



Senior Thesis in Computer Engineering
University of Illinois at Urbana-Champaign
Advisor: Rakesh Kumar

Procurability-Aware Design Space Exploration

Lin Chuan Lee

Submitted in part fulfilment of the requirements for the degree of
Bachelor of Science in Computer Engineering of the
University of Illinois at Urbana-Champaign, May 2023

Abstract

The unpredictable demand, complex supply chain, and high barrier of entry of the semiconductor industry make it prone to shortages. Consequently, creating designs with highly procurable, easily replaceable components has become an increasing priority for any businesses relying on semiconductors: from car makers and home appliance manufacturers to the many companies producing consumer and commercial electronics. While High Bandwidth Memory (HBM) memory offers high bandwidth, low power consumption, and a small form factor, it has lower procurability and higher cost compared to traditional off-chip DRAM DIMM due to the limited number of suppliers and the use of interposers.

In this work, we performed design space exploration (DSE) to find design points using DDR memory with various L3 capacities, number of memory channels, and memory frequency with reasonable performance and cost compared to a system using HBM memory. The DSE is performed through an analytical model that estimates the performance, area, power, and cost of design points. Using a roofline model, we found regions where the system performance is constrained by bandwidth between compute core and L3 cache, bandwidth between the L3 cache and main memory, and compute throughput. We found, for the same L3 capacity, the HBM system has the highest memory cost and total cost but the lowest package cost compared to DDR systems. We demonstrated that, across a range of application profiles with different bandwidth requirements, the HBM system has the highest system cost but outperforms DDR systems in terms of compute die area, package area, and compute die power in iso-performance comparisons. Finally, we found design points using DDR memory that offers the same performance as the reference HBM system at a lower cost, even when lifetime energy costs are considered.

Subject Keywords: architectural design space exploration (DSE); HBM memory; supply chain

Contents

1	Introduction	1
1.1	Thesis Outline	2
2	Background	3
2.1	Procurability-Aware Designs	3
2.2	HBM Memory	4
3	Analytical Model	5
3.1	Performance Model	6
3.2	Enforcing Feasible Designs	7
3.3	Cost Model	10
4	Experimental Setup	12
5	Results	14
5.1	Compute Bound versus Bandwidth Bound	14
5.2	Cost Composition	17
5.3	Cost-Performance Trade-off	19

6	Conclusions	26
	Bibliography	27

Chapter 1

Introduction

As the capability and reach of computing devices continue to rise, procurability becomes an important consideration. The semiconductor industry, with its unpredictable demand, complex supply chain, and high barrier of entry, is prone to shortages [1]. In addition to stockpiling inventory, observing the supply chain in advance, and other marketing strategies, companies using semiconductors can overcome supply chain disruptions by deploying engineering techniques to design products that rely less on difficult-to-procure components and use modular architectures that allow quick substitution of unavailable components [2]. For example, High Bandwidth Memory (HBM) memory offers higher bandwidth, lower power consumption, and a smaller form factor compared to DDR memory [3]. However, HBM is less procurable because of a smaller number of suppliers and a higher integration cost due to the use of interposers. We performed a design space exploration (DSE) to identify design points using DDR memory that can achieve acceptable performance and cost compared to the reference HBM system by having different L3 capacities, number of memory channels, and memory frequencies.

To perform the design space exploration, we constructed an analytical model of the system, including compute cores, cache, I/O, and memory controllers. The performance of each design point is derived from a roofline model that considers L3 hit rate, memory bandwidth, application profile, and compute core parameters. The cost of individual design points includes the cost of compute die, interposer, packaging, and memory. Using parameters from modern systems,

we observed regions where the system performance is bounded by compute core-L3 bandwidth, L3-main memory bandwidth, and compute throughput. For the same L3 capacity, we observed that the reference HBM system has a higher memory cost and total cost but a lower package cost compared to any DDR system. Across a range of application profiles from compute-intensive to memory-intensive, we demonstrated that the HBM system has the highest system cost but outperforms DDR systems in compute die area, package area, and compute die power for iso-performance comparisons. This means there exist design points where HBM memory can be substituted with DDR memory for the same performance at a lower system cost by incorporating a larger L3 cache, more memory channels, or higher memory frequency. This conclusion still holds even when the lifetime energy cost is considered.

1.1 Thesis Outline

This thesis has been organized into six chapters.

- In Chapter 2, we consider the state of the semiconductor industry that makes procurability an important design criterion.
- In Chapter 3, we discuss the formulation of the analytical model and describe how it estimates systems' performance, cost, area, and power.
- In Chapter 4, we discuss the set of design points of interest to be considered in our DSE.
- In Chapter 5, we describe the results from applying our analytical model to the set of design points.
- In Chapter 6, we summarize the main findings of this work.

Chapter 2

Background

2.1 Procurability-Aware Designs

The global chip shortage that began in 2020 impacted more than 169 industries [4], with the automobile industry alone expected to lose \$210 billion in revenue in 2021 [5]. The global shortage was caused by a multitude of factors, including the COVID-19 pandemic’s work-from-home economy boosting demand for consumer electronics [6], political tension between the U.S. and China resulting in restrictions on semiconductor imports and exports [7], proof-of-work cryptocurrency mining increasing demand for GPUs [8], and even natural phenomenon like drought impacting foundry production [9]. The semiconductor industry has unpredictable demand due to the rapid emergence of different applications that require different kinds of silicon chips. The supply is also unpredictable since each part of the complex supply chain can become the bottleneck. The industry’s low-margin nature also means foundries are often operating at close to max capacity [1] without a redundancy buffer. The global chip shortage of 2020 demonstrated that the semiconductor industry is susceptible to shortages whenever there is a mismatch between demand and supply.

Companies from a wide range of industries depend on semiconductors for their products. For these businesses, resilience to disruption in the semiconductor supply chain is key. One of the ways to prepare for these shortages is to consider procurability in product architectures.

Instead of using components with a small number of suppliers, companies design products with modular and flexible architectures that allow the substitution of components when a subset of components becomes unavailable [2].

2.2 HBM Memory

One such example of hard-to-procure components is HBM memory. DRAM dies in HBM are vertically stacked and connected by through-silicon via (TSV) and micro bumps. The HBM DRAM stack is then integrated into the same package as the compute die through a silicon interposer, resulting in a smaller form factor compared to off-chip DRAM DIMMs. HBM offers much higher bandwidth compared to DDR memory by employing a much wider interface (1024 bits for HBM [3] versus 32 bits for GDDR [10]). The lower frequency and shorter interconnect distance through the interposer mean HBM also has lower power consumption. However, HBM memory has low procurability due to a limited number of suppliers (Micron Technology, Samsung, and SK Hynix are the only main manufacturers [11]), which also contributes to its high cost. HBMs are also expensive due to the use of silicon interposers (silicon fabrication is more costly than PCB production). Thus, in this work, we perform a design space exploration to search for procurability-aware designs by substituting HBM with DDR memory. We attempt to compensate for the bandwidth difference by a combination of larger L3 capacities, more memory channels, and high memory frequencies.

Chapter 3

Analytical Model

In this chapter, we describe the formulation of our analytical model to be used in design space exploration (DSE). The system, as modeled in the DSE, includes compute cores, shared L3 cache, memory controllers, and I/O (L1 and L2 caches are private to and part of each compute core). We use the term *memory subsystem* to refer to the memory hierarchy consisting of L3 and the main memory. The analytical model consists of three main parts. First, for each design point, the model estimates the performance via a roofline model that considers L3 cache hit rate, L3 bandwidth, main memory bandwidth, and compute throughput. Second, the model estimates the power and area of each design point to filter out unrealistic designs. Third, for each design point, the model estimates the cost of the compute die, memory, interposer (for HBM systems), and package.

Figures 3.1 and 3.2 show the cross-section of the physical organization of systems using DDR and HBM memory, respectively. HBM systems use a silicon interposer for 2.5D integration of the compute die and HBM stack. DRAM dies for DDR systems are located in off-chip DIMMs. Bump pitch and bump current rating values used in the model are labeled.

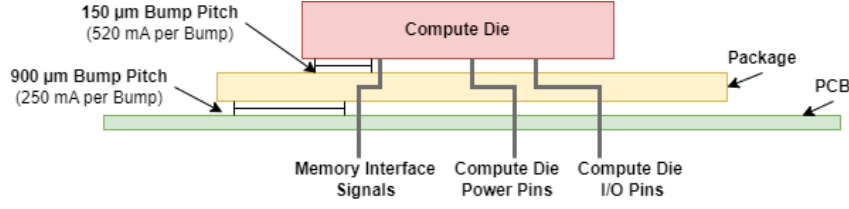


Figure 3.1: Cross-section of systems using DDR memory is shown, with bump pitch and bump current rating labeled.

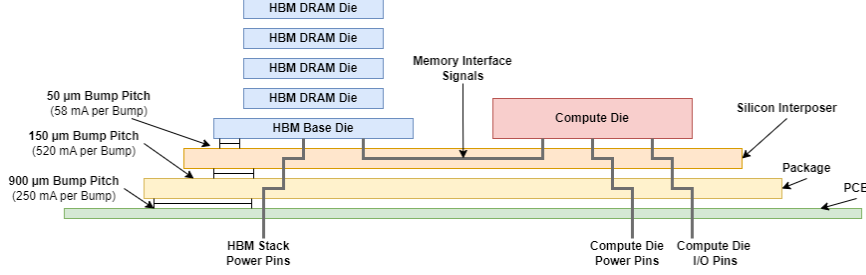


Figure 3.2: Cross-section of systems using HBM memory is shown, with bump pitch and bump current rating labeled.

3.1 Performance Model

Since the performance of the system is either compute-bound or memory-bound, we need to model both the compute throughput and the memory bandwidth of each design point. The compute throughput of the system is given by

$$\pi = CPI \cdot f_{\text{core}} \cdot N_{\text{core}} \quad (3.1)$$

where CPI is the cycles per instruction, f_{core} is the frequency of the compute cores, and N_{core} is the number of compute cores. We assume all compute cores operate at the same frequency.

The bandwidth of the memory subsystem is either bounded by the bandwidth between the compute cores and L3 or the bandwidth between L3 and the main memory. Hence, the memory subsystem bandwidth is given as

$$B_{\text{msys}} = \min \left(N_{\text{L3}} \cdot B_{\text{L3}}, \frac{N_{\text{MC}} \cdot B_{\text{MC}}}{\text{L3 Miss Rate}} \right) \quad (3.2)$$

where N_{L3} is the number of L3 slices, B_{L3} is the bandwidth per L3 slice, N_{MC} is the number of memory controllers, and B_{MC} is the bandwidth per memory controller. The first term is the bandwidth between compute cores and L3 while the second term is the bandwidth between L3 and main memory. The bandwidth between L3 and the main memory depends on the L3 miss rate because we assume that all memory accesses from the compute cores are serviced by L3, and only L3 misses are serviced by the main memory. A higher L3 hit rate implies a better utilization of the L3-main memory bandwidth. The L3 hit rate depends on the relative sizes of the application working set and L3. The hit rate scales linearly with the ratio of the L3 capacity to the working set size until the entire working set fits in L3, at which point we assume a nominal hit rate of 90%.

With the compute throughput and bandwidth of the memory subsystem given by Equations 3.1 and 3.2, respectively, the performance of the system is given as

$$Perf = \min(\pi, B_{msys} \cdot I_{msys}) \quad (3.3)$$

$$I_{msys} = \frac{Cap_{workset}}{Cap_{workset} - (Cap_{L1} + Cap_{L2})} \cdot I_{app} \quad (3.4)$$

where I_{msys} is the arithmetic intensity observed by the memory subsystem and I_{app} is the arithmetic intensity inherent to the applications. Since memory accesses are filtered by L1 and L2 caches before being serviced by the memory subsystem, the memory subsystem observes a higher arithmetic intensity than I_{app} for nonzero L1 and L2 capacities. We assume L1 and L2 cache to be exclusive of one another.

3.2 Enforcing Feasible Designs

To ensure all evaluated design points have reasonable power and area overhead, we developed models to estimate the power and area of each design point.

The thermal energy generated by the system can be dissipated either through the pack-

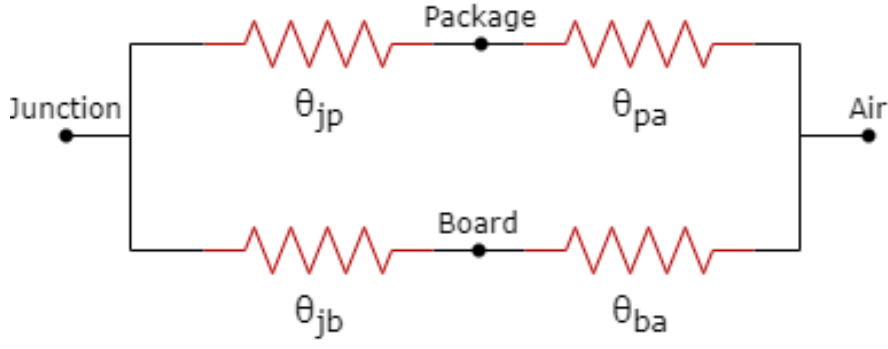


Figure 3.3: Thermal resistance network for the system.

age casing or through the back of the PCB [12]. Based on the configuration of the thermal resistances shown in Figure 3.3, the system thermal resistance is given as

$$\theta_{ja} = \frac{(\theta_{jp} + \theta_{pa})(\theta_{jb} + \theta_{ba})}{\theta_{jp} + \theta_{pa} + \theta_{jb} + \theta_{ba}} \quad (3.5)$$

where θ_{jp} , θ_{pa} , θ_{jb} , and θ_{ba} denote the thermal resistance from junction to package, package to air, junction to PCB, and PCB to air, respectively. Assuming an ambient operating temperature of 25 °C and a maximum junction temperature of 110 °C, the maximum power consumption of the system is given as

$$P_{\max} = \frac{T_{j, \max} - T_{\text{ambient}}}{\theta_{ja}} \quad (3.6)$$

Then, we compute the power consumption of the system to ensure it does not exceed P_{\max} . First, the power of the compute cores is computed as $C \cdot V^2 \cdot f$ where C , V , and f are the capacitance, voltage, and frequency of the compute cores, respectively. The model also implements linear voltage-frequency scaling such that a reduction in frequency gives a cubic reduction in power. Second, the power of the memory controller is given as

$$P_{\text{MC}} = E_{\text{bit}} \cdot f_{\text{MC}} \cdot \#Wire_{\text{MC}} \cdot \left(\frac{V_{\text{MC}}}{V_{\text{MC, nominal}}} \right)^2 + \frac{f_{\text{MC}}}{f_{\text{MC, nominal}}} \cdot P_{\text{controller, nominal}} \quad (3.7)$$

where E_{bit} is the energy required to signal a bit from the compute die to the DRAM die,

f_{MC} is the frequency of the memory controller, $\#Wire_{MC}$ is the number of wires per memory controller, V_{MC} is the voltage of the memory controller. The first term is the physical power required to drive signals from the compute die to the DRAM die, and the second term is the power consumption of the memory controller logic. The memory controller operates at voltage and frequency values independent of the compute cores. The memory controller voltage also scales linearly with the memory controller frequency. Third, the power of the L3 cache is a linear function of L3 capacity. For systems with HBM memory, the power consumption of the DRAM stack also contributes to total system power consumption since the DRAM stack is integrated within the package.

The area of the system is computed as follows. We scale the area of the computing cores linearly with core frequency when the frequency exceeds some baseline which is a model parameter. This is important to account for the microarchitecture changes required to enable higher operating frequency. A 5% increase in core frequency results in a 10% increase in compute core area, with the exception that L1 and L2 area is only scaled by 2%. The final die area is the dot product between the vector of component counts and the vector of component areas.

Since bumps for the memory interface, I/O (PCIe lanes), and power delivery need to fit on the bottom of the die, the bumps enforce a lower bound on the die area. The lower bound is given as

$$A_{die} \geq (\text{Bump Pitch})^2 \cdot (\#Bump_{power} + \#Bump_{MC} \cdot N_{MC} + \#Bump_{I/O}) \quad (3.8)$$

$$\#Bump_{power} = \frac{P_{die}}{V_{die} \cdot I_{bump}} \cdot 2 \quad (3.9)$$

where $\#Bump_{power}$ is the number of bumps for power delivery to the die, $\#Bump_{MC}$ is the number of bumps for each memory controller, $\#Bump_{I/O}$ is the number of bumps for I/O, P_{die} is the power of the die, V_{die} is the voltage of the die, and I_{bump} is the current rating of each bump. If the die area is too small to fit all the bumps, we allocate as much dead space on the die as required. For systems using DDR memory, we assume a compute die to package bump

pitch of 150 μm with a current rating of 520 mA (see Figure 3.1). For systems using HBM memory, we assume a compute die to interposer bump pitch of 50 μm with a current rating of 58 mA (see Figure 3.2). Current rating per bump scales with bump pitch squared. For the upper bound on die area, we only consider systems with die areas less than 1000 mm^2 in our DSE.

Assuming the die has a 3:2 aspect ratio, the die area must be sufficiently large such that there is enough length of the perimeter of the die for wires. The model assumes wires are only connected to the two longer sides of the die. The maximum number of wires that can be connected to the die perimeter is given by

$$\#Wire_{\max} = \left(6 \cdot \sqrt{\frac{A_{\text{die}}}{6}} \right) \cdot \text{PCB Layer} \cdot \text{Link Pitch} \quad (3.10)$$

where the expression in the parenthesis is the combined length of the two longer sides of the die. Since the memory interface accounts for most of the wires that connect to the die perimeter, we require $\#Wire_{\max} \geq \#Wire_{\text{MC}} \cdot N_{\text{MC}}$ where $\#Wire_{\text{MC}}$ is the number of wires per memory controller and N_{MC} is the number of memory controllers.

3.3 Cost Model

To meaningfully perform DSE, we must model the cost associated with each design point. The total cost is the sum of the cost of compute die, memory, interposer (for HBM systems), and package.

To estimate the cost of the compute die that includes compute cores, L3, memory controllers, and I/O, we must model the yield. Since the impact on yield for logic versus memory is different, we use CACTI [13] to estimate the percentage of L1, L2, and L3 area that is dedicated to SRAM cell versus peripheral circuitry, which enables us to accurately define the portion of die area that will significantly impact yield. Then, the yield is given by

$$Y_{\text{die}} = \left(1 + \frac{A_{\text{die, yield}} \cdot D_0}{\alpha} \right)^{-\alpha} \quad (3.11)$$

where $A_{\text{die, yield}}$ is the die area for yield calculation purposes, D_0 is the defect density, and α is the clustering factor [14]. Then, we estimate the number of dies that can be cut from a wafer as

$$\#Die_{\text{wafer}} = d \cdot \pi \left(\frac{d}{4 \cdot A_{\text{die}}} - \frac{1}{\sqrt{2 \cdot A_{\text{die}}}} \right) \quad (3.12)$$

where d is the wafer diameter (300 mm) and A_{die} is the die area [15]. Finally, the cost per die is given as $\frac{Cost_{\text{wafer}}}{\#Die_{\text{wafer}} \cdot Y_{\text{die}}}$ where $Cost_{\text{wafer}}$ is the wafer cost.

The cost of interposers, for systems using HBM memory, is similarly computed using Equations 3.11 and 3.12 with die area replaced with interposer area. The interposer area is the sum of the die area and the HBM DRAM stack area. We also include the cost of assembling the die and the interposer.

For the cost of the package, we first estimate the area of the package. The package area is similarly computed using Equations 3.8 and 3.9. As shown in Figures 3.1 and 3.2, for systems using either DDR or HBM memory, we assume a package to PCB bump pitch of 900 μm with a current rating of 250 mA [12]. For HBM systems, the $(\#Bump_{\text{MC}} \cdot N_{\text{MC}})$ term is excluded from Equation 3.8 since the memory interface wires connecting the HBM memory controller on the compute die with the HBM DRAM stack lie within the interposer without exiting the package. For HBM systems, the P_{die} in Equation 3.9 also includes the power consumption of the DRAM stack since the DRAM stack is on-chip. With the package area computed, the package cost is estimated by assuming the cost-to-area ratio of \$0.02 per mm^2 .

With memory prices from commercially available products [16, 17, 18, 19], we compute the total cost of the system as the sum of compute die, memory, interposer (for HBM systems), and package costs.

Chapter 4

Experimental Setup

With the analytical model described in Chapter 3, the next step is to establish the set of design points of interest to explore in the DSE.

The compute core parameters are based on the Intel Xeon Platinum 8380 CPU. Since we are interested in the performance and cost tradeoff between HBM and DDR memory, a high throughput CPU will ensure most design points will not be compute-bound. For the area of L1, L2, and L3 cache, we assume an L3 cache density of 0.5 MB/mm² with L2 and L1 cache being 1.5x and 4x less dense to account for the larger peripheral circuitry. We verified the area of the L1, L2, and L3 cache using CACTI [13] with cache organization parameters based on an Intel Skylake processor [20]. Since the Intel Xeon Platinum 8380 CPU uses a 10 nm process, we estimate the wafer cost based on TSMC’s pricing for 300 mm wafers [21].

For L3 cache, we consider L3 capacity from 2 MB to 200 MB with a step size of 2 MB (each L3 slice is 2 MB). For memory, we consider a number of DDR and HBM memories with different frequencies and channel counts (see Table 4.1).

Then, the set of design points is the full factorial combination of different L3 capacities, memory types, memory frequencies, and memory channel counts. We evaluate each of these design points under different combinations of arithmetic intensities and working set sizes (see Table 4.2). Arithmetic intensity determines the regions in which the system is compute bound

Table 4.1: Memory Types Considered for DSE

Memory Type	Frequency (MHz)	Channel Counts
DDR4	2400, 3200	4, 6
DDR5	4800, 5600	4, 6
HBM2	1000	4

Table 4.2: Application Profiles Considered for DSE

Arithmetic Intensity (FLOPs/byte)	0.125, 0.25, 0.5, 1
Working Set Size (MB)	25, 50, 100, 150

versus bandwidth bound. Working set size determines (i) the degree to which memory accesses from the core are filtered by L1 and L2 cache before being serviced by the memory subsystem and (ii) the hit rate of the L3 cache in the memory subsystem.

Chapter 5

Results

In this chapter, we describe the results of applying our analytical model to the set of design points. We start by examining the attainable performance of design points. Then, we incorporate the cost model to identify design points with reasonable performance and cost.

5.1 Compute Bound versus Bandwidth Bound

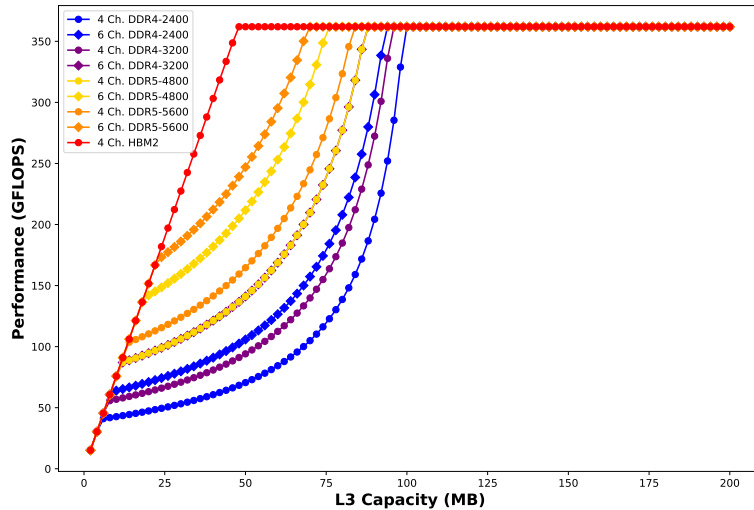


Figure 5.1: Line curve showing overall system performance versus L3 capacity. We show the curves for different memory types, frequencies, and channel counts. The plot is drawn for application arithmetic intensity of 0.5 FLOPs/byte and working set size of 100 MB.

Figure 5.1 shows the system performance of design points as a function of L3 capacity. A curve is drawn for a different combination of memory frequency and channel count. For the same L3 capacity, systems with higher memory frequency or more memory channels have better performance, as expected. Systems with HBM2 memory outperform, or have identical performance, compared to any DDR systems for any L3 capacity. To identify regions where the system performance is bounded by compute throughput, compute core-L3 bandwidth, or effective L3-main memory bandwidth, we have to examine the relationship between each of these quantities.

Figure 5.2 shows, as a function of L3 capacity, the bandwidth between compute core and L3 and the effective bandwidth between L3 and main memory for 4 channels of DDR5-4800 memory. The compute core-L3 bandwidth increases linearly as L3 capacity increases since adding additional slices of L3 cache increases the number of R/W ports. The effective L3-main memory bandwidth increases super-linearly because as L3 capacity increases for fixed working set size, the L3 hit rate increases, and the effective L3-main memory bandwidth has a $\frac{1}{1-x}$ dependence on the L3 hit rate (see Equation 3.2). However, the effective L3-main memory bandwidth stops increasing for L3 capacity larger than 100 MB because at that point the entire working set fits into the L3 cache.

Figure 5.3 shows, as a function of L3 capacity, the effective memory subsystem bandwidth and compute throughput also for 4 channels of DDR5-4800 memory (again, *memory subsystem* refer to the memory hierarchy consisting of L3 and the main memory). As shown in Equations 3.2 and 3.3, the effective memory subsystem bandwidth curve is the minimum of the two curves in Figure 5.2 multiplied by the arithmetic intensity. The compute throughput curve is constant since compute core parameters are not varied during the DSE. The minimum of the effective memory subsystem bandwidth and compute throughput curves gives the system performance curve for 4 channels of DDR5-4800 memory that was shown in Figure 5.1.

Based on these observations, we see that the system with 4 channels of DDR5-4800 memory is bounded by compute core-L3 bandwidth for L3 capacities from 2 MB to 12 MB. As we continue to add more slices of L3, the system performance becomes bounded by the effective

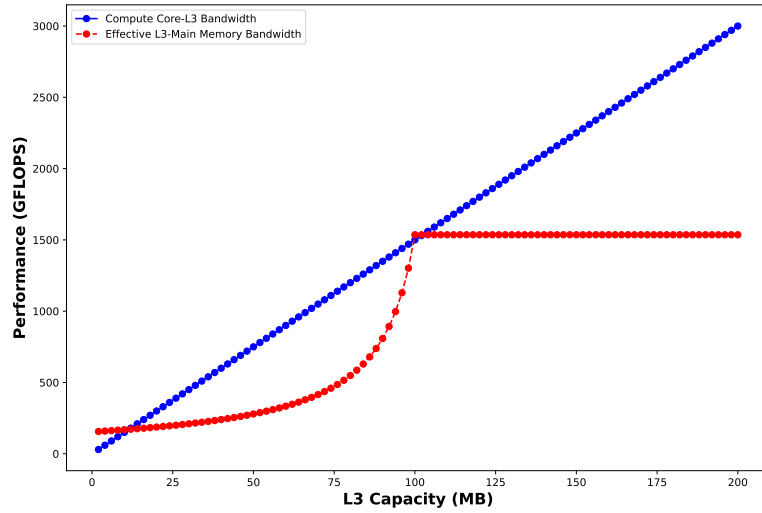


Figure 5.2: Line curve showing compute core-L3 bandwidth and effective L3-main memory bandwidth versus L3 capacity for 4 channels of DDR5-4800 memory. The plot is drawn for application arithmetic intensity of 0.5 FLOPs/byte and working set size of 100 MB.

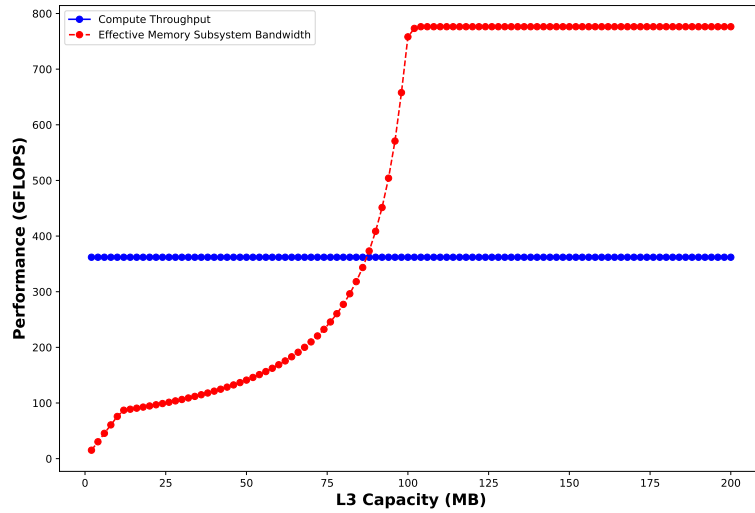


Figure 5.3: Line curve showing compute throughput and effective memory subsystem bandwidth versus L3 capacity for 4 channels of DDR5-4800 memory. The plot is drawn for application arithmetic intensity of 0.5 FLOPs/byte and working set size of 100 MB.

L3-main memory bandwidth. This is the case for L3 capacities from 12 MB to 88 MB. Further adding L3 slices increases L3 hit rate and effective L3-main memory bandwidth until the system performance becomes bounded by the compute throughput of the compute cores. This is true for L3 cache capacity greater than 88 MB. The relative positioning of the compute core-L3

bandwidth, effective L3-main memory bandwidth, effective memory subsystem bandwidth, and compute throughput curves change for different memory types, frequencies, and channel counts. In other words, for each memory configuration, the regions in which system performance is bounded by compute core-L3 bandwidth, effective L3-main memory bandwidth, and compute throughput differ. For instance, as shown in Figure 5.1, the system using 6 channels of DDR5-5600 memory has a smaller interval of L3 capacity during which the system performance is bounded by effective L3-main memory bandwidth since it has higher memory channel count and frequency.

5.2 Cost Composition

Next, we can understand the cost composition of each design point by examining the compute die, memory, interposer (for HBM systems), and package cost separately. We will compare the cost for systems with different memory configurations at iso-L3 capacity since the effect of adding an additional slice of L3 is the same across memory configurations. The effect of adding one additional slice of L3 is as follows. Compute die and interposer area (for HBM systems) increase marginally which lowers yield for compute die and interposer. It also decreases the number of compute dies and interposers that can be cut from a wafer. Consequently, compute die and interposer costs increase. Since a slightly larger L3 cache consumes more power, the area and cost of the package also increase marginally.

Figure 5.4 shows the compute die, memory, interposer, package, and total cost of systems with different memory configurations at iso-L3 capacity of 60 MB. We observe that the cost of compute die is greater for systems with six memory channels compared to four since the additional channels require additional memory controllers on the compute die, which takes up additional area on the wafer, reduces yield, and lowers the number of dies that can be harvested from a wafer. We see that memory cost also increases with the number of channels and frequency with HBM memory costing the most. Next, we consider the package cost. The package cost increases with the channel count since additional memory controllers require additional power

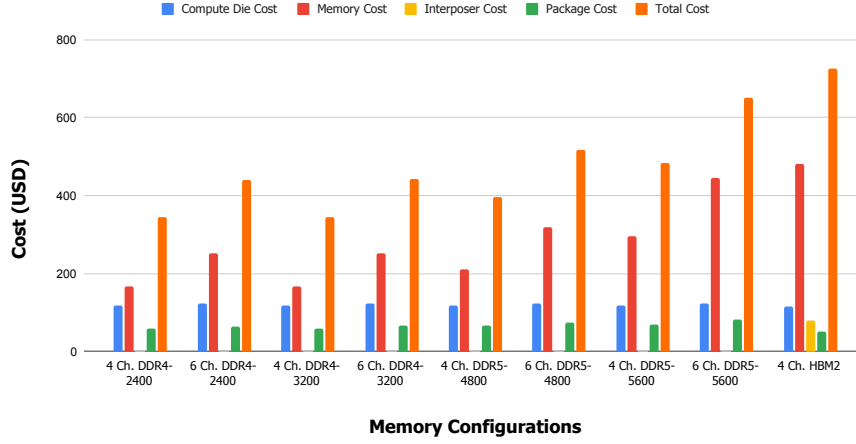


Figure 5.4: Compute die, memory, interposer, package, and total cost of systems with different memory configurations. All systems plotted at an L3 capacity of 60 MB.

and additional bumps on the package to transmit data to off-chip DRAM dies. The package cost also increases with frequency since the memory controller’s operating voltage scales linearly with frequency. We observe that the system with HBM2 memory has the cheapest package. First, HBM’s wide interface enables it to operate at a frequency lower than any DDR memories we considered while still achieving the highest bandwidth (we assume an operating frequency of 1000 MHz for HBM2). Second, since the HBM stack is on-chip compared to the off-chip DDR DIMM, the amount of energy required for the memory controller to signal a bit of data is approximately 4.29x less. Both of these factors reduce the power consumption of the memory controller and thus the number of package bumps required for power delivery. Third, the integration of the DRAM stack within the package means memory interface wires connecting the HBM memory controller with the DRAM stack do not exit the package, which means fewer package-to-PCB bumps for signaling. These three factors outweigh the fact that the package for HBM systems needs to include bumps for power delivery to the HBM DRAM stack. For instance, compared to the 4-channel DDR4-3200 system, the 4-channel HBM2 system uses 26 fewer package bumps for power delivery to the memory controllers (due to lower memory frequency and energy per bit), 640 fewer package bumps for connections to off-chip DRAM DIMM, and 60 more bumps for power delivery to the HBM DRAM stack.

Overall, the system using HBM2 memory has the highest total cost with compute die, memory, interposer, and package accounting for 15.77%, 66.14%, 10.96%, and 7.14% of the

total cost, respectively. The high cost of the HBM system relative to DDR systems is mostly attributed to (i) the inherently high cost of the HBM DRAM stacks which require the use of through-silicon vias and micro bumps compared to the widely available DRAM DIMMs and (ii) the high cost of interposer fabrication and assembly. In fact, in our HBM2 system with 60 MB of L3 cache, the interposer cost, including assembly, is approximately 69.49% of the compute die cost.

5.3 Cost-Performance Trade-off

Now that we have examined the performance behavior and cost composition of the design points, we can evaluate the cost and performance trade-off. Figure 5.5 shows overall system cost versus performance where each design point is represented by a marker on the plot. For a given memory configuration (represented as a unique combination of marker color and shape), the leftmost marker is the design point with 2 MB of L3 cache while the rightmost marker is the design point with 200 MB of L3 cache. The vertically stacked markers on the far right represent systems that are compute-bound, at which point adding additional L3 slices only

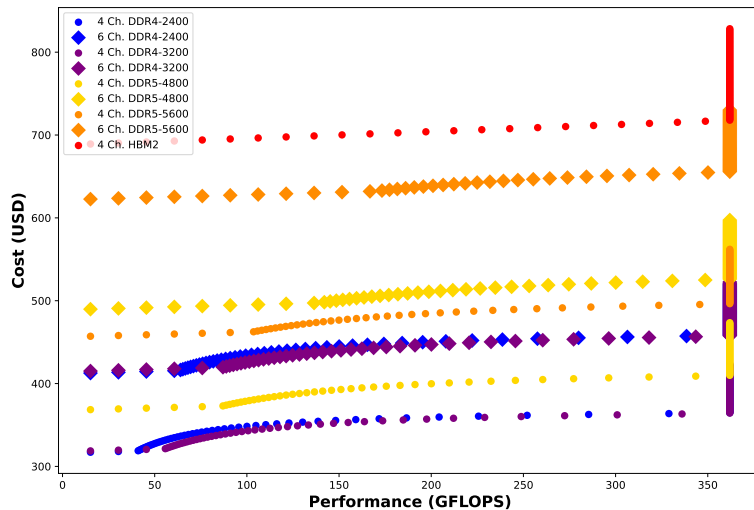


Figure 5.5: Scatter plot showing overall system cost versus performance. We show the data points for different memory types, frequencies, and channel counts. The plot is drawn for application arithmetic intensity of 0.5 FLOPs/byte and working set size of 100 MB.

increases cost without improving performance.

From Figure 5.5, we can see that the same performance can be achieved using different combinations of L3 capacity, memory type, memory channel count, and memory frequency with very different system costs. For instance, Table 5.1 shows, for each memory configuration, the L3 capacity required to achieve 200 GFLOPS of performance (under this specific application profile of 0.5 FLOPs/byte arithmetic intensity and 100 MB working set) and the associated system cost. Therefore, the optimal system cost configuration is to use 4 channels of DDR4-3200 memory with 82 MB of L3 cache. Other system configurations may not have the best performance and cost trade-off but may be superior in other metrics such as area and power consumption. For each system configuration in Table 5.1, Figure 5.6 compares the compute die area, package area, and compute die power consumption (excluding power of HBM DRAM stack) normalized against the system with 4 channels of HBM2 memory and 26 MB of L3 cache. From Table 5.1 and Figure 5.6, we can conclude the 4-channel HBM2 system with 26 MB of L3 cache, despite having the highest system cost, outperforms all other DDR systems in Table 5.1 in compute die area, package area, and compute die power. In the iso-L3 capacity cost comparison in Section 5.2, we explained why HBM systems have small package areas and low compute die power. Another major contributing factor, in this case, is that the HBM system requires a much smaller L3 capacity to achieve 200 GFLOPs of performance, which lowers area and power consumption. This makes HBM a high-cost, low-procurability solution for systems with strict power and area constraints. Although Table 5.1 and Figure 5.6 are based on a target

Table 5.1: L3 Capacity Required for 200 GFLOPS Performance and the Associated System Cost (Normalized against 4 Channel HBM2 System) for All Memory Configurations

Memory Configuration	L3 Capacity (MB)	Normalized System Cost
4 Ch. DDR4-2400	90	0.511
6 Ch. DDR4-2400	78	0.639
4 Ch. DDR4-3200	82	0.507
6 Ch. DDR4-3200	68	0.635
4 Ch. DDR5-4800	68	0.568
6 Ch. DDR5-4800	46	0.726
4 Ch. DDR5-5600	60	0.688
6 Ch. DDR5-5600	36	0.907
4 Ch. HBM2	26	1

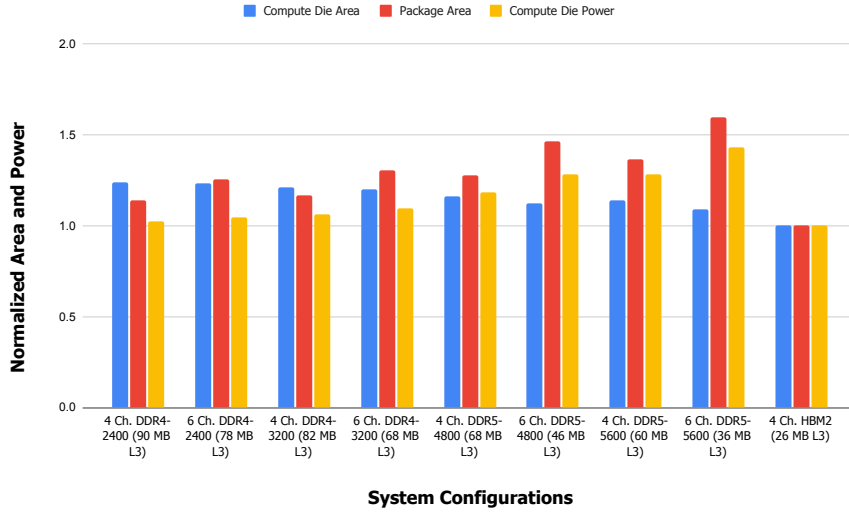


Figure 5.6: Compute die area, package area, and compute die power consumption (excluding power of HBM DRAM stack) for each design point shown in Table 5.1 normalized against the system with 4 channels of HBM2 memory and 26 MB of L3 cache.

performance of 200 GFLOPS, the same conclusion (the HBM system has a higher system cost but lower compute die area, package area, and compute die power than any DDR system in iso-performance comparison), in fact, generalizes to all performance thresholds shown in Figure 5.5.

Since HBM systems have lower compute die power consumption than DDR systems for iso-performance comparisons, we analyzed the energy operational expenditure (OPEX) to understand if HBM systems have lower lifetime costs despite the higher upfront costs. For each system configuration in Table 5.1, Figure 5.7 and Figure 5.8 compares the system cost, energy OPEX, and lifetime cost with an assumed energy cost of 0.05 and 0.2 USD/kWh, respectively. We also assume five years of operation in the systems' lifetime. The trend of OPEX follows the trend of the compute die power consumption, as expected. The lifetime cost of the 4-channel HBM2 system is the third and sixth highest for energy costs of 0.05 and 0.2 USD/kWh, respectively. In other words, given the current cost of HBM DRAM stacks, DDR DIMMs, interposers, and electricity, DDR systems running at a low frequency with a large L3 cache still offer better economics over the system lifetime.

Since the attainable performance of a given system configuration depends significantly on the application profile, namely the arithmetic intensity and working set size, we next see if the

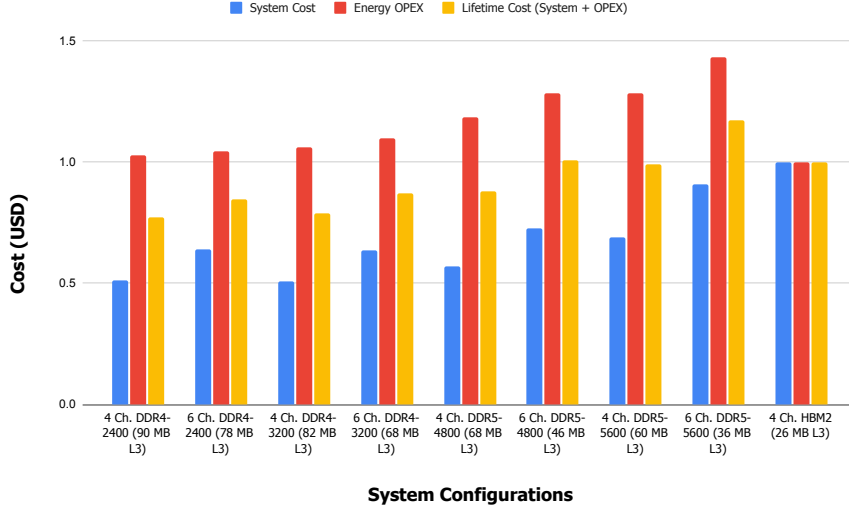


Figure 5.7: System cost, energy OPEX, and lifetime (system + OPEX) cost for each design point shown in Table 5.1 normalized against the system with 4 channels of HBM2 memory and 26 MB of L3 cache. Energy OPEX is estimated based on an energy cost of 0.05 USD/kWh and five years of operation. Only compute die power is considered.

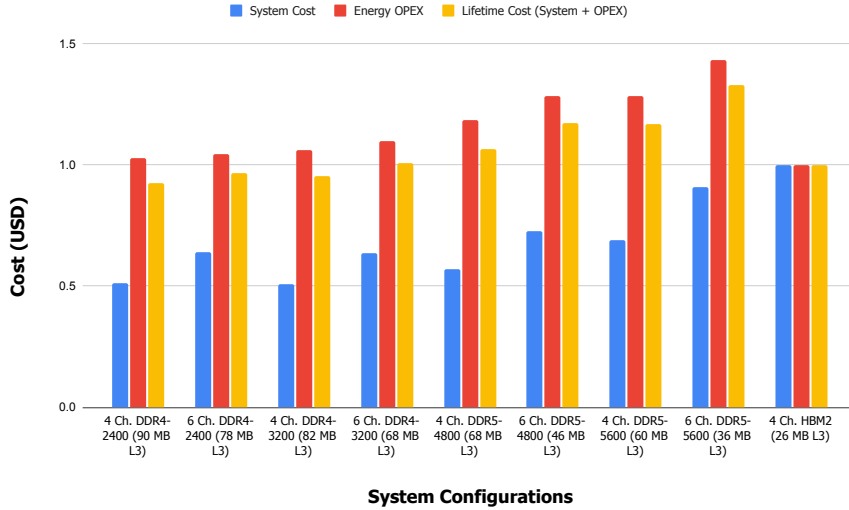


Figure 5.8: System cost, energy OPEX, and lifetime (system + OPEX) cost for each design point shown in Table 5.1 normalized against the system with 4 channels of HBM2 memory and 26 MB of L3 cache. Energy OPEX is estimated based on an energy cost of 0.2 USD/kWh and five years of operation. Only compute die power is considered.

same conclusion (the HBM system has a higher system cost but lower compute die area, package area, and compute die power than any DDR system in iso-performance comparison) generalizes to different application profiles. We reduce arithmetic intensity from 0.5 to 0.125 FLOPs/byte and increase the working set size from 100 MB to 150 MB to consider a more memory-intensive application profile. A larger working set implies a lower L3 hit rate and less filtering from

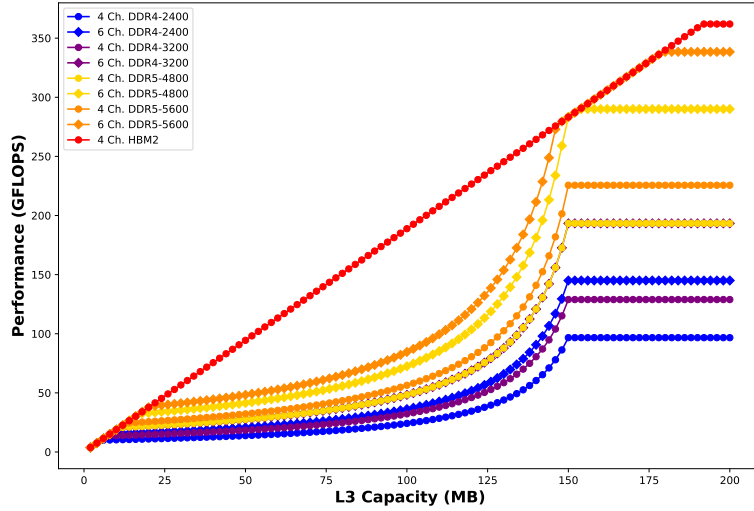


Figure 5.9: Line curve showing overall system performance versus L3 capacity. We show the curves for different memory types, frequencies, and channel counts. The plot is drawn for application arithmetic intensity of 0.125 FLOPs/byte and working set size of 150 MB.

L1 and L2 cache. Figure 5.9, similar to Figure 5.1, shows the system performance of design points versus L3 capacity. A line is drawn for a different combination of memory frequency and channel count. With a more memory-intensive application, only systems with 4 channels of HBM2 memory and more than 192 MB of L3 cache are able to fully utilize the compute throughput of the compute cores. For systems with any other memory configurations, the system performance eventually becomes bounded by the effective L3-main memory bandwidth. We also see alternating intervals of L3 capacity where the system performance is bounded by compute core-L3 bandwidth and effective L3-main memory bandwidth. For instance, the system with 6 channels DDR5-5600 memory is bounded by compute core-L3 bandwidth from 2 MB to 20 MB, bounded by effective L3-main memory bandwidth from 20 MB to 146 MB, bounded by compute core-L3 bandwidth from 146 MB to 178 MB, and finally bounded by effective L3-main memory bandwidth for L3 cache larger than 178 MB.

Figure 5.10, similar to Figure 5.5, shows overall system cost versus performance where each design point is represented by a marker on the plot. Similar to the case with 0.5 FLOPs/byte arithmetic intensity and 100 MB working set size, we observe that HBM systems have the highest system cost for all performance thresholds. Our model also shows that HBM systems have

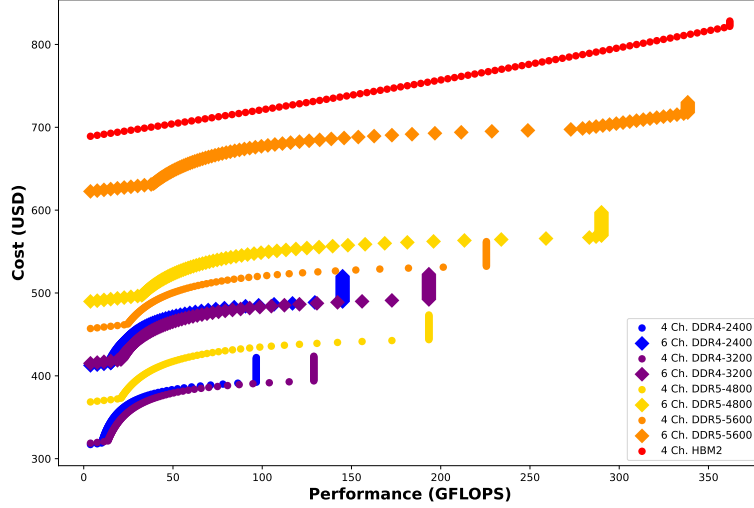


Figure 5.10: Scatter plot showing overall system cost versus performance. We show the data points for different memory types, frequencies, and channel counts. The plot is drawn for application arithmetic intensity of 0.125 FLOPs/byte and working set size of 150 MB.

the lowest compute die area, package area, and compute die power for all performance thresholds compared to any systems using DDR memory. However, for the more memory-intensive application profile, the key difference is that there are performance levels not achievable using DDR memories regardless of channel count, frequency, or L3 capacity, namely performance greater than 338 GFLOPS.

For completeness, we consider a compute-intensive application profile with arithmetic intensity of 1 FLOPs/byte and a working set size of 50 MB. Figure 5.11, similar to Figures 5.1 and 5.9, shows the system performance of design points versus L3 capacity. We see that most memory configurations are compute-bound for most L3 capacities. Figure 5.12, similar to Figures 5.5 and 5.10, shows overall system cost versus performance where each design point is represented by a marker on the plot. We see the same conclusion holds that the HBM system has a higher system cost but lower compute die area, package area, and compute die power than any DDR system in iso-performance comparison (die area, package area, and compute die power scatter plots not shown).

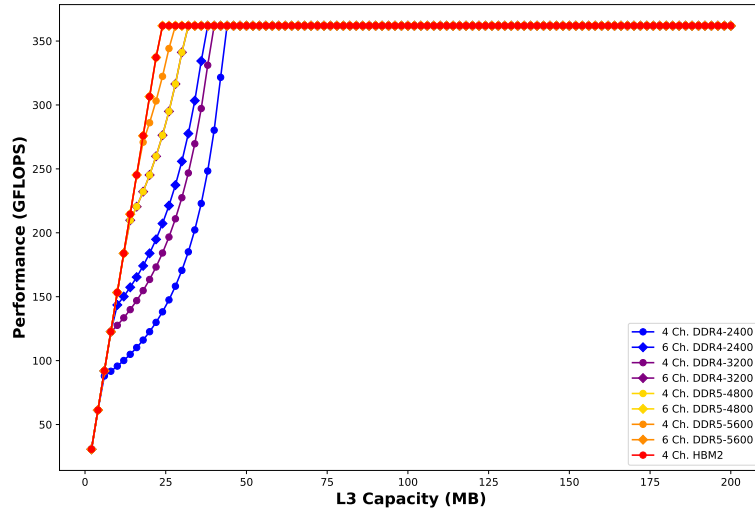


Figure 5.11: Line curve showing overall system performance versus L3 capacity. We show the curves for different memory types, frequencies, and channel counts. The plot is drawn for application arithmetic intensity of 1 FLOPs/byte and working set size of 50 MB.

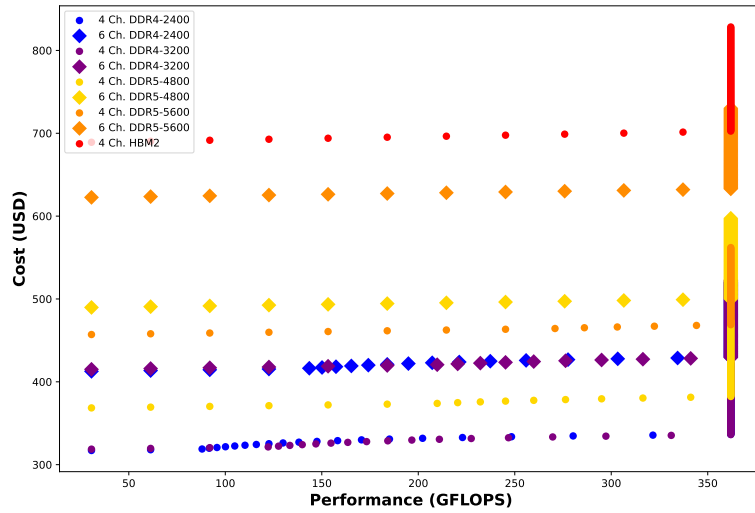


Figure 5.12: Scatter plot showing overall system cost versus performance. We show the data points for different memory types, frequencies, and channel counts. The plot is drawn for application arithmetic intensity of 1 FLOPs/byte and working set size of 50 MB.

Chapter 6

Conclusions

The unpredictable demand, complex supply chain, and high barrier of entry of the semiconductor industry make it prone to shortages. For companies from a wide range of industries relying on semiconductors for their products, designing with procurability and flexibility as a core architectural principle can be immensely valuable when shortages do occur. While HBM memory is a promising technology for applications requiring high DRAM bandwidth, it has lower procurability and higher cost compared to traditional off-chip DRAM DIMM due to the limited number of suppliers and the use of interposers.

In this study, applying our analytical model to design points with different L3 capacities, memory types, memory channel counts, and memory frequencies, we analyzed the performance characteristics of systems and identified regions where the performance is bounded by the compute core-L3 bandwidth, L3-main memory bandwidth, or compute throughput. On an iso-L3 capacity basis, we analyzed the cost composition of DDR versus HBM systems with HBM systems having the highest memory cost and total cost but the lowest package cost. Finally, combining performance and cost modeling, we identified design points where the same performance could be achieved at a lower cost by replacing HBM with DDR memory but with different combinations of a larger L3 cache, more memory channels, and higher memory frequency. Across a range of application profiles with different bandwidth requirements, we have shown that HBM systems have the highest system cost but the lowest compute die area,

package area, and compute die power in iso-performance comparisons. OPEX analysis at different energy cost levels also shows that, despite the lower compute die power of the HBM systems, DDR systems operating at a low frequency with a large L3 cache still offer better lifetime costs.

Bibliography

- [1] R. Kumar, “We’re going to see another chip shortage— despite the chips and science act,” *Fortune*, March 2023. [Online]. Available: fortune.com/2023/03/11/chips-and-science-act-semiconductor-shortage-rakesh-kumar/
- [2] P. Hanbury, B. Radzevych, and B. Grant, “Engineering your way out of the global chip shortage,” *Harvard Business Review*, December 2021. [Online]. Available: hbr.org/2021/12/engineering-your-way-out-of-the-global-chip-shortage
- [3] JEDEC, “Jedec standard 235: High bandwidth memory (hbm) dram,” JEDEC, Tech. Rep., October 2013.
- [4] D. Howley, “These 169 industries are being hit by the global chip shortage,” *Yahoo Finance*, April 2021. [Online]. Available: finance.yahoo.com/news/these-industries-are-hit-hardest-by-the-global-chip-shortage-122854251.html
- [5] M. Wayland, “Chip shortage expected to cost auto industry \$210 billion in revenue in 2021,” *CNBC*, September 2021. [Online]. Available: cnbc.com/2021/09/23/chip-shortage-expected-to-cost-auto-industry-210-billion-in-2021.html
- [6] D. Abril, “Computer monitors and other work-at-home essentials are in big demand in the coronavirus era,” *Fortune*, March 2020. [Online]. Available: fortune.com/2020/03/16/work-from-home-tech-products-sales/
- [7] K. Lyons, “Us tightens trade restrictions on chinese chipmaker smic,” *The Verge*, September 2020. [Online]. Available: theverge.com/2020/9/26/21457350/us-tightens-trade-restrictions-china-chipmaker-smic

- [8] “Crypto-miners are probably to blame for the graphics-chip shortage,” *The Economist*, June 2021. [Online]. Available: economist.com/graphic-detail/2021/06/19/crypto-miners-are-probably-to-blame-for-the-graphics-chip-shortage
- [9] E. Barrett, “Taiwan’s drought is exposing just how much water chipmakers like tsmc use (and reuse),” *Fortune*, June 2021. [Online]. Available: fortune.com/2021/06/12/chip-shortage-taiwan-drought-tsmc-water-usage/
- [10] JEDEC, “Jedec standard 212c.01: Graphics double data rate (gddr5) sgram standard,” JEDEC, Tech. Rep., September 2022.
- [11] T. P. Morgan, “What faster and smarter hbm memory means for systems,” *The Next Platform*, July 2021. [Online]. Available: nextplatform.com/2021/07/21/what-faster-and-smarter-hbm-memory-means-for-systems/
- [12] S. Pal, D. Petrisko, A. A. Bajwa, P. Gupta, S. S. Iyer, and R. Kumar, “A case for package-less processors,” in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2018, pp. 466–479.
- [13] S. Wilton and N. Jouppi, “Cacti: an enhanced cache access and cycle time model,” *IEEE Journal of Solid-State Circuits*, vol. 31, no. 5, pp. 677–688, 1996.
- [14] S. Pal, D. Petrisko, R. Kumar, and P. Gupta, “Design space exploration for chiplet-assembly-based processors,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 4, pp. 1062–1073, 2020.
- [15] AnySilicon, “Die per wafer (free) calculator.” [Online]. Available: anysilicon.com/die-per-wafer-formula-free-calculators/
- [16] Crucial, “Crucial 16gb ddr4-2400 udimm.” [Online]. Available: crucial.com/memory/ddr4/ct16g4dfd824a
- [17] —, “Crucial 16gb ddr4-3200 udimm.” [Online]. Available: crucial.com/memory/ddr4/ct16g4dfra32a

- [18] —, “Crucial 16gb ddr5-4800 udimm.” [Online]. Available: crucial.com/memory/ddr5/ct16g48c40u5
- [19] —, “Crucial 16gb ddr5-5600 udimm.” [Online]. Available: crucial.com/memory/ddr5/ct16g56c46u5
- [20] WikiChip, “Skylake (server) - microarchitectures - intel,” May 2017. [Online]. Available: [en.wikichip.org/wiki/intel/microarchitectures/skylake_\(server\)](https://en.wikichip.org/wiki/intel/microarchitectures/skylake_(server))
- [21] S. M. Khan and A. Mann, “Ai chips: What they are and why they matter,” *Center for Security and Emerging Technology*, pp. 44–45, April 2020.