of ports grow with increasing issue width.

The silicon area of a multiported memory, built using conventional approaches, grows quadratically in the number of ports [10]. Therefore, taking into account growth both in storage needs and the number of ports, we should expect that the power portion of multiported on-chip memories will grow rapidly in the future.

In this paper we concentrate on the power dissipation of an integer RF. The developed approach can also be applied to other CPU on-chip multiported memories. Our model will express the RF access energy in terms of the read and write port number, N_{read} and N_{write}, the number of registers, N_{reg}, and several other relatively simple system and technology parameters. It tries to find the lower bound of the RF power that can be approached by different implementations. Such a model is badly needed for architectural studies, where we are mostly interested in relative energy (power) estimates that would allow us to compare energy complexity of different architectures.

The organization of the paper is as follows: Sections 1 through 5 give an energy analysis of various RF circuitry techniques. Section 6 applies the developed RF energy model to the superscalar architecture, and Section 7 analyses a split register file architecture as a possible solution to the RF power growth problem. Section 8 summarizes the paper.

*This work was supported in part by the National Science Foundation under Grant No.MIP–95–03682.

1 Conventional approach

The conventional multiported memory cell for a RF typically uses two bit lines per write port and one bit line per read port, as well as one word line for every port to control the connection of the cell to the bit lines of the corresponding port [8], [10], [6]. Multiple word lines can go high at the same time in case of simultaneous access through several ports to the same cell. Therefore the cell must be capable of driving significant current which is proportional to the number of read ports. To protect the data stored in the cell during such multiple read accesses, an additional buffer is typically inserted between the cell flip flop and the read pass transistor further referred to as a decoupling buffer. Because of this decoupling buffer, the read bit line cannot serve as a write bit line, and thus the total of $N_{read} + 2N_{write}$ bit lines are needed.

1.1 Read access energy

We will consider four terms in the read access energy: the word line energy, $E_{wl,read}$, bit line energy, $E_{bl,read}$, sense amplifier energy, E_{SA}, and energy for driving control signals, $E_{control}$.

1.1.1. Word Line Energy. Assuming a word line swing of V_{dd}, we have for the word line energy $E_{wl,read} = V_{dd}^2 N_{bits}\left(C_{gate}W_{pass,r} + W_{cell}C_{metal}\right)$, where C_{gate} is gate capacitance per unit width, and $W_{pass,r}$ is the width of the cell read pass transistor. Here and in the following the value of $W_{pass,r}$ is chosen in such a way that the bit line signal sufficient for sensing V_{sense} can be developed in at least $\frac{T_{period}}{4}$ time, where T_{period} is a typical CPU clocking period for a given technology level. C_{metal} is the metal word line capacitance per unit length, and W_{cell} is the cell width. If the number of ports is large enough, then the cell sizes are determined by the bit line and word line crossing area, therefore, $W_{cell} = (2N_{write} + N_{read}) * W_{pitch}$.

1.1.2. Bit Line Energy. In estimating bit line energy we assume that the word line is pulsed for the minimum time required for reading data from a cell [4]. There is, however, a limitation on how short the word line activation pulse can be. We assume that for robustness reasons, the word line activation pulse cannot be made any shorter than $\frac{T_{period}}{4}$, where T_{period} is the CPU clocking period. If bit lines are so short that the signal V_{sense} can be developed in a shorter period, then weaker cell transistors should be used to avoid energy waste, or a word line swing control circuitry can be used, as in [5].

In addition, a safety margin must be provided to ensure that the word line pulse is long enough even under process corner conditions. We assume that a 30% margin is sufficient: $M_{margin} = 1.3$ which means that if the bit line signal $V_{bl} = V_{sense}$ is sufficient for reliable sensing under nominal conditions, then the word line should be activated for a longer interval, such that the bit line signal $V_{bl} = M_{margin}V_{sense}$ is developed under nominal conditions. Assuming that bit line loads are entirely cut off during

reading, and that the precharge transistors are the only current source that drives bit lines during the precharge phase, $E_{bl,read} = V_{dd} M_{margin} V_{sense} C_{bl,read} N_{bits}$. The bit line capacitance is estimated as $C_{bl,read} = N_{reg}\left(C_{metal} H_{cell} + C_{drain} W_{pass,r}\right)$, where H_{cell} is the cell height. If the number of ports is large enough, then $H_{cell} = (N_{write} + N_{read} + 2) * W_{pitch}$, with the constant '2' standing for the power and ground lines.

Energy can be reduced if the bit line precharge voltage is $V_{dd} - V_{TH}$ or higher, where V_{TH} is the threshold of the n-channel read pass transistor. In this case the bit line is discharged only if zero is read from the cell. If zero is read on the average in P_{zero} percent of all read accesses, then the bit line energy is reduced by the factor P_{zero}. We have measured that for the Sparc-V8 architecture, on integer programs, $65 - 75\%$ of all bits read from the RF were zeros (not counting bits read from register $R0$). This result means that if we store in the cells the complement values of data, and precharge bit lines to an appropriate level, we reduce the average energy of read accesses by a factor of three or even four. We will extrapolate this result to 64-bit architectures, and use $P_{zero} = 0.3$ in the following.

1.1.3. Bit Line Signal. To reduce the bit line swing sufficient for sensing V_{sense}, the bit line precharge voltage, $V_{precharge}$, should be as close as possible to the sense amplifier (SA) threshold. However, $V_{precharge}$ should be higher than $V_{threshold}$ by a sufficient margin to provide reliable operation in process corners and in the presence of noise. To reduce the necessary margin, the $V_{precharge}$ and $V_{threshold}$ must change in the same way as temperature and V_{dd} change, and with technology. We assume that with the use of a feedback circuitry built of CMOS transistors only, the biasing $V_{precharge} - V_{threshold}$ can be controlled within a range of $300mV$. This value depends on the range of technology deviations, temperature and V_{dd} changes, temperature gradients. It is not likely to scale down with the feature size or with V_{dd}.

Another component to the V_{sense} is noise on the bit lines. This scales down with V_{dd} reduction, and we will use for estimates the $0.1V_{dd}$ value for the sufficient noise margin. Thus, we will use for V_{sense} in single-ended sensing the estimate $V_{sense} = 2{*}300mV + 0.1V_{dd}$. For comparison, in the UltraSparc RF [10] $V_{dd} = 3.3V$, $V_{precharge} = 1.31V, V_{threshold} = 0.70V$ and $V_{sense} = 1.31V$. In the RF of iWarp [6] $V_{dd} = 5V$ and $V_{sense} = 1.5V$.

1.1.4. Energy of Sensing Circuitry. The typical sensing scheme is basically an inverter with the input connected to the bit line. Since the bit line voltage is close to the inverter threshold, there is a short-circuit current flowing through the inverter while the SA is activated which can be estimated, assuming minimum size transistors, as $I_{SA} = \frac{1}{2} I_{dsat}$. Here and in the following we assume that the value of V_{dd} scales according to the "High-Speed Scenario" [3], and the drain saturation current per unit width is $\frac{I_{dsat}}{W} = 0.5mA$, independent of the feature size [3].

In order to save energy, it is desirable to activate SA's for as short interval as possible. We assume, as for the word line, that the shortest SA activation pulse we can afford without sacrificing robustness is $\frac{T_{period}}{4}$. Then, $E_{SA,inv} = \frac{1}{8} V_{dd} T_{period} I_{dsat}$.

1.1.5. Control Signals. We estimate the energy dissipated driving SA and precharge control signals similarly to that of the word line: $E_{SA,ctrl} = V_{dd}^2 N_{bits}(C_{gate} W_{SA,ctrl} + W_{cell} C_{metal})$, $E_{precharge,ctrl} = V_{dd}^2 N_{bits}\left(\frac{C_{bl,read}}{40} + W_{cell} C_{metal}\right)$. It is assumed that the size of the precharge transistors is proportional to the bit line capacitance, $W_{precharge} = \frac{C_{bl,read}}{40 C_{gate}}$, in order for the precharge time to be independent of the bit line capacitance.

Figure 1 shows the main components to the read access energy versus the number of registers, and the number of ports. Here and in the following we assume $N_{bits} = 64$, $N_{read} = 2N_{write}$. For large register files the bit line energy is the dominant component; for small register files the sense amplifier energy dominates.

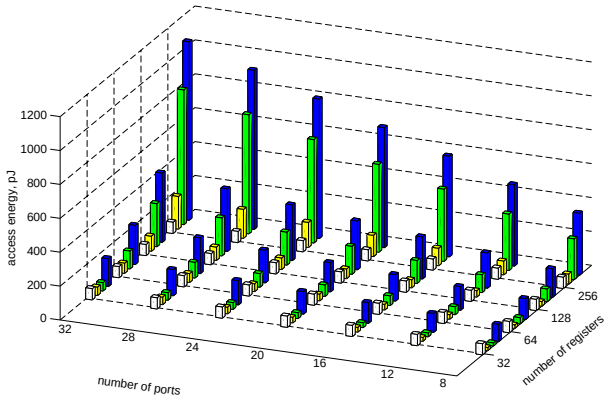

Figure 1: Components to the read access energy for the conventional RF architecture, the 0.5um technology, $V_{dd} = 3.3V$. From front to back bars represent 1.–SA energy; 2.–word line energy plus energy of control signals; 3.–bit line energy; 4.–sum of all components.

1.2 Write access energy

For write access energy we consider three terms: word line energy $E_{wl,write}$, bit line energy $E_{bl,write}$, and the energy dissipated driving control signals $E_{ctrl,write}$. The terms $E_{wl,write}$ and $E_{ctrl,write}$ are estimated similarly to those in the read access [11].

Writes to a cell are usually done by a full-swing differential signal to ensure fast write operation and robust noise margins [10], which results in energy dissipation of $C_{bl,write} V_{dd}^2$. The write energy can be reduced if after every write operation we equalize the write bit lines through an equalizing transistor. In this case, in the limit, we save half of the energy, so that $E_{bl,write} = \frac{1}{2} C_{bl,write} N_{bits} V_{dd}^2$.

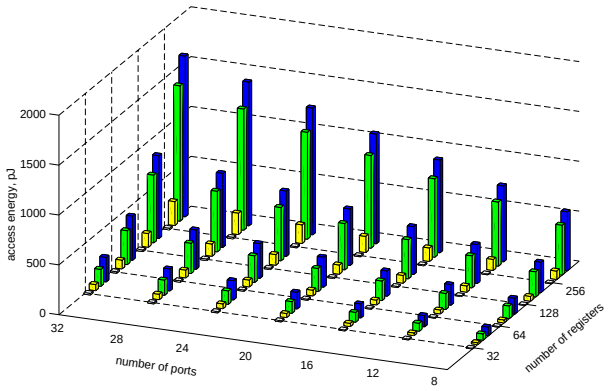

Figure 2: Components to the write access energy of the conventional RF architecture. From front to back bars represent 1.–word line energy; 2.–energy of write driver and precharge control signals; 3.–bit line energy; 4.–sum of all components.

Figure 2 shows the main components to the write access energy. The average access energy per instruction is a weighted sum of the read and write access energies. We have measured on integer programs that for the Sparc - V8 architecture there are on the average 0.95 RF read and 0.6 write accesses per instruction, therefore, $E_{average} = 0.95 E_{read} + 0.60 E_{write}$. The worst case energy per instruction is about twice as much as the average energy. To estimate the worst case access energy we assumed that every instruction issued reads two operand from and writes a result to the

RF, besides the values read from the RF are such that read bit lines are discharged every other read access.

1.3 Possible improvements

There are at least two modifications to the conventional RF architecture that could improve the energy efficiency. The first one reduces the average energy of the write access, taking advantage of the correlation between consecutive writes. If we choose not to precharge write bit lines after writes, the energy is dissipated only when the value being written by a particular port is different from the previous value written through the same port, so that $E_{bl,write} = \alpha C_{bl,write} V_{dd}^2$, where α is the percentage of the writes to individual cells in which the written value is different from the previous value written by the same port. We have measured that for the SPARC-V8 architecture α is in the range from 0.25 to 0.35 for most integer applications.

The second modification is to use a synchronous latch-type SA that eliminates the dc power component. For this kind of SA, when used in the single-ended read scheme, the reference voltage must be generated however. The power of the latch-type SA consists of the power due to the short-circuit current that flows through the SA while it is in a metastable state, and the power due to capacitive loading. In [11] we give a simple estimate for the SA energy, $E_{sa,latch} = \frac{1}{15} V_{dd} I_{dsat} T_{period}$.

Overall, we observe an almost 40% improvement in energy efficiency compared to the original version for small RF's due to the use of a more energy-efficient sensing scheme, and about 30% improvement for large RF's due to energy savings during write operation.

2 Current - Direction Sensing

In [1] the authors justify the use of the current sensing scheme to increase sensing speed in the case of heavily loaded bit lines. This idea of using a very low input impedance SA is also very attractive from the energy prospective.

Due to the small impedance at the sensing node, the signal current from the memory cell can be injected into the sense amplifier without the need for charging or discharging the bit line capacitance. As a consequence, the voltage change on the bit lines during cell access is extremely low, eliminating the source of most voltage noise coupling problems, and yielding low power dissipation during the sensing operation.

The word line energy, $E_{wl,read}$, and the control signal energy, $E_{control}$ are estimated similarly to those in the previous section. In [11] we derive a simple estimate for the SA energy, $E_{SA} = \frac{1}{4} V_{dd} T_{period} I_{dsat}$. The bit line energy can be estimated as $E_{bl,read} = I_{cell} V_{dd} T_{wl}$, where I_{cell} is the cell current during the read access, T_{wl} is the word line activation pulse length.

To minimize the bit line energy we need to minimize both I_{cell} and T_{wl}. The minimum value of the SA input current sufficient for reliable sensing, I_{sense} is determined by the sensitivity of the SA which, in turn, depends on mismatches in transistor parameters in the SA. We will use the value of $I_{sense} = 100uA$ in our estimates [11] [5].

As before, we require the sensing to be done in $T_{wl} = \frac{T_{period}}{4}$ time, therefore we need that the cell current be at least $I_{cell} = \frac{I_{sense}}{1 - exp(-\frac{T_{period}}{4(r_{bl}+r_{SA})C_{bl}})} M_{margin}$, where r_{bl} is the bit line resistance, and r_{SA} is the sense amplifier input resistance. The estimates for these values are given in [11]. We assume, as before, that the 30% safety margin should be provided to ensure that cell supplies sufficient current even in the process corner conditions. The restriction on the minimum cell current above makes it sure that sufficient current flows through the SA by the end of the word line activation period. Thus, we have for the bit line energy of the read access: $E_{bl,read} = I_{cell} V_{dd} \frac{T_{period}}{4} N_{bits}$.

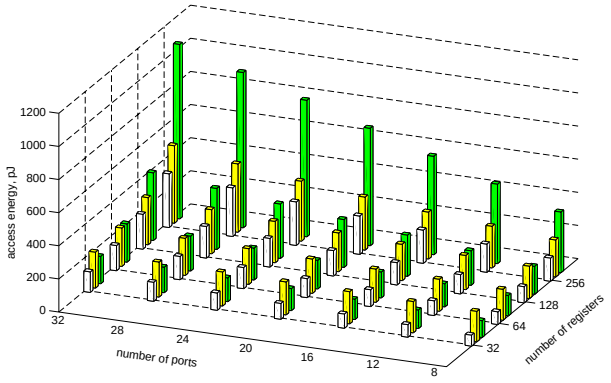

Figure 3: Comparative read access energy of the RF with the the double-ended sensing (1st bar); current-direction sensing (2nd bar), and the conventional RF architecture (3d bar). $V_{dd} = 3.3V$.

Figure 3 compares the average read access energy of the RF with the current direction sensing scheme (the 2nd bar) and that of the conventional RF (the 3d bar). For small register files, where the SA energy is the dominant component (Fig. 1) the use of the current-direction sensing technique results in increased read access energy, while for large RF, where the bit line energy dominates, the read access energy is significantly reduced.

3 Differential Sensing Scheme

The differential sensing scheme has the ability to sense small bit line swings, resulting in lower energy, and higher speed.

The derivation of the formulas for read and write access energy of a double-ended sensing scheme follows the one for the single-ended scheme in Section 1. The cell width in for the double-ended sensing is larger, however, $W_{cell} = (2N_{write} + 2N_{read}) * W_{pitch}$, resulting in a higher energy for all signals running across the RF, such as $E_{wl,read}$, $E_{wl,write}$, $E_{control}$. Also, additional pass transistors represent an extra load on read word lines, as compared to the single-ended scheme, though they do not need to be as large as in the single-ended scheme.

In estimating the energy dissipated in bit lines we make the same assumptions as for the single-ended scheme in Subsection 1.1. The bit line signal sufficient for reliable differential sensing, V_{sense}, is much less than that in the case of using the single-ended sensing scheme. There are three components to the V_{sense}: offset voltage due to transistor parameter mismatches, capacitive asymmetry and the coupling noise. To be specific, we assume that the cross-coupled latch-type SA is used, because it is both energy efficient and fast [4]. In [11] we show that the minimum sensing signal for this kind of SA can be estimated as $V_{sense} = 150mV$, of which a $60mV$ component is assumed to scale down with V_{dd}, and a $90mV$ is assumed not to scale.

We should notice that with the use of offset-compensating techniques or/and careful layout used for SRAM memories, the offset voltage can be significantly reduced. However, taking into account the huge number of SA's in register files, and area and/or power penalties of these techniques, we assume that they will not be used for RF's in the nearest future.

Figure 3 compares the average read access energy of the RF with the double-ended sensing scheme (1st bar), current-direction scheme (2nd bar), and that of the conventional RF (3d bar). The double-ended sensing yields improvement in energy efficiency for all RF sizes due to the reduced read bit line swing. The write access

energy is slightly more than that in the conventional RF architecture because of the increase in capacitance of all lines running across the RF. The primary disadvantage of the differential sensing approach is the area penalty which is approximately 50%. The increase in area results in higher energy dissipation of other components to the total CPU energy, not taken into account in the present model, such as clock distribution energy and energy of driving longer data busses.

4 Low Swing Write Technique

According to the two previous sections, the energy of the read access can be reduced by decreasing the swing on read bit lines. Then, the write cycle energy begins to determine the average access energy per instruction. We saw in Section 1 that in the conventional RF architecture the energy during the write operation is dissipated primarily in write bit lines. Consequently, write swing reduction is essential for reducing the RF access energy.

One way to reduce the write swing is to use the Driving Source Line (DSL) cell architecture, [7]. The basic idea is to connect the sources of the n-channel transistors in the cell to the source line, Fig.5. During reads the source line is driven low, and the cell operates as a conventional cell. During write operation the source line is either left floating, or driven high, depending on the bit line precharge voltage, so that a small swing on bit lines is sufficient to change the potential at the nodes inside the cell. At the end of the write cycle the source line is driven low, and the cell works as a latch-type SA, latching the new data.

The write access energy consists of the same components as before, plus the energy for driving the source lines which is estimated similarly to the word line energy. The energy dissipated in the cell during low-swing write operation is estimated to be the same as that of the latch-type SA. To estimate the bit line energy it is necessary to know the bit line swing sufficient for reliable writes to a memory cell, V_{write}. It is estimated similarly to V_{sense} for a latch-type SA in the previous section. Some of the terms to the V_{write} for a cell are larger, however, than the corresponding terms to the V_{sense} for a SA. In [11] we estimate the value V_{write} for the RF cells to be $250mV$, of which a $100mV$ component is assumed to scale down with V_{dd}, while the other $150mV$ component is assumed not to scale.

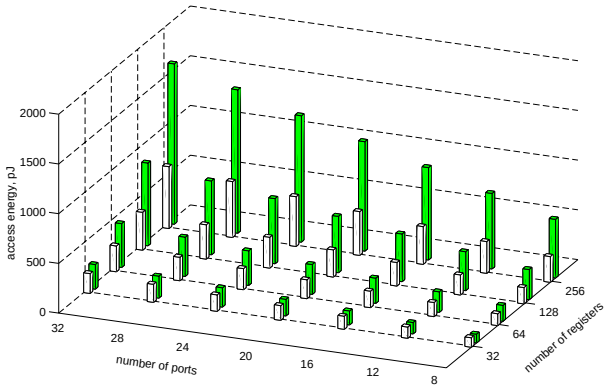

Figure 4: Write access energy in the RF with the low-swing write technique (1st bar), and the original architecture (2nd bar).

Figure 4 compares the write access energy in the RF with the low-swing write technique and that of the conventional RF. We observe an improvement in energy efficiency of the write access for all RF sizes. The low-swing write technique can be combined with various read sensing techniques, as described before. We expect that low-swing writes will become a common technique in multi-ported register files.

5 Port Priority Selection technique

The Port Priority Selection (PPS) technique [9] allows us to significantly reduce the register file area and, consequently power, by reducing the number of word lines and bit lines. The idea of the PPS technique is based on the observation that when several ports need data from the same location, we do not actually need to read multiple copies of data from the cell. One copy would be sufficient if we could distribute the data among all the ports that need it. Based on this observation, we can prohibit the simultaneous read access of more than one port to the same cell. If more than one port tries to read data from the same cell, a special priority mechanism chooses among these ports the one with the highest priority, and allows this port to access the cell. All the other ports get the data from the bit lines of the port with the highest priority. For efficient realization of the priority circuitry and the data steering mechanism we refer the reader to [9].

Once the cell no longer needs to be capable of driving more than one pair of bit lines at a time, the decoupling buffer between the cell flip flop and the read pass transistor which serves to protect the cell data in the conventional design is no longer needed. With this removed, the same bit lines can be used both for read and write operations, assuming they are separated in time.

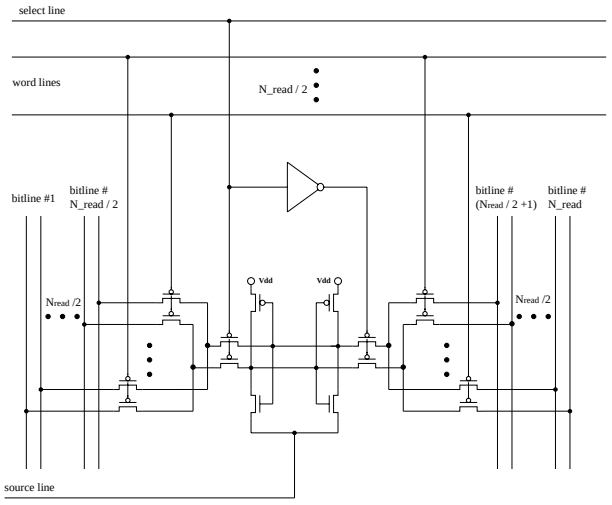

Figure 5: RF memory cell using the PPS technique with N_{read} read ports and N_{write} write ports $(N_{read} \geq N_{write})$.

The combination of differential reads (Section 3) and low swing writes (Section 4) appears particularly natural. The only difference between the read and write operations is that the source line is at the ground during read operation, and it goes high during the write operation (assuming V_{dd} precharge of the bit lines). The use of the same bit line pairs both for reads and writes allows us to reduce the number of bit lines from $2N_{read} + 2N_{write}$ to $2N_{read}$.

With the use of the PPS technique, the only information that really needs to be passed to the cell through word lines during read or write operation is the number of which of the bit line pairs the cell needs to be connected to. In the limit, $\log_2 N_{read}$ word lines would be sufficient to pass this information. In this case, however, this information would need to be fully decoded within the cell, which might not be area efficient and would cause extra delay in the access time. As a compromise, partial port number decoding within the cell allows us to reduce the number of bit lines and optimize the cell area.

In [9] the $\left(\frac{N_{read}}{2}+1\right) \rightarrow N_{read}$ partial decoding scheme is used, such that all bit line pairs are divided into two halves, and the signal on an additional select line (Fig.5) indicates to which half of the bit lines the cell is to be connected, so that a total of $\left(\frac{N_{read}}{2}+1\right)$ lines are needed. This scheme appears to be optimal for a wide range of the number of ports. Therefore, we will assume this scheme in the following estimates.

The energy estimation for the RF using the PPS technique is, basically, the combination of the results for the differential read and low-swing write schemes. The cell width and height are significantly reduced compared to Sections 3 and 4. Now $W_{cell} = 2N_{read}W_{pitch}$, and $H_{cell} = \left(\frac{N_{read}}{2}+1+2+1\right)W_{pitch}$, where the term $\frac{N_{read}}{2}$ stands for the number of one-hot word lines, '1' stands for the select line, '2' stands for power and ground lines, and an additional '1' stand for the source line. The port priority and data steering circuits consume additional energy which is estimated to be of the order of 10% of the RF array energy for a small number of ports. Moreover, the asymptotic growth of the energy of these units as the number of ports and registers in the RF grows is less than that of the RF array energy.

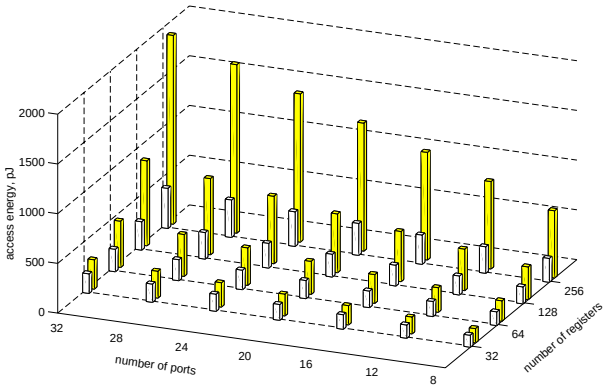

Figure 6: Average access energy per instruction of the RF using the PPS technique (1st bar) and the conventional RF architecture (2nd bar).

Fig. 6 compares the average RF access energy per instruction of the PPS RF architecture with the conventional RF architecture (Section 1). The PPS RF architecture results in significant improvements in energy efficiency for large register files. In addition, for large number of ports, the cell area is reduced almost three times, resulting in reduction in energy of other components not taken into account in this model, such as clock distribution and data routing energy. Thus the PPS register file architecture combined with differential reads and low-swing writes is a very worthy candidate for future low power register files.

6 Application to Superscalar Architecture

In this section we apply the energy model to superscalar architectures to analyze the dependence of the register file energy upon the processor issue width (IW). We assume for the number of ports: $N_{read} = 2IW$, and $N_{write} = IW$. To estimate the number of registers we use the results in [2], where it was found that for a four-issue and eight-issue machines the performance saturates around 80 and 128 registers, respectively. Based on this data, and assuming that 40 registers is sufficient for a single issue machine, we extrapolate the dependence linearly to two- and 16-issue machines.

The average RF access energy per instruction versus the microprocessor issue width is plotted in Fig. 7a, both for the PPS and conventional RF architectures. The equations listed for each case are approximate curve fits for the region $IW = 4$ to $IW = 16$. In

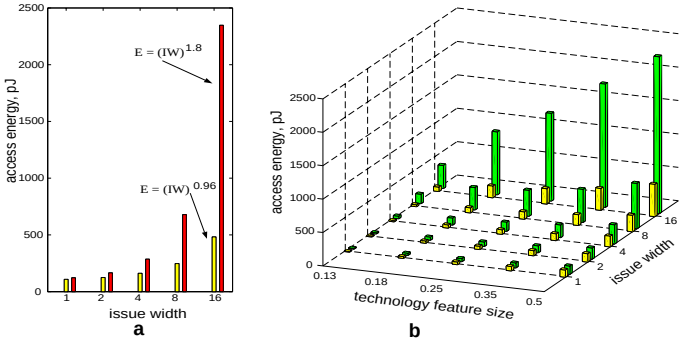

Figure 7: Average access energy per instruction for the PPS (1st bar) and the conventional RF architecture (2nd bar) **a** - versus the issue width, 0.5μ feature size, $V_{dd} = 3.3V$; **b** - versus issue width and technology feature size.

this range the PPS case grows at about square root of the conventional RF architecture case.

The chart in Fig. 7a does not take into account technology scaling, whereas machines with higher issue width are usually built on more advanced processes. To take into account technology scaling we plotted in Fig. 7b the average access energy as a function of issue width and technology feature size. High-speed scenario [3] scaling was assumed, and all other scaling assumptions are as stated in the preceding sections. Fig. 7b shows that if every new machine with higher IW is built using a more advanced process or, in other words, if we are moving along the diagonal going through point $(\lambda = 0.5\mu, IW = 1)$ to $(\lambda = 0.13, IW = 16)$ in the coordinate plane, then the average access energy per instruction can be moderated for the RF using the PPS technique.

If we are interested in dissipated power, however, we need to take into account the increase in the clocking rate for smaller technology feature sizes. Also, we must be able to estimate the maximum sustained power of the RF, dissipated if the maximum number of instructions, determined by the IW of the processor, are issued every clock cycle. The result is that the maximum power grows rapidly with increasing IW, even for the PPS RF architecture built using technology that tracks the growth in IW.

It must also be stressed that all the analysis discussed above assumed the very aggressive energy management techniques described in this paper, including pulse word line activation technique for reducing the bit line swing to the minimum, pulse activation of the sensing circuitry, fully cutting off precharge during reading and writing, taking advantage of the statistics of the data stored in RF memory cells, minimum transistor sizing wherever possible, use of equalizing transistors to save bit line energy during precharge, also the short-circuit currents were assumed to be negligible. In real wold CPUs the desire for speed will often not permit some of these techniques, meaning that real register file powers are even higher. The conclusion is that none of the known circuit techniques solves the problem of rapid RF power growth for the machines with increasing IW. Neither does aggressive technology scaling solve the problem for high-speed microprocessors. This result should motivate the development of inter-instruction communication mechanisms alternative to the centralized register file.

7 Split Register File Architectures

In a recent work [12] we studied split register file architectures as an alternative to the conventional centralized register file architecture. Our approach to the register file decentralization is based on a hypothesis that there exist certain groups of instructions in the dynamic instruction sequence such that the inter-instruction com-

munication is likely to be mostly local within each group. If this is the case, then we can implement a CPU as a collection of processing unit clusters and provide each cluster with a local physical register file which will be small, fast and low-power. At run-time, instruction despatch logic tries to steer every instructions to the cluster where instructions producing its register source operands were executed, meanwhile exploiting as much of the available ILP as possible. A desired result of such partitioning would be that instructions access local register files most of the time. Additional paths are provided for inter-cluster traffic.

We verified the above hypothesis by considering an optimal partitioning of instructions in the dynamic code sequence into groups such that inter-instruction communication across group boundaries is minimized, while executing instructions from different groups in parallel would exploit most of the ILP available in a program. A partitioning was constructed using an algorithm [12] that approximately solves the problem of optimal partitioning of a program graph into n subgraphs (n equals the number of clusters) such that the number of inter-subgraph edges (inter-cluster RF accesses) is constrained to be below a specified limit k. The sets of instructions resulting from this partitioning are dispatched to different cluster, and an execution-driven simulator measures the achievable IPC for the specified n and k. Some of the results for a few integer benchmarks are given in Table 1. Entries in the Table columns show the measured performance for specified constraints on the rate of inter-cluster register file accesses, divided by the maximum achievable performance for the given number of clusters n (when traffic constraint k is disregarded).

Performance ratio, $\frac{\text{IPC}}{\text{IPC}_{\max}}$				
n: number of clusters	k: constraint on inter-cluster RF accesses per instruction			
	0.20	**0.15**	**0.10**	**0.05**
2	0.99	0.98	0.96	0.82
4	0.96	0.91	0.77	0.60
8	0.91	0.74	0.60	0.45

Table 1: Performance and inter-cluster register file traffic on a per-instruction basis, for optimal partitioning of instructions.

The results show that even with the existing compiler which performs conventional performance optimizations only, it is possible to partition instruction into a reasonably large number of groups such that inter-instruction communication tends to be mostly local within each group, without a huge impact on IPC. The discovered properties open up an opportunity for architectural solutions that could keep the RF from becoming a bottleneck in the power budget *without* using complicated circuit techniques. The actual amount of energy savings will depend on a particular implementation, and its estimation is a target of our current work.

There are additional energy benefits of implementing a CPU as a collection of processing unit clusters. First, the energy efficiency of bypassing can be improved if we implement full bypassing within each cluster only. Second, by steering instructions that access the same registers to the same cluster we increase the probability that operand values of these instructions are correlated, and thereby reduce the average number of bit transitions at the inputs of functional units. Indeed, we observed a 9% to 10% reduction in the data switching activity at the inputs of functional units with the corresponding reduction in the data switching activity at the outputs of the functional unit of from 12% to 14%. The reduction in the switching activities at the outputs of local register files was even more significant, up to 17%, which could be used to further reduce the access energy to local register files.

8 Conclusions

In this paper we have developed energy models for multiported registers files with a variety of architectural variations. The port priority selection technique was found to be the most energy efficient, and seems to provide significant energy savings in comparison to traditional approaches, especially for large register files.

Given the critical role played by centralized register files in modern superscalar computer architectures, we used the developed model to express the register file power as a function of processor issue width. Even assuming aggressive technology scaling to track the growth in register file requirements, the resulting power growth is huge, and may begin to swamp the power budgets for future microprocessors. This leads to the inescapable conclusion that the use of a centralized register file as an inter-instruction communication mechanism is going to become prohibitively expensive. Alternative techniques involving more than just circuit tricks are going to be necessary. Split register file architecture is studied as a possible alternative, and is shown to hold certain promise. More detailed analysis of this approach is a target of our future work.

References

[1] T. Blalock and R. Jaeger, "A High-Speed Clamped Bit-Line Current-Mode Sense Amplifier", *IEEE Journal of Solid-State Circuits*, Vol.26, No.4, April 1991.

[2] K. Farkas, N. Jouppi, P. Chow, "Register File Design Considerations in Dynamically Scheduled Processors." Technical Report 95/10, Digital Equipment Corporation Western Research Lab, November 1995.

[3] C. Hu, "Future CMOS Scaling and Reliability." *Proceedings of the IEEE*, Vol.81, No.5, May 1993.

[4] K. Itoh, K. Sasaki and Y. Nakagome, "Trends in Low-Power RAM Circuit Technologies." In: *Proceedings of the IEEE*, Vol. 83, No.4, April 1995.

[5] M. Izumikawa and M. Yamashina, "A Current Direction Sense Technique for Multiport SRAM's." *IEEE Journal of Solid-State Circuits*, Vol.31, No.4, April 1996.

[6] R. Jolly, "A 9-ns, 1.4-Gigabyte/s, 17-Ported CMOS Register File." *IEEE Journal of Solid-State Circuits*, Vol.26, No.10, October 1991.

[7] H. Mizuno and T. Nagano, "Driving Source-Line Cell Architecture for Sub-1-V High-Speed Low-Power Applications." *IEEE Journal of Solid-State Circuits*, Vol.31, No.4, April 1996.

[8] S. Palacharla, N. Jouppi, J. Smith, "Complexity-Effective Superscalar Processor." In: *Proceedings of the 24th Annual International Symposium on Computer Architecture*. pp. 206–218, June 1997.

[9] U.S. Patent No. 5,657,291, issued Aug. 12, 1997 to A. Podlesny, G. Kristovsky, A. Malshin, "Multiport Register File Memory Cell Configuration for Read Operation."

[10] M. Tremblay, B. Joy and K. Shin, "A Three Dimensional Register File For Superscalar Processors." In: *Proceedings of the 28th Annual Hawaii International Conference on System Sciences*. pp. 191–201, January 1995.

[11] V. Zyuban, P. Kogge, "The Energy Complexity of Register Files", *Notre Dame CSE Technical Report No.97-20*, December 1997.

[12] V. Zyuban, P. Kogge, "Split Register File Architectures for Inherently Lower Power Microprocessors," In: *Power-Driven Microarchitecture Workshop, in conjunction with ISCA'98*, June 1998. available at http://www.cs.colorado.edu/ $\sim$ grunwald/LowPowerWorkshop