An Embedded DRAM for CMOS ASICs

John Poulton
Department of Computer Science
University of North Carolina at Chapel Hill

Abstract

The growing gap between on-chip gates and off-chip I/O bandwidth argues for ever larger amounts of on-chip memory. Emerging portable consumer technology, such as digital cameras, will also require more memory than can easily be supported on logic-oriented ASIC processes. Most ASIC memory systems are P-load SRAM, but this circuit technology is neither dense nor power efficient. This paper describes development of a DRAM, compatible with a standard CMOS ASIC process, that provides memory density at least 4x improved over P-load SRAM in the same layout rules. It runs at speeds comparable to logic in the same process and uses circuitry that is reasonably simple and portable. The design employs Vdd-precharge bit lines, half-capacitance, full-voltage dummy cells, and a simple complementary sense amp. DRAM is organized as a number of small pages, allowing simple circuit design and low-power operation at modest expense in area overhead. The paper also described a power-conserving low-voltage-swing bus design that interfaces multiple pages to full-voltage-swing circuitry. Circuit and layout details are provided, along with experimental results for a 100MHz 786K-bit embedded DRAM in a 0.5μ process.

1. Introduction

Digital system designers, faced with a large and rapidly growing gap between available chip I/O bandwidth and demand for bandwidth, attempt to find clever system partitioning schemes to reduce demand. In a design environment in which millions of devices can be economically integrated onto a chip, but only a few hundred I/O pins can be supported, efficient system partitioning requires ever more on-chip storage.

ASIC designs now commonly require large amounts of on-chip memory, usually implemented as static random-access memory (SRAM). SRAM with PFET loads can be supported on any CMOS process but requires about 1,000 λ^2 of area per bit. Some vendors offer special processes, adapted from commercial SRAM manufacture, that include polysilicon resistor loads; resistor-load SRAM cells can be made as small as a few hundred λ^2. Dynamic memory (DRAM) circuits, while potentially much more compact than SRAMs, have fallen out of favor with ASIC designers. We speculate that this is partly because of the perceived complexity of DRAM circuit design and partly because of the unfavorable scaling of FET leakage currents in sub-micron CMOS that makes dynamic circuitry more problematic.

Few computer systems requires as much memory bandwidth per bit as hardware accelerators for interactive 3D graphics. Current high-end graphics hardware must support bandwidth of order 10 Gbytes/second into relatively small amounts of total storage, perhaps a few 10s of MBytes. Graphics systems built with commercial RAM chips therefore feature many-way partitioning and considerable replication of data across partitions. Our research for the past 15 years has focussed on ways to remove the memory bandwidth bottleneck by combining graphics processors with on-chip RAM. Large improvements can be realized simply by removing the column decoder of conventional RAM designs, thereby liberating the huge bandwidth from many-way parallel access inherent in rectangular memory organizations. We have built and fielded three generations of experimental systems to examine the validity of this idea (references [1-3]). Within the past couple of years, this idea has become mainstream, particularly in cost-sensitive applications such as graphics accelerators for PCs.

An efficient implementation of our latest system, *PixelFlow*, required on-chip memory with much higher density than could be achieved with conventional SRAM. To satisfy this requirement we developed the 1-transistor embedded DRAM that is the subject of this paper. Designed in scalable geometric rules, the DRAM is about four times denser than P-load SRAM in the same rules, dense enough to permit about 1Mb of RAM on a 10mm square chip in 0.5μ CMOS. It is reasonably well optimized for low power consumption, runs at the processors' speed (100MHz), and is fairly simple and portable to other applications.

Section 2 of the paper briefly describes the application environment in which the DRAM design was used. Section 3 describes the memory circuit design and operation. Section 4 takes up the power conservation and interface bus design issues that consumed most of the design time for the DRAM. Section 5 describes additional details of layout, circuit design and simulation of the memory itself. Section 6 outlines experimental results, and Section 7 presents conclusions, briefly reviews previous and current work in this area, and acknowledges many sources of help.

2. Application

The embedded DRAM serves as a register file for an array of 256 8-bit processors in a graphics 'enhanced memory chip' (EMC) [3]; each processor 'owns' 384 bytes of memory. The memory layout is bit-sliced, so the organization is 2,048 bits (columns) by 384 words (rows). In the word dimension, the memory is composed of 'pages', each a block of 32 words. Each page is a self-contained memory system with bit cells arrayed along a pair of bit lines, a local sense amp and precharger, and an interface to a (differential) data bus that delivers data between the (12) pages and the processor. The column dimension has no decoder; on every cycle of chip operation, data is read from or written to 2,048 bits of memory. A block diagram of a bit slice through the DRAM is shown in Figure 1; the column dimension is horizontal in the figure.

The organization into many small pages is the central design feature of this DRAM, inspired by Don Speck's observation in the design of the MOSAIC DRAM [4] that short, low-capacitance bit lines allow large voltage differences that can be sensed with simple sense amps. Simple sense amps, in turn, allow compact realizations of small memory modules. This advantageous design 'spiral' also leads to considerable power savings. If data is fetched from one of an array of small modules, the modules that are not accessed remain quiescent and burn no power.

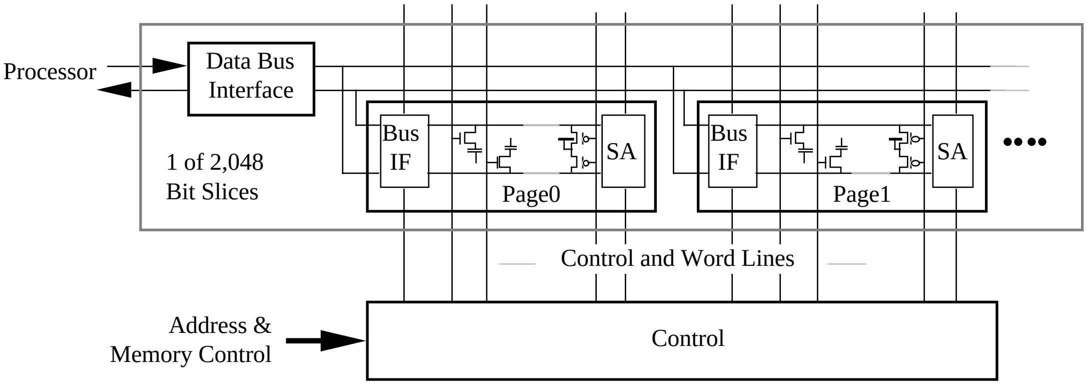

Figure 1. Block diagram of one bit-slice through the DRAM.

Another way to think about the power savings in this organization is as follows: Most of the power in a DRAM is consumed in charging the bit-line capacitance. This large capacitance is mainly composed of the parasitic drain capacitance of the many bit access transistors attached to the line. By breaking the bit-lines into many small pieces, one per page, the power required for a cycle of operation is divided by the number of pages, with some additional overhead incurred in the data bus that connects the pages. Since the data bus itself is lightly loaded (one connection per page), this is a winning strategy.

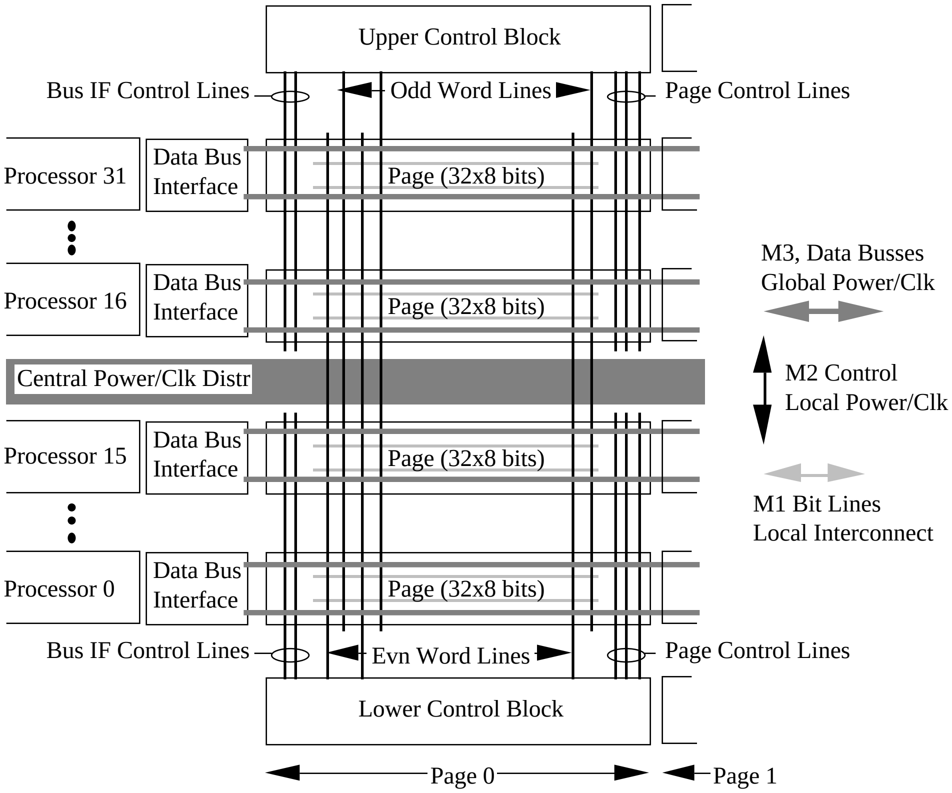

Figure 2. Physical arrangement and assignment of wiring layers within a panel of memory.

The memory and processors are physically arranged in eight ‘panels’, each with 32 processors and 256 x 384 bits of DRAM. A sketch of this layout in Figure 2 shows how each of three layers of metal interconnect are used in the layout.

The horizontal (bits along bit lines) pitch of the memory cell layout is quite tight, adding considerable entertainment value to the physical design of the memory control block. To help with this problem, the control block is split into two parts, each responsible for half the bits in a page. Odd word lines are generated in the top block, even in the bottom, and they are driven across the entire array of memory cells in each panel. Control lines for the page data bus interface and for the page sense amps, prechargers, and dummy cells are more heavily loaded than word lines. Copies of these controls are generated in both top and bottom control blocks and driven half-way across the array, stopping just short of the central power/clock bus.

It may be worth noting that in the application of this DRAM, eight of the memory pages are general purpose, while four are special purpose, augmented with a second data port. These four pages serve as I/O buffers that support communication between the processors and off-chip communication networks. As will be clear from the next section, it is easy to add a second data port to this design without greatly affecting performance or layout.

3. Memory circuits

The internal circuitry for a bit slice of a page of memory is shown in Figure 3.

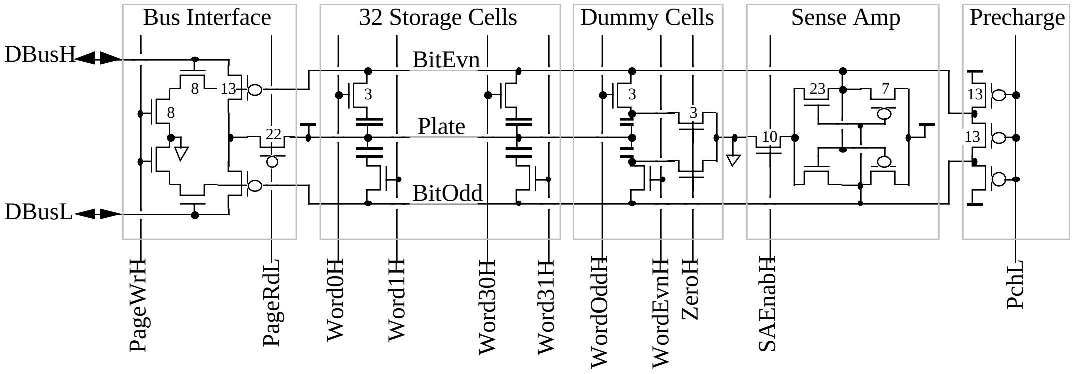

Figure 3. Internal circuitry of a memory page (numbers near devices indicate width in λ; all are minimum length).

The DRAM is a folded bit line design with full Vdd precharge. Differential bit-line voltage is generated by means of half-capacitance, full-voltage dummy cells. Storage and dummy cells use MOS storage capacitors, described in more detail in Section 4; these capacitors are referenced to a common terminal, labelled “Plate”, connected to Vdd. The sense amp is a simple cross-coupled differential CMOS pair enabled by a clocked current source. A triplet of precharge PFETs ensures that the bit-lines are pulled very nearly to Vdd, and (more importantly) that they are at the same voltage, after a precharge.

At the beginning of each cycle of operation of the memory, the bit-lines are precharged HI. Quiescent pages (pages not being referenced on a given cycle) have PchL asserted continuously. As noted in [4], this has the beneficial side effect of adding all of the quiescent bit-line capacitance to the Vdd-to-GND bypass capacitance and helps considerably in controlling noise on the supply rails. During a memory operation, bit-lines operate monotonically LO (they either remain HI or go LO), and this leads to the particularly simple implementation of the bus interface shown in the figure, by virtue of the signaling convention on the data bus DBus{H,L}: it is precharged LO, and one of the DBus pair goes monotonically HI during a cycle of operation. DBus is a small-signal-swing differential bus, whose details are described in Section 4.

The timing of the sequence of events in memory reads and writes is shown in Figure 4.

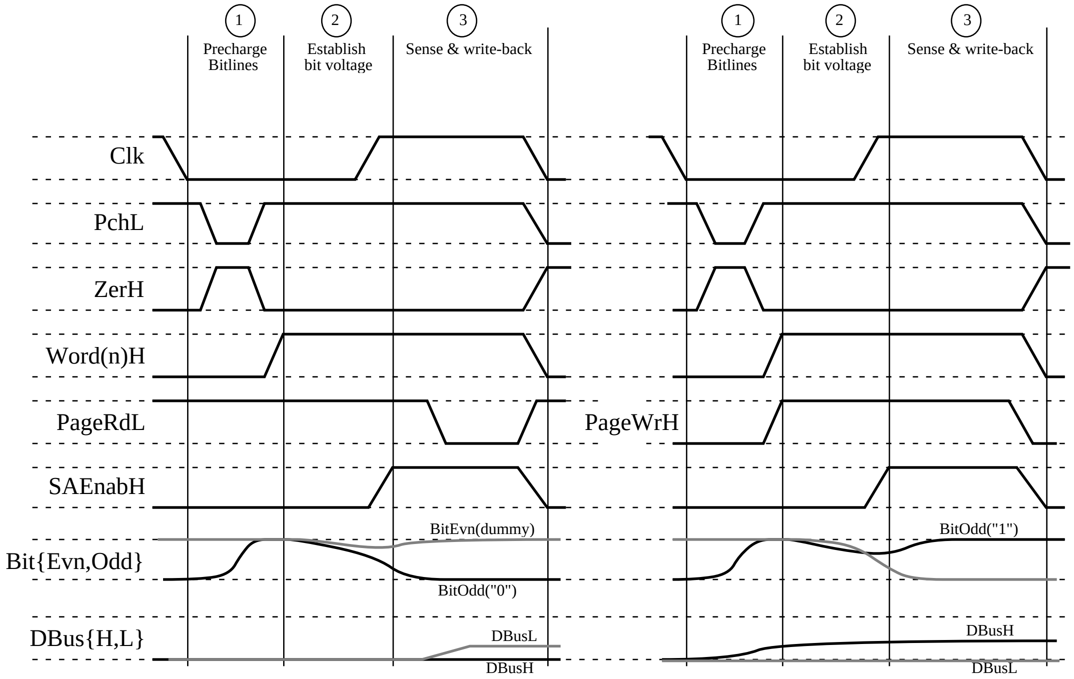

Figure 4. Timing of events during read and write operations.

Memory operations are coordinated with the chip's clock. For convenience in register construction, most chip events are timed relative to the falling edge of the clock. Each of the operations proceeds in three phases. For a read operation,

1. Bit lines are precharged to Vdd during the first quarter of the clock cycle by pulsing PchL low. The control ZerH is asserted during precharge, discharging the half-capacitance dummy storage cells. The DBus{H,L} pair is precharged low by the processor's data bus interface.

2. De-assertion of PchL leads to assertion of one of the word lines, which connects one of the storage cells to its bit line. In this example, we assume that an odd word is enabled onto BitOdd, and that the storage cell on that word-line is storing a "0". At the same time, the control WordEvnH (not shown, but identical in timing to the normal word line) is asserted, connecting the even dummy cell to BitEvn. Both Bit{Evn,Odd} head low, since both storage and dummy cells have "0's" stored, but BitOdd is pulled down more strongly because it is charge-sharing with a capacitance twice as big as that in the dummy cell. At the end of this phase of operation, charge sharing between storage cells and bit lines is essentially complete, and there is now a voltage difference between the bit lines. The data in the storage cell has been destroyed, since the bit cell capacitance has been charged high by the much larger capacitance on the bit line. When reading a cell that is storing a "1", the dummy cell pulls its bit line downward as in this example, but the other

bit line remains at Vdd. Since the dummy cell is half the capacitance of a storage cell, the bit-line voltage difference is roughly the same for reading “1’s” and “0’s”.

3. Just after the rising edge of Clk, the sense amp is turned on by asserting SAEnabH. The gates of the two source-coupled NFETs are both well above the NFET threshold and at slightly different voltages, so the NFET pair amplifies the voltage difference and begins to pull BitOdd down rapidly. Once BitOdd’s voltage is below the PFET threshold, the cross-coupled PFET pair rapidly pulls BitEvn up toward Vdd. The word line is still asserted, so the low-going BitOdd restores, or ‘writes back’ the “0” originally stored in the bit cell. The rapidly low-going BitOdd turns on its PFET in the data bus interface and pulls DBusL high, since by now PageRdL is asserted. The duration of PageRdL is set by a timer adjusted so that the high-going DBus voltage is less than Vdd/2 under all conditions.

Writes proceed in a similar fashion:

1. Bit lines are precharged to Vdd. One of DBus{H,L} is driven to a weak high, roughly Vdd/2, by the processor’s data bus interface.

2. One of the word-lines is asserted. We again assume that a cell storing a “0” is connected to BitOdd. BitOdd heads low, just as in the read operation described above. Initially the dummy cell pulls BitEvn low, as in the read, but during a write, current flowing in the NFET stack in the page bus interface, enabled by the assertion of PageWrH, causes BitEvn to be driven low much faster, so that by the end of this phase, BitEvn is even lower than the low-going BitOdd. The voltage difference on the bit lines at the end of this phase is therefore inverted from its value during a read-”0” operation.

3. The sense amp is turned on, pulls BitOdd high, and overwrites the bit cell with a “1”.

There is a third type of operation, *refresh*, for the DRAM. In the PixelFlow EMC application, refresh is scheduled both *opportunistically* (a refresh operation happens on cycles when the processor array does not need to access memory) and *prejudicially* (when too much time has gone by since the last refresh, processor operation is halted for a cycle, and a refresh operation is inserted). A refresh operation is identical to a read, with the exception that there is no need to spend the power to change the state of DBus, so the page bus interface is turned off during refresh. This aggressive refresh policy was designed into our system because of worries about data retention time in the DRAM; these worries fortunately turned out to be unfounded, as we will discuss in Section 6, so other applications of this memory might get by with less aggressive refresh policies.

Generating the control signals shown in Figure 4 requires timing on a finer scale than the chip clock. We used a very simple scheme in which a single inverted, delayed clock is combined logically in various ways with the main clock to generate the several timing edges. At the fast corner, the duration of PchL and PageRdL, for example, is quite short, but the devices that are driven by the controls have correspondingly higher conduction and can get their job done in less time. This simple approach held power consumption constant over simulated wide variations in supply, temperature, and process corner. ‘Tuning’ the single delay circuit (composed entirely of minimum-length devices for accurate tracking) was easy, simply by setting it to ‘just long enough’ at the worst-case corner at the highest clock speed.

4. Data bus circuitry

More than half of the design time was spent investigating various alternatives for the data bus and its interfaces. Before describing some of these alternatives, we motivate the discussion by outlining the main design goal: reducing bus power to an acceptable level. In this application, 2,048 bus pairs carry data between processors and memory, and data on the busses can change on each cycle. These metal-3 busses are about 2.2 mm long and pass over dense arrays of metal-2 wires running at right angles; capacitance is about 750fF. Bus power for full-swing signals would be 2048 x 750fF x $(3.6V)^2$ x 100MHz = 2 watts. The only practical way to reduce power was to reduce bus voltage swing. 1 watt was the power budget, so bus voltage swing had to be less than Vdd/2.

The low-voltage bus scheme uses a low-precharge, as previously described. Precharge is actually only required for memory reads. On writes, the bus is simply driven to the correct data value directly; the bus write circuitry also takes care of read precharge. The processor's data bus interface circuitry is shown in Figure 5.

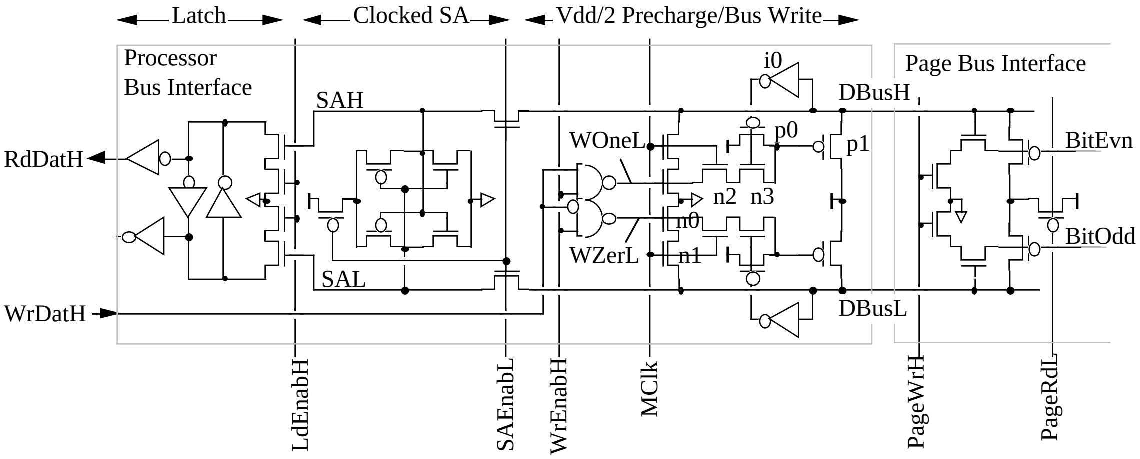

Figure 5. Processor bus interface circuitry.

Considering first the precharge/write circuitry, during a read WrEnabH is low, so both W{One,Zer}L are high. MClk, an inverted version of Clk, precharges DBus{H,L} low during the first half of the clock cycle, and releases the bus for the read during the second half of the cycle. Writes work as follows: suppose the data on WrDatH is high, then WOneL is low and WZerL is high. The NFET stack n0,n1 pulls DBusL low on MClk = high. The N-passgate pair n2,n3 pulls the gate of p1 low, and p1 moves DBusH toward Vdd. The inverter i0 is ratioed heavily in favor of its NFET; when the voltage on DBusH gets to i0's threshold, i0 turns passgate n3 off, and snubber p0 on, rapidly pulling p1's gate high, shutting p1 off, and thereby holding the bus high voltage somewhat below Vdd/2. The idea for a feedback circuit to control bus voltage was inspired by [5] though our implementation differs considerably in detail. Note that there is some opportunistic power saving in this design; when writes fall on successive cycles and the data value does not change, the bus voltage remains fixed.

Bus reads require amplifying the small DBus voltage difference (typically a few hundred millivolts) to a full-swing signal. A simple clocked sense amp with a source-coupled PFET pair performs this job. While SAEnabL is high, the bus voltage is transferred onto the sense amp terminals SA{H,L}. When SAEnabL is asserted, the N-passgates disconnect the sense amp from the bus, and the PFET current tail transistor pulls current through the sense amp, which rapidly amplifies the voltage difference. The sense amp's output is latched onto RdDatH when the latch enable LdEnabH is asserted.

5. Layout and simulation details

The DRAM was fabricated on a 0.5µ process, Hewlett-Packard's CMOS14. The process has three levels of metal interconnect, 0.6µ drawn and 0.35µ effective minimum device length, and is intended for 3.3-volt operation. The DRAM (and all the other parts of the EMC) design was handcrafted using MAGIC, with a locally improved interconnect capacitance extractor and a very carefully generated technology file, in which considerable attention was devoted to wiring parasitics. Our technology file's geometric design rules are very similar to MOSIS's "scmos-tm" (tight-metal) rules, with $\lambda = 0.35\mu$, and 1.75µ (m1), 2.1µ (m2), and 3.5µ (m3) pitches. Circuit simulation was performed using Meta-Software's HSPICE, and switch level simulations were done with IRSIM. We used a set of locally-constructed 'wrapper' programs that coordinate and compare circuit and switch level simulations and that invoke appropriate FET models, supply voltages, temperatures and clock waveforms for extreme-case simulation. We estimate these simple tool extensions saved our design team perhaps a man-year or so of effort over the course of five custom chip designs.

Switch-level simulation of a 1-transistor DRAM is somewhat problematic. IRSIM has some notion of how charge is shared between capacitors of different values, but we were unable to persuade it to handle both reads and writes correctly. We ended up using a fairly brutal hack to solve this problem. A simple preprocessor substitutes a 4-transistor DRAM cell, which IRSIM can handle without difficulty, into the .sim file at each occurrence of a DRAM cell. The dummy cells were eliminated entirely from switch level simulation, since the 4-T DRAM provides both halves of a differential signal pair.

Turning to the memory cell itself, we use an NMOS capacitor to store data, similar to early planar DRAMs. Charge is stored in the channel of the device, and its 'gate' electrode, or plate, is connected to Vdd to ensure that the channel is in saturation when storing a '0'. In a Vdd-precharge design, the capacitance for storing a '1' is not critical, since we do not want to move the bit line voltage from its precharged value. The 'drain' terminal of the device is therefore the storage node, and storage capacitance is the sum of the channel/gate, channel/substrate, and drain diffusion junction capacitances. A cross-section of the storage capacitor and bit access transistor is shown in Figure 6.

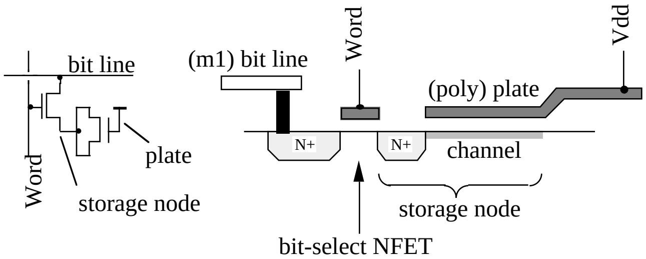

Figure 6. Cross-section of a bit cell.

An important consideration in the design of the storage capacitors was the “channel length” of the MOS storage capacitor. Capacitors that are drawn long cannot be read quickly, because significant time is needed to move the charge from the far end of the channel to the single ‘drain’ connection through the ‘resistance’ of the channel. For this reason, the capacitor is drawn so that its diffusion terminal (salicided in this process to reduce resistance) straps a large width of channel, thereby reducing the equivalent resistance to the various parts of the storage node. The layout of the storage and dummy capacitors is shown in Figure 7.

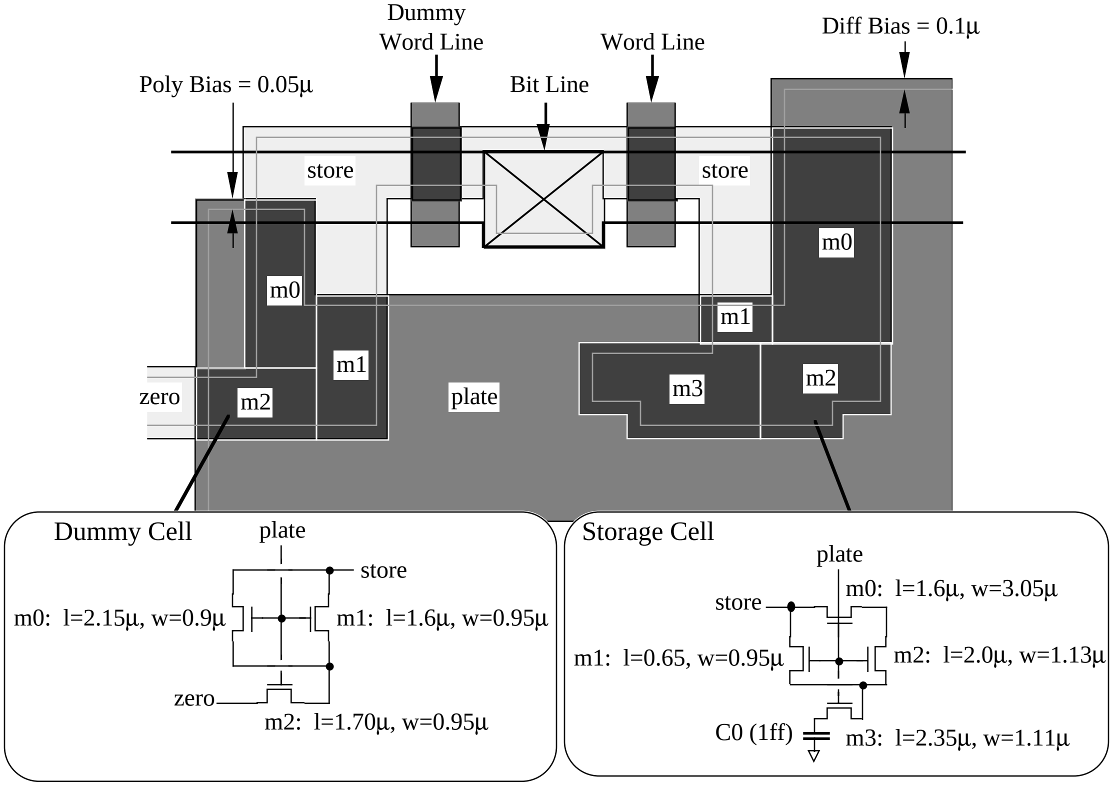

Figure 7. Layout and circuit modeling details of storage and dummy bit cells.

The complexity of the shape of the storage node results from two competing optimizations: maximizing the storage node area, while minimizing the overall layout dimensions. These dimensions were governed mainly by the sizing of peripheral circuits around the memory. The processor could not be laid out easily in less than 36λ per bit slice, so that dimension established the vertical pitch of memory. The word-line and control lines are generated in huge inverters running along the horizontal dimension of the memory system, and these are best laid out on a pitch of 16λ per inverter. This dimension set the overall horizontal pitch of the memory (2 cells per 16λ, one word-line per 8λ).

Because the shape of the storage capacitors is complex, it is necessary to intervene in the circuit extraction process to build a more accurate model of the storage nodes. We model these nodes as a collection of NFETs, interconnected as shown at the bottom of Figure 7. Small capacitors terminate dangling source/drain terminals to keep HSPICE from complaining about unconnected nodes (HSPICE does not have a separate model for MOS capacitors). Source/drain areas and squares in the circuit listing are set to zero, and all the parasitics associated with the diffusion part of the storage node are attached to the access transistor. This crude approximation does not model all the subtleties of this layout, and indeed it would be difficult to do so. Fortunately, the storage capacitances are fairly large and the bit line capacitances low by commercial DRAM standards, so the approach appears sufficiently accurate.

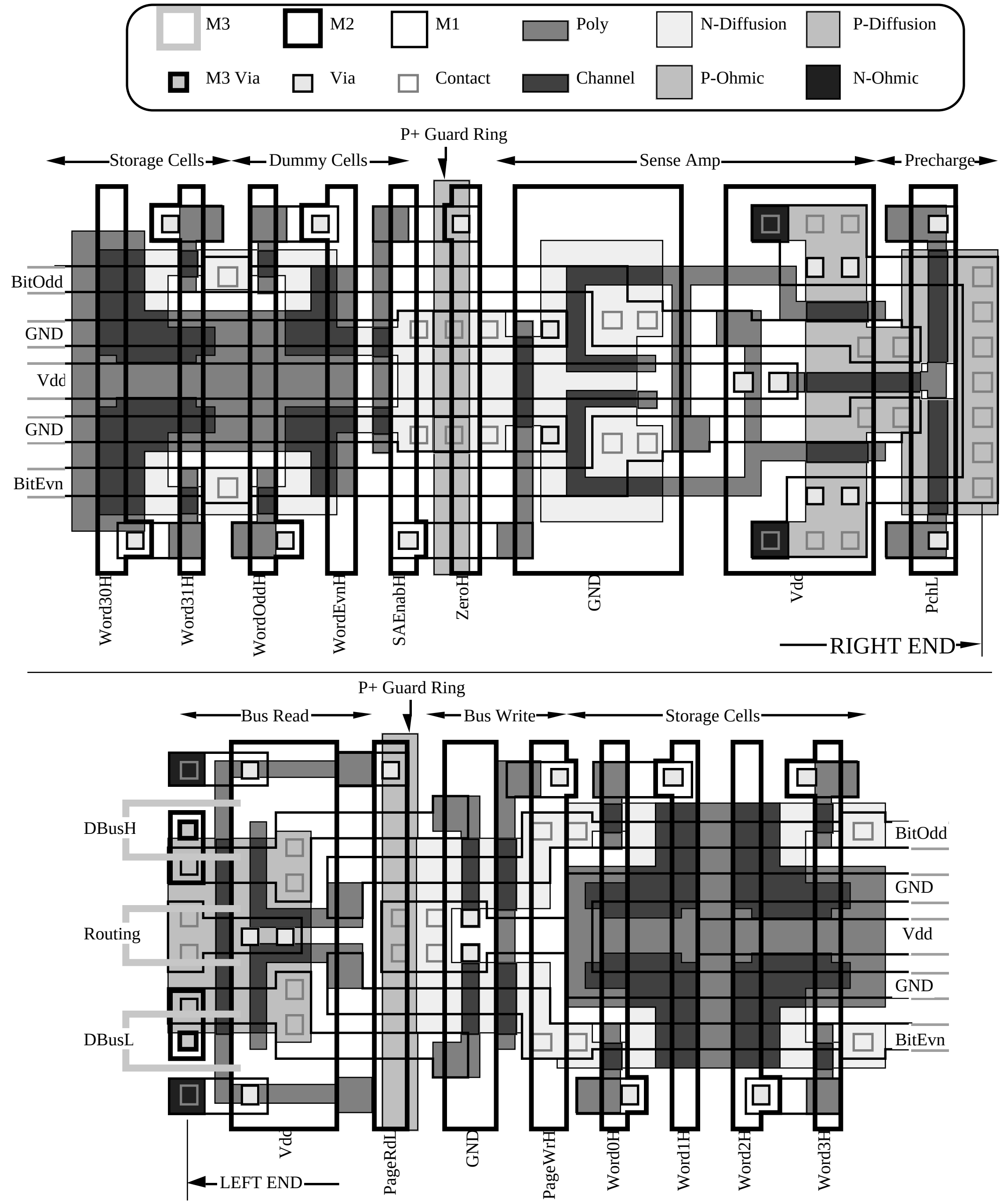

Figure 8. Layout details of DRAM bit slice.

Figure 8 shows the gruesome details of the layout of a bit slice. Salient features include:

Power/Substrate Noise Control. The sense amp detects voltage differences that arise from charge sharing between bit cells and bit lines. It is essential to arrange the circuit so that any noise appearing on a supply terminal or the substrate appears as a common mode voltage to the sense amp. To this end, a central metal-1 strap runs from the sense amp's Vdd supply rail across the polysilicon plate, with periodic contacts to the plate. Thus, the plate (one of two terminals to which the storage capacitance is referred) is connected via a low impedance to a common supply with the sense amp. There is no room within the dense cell array to place substrate contacts, so the entire cell area of the memory array is surround by a continuous P+ guard ring. This ring is connected on both edges of the cell array within each slice by a pair (for symmetry) of metal-1 straps to the GND terminal of the sense amp, in an effort to keep the substrate capacitance return terminal common to the sense amp.

Word Lines. The pitch of the word lines, 8λ, is larger than the minimum pitch for metal-2; we use the extra space for separation between wires, rather than additional width. Though this increases the resistance of the word lines, the reduction in coupling capacitance to neighboring word lines is (as found through simulation) a better use of the space. Since word lines have to drop down two levels to poly, the large contact area is shared between neighboring bit slices (every other slice is flipped vertically).

Bus Lines. The data bus lines pass over the entire slice in metal-3 (they are shown only at the left end for clarity). There is the possibility that these lines could couple differentially to the bit lines that lie underneath. However, the m3-m1 coupling capacitance is greatly reduced by the dense vertical array of metal-2 (word and control) wires. Further, the bus voltage swing is small and edge rates are slow. Though there was no easy way to model this coupling (a complex 3D capacitance problem), we believed the risk was small, and experimental results appear to justify this guess. Note that there is a third metal-3 rail that passes over the geometric center of each slice. This wire, which should couple only in common mode to the symmetric layout, carries an unrelated signal across the slice. Without the cancellation due to symmetry, noise coupling from the full-swing signal on this wire might have been problematic.

Guard Rings. A series of guard rings surrounds each panel's array of memory pages, in the order N+/NWell, P+/Substrate, N+/NWell, biased at Vdd,GND,Vdd respectively. The P+ guard ring shown in Figure 9 around each page's memory cell array is inside these outer guard rings. The outer guard rings are intended to capture minority carriers that may travel for long distances in the low-resistance P+ base wafer material that underlies the P- epitaxial substrate in this process (and most other modern processes). Though these rings were designed several years ago, they appear to conform to recent recommendations for protecting analog circuitry from substrate effects in mixed-mode ASICs [7].

Control Wire Modeling. Though controls run vertically in relatively low-resistance metal-2, the total resistance of a minimum-width control wire is of order of 500Ω, and the distributed load is about 2.5pF, so the delay is not negligible with respect to the 10nsec clock period. Referring to Figure 2, memory slices near the top and bottom of the array receive half their word lines late, across maximum-length wires, but get their controls early, from neighboring control blocks. Memory slices near the central power bus get moderately delayed word lines but maximally delayed controls. To deal with this in simulation, we introduced fairly detailed RC ladder models for each control wire, then performed extensive memory simulations for each of the four worst-case positions for a slice.

Noise Modeling. Since the memory slices are in the central core of a chip that may have large power supply current fluctuations, we were concerned about the memory's performance in the face of large amounts of power supply and substrate noise. To model this effect, we ran extensive simulations with large resistances inserted in the supply nets to allow significant (0.5 volt or so) noise to develop on supply and substrate terminals. Memories were simulated with processors running scraps of code, chosen to stress memory function as fully as possible. In addition to verifying that the memory system delivered correct digital values back to the processor during a read, we also extracted the bit line voltage difference from the simulation whenever SAEnabH passed upward through the NFET threshold. In all cases, the voltage difference at the sense amp was greater than 140 mV. We estimated the sense amp's offset voltage due to transistor and capacitance mismatches at no more than 30 mV, so this voltage difference appeared reasonably safe.

Leakage and Data Retention Time. We close this section with a discussion of data retention time in the face of leakage currents, both subthreshold and junction (experimental results are outlined in the next section).

Drain current in a MOSFET is not zero even when Vgs = 0. Under these conditions, a small *subthreshold* current can flow between source and drain via a diffusion mechanism much like collector current in a bipolar transistor. This current is enhanced by the presence of the gate electrode, increases by about an order of magnitude for each 100mV increase in gate voltage, and scales with the shape factor of the device. Subthreshold leakage is not a concern in the MOS storage capacitor in our design, because there is no drain to collect current from the diffusion terminal. There is the possibility, however, that leakage in the bit-cell select device could charge an initially-low storage node toward the Vdd-precharged bit line. In a quiescent memory slice, the bit lines are held at Vdd and the word lines are held (approximately) at GND. Any noise on the word line will exponentially increase the leakage in the select device. However, we hypothesized that once the storage node rises by a small voltage, the exponential factor in Vgs reduces the leakage current to a negligible value.

The remaining concern is reverse junction leakage in the N+ storage node diffusion; this leakage current discharges an initially-high storage node. Junction leakage has both area (bottom) and linear (junction side-wall) components. In our process, these components are worst-case estimated at about 1fA/μ^2 and 2fA/µ respectively at room temperature. The diffusion terminal of the storage node has area and perimeter of 3μ^2 and 7µ respectively, so room temperature leakage current is about 17fA. Junction leakage increases exponentially with temperature at about a decade every 30°C, so at the maximum allowed junction temperature of 110°C, we would expect leakage to be nearly three orders of magnitude higher, about 6pA. From the storage cell dimensions and Tox = 9nM, we estimate the storage capacitance at about 50fF, so the rate of change of the voltage on the storage node is about 120V/sec, and the storage time is about 8msec. Our refresh policy was designed to work with data retention times of less than 1msec, so this estimate appeared safe at design time for the DRAM.

6. Experimental results

The DRAM has been fabricated on several wafer lots over the past year, and full functionality has been demonstrated. Test performed on wafers with near-nominal fabrication parameters were found to operate at 150MHz, even in packages, where temperatures are generally higher than on a thermally massive wafer test chuck. A schmoo plot for a nominal-fabrication packaged chip is shown in Figure 9, showing its limits of operation over power supply voltage and clock speed.

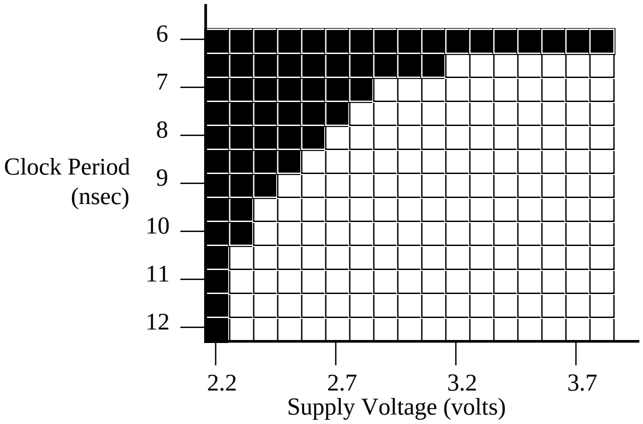

Figure 9. Schmoo plot (supply voltage and clock period) for a nominal chip.

We also have available a *skewed* wafer lot, in which a few wafers in the lot have various parameters purposely biased toward either worst-case (slow) or best-case (fast) corners. These wafers also demonstrated full functionality; the DRAM on the slow skewed wafers operates correctly at over 125MHz (design speed was 100MHz).

To evaluate data retention time, we devised tests that leave one or more pages of memory quiescent (and unrefreshed) for long, variable periods of time. During these time periods, the tests run memory-intensive code on other pages of memory, in order to maximize noise. These tests were run on wafers mounted on a 'hot chuck', a device that allows the wafer temperature to be controlled fairly accurately. All memory data failures were of the form of initially-high storage nodes going low, confirming the hypothesis that junction leakage is the main data-loss mechanism at work in these DRAMs.

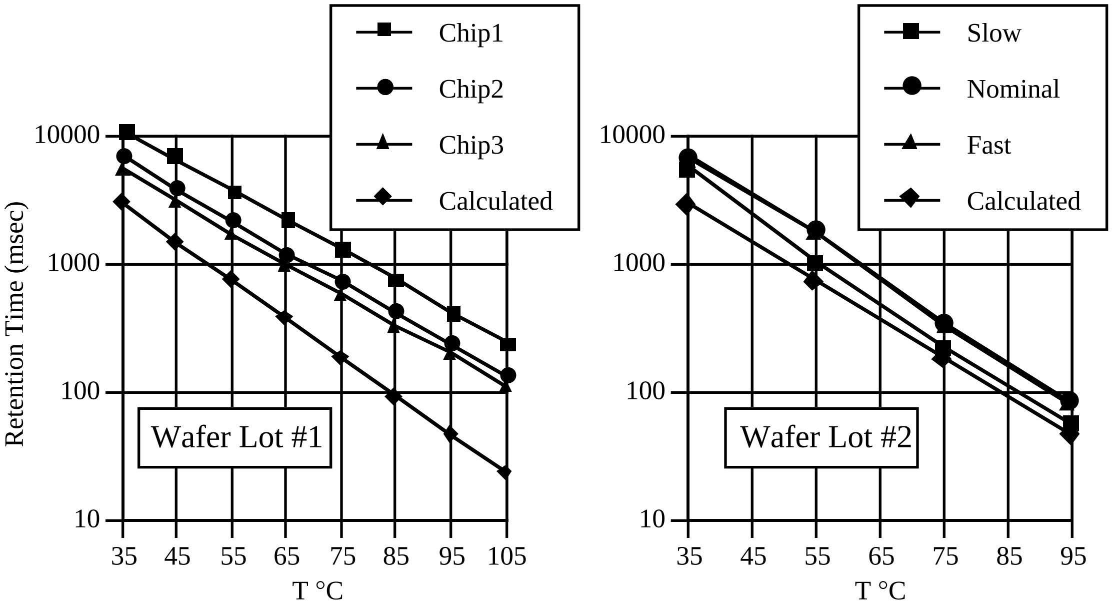

Figure 10. Experimental results for DRAM data retention time.

The results are summarized in the plots shown in Figure 10. Data retention time for three die from an early wafer lot are shown on the left of the figure, along with a calculated value for retention time, computed as in the previous section. Many die were measured in the second, skewed wafer lot; results shown are typical for die on the slow, nominal, and fast wafers. Generally, leakage on the second wafer lot was found to be higher than on the first. However, even the worst-case high-temperature data retention time was larger than 10msec and was easily accommodated by the refresh scheme.

While we could not measure the power consumption of the memory system directly, overall chip power consumption agreed well with estimates based on circuit simulation. These estimates indicate that the memory itself consumes about 800mW, and the low-voltage-swing busses and their controllers consume another 780mW.

Chips from various lots have been integrated into a pre-production prototype graphics system at Hewlett-Packard’s Chapel Hill Graphics Laboratory; a product announcement is expected mid-summer 1997.

7. Summary, previous work, and acknowledgements

This paper has described a DRAM compatible with a standard CMOS ASIC process. It provides memory density at least 4x improved over P-load SRAM in the same layout rules, runs at speeds comparable to logic in the same process, and uses circuitry that is reasonably simple and straightforward. The design employs Vdd-precharge bit lines, half-capacitance, full-voltage dummy cells, and a simple complementary sense amp. DRAM is organized as a number of small pages, allowing simple circuit design and low-power operation at modest expense in area overhead. The paper also described a power-conserving low-voltage-swing bus design that connects multiple pages to a full-voltage-swing interface. This work shows that it is not only possible to integrate fairly dense DRAM with logic on an ASIC process, but that it is relatively straightforward to do so, as postulated in [7].

This work was inspired to a great extent by the experience of others in the university VLSI community, notably those described in [4] and [6]. The painstaking work of Speck [4], in particular, allowed us to avoid some nasty pitfalls. For example, our DRAM uses a folded bit line arrangement, in contrast to the open bit line layout in the MOSAIC DRAM. Folded bit-line layouts avoid many of the serious noise and voltage shift problems outlined in [4], and are always preferred in applications where there is room to fit them.

There has been considerable recent renewed interest in the problem [7-9], including an entire session at the 1996 International Solid State Circuits Conference. References [8] and [9] described Vdd/2-precharge DRAMs, which offer potential power savings over our full-Vdd-precharge approach, but at the expense of more complex circuitry. These references describe rather different bit-cell circuits that attempt to deal with junction leakage, for example by isolating the storage node in a biased NWell. Our results suggest that, in applications where refresh times are of order 10msec, no such heroic means are necessary, and the simpler and much denser approach, using an NMOS capacitor, is satisfactory.

We gratefully acknowledge the support of the Defense Advanced Research Projects Agency (Order No. A410) and the National Science Foundation (MIP-9306208) for the early development of PixelFlow chips, systems, and architecture. We also wish to thank Division, Ltd. (Bristol, UK), which joined the University of North Carolina in a co-development effort to complete the design of the PixelFlow components, and Hewlett-Packard Corporation, which has recently undertaken the completion of the system.

The author also wishes to thank John Eyles and Steve Molnar (both currently at Hewlett-Packard Chapel Hill Graphics Lab) and Steve Tell, collaborators for many years, without whom these designs would have been impossible, and Randy Bailey (Hewlett-Packard, Fort Collins, CO), who performed all of the wafer-level DRAM tests.

8. References

[1] Poulton, J., Fuchs, H., Austin, J., Eyles, J., Heinecke, J., Hsieh, C-H, Goldfeather, J., Hultquist, J., and Spach, S., "PIXEL-PLANES: Building a VLSI-Based Graphic System," Proceedings of Conference on Advanced Research in VLSI, 1985, pp 35-60.

[2] Fuchs, H., Poulton, J., Eyles, J., Greer, T., Goldfeather, J., Ellsworth, D., Molnar, S., Turk, G., and Israel, L., "A Heterogeneous Multiprocessor Graphics System Using Processor-Enhanced Memories," Computer Graphics (Proc. of SIGGRAPH '89), Vol. 23, No. 3, pp 79-88.

[3] Molnar, S., J. Eyles, and J. Poulton, "PixelFlow: High-Speed Rendering Using Image Composition," Computer Graphics (Proc. of SIGGRAPH '92), Vol. 26, No. 2, pp. 231-240.

[4] Speck, D., “The Mosaic Fast 512K Scalable CMOS dRAM,” Proceedings of Conference on Advanced Research in VLSI, 1991, pp 229-244.

[5] Bakoglu, H.B., “Circuits, Interconnections, and Packaging for VLSI,” Addison-Wesley, New York, N.Y., 1990, pp 175-178.

[6] Gasboro, J., and M. Horowitz, “A Single-Chip Functional Tester for VLSI Circuits,” 1990 ISSCC Digest of Technical Papers, pp 84-85.

[7] Verhhese, N., T. Schmerbeck, and D. Allstot, “Simulation Techniques and Solutions for Mixed-Signal Coupling in Analog Circuits,” Kluwer Academic Publishers, Norwell, MA, 1995, Chapter 11.

[8] Foss, R., “Implementing Application Specific Memory,” 1996 ISSCC Digest of Technical Papers, pp 260-261.

[9] Gillingham, P., B. Hold, I. Mes, C. O’Connell, P. Schofield, K. Skjaveland, R. Torrance, T. Wojcicki, and H. Chow, “A 768K Embedded DRAM for 1.244Gb/s ATM Switch in a 0.8μm Logic Process,” 1996 ISSCC Digest of Technical Papers, pp 262-263.

[10] Hashimoto, M., K. Abe, and A. Seshadri, “An Embedded DRAM Module using a Dual Sense Amplifier Architecture in a Logic Process,” 1997 ISSCC Digest of Technical Papers, pp 64-65.