IMPACT OF MCMs ON SYSTEM PERFORMANCE OPTIMIZATION*

A. I. Kayssi, K. A. Sakallah, R. B. Brown, R. J. Lomax, T. N. Mudge, and T. R. Huff
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109-2122

Abstract *We present a performance model for the University of Michigan MCM-based GaAs microcomputer. The model takes into account architectural as well as MCM packaging considerations such as the chip-to-substrate bonding method, the dielectric constant of the insulator, and the resistivity of the metal conductor. We illustrate the use of the model by finding the I-cache size which maximizes the MIPS rating for given MCM technology parameters.*

1 Introduction

As the performance goals of computers increase, packaging techniques become critical for reducing the delays that result from chip crossings and interconnect among chips. Packaging technology has developed to the point where it is now possible to integrate a processor and cache onto a single Multi-Chip Module (MCM) a few inches on a side. MCMs offer the following advantages over printed circuit boards. First, they eliminate one level of packaging, which also eliminates the parasitic delays, and improves overall reliability. Second, they allow chips to be placed closer together, improving the area utilization, and decreasing the length of the interconnect lines between the chips. Finally, denser lines can be used on the MCM, because the line width, pitch, and thickness on MCMs are much smaller than those on PCBs. All of these factors contribute to an improvement in signal delay times on the MCM substrate as compared to a PCB, which directly affects system performance. Simulations of a GaAs microcomputer currently under development at the University of Michigan [1] show that using an MCM for packaging improves system performance by approximately 50% as compared to a PCB, due to a factor of three decrease in the signal propagation delay from CPU to memory. This high-performance microcomputer, which implements the MIPS RISC architecture [2], is being designed using GaAs direct-coupled FET logic (DCFL) technology from Vitesse Semiconductor Corporation. The system components mounted on the MCM are the CPU and floating-point accelerator, the primary instruction- and data-caches, and a memory management unit.

This paper presents, and demonstrates the application of a performance model of the above mentioned GaAs microcomputer. The model takes into account architectural as well as packaging considerations. Specifically, the model expresses the *MIPS* rating of the computer in terms of the first level instruction cache (I-cache) size, the chip-to-substrate bonding method, the dielectric constant of the insulator, and the resistivity of the metal conductor. We illustrate the use of the model by finding the I-cache size which maximizes the *MIPS* rating for given packaging parameters.

2 Performance Model

The memory organization of the GaAs microcomputer consists of a two-level cache with split instruction and data caches. The *MIPS* figure (millions of instructions per second) is a suitable measure of system performance, and is defined as

$$MIPS = \frac{1000}{T_c \times CPI}$$

where T_c is the processor clock period in nanoseconds, and CPI is clock cycles per instruction for a representative set of application programs. Both of these quantities are functions of, among other things, the sizes of the instruction and data caches. In our model, however, architectural constraints dictate a fixed data cache size of 4K words [3], leaving the I-cache size as the only free architectural parameter on the MCM. The variation of T_c and CPI with I-cache size was obtained after MCM layouts for the system were generated for 2K, 4K, 8K, and 16K I-cache sizes. The cache memory is implemented using custom $1K \times 32$ bit GaAs SRAM chips, currently being designed at the University of Michigan. For each cache size, the chip bonding technology, the relative dielectric constant of the insulator, and the conductivity of the interconnect metal were all varied to cover the range of MCM technologies available from MCM vendors. Circuit simulation was then used to obtain the

*This work was supported by the Defense Advanced Research Projects Agency under DARPA/ARO Contract DAAL03-90-C-0028, and by the U.S. Army Research Office under the URI Program, Contract DAAL03-87-K-0007

0-7803-0593-0/92 $3.00 1992 IEEE

clock period T_c, and results from architectural simulations were used to determine the corresponding CPI.

As detailed in [3], the clock period T_c of the processor is

$$T_c = \frac{1}{2}(T_{MCM} + T_{CPU} + T_{MEM})$$

where T_{CPU} is the critical delay through the processor chip, T_{MEM} is the access time of the cache, and T_{MCM} is the round trip signal propagation delay on the MCM, including input and output buffers on the CPU and cache chips. The worst-case delay through the processor chip, as simulated in HSPICE [4], is $T_{CPU} = 3.0$ ns and the access time for the memory is expected to be $T_{MEM} = 3.0$ ns. T_{MCM} was obtained from the actual MCM layouts; line lengths were extracted and modeled by lossy transmission lines. We varied ϵ_r, the relative dielectric constant of the insulator, from 2.5 to 4.0, and ρ, the resistivity of the conductor, from $10n\Omega m$ to $40n\Omega m$. The line dimensions are shown in Fig. 1. Output pad transmitters included pull-up and pull-down transistors with a gate width of $400\mu m$. The input pad receivers consist of two voltage limiting diodes and one pass transistor. No line terminating resistors were used, as the diodes from gate to source of the MESFETs at the receiving ends of the lines present an adequate termination.

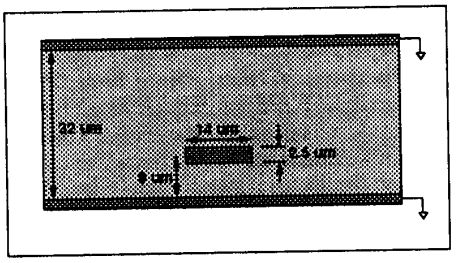

Figure 1: MCM Line Geometry.

The chip bonding method (CBM) was assumed to be one of flip-chip, TAB, or wire bonding. The die-to-substrate bond parasitics were modeled by a low-pass pi-section with two capacitances and one inductance. The values of the elements are shown in Fig. 2.

The resulting circuits for a one bit line going from the CPU to memory and back were simulated in HSPICE for each I-cache size considered, using the HSPICE built-in lossy transmission line model. Time delays were calculated at the $V_{in} = V_{out}$ point for a GaAs DCFL inverter (400 millivolts.) The delay on the MCM, T_{MCM}, and therefore the clock period T_c, can then be expressed as a function of the size of the I-cache, S_I, the relative dielectric constant of the insulator, ϵ_r, the resistivity of the interconnect conductor, ρ, and the chip bonding method (CBM)

$$T_c = f(S_I, CBM, \epsilon_r, \rho)$$

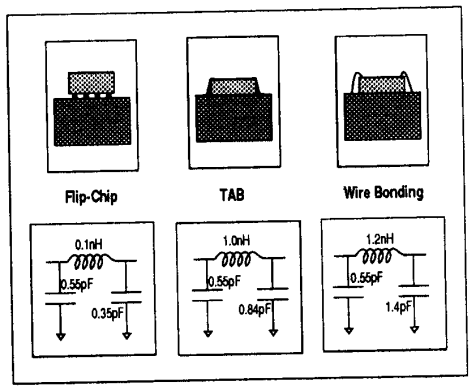

Figure 2: Parasitic Elements Associated with Chip Bonding.

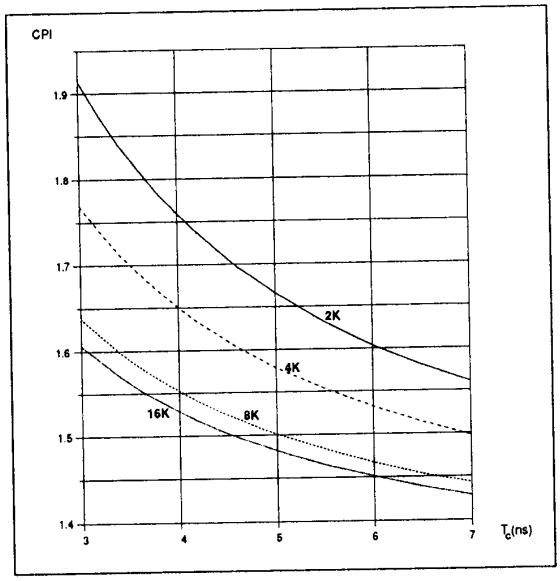

Figure 3: Variation of CPI with I-Cache Size and Clock Period.

The clock cycles per instruction (CPI) metric was obtained for various I-cache sizes from the cache simulator **cacheUM**, developed at the University of Michigan. Figure 3 shows the variation of CPI with cache size and clock period. The cache simulations were performed for sixteen benchmarks reflecting the typical workload of an engineering workstation [3]. The functional dependence

of CPI on S_I and T_c can be expressed as

$$CPI = g(S_I, T_c)$$

The $MIPS$ rating is

$$\begin{aligned} MIPS &= \frac{1000}{T_c \times CPI} \\ &= \frac{1000}{f(S_I, CBM, \epsilon_r, \rho) g(S_I, f(S_I, CBM, \epsilon_r, \rho))} \\ &= h(S_I, CBM, \epsilon_r, \rho) \end{aligned}$$

The function h is plotted in Fig. 4 for all the I-cache sizes used in the simulations. Only two metals are shown: copper (Cu) and aluminum (Al). The plots show how the expected system performance is affected by the choice of insulator material, metal conductor, and CBM . By taking the maximum $MIPS$ rating from each of these plots we can derive a combined summary plot, shown in Fig. 5, which allows us to identify the I-cache size that maximizes $MIPS$ for given packaging parameters.

3 Remarks and Conclusions

We can draw several conclusions from the data in figures 4 and 5:

- The smallest and largest cache sizes considered yield sub-optimal performance. Specifically, the 2K cache results in too high a CPI while the 16K cache requires a long cycle time due to its large footprint on the MCM.
- For the range of insulators and conductors considered, an 8K cache is always better than a 4K cache if flip-chip bonding is used. Otherwise, the cache size that yields the best performance depends on the other packaging parameters. For example, a system built using copper conductors and wire bonding requires an 8K cache for insulators with $\epsilon_r \leq 3.2$ and a 4K cache for insulators with $\epsilon_r \geq 3.2$. As another example, a system using polyimide ($\epsilon_r = 3.0$) as an insulator and wire bonding requires an 8K cache if copper conductors are used; if aluminum is used, however, optimal performance requires a smaller 4K cache!
- Figure 4 clearly shows that the curves have a larger slope, and become more dispersed as the number of cache chips increases. This implies that the dependence of $MIPS$ on ϵ_r, ρ, and CBM becomes greater for larger cache sizes. This is due to the fact that the line lengths and the number of loads on the lines are larger, which makes the delay on the MCM an increasingly important factor in determining the clock cycle time.

The choice between a 4K cache size and an 8K cache size is also affected by other criteria such as power dissipation, area, and reliability. The number of memory chips for an 8K I-cache is twice that of a 4K. This not only occupies more of the MCM area, but also implies that the power dissipated in the I-cache will be doubled for the 8K cache. With more power dissipated, the substrate temperature increases, thereby increasing the temperature of all MCM-mounted components. Higher temperatures increase the thermal shear stresses due to the thermal coefficient of expansion mismatch, decrease noise margin, and decrease system reliability and MTTF. Since power supply current increases with the number of memory chips, larger power and ground buses should be used in the 8K cache, again occupying more area on the MCM. Finally, the signal wave shape becomes more distorted as the line length and the number of loads on the line increase. All the above factors favor the use of a smaller cache size.

To conclude, this study has demonstrated that system performance cannot be optimized without *simultaneously* considering both the architectural as well as the technological parameters. The need to do so will grow in importance if significant loss in potential system performance is to be avoided.

References

[1] T. N. Mudge, R. B. Brown, W. P. Birmingham, J. A. Dykstra, A. I. Kayssi, R. J. Lomax, O. A. Olukotun, and K. A. Sakallah, "The Design of a Microsupercomputer," *IEEE Computer*, pp. 57–64, January 1991.

[2] G. Kane, *MIPS RISC Architecture*, Prentice-Hall, 1989.

[3] O. A. Olukotun, R. B. Brown, R. J. Lomax, T. N. Mudge, and K. A. Sakallah, "Multilevel Optimization in the Design of a High-Performance GaAs Microcomputer," *IEEE Journal of Solid State Circuits*, May 1991.

[4] Meta-Software Staff, *HSPICE User's Manual*, Meta-Software, 1991.

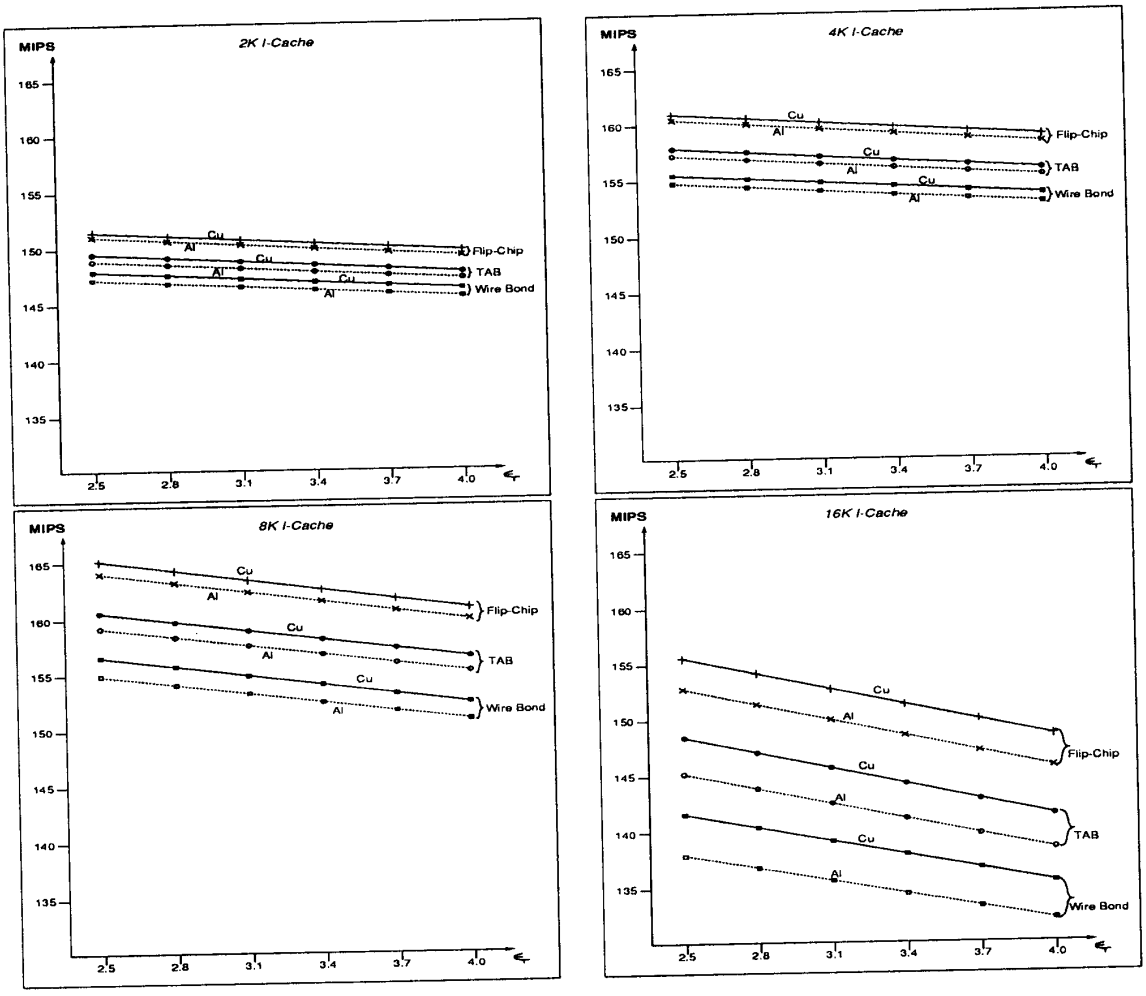

Figure 4: System Performance for Different I-Cache Sizes

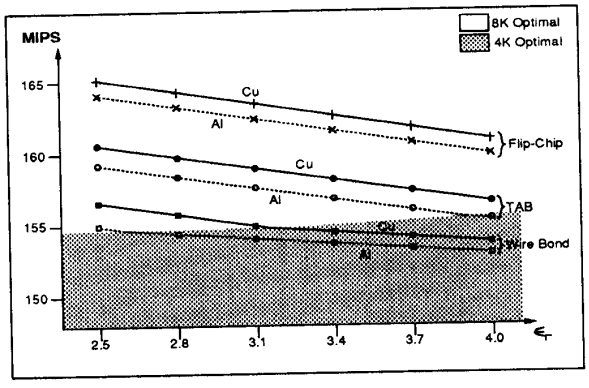

Figure 5: Optimal Performance and Corresponding Cache Size