AN ANALYTICAL AREA, ACCESS AND CYCLE TIME MODEL FOR ON-CHIP MULTIPORTED MEMORIES *

Gabriel P. Silva[+#]; Eliseu M. Chaves[#]; Victor M. Goulart[@]; Vladimir C. Alves[@]

NCE[+] - COPPE Sistemas[#] - COPPE Elétrica[@]
Universidade Federal do Rio de Janeiro
P.O.Box 68511 Cep 21945-970 - Rio de Janeiro - Brasil
e-mail:gabriel@cos.ufrj.br

Abstract

This article presents an analytical model for the area, access and cycle times of on-chip multiported memories. This model can be used to predict the behavior of register files and register renaming mapping tables and other similar structures on superscalar processors. The inputs to the model are the number of words, output width, number of read ports, and number of write ports. Software implementing the model is available on request through e-mail.

1 Introduction

The design of a superscalar processor depends on investigating several alternatives. An adequated design decision requires the evaluation of the costs of each alternative: access and cycle times, chip area and power requirements need to be estimated. One practical solution is to employ analytical models that predict costs based on some architectural parameters.

In this article, we are reviewing some access time models [1] [2] and chip area models [3] for on-chip memory devices. We present some improvements to these models by considering the use of distinct memory cells circuits and multiple ports.

Multiported memories are needed in superscalar processors to increase bandwidth enough to provide operands and store results for several instructions being executed concurrently. Register files and register renaming mapping tables are some examples of on-chip memories.

The goal of this paper is to present relatively simple equations that predict the access/cycle time and area cost of on-chip multiported memories. The delay of each component was estimated by decomposing each component into several equivalent RC circuits, and using simple RC equations to estimate the delay of each stage. This cost is evaluated as a function of several device parameters, process parameters, and array structure parameters. The models shown here are enhanced in the following sense:

- The memory can have an arbitrary number of read and write ports;
- It considers the use of both single ended and differential sense amplifiers;
- The use of different designs for the multiported memory cell are considered;
- An area estimation is provided.

*This work was partially granted by CNPq PROTEM III program (MULFLUX project, n° 680096/95.7).

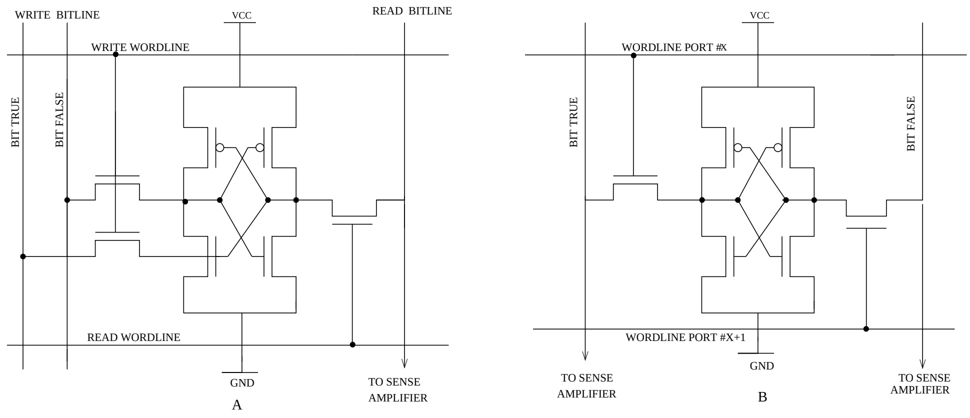

Figure 1: Memory Cells

When designing a real memory, many design techniques could be employed to optimize the delay of certain stages. However, the relative access times and area costs between different configurations should be more accurate than the absolute values, and this is often more important for optimization studies.

We intend to use these models to help the design of the register file, and the register renaming mapping table, which are very important in the final processor cycle performance. These multiported memories will be used in the design of a superscalar processor, named Superflux, that is under development at COPPE/UFRJ.

The multiported memory model explained in this report was implemented, and the software is available by e-mailing the authors.

2 Memory Description

2.1 Memory Structures

Figure 1 shows two static memory cells. There are many possible organizations for a static memory cell, but we will concentrate our studies on those two. The final memory cell layout involves a lot of design techniques to maximize speed and minimize noise. Some examples are: cell diffusion contacts are shared between adjacent cells to reduce bitline load; bitline metals are minimized and spaced to reduce line-to-line coupling; ground lines running along two adjacent rows of cells are also used. The layout depends strongly on the number of metal layers available to the designer.

Beyond the memory cell, there are many other structures that compose a multiported memory: decoders, wordline drivers, precharge drivers, write buffers, multiplexors, sense amplifiers and output drivers. A differential or a single ended sense amplifier could be used for reading/writing the memory cell. Differential sense amplifiers usually provide better speed and noise immunity [4], but it requires a reference voltage level, which could be difficult to provide and costs extra die area. We found also some designs where a non-differential writing scheme [5] is used, but the most reliable and ordinary scheme is the differential writing.

All these options make difficult to have precise area and time estimations of the final memory layout, but some effort could be done to achieve this goal. Initially, let us consider the basic functionality of a static memory, using the circuitry showed in Fig. 1.a. This memory cell has separate bitlines for reading and writing, and uses differential write buffers and single ended sense amplifiers. The decoder first decodes the address and selects the appropriate row by driving one read wordline in the data array. Each array contains as many read wordlines as there are rows in the array, but only one read wordline in each array can go high at a time. Each memory cell along the selected row is associated with one read bitline; and each read bitline is initially precharged high. When a read wordline goes high, the value stored in the memory cell determines if the read bitline goes high or low.

If the memory cell has bitlines shared for read and write ports, it could have a circuit schematic as showed in Fig. 1.b. It can use either one single ended sense amplifier for each bitline or one differential sense amplifier for both bitlines. In superscalar processors, it is usual to find a 2:1 relation in the number of read and write ports in the register file, thus a cell design with single ended amplifiers fits better to this kind of application.

If we split the array conveniently, it is possible for one sense amplifier to be shared among several bitlines. In this case, a multiplexor is inserted before the sense amps; the select lines of the multiplexor are driven by the decoder. The number of read bitlines that share a sense amplifier depends on the layout parameters, that will be described in the next section.

2.2 Memory Organization Parameters

The following memory parameters are used as inputs to the model: number of words, number of read ports, number of write ports, memory size in words, size of the word in bits.

In the cache organization discussed by Wada [2], all memory cells in a line share a common wordline. Clearly, such an organization could result in an inadequate aspect ratio, causing either the bitlines or wordlines to be very long and slow. This could result in a longer-than-necessary access time. To alleviate this problem, Wada describes how the array can be broken horizontally and vertically and defines two parameters: N_{dwl} and N_{dbl} that indicates to what extent the array has been divided. The parameter N_{dwl} indicates how many times the array has been split with vertical cut lines (creating more, but shorter, wordlines), while N_{dbl} indicates how many times the array has been split with horizontal cut lines (causing shorter bitlines). The total number of subarrays is $N_{dwl} \times N_{dbl}$.

Wilton and Jouppi [1] introduced another organization parameter, N_{spd}. This parameter indicates how many cache/memory lines are mapped to a single wordline, and allows the overall access time of the array to be changed without breaking it into smaller subarrays. In this case, two or more memory lines could share the same wordline, being selected later by a column multiplexor. This could reduce the size of the first stage of the address decoder, improving the memory access time.

The optimum values of N_{dwl}, N_{dbl}, and Nspd depend on the memory size and number of ports. Notice that increasing these parameters is not free in terms of area. Increasing N_{dbl} or Nspd beyond one increases the number of column multiplexors required, and increasing the number of ports increase the number of sense amplifiers. Also, increasing N_{dwl} means more wordline drivers are required. Because we are modeling a memory, that is equivalent to a direct mapped cache, there is no need for a multiplexor to select the appropriate sense amplifier output to return to the processor. Finally, increasing N_{dbl} or Nspd increases the size of the multiplexor driver.

Using these organizational parameters, each subarray contains $8 \times B \times A \times N_{spd}/N_{dwl}$ columns and $C/B \times A \times N_{dbl} \times N_{spd}$ rows. This information will be used throughout this article.

3 Model Derivation

Precisamos reescrever esta secao pois esta igual ao orginal de Wilton e Jouppi.

The model uses RC approximations described in [6] for estimating the delay of each stage. The stage delay in our model depends on the slope of its inputs, as needed to accurately estimate submicron technologies delays.

3.1 Estimating Resistances

To use these RC approximations, the model estimates the full-on resistance of a transistor. This is the resistance seen between drain and source of a transistor, if the gate voltage is constant and the gate is fully conducting. This resistance can also be used for pass transistors that (as far as the critical path is concerned) are fully conducting.

It is assumed that the equivalent resistance of a conducting transistor is inversely proportional to the transistor width (only minimum-length transistors were used). Thus, equivalent resistance = R/W, where R is a constant (different for NMOS and PMOS transistors) and W is the transistor width.

3.2 Estimating Gate Capacitances

The RC approximations used also require an estimation of a transistor's gate and drain capacitances. The gate capacitance of a transistor consists of two parts: the capacitance of the gate itself, and the capacitance of the polysilicon line going into the gate. The value of Cgate depends on whether the transistor is being used as a pass transistor, or as a pull-up or pull-down transistor in a static gate [1].

If Leff is the effective length of the transistor, Lpoly is the length of the poly line going into the gate, Cgate is the capacitance of the gate per unit area, and Cpolywire is the poly line capacitance per unit area, then a transistor of width W has a gate capacitance of:

$gatecap(W) = W \times Leff \times Cgate + Lpoly \times Leff \times Cpolywire$

The same formula holds for both NMOS and PMOS transistors. The value of Cgate depends on whether the transistor is being used as a pass transistor, or as a pull-up or pull-down transistor in a static gate. Thus, two values of Cgate are required.

3.3 Drain Capacitances

The drain capacitance is composed of both an area and perimeter component. Two equations are used in the model: one if the width is less than $10\mu m$, and other if the width is larger than this value.

$draincap(W) = 3Leff \times W \times Cdiffarea + (6Leff + W) \times Cdiffside + W \times Cdiffgate$

where Cdiffarea, Cdiffside, and Cdiffgate are process dependent parameters (there are two values for each of these: one for NMOS and one for PMOS transistors). Cdiffgate is the sum of the junction capacitance due to the diffusion and the oxide capacitance due to the gate/source or gate/drain overlap. If the width is larger than $10\mu m$, it is assumed that the transistor is folded, reducing the drain capacitance to:

$draincap(W) = 3Leff \times W/2 \times Cdiffarea + 6Leff \times Cdiffside + W \times Cdiffgate$

Now, consider two transistors (with widths less than $10\mu m$) connected in series, with only a single Leff × W wide region acting as both the source of the first transistor and the drain of the second. If the first transistor is on, and the second transistor is off, the capacitance seen looking into the drain of the first is:

$draincap(W) = 4Leff \times W \times Cdiffarea + (8Leff + W) \times Cdiffside + 3W \times Cdiffgate$

If the transistors are wider than $10\mu m$, the capacitance seen looking into the drain of the inner transistor assuming it is on but the outer transistor is off is:

$draincap(W) = 5Leff \times W/2 \times Cdiffarea + 10Leff \times Cdiffside + 3W \times Cdiffgate$

3.4 Other parasitic effects

Parasitic resistances and capacitances of the bitlines, wordlines, predecode lines, and various other signals within the memory are also modeled. These resistances and capacitances are fixed values per unit length; the capacitance includes an expected value for the area and sidewall capacitances to the substrate and other layers.

The delay of a gate is defined as the time between the input reaching the switching voltage (threshold voltage) of the gate, and the output reaching the threshold voltage of the following gate. Detailed information is provided in [1] about how the delay of a stage, which depends on the slope of its inputs, is calculated.

4 Model Components

The access and cycle times were derived by estimating delays due to the following components:

1. decoder;
2. wordlines;
3. bitlines and memory cell (for both types of memory cell);
4. sense amplifiers (differential and single);
5. output drivers (data output and valid signal output).

The delay of each of these components is estimated separately and the results combined to estimate the access and cycle time of the entire multiported memory. These estimations will be affected by the number of ports of the memory and the type of the sense amplifier used.

The use of multiple ports impacts mostly in the design of the memory cell. There is the need to use additional wordlines, bitlines and pass transistors. For each new metal line (bitline or wordline) added, the cell's dimensions will increase of a certain amount. This amount depends on many factors, as the number of metal layers available, the relative size of the metal lines compared to the transistors, and the designers skill.

Rever estas estimativas de aumento de area com os novos layouts ..

Mulder [3] estimates a linear increase around 25% in each dimension for each new metal line added. But the technologies he analyzed were between 1.0 μm and 2.0 μm, with 2 metal layers. Our measurements indicate a linear increase of 33% for a 0.5 μm technology, with 2 metal layers.

In the work of Wilton and Jouppi [1], the memory cell used was the one presented in Fig 1-b, with differential sense amplifiers. Considering the equations used in this work, the following parameters are directly affected by this memory cell area increase:

- $New\_Cwordmetal = Cwordmetal(1 + PORTFACTOR * Cbitl)$ (wordline capacitance of a metal wire per bit width);
- $New\_Rwordmetal = Rwordmetal(1 + PORTFACTOR * Cbitl)$ (wordline resistance of a metal line per bit width);
- $New\_Cbitmetal = Cbitmetal(1 + PORTFACTOR * Cwordl)$ (bitline capacitance of a metal wire per bit height)
- $New\_Rbitmetal = Rbitmetal(1+PORTFACTOR*Cwordl)$ (bitline resistance of a metal line per bit height)
- $New\_Bitwidth = Bitwidth(1 + PORTFACTOR * Cwordl)$ (width of a memory cell)

where:

- PORTFACTOR: an empirical scaling factor due to the inclusion of a new metal line (see section 5);
- Cbitl: number of extra bitlines per memory cell;
- Cwordl: number of extra wordlines per memory cell.

In the following sections we will explain the changes introduced with the use of multiported memory cells.

4.1 Decoder

This model consists of a detailed transistor-level decoder that includes both parasitic capacitances and resistances. It is assumed that subarrays are placed in a two-dimensional array to minimize critical wiring parasitics.

The decoder in this model contains three stages. Each block in the first stage takes three address bits (in true and complement), and generates a 1-of-8 code, driving a precharged decoder bus. These 1-of-8 codes are combined using NOR gates in the second stage. The final stage is an inverter that drives each wordline driver. Separate decoder driver buffers for driving the 3-to-8 decoders of the data array are also modeled.

Estimating the wire lengths in the decoder requires knowledge of the memory tile layout.

As mentioned before, the memory is divided into $N_{dwl} \times N_{dbl}$ subarrays; each of these arrays is $8 \times B \times A \times N_{spd}/N_{dwl}$ cells wide. It is assumed that they are grouped in two-by-two blocks, with the 3-to-8 predecode NAND gates at the center of each block; Figure 4 shows one of these blocks. This reduces the length of the connection between the decoder driver and the predecode block to approximately one quarter of the total memory width, or $2 \times B \times A \times N_{dbl} \times N_{spd}$. The length of the connection between the predecode block and the NOR gate is then (on average) half of the subarray height, which is $C \times B \times A \times N_{dbl} \times N_{spd}$ cells. In large memories with many groups the bits in the memory are arranged so that all bits driving the same data output bus are in the same group, shortening the data bus.

Figure 4: Memory block tiling assumptions

Changing the number of memory ports will change the number of decoders needed, since an extra decoder is requird for each new port added. This will have an impact in the total area, but a negligible change in its delay.

4.2 Variable Size Wordline Driver

The size of the wordline driver in Wada's model [2] does not depend on the number of cells attached to the wordline; this severely overestimates the wordline delay of large arrays. The model used here is the same from Wilton and Jouppi and assumes a variable-sized wordline driver. Normally, a memory designer would choose a target wordline rise time, and adjust the driver size appropriately. Rather than assuming a constant rise time, we assume the desired rise time (to a 50% word line swing)is:

$desiredrisetime = krise \times ln(cols) \times 0.5$

$wherecols = 8 \times B \times A \times N_{spd}/N_{dwl}$

and *krise* is a constant that depends on the implementation technology. To obtain the transistor size that would give this rise time, it is necessary to work backwards, using an equivalent RC circuit to find the required driver resistance, and then finding the transistor width that would give this resistance. More details are given in [1].

For the wordline delay, the new values of *Cwordmetal* and *Rwordline* will affect the wordline capacitance to calculate the desired time:

$Cline = (gatecappass(Wa, 0.0) + CWORDMETAL) * cols$

And affect also the final wordline delay, as expressed bellow:

$Cline = (gatecappass(Wa, (BITWIDTH - 2 * Wa)/2.0) + CWORDMETAL) * cols + draincap(nsize, NCH, 1) + draincap(psize, PCH, 1); / * Wordlinecapacitance * /$

$Rline = RWORDLINE * cols/2$

4.3 Bitlines and Memory Cells

Wada's model does not apply to memories with column multiplexing. Wilton and Jouppi model allow column multiplexing using NMOS pass transistors between several pairs of bitlines and a shared sense amplifier. In this model, the degree of column multiplexing (number of pairs of bitlines per sense amplifier) is $N_{spd} \times N_{dbl}$.The bitlines are precharged with two NMOS diodes to less than Vdd since the differential sense amplifier performs poorly with a common-mode voltage of Vdd. We are considering the same voltage level even for the single ended sense amplifier.

None of those models considers the use of multiported memory cells. Depending on the memory cell design, the addition of one port will, at least, imply in the need of one extra wordline and one or two extra bitlines.

If a memory cell as in Fig. 1-a is used, for each new **port** that is added to the memory device, one wordline and two bitlines are added to the memory cell. Also, one wordline and one bitline are added to the memory cell for each new **read** port. The formulas that express this sentence are:

$number\_of\_wordlines = read\_ports + write\_ports$

$number\_of\_bitlines = read\_ports + 2 * write\_ports$

If the memory cell is as the one in Fig. 1-b, for each new write port that is added to memory, the number of wordlines and bitlines increases of one and two, respectively. If a new read port is added, the situation is a bit more complex, since we can share wordlines and bitlines, and single ended or differential sense amplifiers could be used. If differential sense amplifiers are used, for each new read port added, one wordline and two bitlines will be added, ONLY if the number of read ports is greater than the number of write ports. If it is less or equal than, the wordlines and bitlines of the new port can be shared with the ones of an existing write port. See the equations bellow:

$number\_of\_wordlines = MAX(read\_ports, write\_ports)$

$number\_of\_bitlines = MAX(2 * read\_ports, 2 * write\_ports)$

If single ended sense amplifiers are used, and a new port is added to the memory, two situations arise. First, if the number of read ports is greater than twice the number of write ports, the number of wordlines and bitlines will increase of one unit. Or, if it is equal or less than, the wordlines and bitlines of the new read port can be shared with existing wordlines and bitlines of a write port. The exact formula can be seen bellow:

$number\_of\_bitlines = MAX(read\_ports, 2 * write\_ports)$;

$number\_of\_wordlines = MAX(read\_ports, ceil((read\_ports + 2 * write\_ports)/2))$

Finally, the number of additional wordlines and bitlines is calculated as follows:

$Cwordl = number\_of\_wordlines - 1$;

$Cbitl = number\_of\_bitlines - 2$;

The addition of either a new wordline or a bitline has an impact on the memory layout and in the final memory performance. As each new (metal) line is added, the memory array dimensions increase, increasing its corresponding capacitance and resistance. This will affects directly wordline and bitline delays.

Also, some transistor dimensions in the memory cell need to be altered to support the new pass transistors added for each read/write port. Particularly, the pull-up and pull-down transistors of the memory cell are increased in their sizes, in order to avoid a destructive read of the memory cell. The critical sizing ratio in the cell is between the parallel combination of pass transistors and the n-channel pull-down of the static memory cell. A minimum ratio of 2 is a safe target [4].

The capacitance seen by the wordline driver of each memory cell, does not change with the use of multiple ports, since there is also a shared contact between adjacent cells, and the pass gate is of the same size, but both bitline resistance and capacitance are altered.

The final capacitance seen by the bitlines depends on the type of sense amplifier. For the single ended amplifier, there is an increase in the capacitance, because there is an extra pass transistor. The bitline resistance increases because the memory cell has a correspondent increase in its dimension.

4.3.1 Adapting the Formulas

The pieces of code that require modifications are shown bellow:

Cbitrow, the capacitance of the memory cell, does not change with multiple ports, since there is also a shared contact between adjacent cells, and Wa is of the same size. But both bitline resistance and capacitance are altered:

$Cline = rows * (Cbitrow + CBITMETAL) + 2 * draincap(Wequ, PCH, 1)$

The final capacitance seen the bitlines depends on the type of sense amplifier. For the differential amplifier the formula is:

$Ccolmux = 2 * gatecap(WsenseQ1to4, 10.0)$

For the single ended amplifier, there is an increase in the capacitance, because there is an extra pass transistor, as showed in the next item.:

$Ccolmux = 2 * gatecap(Wsenseinv, 10.0) + draincap(Wsensepass, NCH, 1)$

The bitline resistance increases because the memory cell has a correspondent increase in its dimension:

$Rlineb = RBITMETAL * rows/2.0$

4.4 Sense Amplifiers and Output Drivers

There are two basic sensing schemes of the memory cell: differential and single ended. A differential sensing scheme usually provides better speed and noise immunity [4], but there is the need to generate a reference level, which could be difficult under certain circumstances and costs extra die area. The single ended scheme needs around 80% more time to sense the memory cell [4], but is simpler and smaller. Some circuits [5] could be employed to enhance its noise immunity, at the expense of some area. The delay is approximated by a constant, which is obtained through spice simulations for both types of sense amplifiers.

The output driver for a multiported memory is simpler than the one used for a cache [1]. The output driver delay also changes because the memory cell changes its dimension with the number of ports.

Figure ?: Overview of data bus output driver (FALTA)

The output driver delay also changes because the basic memory cell changes its dimension with the number of ports:

$Ceq = (draincap(Woutn, NCH, 1) + draincap(Woutp, PCH, 1)) * ((8 * B * A)/BITOUT) + CWORDMETAL * (8 * B * A * N_{spd} * (vstack)) + Cout$

$Rwire = RWORDMETAL * (8 * B * A * N_{spd} * (vstack))/2$

4.5 Total Access and Cycle Time

Esta secao esta identica ao trabalho de Wilton e Joupi tambem.

The access time can be derived from the following formula:

$Taccess = Tdecoder + Twordline + Tbitline + Tsense + Toutdrive$

The model first attempts to find the array organization parameters that resulted in the lowest access time via exhaustive search for each memory parameter, N$_{spd}$, N$_{dwl}$ and N$_{dbl}$. So, the corresponding area is determined and stored to be printed later with the cycle time results.

The difference between the access and cycle time of a memory varies widely depending on the circuit techniques used. We have chosen to model a conventional structure with the cycle time equal to the access time plus the precharge.

There are three elements in our assumed memory organization that need to be precharged: the decoders, the bitlines, and the comparator. The precharge times for these elements are somewhat arbitrary, since the precharging transistors can be scaled in proportion to the loads they are driving. We have assumed that the time for the wordline to fall and bitline to rise in the data array is the dominant part of the precharge delay. Assuming properly ratioed transistors in the wordline drivers, the wordline fall time is approximately the same as the wordline rise time. It is assumed that the bitline precharging transistors are scaled such that a constant (over all memory organizations) bitline charge time is obtained.

This constant will, of course, be technology dependent. In the model, we assume that this constant is equal to four inverter delays (each with a fanout of four). Thus, the cycle time of the cache can be written as:

$Tmemory = Taccess + Twordlinedelay + 4 \times (inverterdelay)$

5 Area Estimation

The initial consideration starts with the memory cell area. This area varies widely depending on the number of ports. We considered a basic memory cell, with one wordline and two bitlines, with the dimensions *BitHeight* and *BitWidth*. As we increase the number of ports, depending on the memory cell type, the number of bitlines and wordlines will increase also. We are estimating an empirical factor PORTFACTOR increasing the size of each dimension of the memory cell for each additional wordline/bitline added. Using the values of $number\_of\_bitlines$ and $number\_of\_wordlines$ previously calculated, we have:

$cell\_area = (1 + PORTFACTOR * (number\_of\_bitlines - 2)) * BitWidth * (1+$
$PORTFACTOR * (number\_of\_wordlines - 1)) * BitHeight$

We are estimating PORTFACTOR as 33% for the cell layouts we have investigated, using a 0.5 μm CMOS technology, with 2 metal layers. The next step includes calculating the total data array area, that can be simple expressed by:

$data\_area = cell\_area * nwords * bitout$

Where *nwords* equals the number of words in the memory array and *bitout* is the output width in bits.

We need also to include the area of the decoders, wordline drivers, write buffers and sense amplifiers. The data array organization has direct impact over the wordline driver. The number of decoders is proportional to the number of data subarrays, and their size is proportional to the size of the whole array. The number of sense amplifiers depends on the sensing scheme and on the number of bitlines. When using single ended sense amplifiers, the sense amplifier area is reduced by a factor of two, when compared to the case where differential sense amplifier are used. Depending on the memory organization parameters, we also need to add the area due to the column multiplexor,just a pass gate and respective wiring for each column.

The write buffer and precharge circuitry areas are dominated by the transistor size, rather than wiring and component spacing.So we can approximate that area by the following formula:

$wrbuff\_size = 4 * Wbitpreequ * Leff * ROUTEFACTOR2$ (Two inverters)

$prechg\_size = 7 * Wbitpreequ * Leff * ROUTEFACTOR2$

$prechg\_area = prechg\_size * (nbitlines/2) * cols + (wrbuff\_size * nwports * cols)$

For this situation and all bellow we are estimating ROUTEFACTOR2 as 1.5.

The wordline driver is as larger as wider is the memory subarray it is driving. Thus, its size varies with the number of subarrays and is recalculated at each iteration, for each new value of N_{dbl}, N_{dwl} and N_{spd}:

$wordrv\_size = (((psize+nsize)*Leff)+((W_{decinvp}+W_{decinvn})*Leff))*ROUTEFACTOR2$

$wordrv\_area = wordrv\_size * nwordlines * rows$

The decoder area varies with the number of subarrays and the number of rows of each subarray.We can express this formula as:

$wdecdrv\_size = ((Wdecdrivep + Wdecdriven) * Leff) * ROUTEFACTOR2$

$wdec3to8\_size = ((Wdec3to8n + Wdec3to8p) * Leff) * ROUTEFACTOR2$

$wdecnor\_size = ((WdecNORn + WdecNORp) * Leff) * ROUTEFACTOR2$

There are as many drivers as the number of address bits plus two, since we need true and inverted address lines:

$wdecdrv\_area = wdecdrv\_size * logtwo((double)nsets) * 2$

The variable numstack here replicates the formula from Wilton and Jouppi:

$numstack = ceil((1.0/3.0) * logtwo((double)((double)nsets/(double)(N_{dbl} * N_{spd}))))$

$if(numstack == 0)numstack = 1$

$if(numstack > 5)numstack = 5$

Each NAND in the second stage of the decoder has three inputs and there are 8 NANDs per block, and there are a total of numstack blocks in each subarray. There are $N_{dwl} \times N_{dbl}$ subarrays, so we can estimate the area of this decoder stage as:

$wdec3\_to8\_area = 8 * 3 * wdec3to8\_size * numstack * N_{dwl} * N_{dbl}$

In the last stage of the decoder, there is one NOR driver per wordline in every subarray. Each of these NOR has numstack inputs:

$wdec\_nor\_area = numstack * wdecnor\_size * nsets * nwordlines$

We also need to consider the output driver area, one driver for each output bit of the memory array:

$woutnor\_size = (Woutdrvnorn + Woutdrvnorp) * Leff) * ROUTEFACTOR2$

$woutdrv\_size = (Woutdrivern + Woutdriverp) * Leff) * ROUTEFACTOR2$

$wout\_area = (woutnor\_size * 2.0 + woutdrv\_size) * bitout$

The total area of the memory device is the sum of all these stages, as shown bellow:

$total\_area = wdec\_nor\_area + wdec3to8\_area + wdecdrv\_area + wout\_area +$

$wordrv\_area + prechg\_area + data\_area + sense\_area;$

This model is an approximation, and its most important characteristic is that it considers the impact of multiple ports in the memory cell layout, and it allows quick and accurate comparisons between different implementation alternatives.

6 Result Analysis

The time and area models presented are based on the fact that the dimensions of the memory cell transistors will not change as your design scales up. Designs that are faster can be achieved increasing memory cell transistor sizes and/or wordline drivers, but this will have an area penalty in the final design, that can lead to a prohibitive size as the number of registers and/or ports increase.

This section shows some results of time and area for multiported memory configurations that can be used in register files of superscalar processors. In this case, the number of read ports equals the twice the number of write ports. In all cases analyzed, we are considering 32-bit word memory.

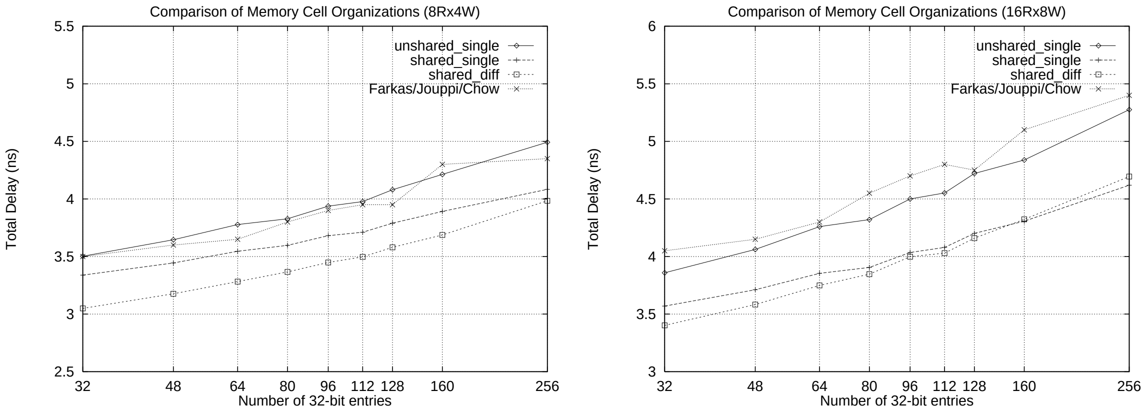

Figure 2: Cycle Time

6.1 Cycle Time Analysis

The graphics plotted in Fig. 2 show the total cycle time of the memory versus number of 32-bit words for two configurations of read/write ports: 8 read ports and 4 write ports; and 16 read ports and 8 write ports. All the configurations have a 2:1 relation between read and write ports, as it usually happens in superscalar processors.

The graphics in Fig. 2 were obtained from the simulation of a memory with 3 types of memory cell: one as the memory cell in Fig. 1-a, with unshared lines (bitlines and wordlines) and with single ended sense amplifiers; the other with a memory cell as in Fig. 1-b with shared lines and differential sense amplifiers; and finally a memory cell as in Fig. 1-b with shared lines and single ended sense amplifiers.

Farkas et al. [7] used the same port configurations in their model analysis, and the memory cell used by Farkas is identical to that presented in Fig 1-a. Our plotting shows that our model is very similar to theirs, with minimal delay difference, using the same memory cell organization (unshared_single).

Using the memory cell of Fig. 1-b the results are significantly improved. The memory cell using differential sense amplifier (shared_diff) has a better performance due to its faster sensing scheme. But, although the single ended amplifier is slower, the number of bitlines and wordlines per memory cell required is smaller, and the result with the shared_single memory cell is practically identical to the obtained with the shared_diff cell.

6.2 Area Analysis

Area estimations for the memory organizations analyzed in the previous section can be found in Fig. 3. The graphics plot the total memory area versus number of 32 bit words, for two memory port configurations.

The graphics show that doubling the number of words is less significant in terms of area than doubling the number of ports. We found that the unshared_single and the shared_diff memory cells are the most area consumer. The shared_single memory cell presents the best time/area relation among all three memory types and will be used in the Superflux register file design.

Mulder presented an area model and checked it against some designs using old technologies. This does not apply to the model presented here, which models submicron technology. We checked our model with the design [4], and our estimations perform very well. We predicted a total area of 1.2 mm^2, and they claimed a total area of 1.4 mm^2, that included some dedicated registers.

Most of the designs present unique characteristics that makes difficult to predict their behavior. We present an open model, so the designer can modify it according to his needs. Our experiments show that the model is accurate concerning comparisons made within the same technology parameters.

7 Conclusions

In this paper, we have presented an analytical model for both the area and cycle time of a multiported register file. The computational complexity is considerably lower than an electrical simulation. Although previous models have been proposed for cache memories, our model predicts the behavior of a multiported memory, that has briefly mentioned in [7], but with no details. Our time model has the same level of detail as the work from Wilton and Jouppi, which includes: non-step stage input slopes; rectangular stacking of memory subarrays; a transistor-level decoder model; column-multiplexed bitlines; modern array organization parameters and load-dependent transistor sizes for wordline drivers. It also produces cycle times as well as access times. This makes the model much closer to the real memory behavior.

Our model improves previous models, since it takes into account three basic cell memory designs, involving two types of sense amplifiers. We also included an area model that can be used to make comparisons among the various implementation alternatives. This provides enough information to the designer to make a decision considering not only the cycle times, but also the relative area costs involved.

Our next step is to make a better validation of this model, building spice models and some memory layouts of multiported memories. Considering that the model is based on well validated and accurate model, we believe we will not have meaningful changes from the model presented here.

References

[1] S. J. Wilton and N. P. Jouppi. An access and cycle time model for on-chip cache. Technical Report

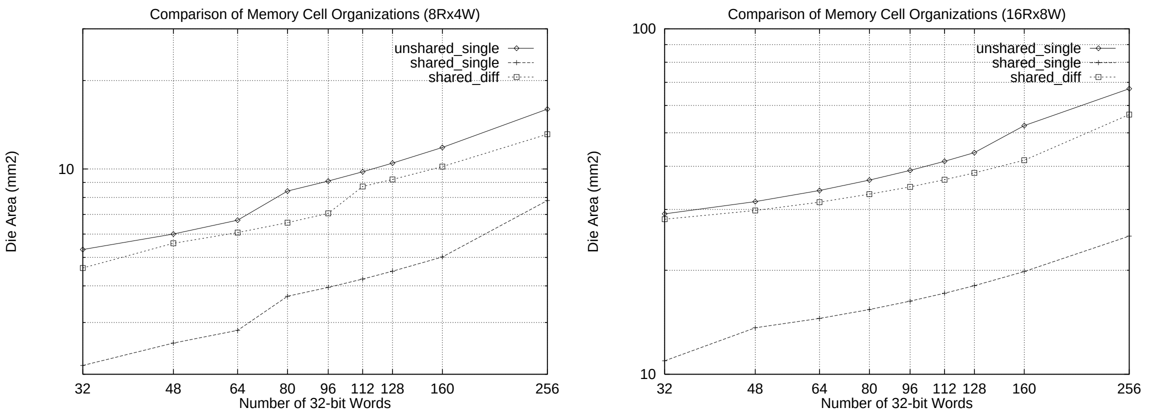

Figure 3: Memory Area

93/5, Digital Equipment Corporation Western Research Lab, 1994.

[2] T. Wada; S. Rajan and S. A. Przybylski. An analytical access time model for on-chip cache memories. *IEEE Journal of Solid-State Circuits*, 27(8):1147–156, Aug. 1992.

[3] J. M. Mulder ; N. T. Quach and M. J. Flynn. An area model for on-chip memories and its application. *IEEE Journal of Solid-State Circuits*, 26(2):98–106, Feb. 1991.

[4] R. D. Jolly. A 9-ns, 1.4 gbytes/s, 17-ported cmos register file. *IEEE Journal of Solid State Circuits*, 26(10), Nov 1991.

[5] L. A. Lev et alli. A 64-bit microprocessor with multimedia support. *Journal of Solid State Circuits*, 1995.

[6] S. J. Wilton and N. P. Jouppi. Timing models for mos circuits. Technical report, Integrated Circuits Laboratory, Stanford University, 1983.

[7] K. I. Farkas; N. P. Jouppi and P.Chow. Register file design considerations in dynamically scheduled processor. Technical Report 95/10, Digital Equipment Corporation Western Research Lab, 1995.

8 Appendix: Circuit and Technology Parameters

The transistor sizes and threshold voltages used in the circuits in this report are given in Table I-1 (all transistor lengths are $0.8\mu m$).

Table 1: Transistor sizes and threshold voltages

Table 2: $0.8\mu m$ CMOS process parameters

```
.model nt nmos ( level=3 + vto=0.77 tox=1.65e-8 uo=570 gamma=0.80 +
vmax=2.7e5 theta=0.404 eta=0.04 kappa=1.2 + phi=0.90 nsub=8.8e16 nfs=4e11
xj=0.2u + cj=2e-4 mj=0.389 cjsw=4.00e-10 mjsw=0.26 + pb=0.80 cgso=2.1e-10
cgdo=2.1e-10 delta=0.0 + ld=0.0001u rsh=0.5 ) *
.model pt pmos ( level=3 +vto=-0.87 tox=1.65e-8 uo=145 gamma=0.73 +
vmax=0.00 theta=0.233 eta=0.028 kappa=0.04 + phi=0.90 nsub=9.0e16 nfs=4e11
xj=0.2u + cj=5e-4 mj=0.420 cjsw=4.00e-10 mjsw=0.31 + pb=0.80 cgso=2.7e-10
cgdo=2.7e-10 delta=0.0 + ld=0.0001u rsh=0.5 )
```

Figure 15: Generic $0.8\mu m$ CMOS Spice parameters [3]

Stage	**Symbol**	**Value**
Decoder Driver	Wdecdrivep	$100\mu m$
	Wdecdriven	$50\mu m$
	vthdecdrive	0.438
Decoder NAND	Wdec3to8p	$60\mu m$
	Wdec3to8n	$90\mu m$
	vthdec3to8	0.561
Decoder NOR	Wdecnorp	$12\mu m$
	Wdecnorn	$2.4\mu m$
	vthdecnor	(one input) 0.503 (two inputs) 0.452 (three inputs) 0.417 (four inputs) 0.390
Decoder inverter	Wdecinvp	$10\mu m$
	Wdecinvn	$5\mu m$
	vthdecinv	0.456
Wordline driver	Wworddrivep	varies
	Wworddriven	varies
	vthworddrive	0.456
	krise	0.4ns
Memory Cell (Fig. 1)	Wa	$1\mu m$
	Wb	$3\mu m$
	Wd	$4\mu m$
	vthwordline	0.456
	BitWidth	$8.0\mu m$
	BitHeight	$16.0\mu m$
Bitlines	Wbitpreequ	$80\mu m$
	Wbitmuxn	$10\mu m$
	Vbitpre	3.3 volts
	Vbitsense	0.1 volts
	vt	1.09 volts
Sense Amp (Fig. 6-7)	Q1-Q4	$4\mu m$
	Q5-Q6	$8\mu m$
	Q7-Q10	$8\mu m$
	Q11-Q12	$16\mu m$
	Q13-Q14	$8\mu m$
	Q15	$16\mu m$
	tsense-data	0.58ns
	tfall sense-data	0.70ns
Output Driver NOR	Woutdrvnorp	$40\mu m$
	Woutdrvnorn	$6\mu m$
	vthoutdrvnor	0.431
Output Driver (final)	Woutdriverp	$80\mu m$
	Woutdrivern	$48\mu m$
	vthoutdriver	0.425
	Cout	0.5 pF

Table 1: Table I-1

Parameter	**Value**
Cbitmetal	4.4 fF/bit
Cgate	1.95 fF/μm2
Cgatepass	1.45 fF/μm2
Cndiffarea	0.137 fF/μm2
Cndiffside	0.275 fF/μm
Cndiffgate	0.401 fF/μm
Cpdiffarea	0.343 fF/μm2
Cpdiffside	0.275 fF/μm
Cpdiffgate	0.476 fF/μm
Cpolywire	0.25 fF/μm
Cwordmetal	1.8 fF/bit
Leff	0.8 μm
Rbitmetal	0.320 W/bit
Rn-switching	25800 W*μm
Rn-on	9723 W*μm
Rp-switching	61200 W*μm
Rp-on	22400 W*μm
Rwordmetal	0.080 W/bit
Vdd	5 volts

Table 2: Table Caption