Designing Multi-socket Systems Using Silicon Photonics

Scott Beamer
Krste Asanovi
Chris Batten
Ajay Joshi
Vladimir Stojanovic

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2009-9
http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-9.html

January 21, 2009

Designing Multi-socket Systems Using Silicon Photonics

1 Introduction

Given the difficulties of scaling uniprocessor performance further, most commercial microprocessor manufacturers have instead used increased transistor densities to integrate multiple processor cores on one die [1]. These manycore systems will require increasing memory bandwidth at reasonable energy consumption if they are to deliver improvements in application performance.

Silicon photonics is a promising new technology that may provide considerable bandwidth density and energy efficiency advantages compared to conventional electrical signalling, especially for off-chip connections to memory. Extending an off-chip photonic connection to replace on-chip electrical interconnect can also provide significant additional benefits in a manycore memory system, since most of the energy and latency cost of a photonic connection is at the endpoints [3]. In this paper we explore another benefit of photonics — flexibility in system packaging. The latency and energy of a photonic link is much less sensitive to distance than conventional electric links, and a waveguide can support very high bandwidth density. By lowering the penalty for inter-chip communication, silicon photonics decreases the incentive to integrate. Instead of building multiprocessors with large dies containing many cores, we can implement a large multiprocessor using multiple smaller dies connected with photonics (Figure 1). The benefits include: lower total silicon cost from improved yield; support for a scalable family of multiprocessors built from a single die; lower-cost printed circuit board (PCB) design; and reduced board-level power density.

Decreasing the die size provides a large reduction in cost due to increased yield. Figure 2 shows the relative costs of manufacturing $400\,\mathrm{mm}^2$ of silicon as one whole die or many smaller dies. Although the combined cost of the smaller dies is always cheaper due to increased yield, most of the gain can be had by splitting 4 ways to get a $\approx 3\times$ cost advantage. Figure 2 is from a simple

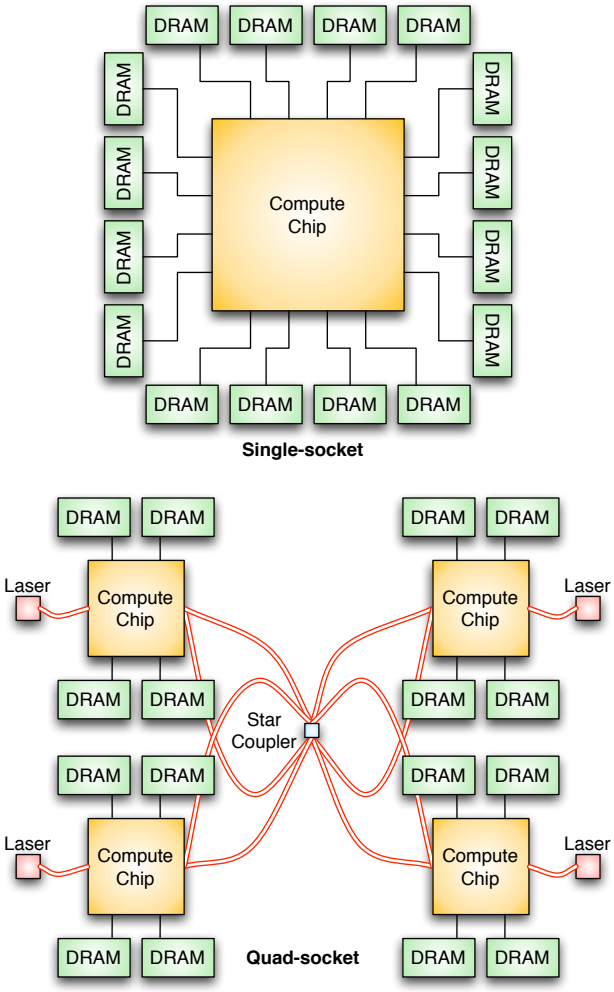

Figure 1: **Same silicon area as one socket or spread over four sockets**

model [7] and only takes into account parameters for area and defect densities. For real world costs there will be also fixed costs (packaging, assembly, and test) per die that will make the systems with the smallest dies less desirable, but there still will be significant advantage to using multiple moderately smaller dies rather than a single large die.

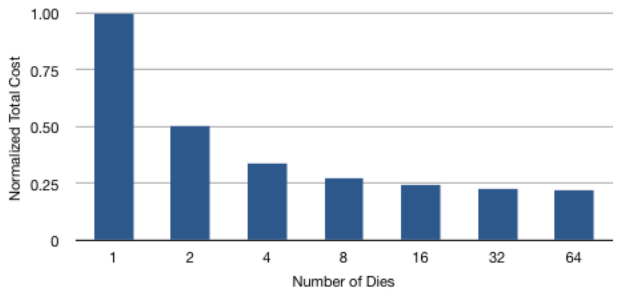

Figure 2: **Relative total costs for 400 mm^2 of total silicon area**

Breaking the single-socket system into multiple smaller sockets also allows a manufacturer to offer a range of systems. Reusing the same compute die in many different system sizes will allow greater amortization of non-recurring engineering (NRE) costs over the increased manufacturing volume. A scalable multi-socket system composed of small dies may also help reduce the impact of process variation. Since each die is smaller, the probability of there being high variance on a die is reduced. Moreover, since the dies are smaller they can be binned on a finer granularity, allowing a greater number of high performance cores to be sold.

A photonic multi-socket design can also be used to scale systems much larger than is practical for electrical multi-socket systems. With electrical interconnect, system bisection bandwidth per core drops off sharply as system size scales due to power and pin limitations. This forces architects to implement a non-uniform memory architecture (NUMA) that requires performance programmers to worry about the system topology, and place data in the memory attached to the same socket as the cores that will access it where possible. Providing a flatter and more uniform performance profile for the memory system will improve programmer productivity and also increase the range of parallel applications that can benefit from increasing core counts. Some multiprogrammed workloads, such as virtual machines running within a datacenter, will also benefit from the scheduling flexibility that bandwidth uniformity provides. When scheduling jobs, a job could be run on the first available

core independent of where the data it needs resides.

In the rest of this paper, we explore the design of a photonic multi-socket system that is optimized to reduce system cost and overall power consumption while providing increased and uniform memory bandwidth.

2 Photonics Background

We begin by reviewing the properties of photonic interconnect. In this paper we select the monolithic technology presented in [3] and explore some of the tradeoffs photonics presents to the system architect. Although we base the rest of this study on this particular monolithically integrated silicon photonics technology, we believe the overall approach is applicable to other proposed photonic technologies.

2.1 System Overview

Silicon photonics has been shown to offer improvements in many important interconnect metrics including energy efficiency, bandwidth density and latency. But the technology is still immature, with many competing implementation proposals, and so projected performance on these important metrics varies significantly. Essentially silicon photonics uses photons to transmit data rather than electrons, and as shown in the Figure 3, a link is comprised of a light source, a modulator, a waveguide, and a photodetector. The light travels down the waveguide past the modulator which does or does not absorb light to encode the signal. At the other end of the waveguide, the photodetector senses the changes in light and decodes the signal. Photonics excels over longer distances, but the conversions at the endpoints (electro-optical and opto-electrical) introduce a latency and energy penalty that limits the minimum distance over which it will be advantageous.

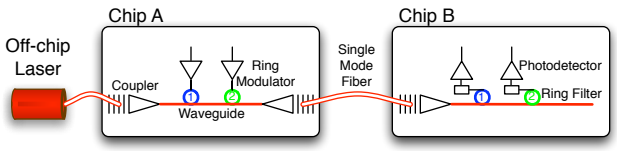

Figure 3: **An inter-socket photonic link**

One nice feature of the selected technology is its ability to be integrated into a current CMOS manufacturing process which makes it much more realizable since there already exists a great deal of manufacturing hardware investment and knowledge about it. Other photonic proposals may be better suited for transmitting light, but they use materials or steps not currently part of a standard CMOS process making them more cost prohibitive to implement. On-chip light travels in Poly-Si which can be made into a usable waveguide by etching an air gap underneath it [5]. The air gap takes up silicon area, so when possible multiple waveguides share one to amortize the overhead. When light needs to go off chip, it goes through a coupler into a strand of optical fiber with very low loss.

The selected technology also provides a bandwidth density advantage enabled by Dense Wave Division Multiplexing (DWDM). DWDM allows light from different wavelengths to share the same waveguide with minimal interference. This allows multiple logical links to share the same physical media without time multiplexing. This is done by putting rings which resonate with a narrow frequency of light onto the waveguide, and when the light resonates with a ring it is pulled off the waveguide. We can make a ring modulator using charge injection to change a ring's resonance to modulate the light. A filter can be made by using these resonant rings, and the selected technology uses two cascaded rings to get additional frequency selectivity. To use a photodetector with DWDM, a double ring filter is placed between it and the waveguide so only the correct wavelength gets through the filter and strikes the photodetector. These resonant rings are sensitive to a variety of environmental factors and manufacturing variations, but this can be combated by thermally tuning the rings with heaters.

2.2 Performance

Table 1 gives gives a rough comparison of photonic and electric links. The electric values are for a projected optimally repeated wire. On-chip photonic links need to go millimeters to be advantageous, but for off-chip it has a dramatic energy advantage. If the electro-optical and opto-electrical conversion cost (200ps and 150fJ) is going to be paid to send data off-chip, it also makes sense to perform the conversions near where the data is produced and consumed to save on on-chip

latency and energy. Our technology assumes a signaling rate of 10Gbps (faster could be possible) and squeezes in 64 wavelengths per direction [6], meaning a single link (fiber or waveguide) has 80GB/s of bidirectional bandwidth. On-chip waveguides take up more area than wires because of a wider pitch and the air gap, but there is so much more bandwidth per link it still obtains a bandwidth density advantage. Off-chip this advantage becomes more significant where they will have comparable pitches, but each photonic link is one fiber that is bidirectional rather than all the uni-directional pins needed to signal electrically.

Table 1: Approximate energy, latency, and area costs per bit

Link Type	On-Chip Energy	Off-Chip Energy	On-Chip Delay	Off-Chip Delay
Electric	$50\frac{fJ}{mm}$	$5000fJ$	$100\frac{ps}{mm}$	$50ps + 5\frac{ps}{mm}$
Photonic	$250fJ$	$150fJ$[1]	$200ps + 10\frac{ps}{mm}$	$200ps + 5\frac{ps}{mm}$

2.3 Tradeoffs and Design Considerations

One of the most often under looked issues when using silicon photonics is laser power. Most other work has not added it to the power total and the justification used is that it is off chip and thus does not contribute to the power density hotspots by the processors [11]. The laser power is how strong the laser needs to be to get through the worst case path with enough strength to be readable by a photodetector at the end. Losses tend to grow exponentially rather than linearly, so a reasonable design can quickly become unreasonable when it is expanded. The network layout and size can contribute to this greatly, so it is essential that the designer reduce loss to save power. It is also important to note that keeping with convention, throughout this work laser power is presented, but this is not the same as the amount of electrical power required to generate it. The rationale behind this is that laser light generation is an orthogonal area of research so converting it to electrical power might be misleading. A conservative estimate of future laser efficiency is 25%, so it can easily non-negligibly contribute to total system's electrical power.

Another important consideration is the non-linearity limit imposed by the Poly-Si waveguide.

[1] $100\frac{fJ}{b}$ (modulator) + $50\frac{fJ}{b}$ (receiver) + $80uW$ (power to thermally tune rings) + laser power

As the combined power of the light inside a waveguide grows, there is a non-linear increase in the amount of light that escapes. To combat the loss more laser power is used which results in even more loss, so its best to keep the total power for a waveguide within reasonable limits. Normally how many wavelengths can be put into a waveguide is set by the frequency selectivity of the photonic components used, but the number of wavelengths used per waveguide may also be set by the path loss which determines the power required per wavelength. The designs presented later in this study were made to have low loss, and they should be able to carry 64 wavelengths per direction without issue.

3 Architecture Description

3.1 Overview

To gain the most from the transistors provided by Moore's law we are envisioning a system comprised of many simple cores. Although these cores will be in order, they may contain small SIMD units to efficiently enable a good deal of numerical performance. Photonics provides many good opportunities, but it is usually not advantageous until it is over a sufficient distance. For this reason we will electrically join groups of 2-16 cores together by a shared L2 cache into clusters. Photonics will be used for the longer links between clusters and memory controllers. Each memory controller may actually be more than one controller or support multiple DRAM channels, but from the point of view of the rest of the system it is a single point of control and arbitration. The connection between a memory controller and the DRAM module will still be electrical. This is done for convenience, but there is no reason why it could not be done with a photonic link and doing so should not change the results of this study. Keeping it electrical allows for current and upcoming DRAM interface technologies to be used. We intend to use photonics to provide a balanced memory bandwidth of one byte per flop and our network is designed to provide uniform memory performance independent of its location. This will simplify programming for performance because there will be no NUMA effects. If desired, more or less bandwidth could be provided and our results should scale accordingly.

3.2 Network Topology

To connect the clusters to memory controllers, we use a fully connected point-to-point network that connects every cluster to every memory controller, but there are no direct connections between any two clusters or between any two memory controllers as shown in Figure 4. All-to-all networks are often avoided for moderate to large networks because of their quadratic growth rate, but silicon photonics increases the size of what is feasible because of its energy efficiency and bandwidth density advantages. DWDM partially enables this by providing the ability to pack many channels onto a few waveguides which simplifies layout and results in a large constant factor reduction in the area required.

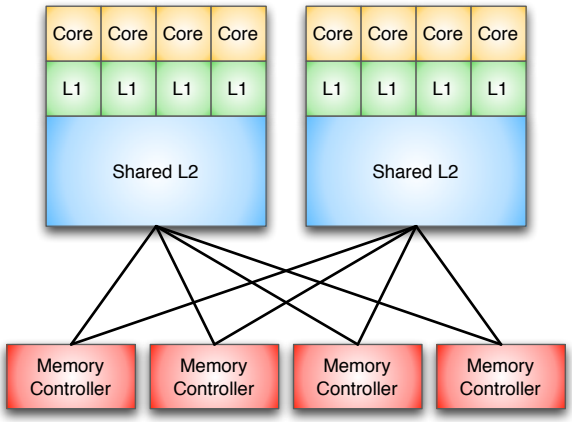

Figure 4: **Logical topology for 2 clusters of 4 cores and 4 memory controllers**

The bandwidth is uniformly provisioned such that every cluster has an equal share of each memory controller's bandwidth. Thus for a cluster to achieve its full bandwidth it must access each memory controller uniformly. Depending on how the interleaving is done across memory controllers, even very localized memory access patterns could be spread out across the memory system. We design our system to provide one byte per flop, so the total memory bandwidth for a cluster is independent of system size. This bandwidth will be supplied by as many logical links as there are memory controllers in the largest supported system. These links will be implemented without time multiplexing by using at least one wavelength if not more. Increasing the maximum supported size will increase the number of logical links which will decrease the bandwidth of each

one. Although the total bandwidth offered to each cluster will be the same, this will result in higher serialization latencies. Using more cores per cluster can improve this since it will give more bandwidth to each cluster and thus more bandwidth per link. Systems that are not fully populated will have multiple logical links between each memory controller and cluster. It might be possible to combine these links to decrease serialization latency.

4 Physical Design

4.1 System Objectives

To ground our argument we present a feasible system that is designed to be scaled up to a maximum size of 1024 cores. In the $22nm$ process using $400mm^2$ it could be possible to build a 256 core system running at 2.5GHz. Each core will have a 4-way SIMD unit with fused multiply adds generating a total of 8 flops/cycle/core. To provide the memory bandwidth there will be 16 memory controllers on chip. We assume clusters along with their caches will form tiles and they will be laid out in a grid on chip. For our designs and analysis we will make the cluster size 8 since it supplied the tolerable channel bandwidth of 8b/cycle when the system is designed for 1024 cores with 64 memory controllers and it has been shown feasible by current systems [9]. Even larger clusters could be feasible and will further reduce serialization latency. For most of the layouts presented later, cluster sizes of 4 or 16 could work without any significant re-routing or restructuring, and the change will only affect the serialization latency.

4.2 Physical Packaging

To simplify manufacturing, we use a star coupler as a hub to provide the needed connectivity between chips without the quadratic number of connections that is normal for fully connected networks. Each die connects to the hub with two ribbons of fibers as shown in Figure 1. One ribbon contains fibers that connect to all the clusters on that die and the other contains fibers that connect to the memory controllers. As shown in Figure 5, all of the cluster ribbons attach to one side of the coupler, and all of the memory controller ribbons attach to the other. The ribbons

from both sides come in orthogonal to each other so each ribbon crosses every other ribbon. In the example shown, 4 ribbons of 4 fibers come in each side, so effectively it is as if there is a fiber between every die including itself (one fiber gets looped back by this). In cases where there are more fibers in the ribbon than ribbons coming in, the same style of connector can be used by making some or all of the ribbons multi-row. Taking the same simplified example, but in the case where only 2 sockets are installed results in 2 ribbons of 4 fibers coming in on both sides. This can be fixed if on one side of the coupler both dies use 2 rows (of 2 fibers) per ribbon and continue to use the same one row ribbon on the other side. Another advantage of using this connector is that it should be comparably inexpensive and along with some of the ribbons are the only things to change between different system sizes.

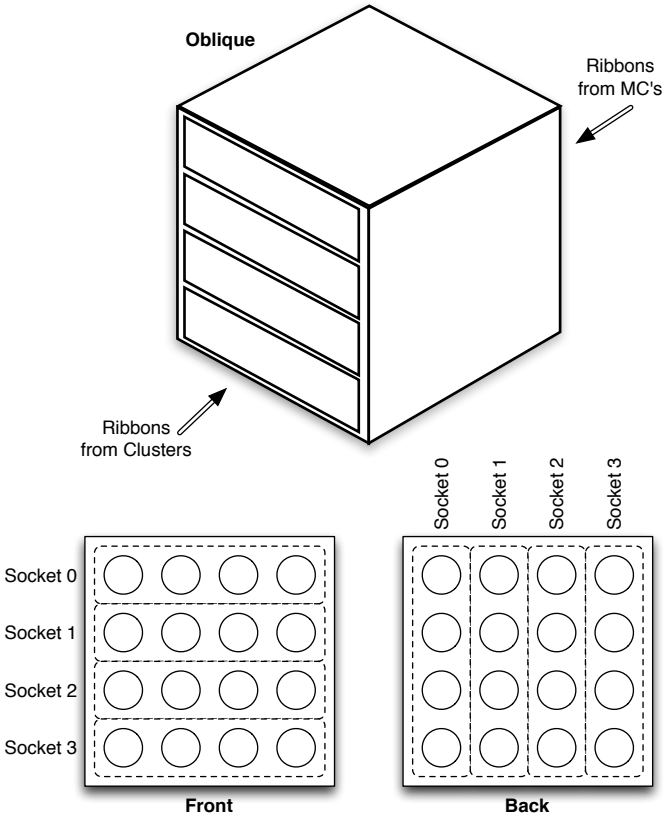

Figure 5: **Star coupler for 4 sockets**

For physical layout, the star coupler will be surrounded by identical compute dies that contain

the clusters and memory controllers as shown in Figure 1. Each of the compute dies is surrounded by its own locally attached DRAM to reduce the electrical links between them. The memory controllers are evenly spaced around the edge of the die to provide the easiest exposure for wiring to DRAM. The ribbons are attached only at the endpoints by vertical couplers and the ribbons will float freely beneath the board (Figure 6), so they can avoid the heat sinks of the compute dies. A more dense board layout might reduce ribbon lengths, but it could significantly complicate the much more costly electrical signaling to DRAM or increase the power density. Extra distance in the ribbon is tolerable since the optical power loss is practically non-existent and the increase in delay is marginal.

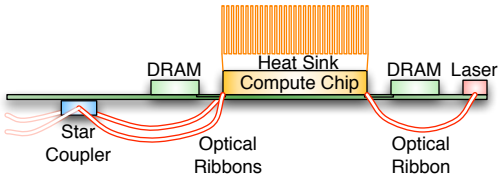

Figure 6: **Partial board layout from side**

4.3 Photonic Structures

For our designs we will need a few more photonic structures than were presented in the background. When using small dies as the basic building block, there is a need for what could best be described as a wavelength splitter. Essentially the component will split the wavelengths of one waveguide over n waveguides such that each output waveguide gets $\frac{64}{n}$ wavelengths. This component is bidirectional, so from one direction it looks like a splitter but from the other it looks like an aggregator. As shown in Figure 7 this can be done without crossings. Alternatively the layout could simply under-fill the waveguides on chip, but this wastes area and the optical loss through this structure will be low.

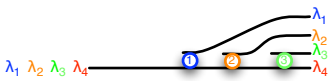

Figure 7: **Wavelength splitter**

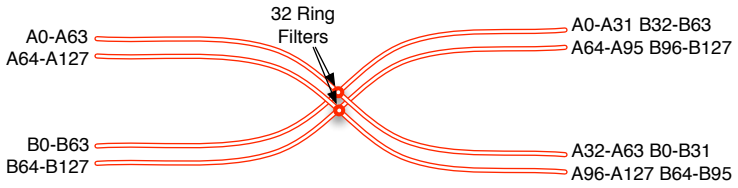

Figure 8: **2x2 mixer (2 wide)**

There are other times when we want to "mix" groups of waveguides. For the designs we use we only need to mix two groups, but each group is 4 or 8 waveguides wide (Figure 8 shows it for 2 wide). For each input group, half of the input wavelengths will end up on each of the output groups.

4.4 Layouts

Responding to our technology selection, we attempted to design our layouts to minimize path loss without using an unreasonable amount of area. We designed layouts with 16, 32, 64, 128, and 256 cores per die to support systems with 64, 128, 256, 512, and 1024 cores total and a representative one is shown in Figure 9. We used the same style of layout for every design but each one was human optimized. For some sizes and situations a different style might be better, but we used the same for all because it is practical for every size and it keeps the study more fair. The challenge for the die layout is to supply at least one fiber for every other die in the largest supported system while ensuring every cluster will be able to reach every memory controller.

The biggest losses our designs attempted to reduce were: waveguide crossings, ring filters, and waveguide length. Any reduction in loss is typically doubled, since after leaving one die and going through the coupler, it must face a similar path on the other. To keep waveguide distances short, we placed the power fiber on one side and the data ribbons on the other so waveguides go across chip only once. To reach items on the edge of the die, a waveguide might have to travel in a U-shape, but this path is no longer than twice the die width. The waveguides were also routed to minimize disruption to the logic circuits, so they were routed between compute tiles or along the edge of the die. Our larger designs (128 cores/die and 256 cores/die) have multiple power fibers

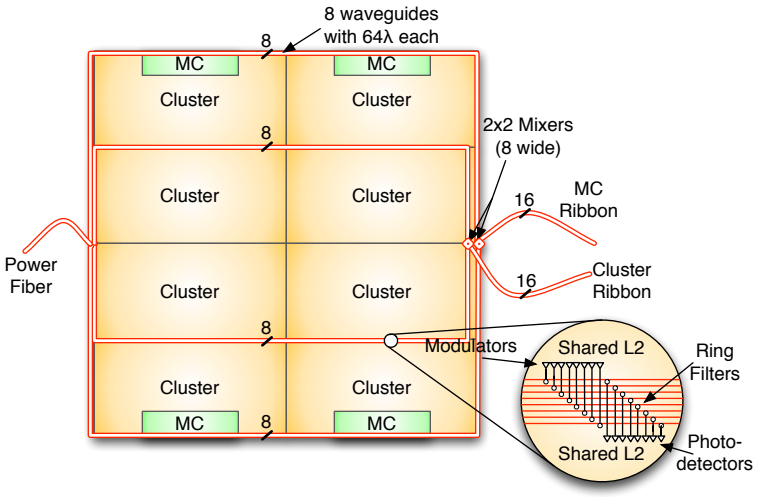

Figure 9: **Layout of a 64 core die than can support a system with at most 1024 cores**

coming in and this also helps to reduce waveguide length. More power fibers could reduce lengths for all the designs, but it comes at the cost of additional manufacturing complexity and laser light is more efficiently generated in modest amounts.

Figure 9 shows the same style that all the designs have of routing the memory controller waveguides on the outside around the cluster waveguides to avoid crossings. A key insight that enabled many of our designs was to spread a cluster's wavelengths over multiple waveguides as they pass the cluster rather than concentrating them all onto a few waveguides and then being forced to mix them. The waveguides already have to travel the distance, but loading them this way saves on crossings and rings later on. This can be seen in Figure 9 where 4 clusters are passed by 8 waveguides, so each cluster uses a quarter of the wavelengths of each waveguide rather than two whole waveguides exclusively.

The 64 cores/die and the 32 cores/die designs need to use the mixers described in 4.3 because they have 2 groups of on chip waveguides but only one fiber per inter-die link. The 16 cores/die and the 32 cores/die designs need to use the wavelength splitters because there are more dies than on chip waveguides, so the wavelengths on chip will need to be fanned out to enough off chip fibers to use the star coupler. The smallest dies may be impractical to use for the largest systems because of manufacturing and assembling complexity but the technology is still too far off to quantify this

cost. For systems with two sockets whether by design or under-utilization, the star coupler is unnecessary and can be replaced with direct ribbons between the two sockets. When a system has only socket populated but is designed for more, a single ribbon is used to loop back.

5 Analysis

In the last section we presented a general architecture, and in this section we will analytically examine how it meets our goals of reusable scalability. There are tradeoffs when choosing the base building block for the system, both in terms of how big it is and how many other blocks it expects. If the maximum system size is designed too small it will not be able to scale to larger systems without penalties, but if it is designed too large, the functionality needed for larger systems will waste area and raise cost when used in smaller systems. Some places where this tradeoff becomes apparent are: off-chip bandwidth, off-chip link organization, and coherency. For our particular family of designs, how populated the system is does not noticeably affect performance once the die size and the maximum system size have been set. Throughout this analysis, it is important to note that the values are strongly influenced by the selected silicon photonic technology and the layouts used to make our systems, but the most important thing to notice is that using smaller dies should perform no worse.

5.1 Power

Power is consumed by our network in three places: the send/receive circuitry, the ring heaters, and the laser. Figure 10 shows a breakdown of where the power is consumed in the network for 1024 cores built from 16 64 core dies. The laser power and ring tuning power make up the static portion, as reduced or increased traffic will not noticeably change them, but the send/recieve power will scale down as utilization decreases. For our analysis we use the impractical 100% utilization to show what the peak power could be. Although laser power is the majority consumer, this power is burned in off chip lasers so it adds to the system wall power but not to the compute die's power density. For the above comparison, we converted laser power to the amount of electrical power required to generate it assuming a conservative laser efficiency of 25%.

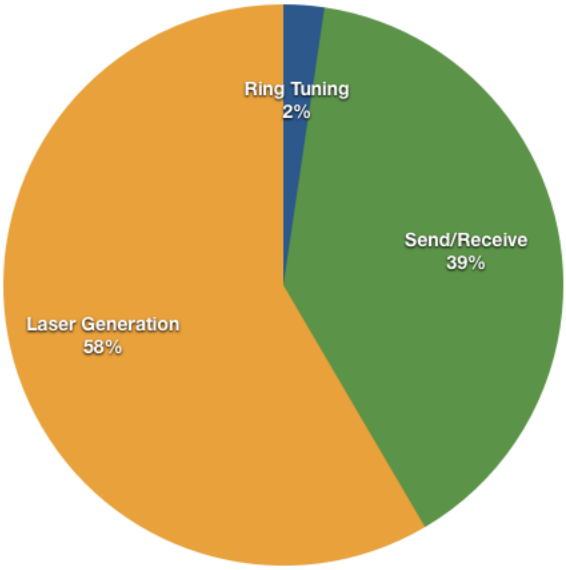

Figure 10: **Breakdown of Total Power for 1024 cores built from 64 cores/die**

Since we keep the bandwidth per core constant, for a given system core count the power for the send/recieve circuitry will remain constant as it is spread across more dies. This power includes the clocking, modulation, and SERDES required to encode/decode the data but not any buffering. For 1024 cores, this amounts to 24.576W total (24mW per core), which will be spread out over at least 4 dies. This power will scale directly with the number of cores in the system.

The power to heat the rings to keep them tuned to the correct wavelengths is mostly set by the number of cores since each core needs a constant number of them to send and receive, however some of the smaller die sizes use additional rings for filters in the interconnect and these also need to be tuned. This power is continuously burned but is not a large overall contributor.

The largest power consumer and the one most sensitive to layout is the optical power. Figure 11 shows the laser power required as a function of die size and maximum system size. For all die sizes as the maximum supported system size is reduced, the required optical power is also reduced, as would be expected. The rate at which it decreases can fluctuate significantly because as the system size is reduced, some components (2x2 mixer, waveguide splitter, star coupler) can be eliminated from the interconnect and the loss rates of these components varies. A more interesting trend is that smaller systems are more efficiently constructed from smaller dies as is visible on the pareto-optimal

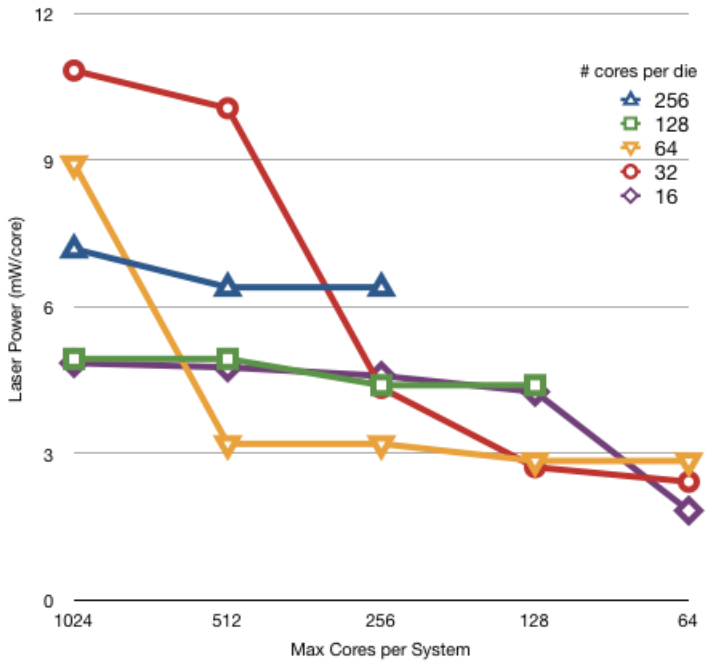

Figure 11: **Laser power per core**

curve (underside of the graph). This appears to indicate that systems with a moderate number of sockets perform best because of the fan-out costs associated with making all to all connectivity. With our selected technology, smaller dies have an advantage of shorter waveguides (less loss) as shown by 16 cores/die. Systems that are not fully populated should require the same laser power per core except for when only 1 or 2 sockets are populated and the star coupler is not needed.

5.2 Area

In general our photonic interconnect fits well within an area budget as shown in Figure 12. Since our technology is using projected values, these overheads could change, but we were pessimistic in our assumptions about sizing resulting in over-estimates for area. Smaller dies use less area for the interconnect, because more of it is off chip and they are small enough that it is still possible to put many or all waveguides over the same air gaps. Although this suggests another reason smaller dies will be more cost-effective is less wasted area, the most important result is that using smaller dies is no worse than using larger ones.

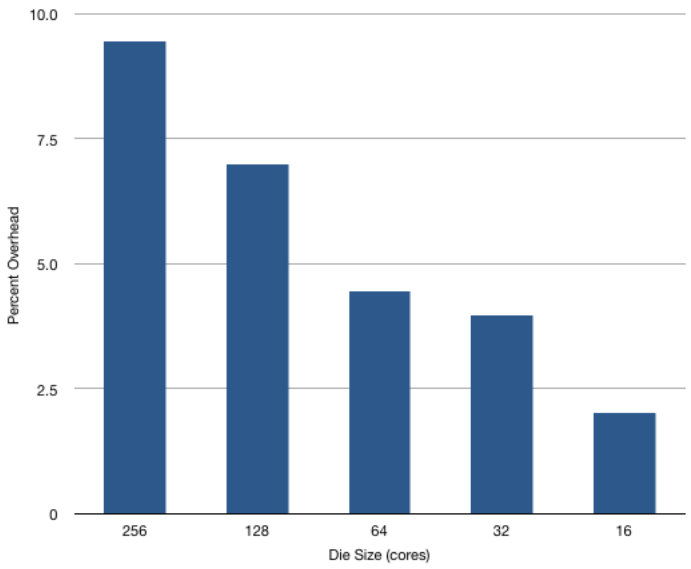

Figure 12: **Percentage die area taken by photonic network**

5.3 Latency

Surprisingly latency does not seem to be too much of an issue for breaking apart sockets, even if electrical links are used off chip. As visible back in Table 1, both technologies get faster off chip after a minimum distance has been traversed to make up for the conversion delay. Splitting a system up into multiple sockets might even make it faster when going across chip because the signal will spend more of its time off chip in a faster medium. Once the overhead of getting onto a fiber is paid, the signal can travel 8cm in a clock cycle of our baseline system, so within less than a few cycles, everything is reachable to everything else on board. The only time link latency is worrisome is when trying to route a signal for a long distance electrically with a normal repeated wire on chip, but this does not happen in our design since all long links are done photonically. Faster wires could be used but at the cost of area and power.

6 Coherency Scheme

To make this system more realizable it will need a coherency scheme (protocol and hardware implementation), which is something past designs have not given much consideration to. Especially

for the general architecture presented in this paper, it is essential that the coherency scheme achieve the goals of reusability and scalability. We want the same design to be able to handle different binary amounts of populated sockets in the system wihout unreasonable overheard. Our system uses shared memory, and coherency is maintained amongst all caches by a two level protocol corresponding to within and between clusters.

6.1 Intra-Cluster Coherence

Within a cluster, each core has its own private L1 cache and they all communicate through a shared L2 cache. The L2 cache is not inclusive of the L1's, but it does store duplicates of the tags. We envision using this with a protocol similar to what was described in Piranha [2]. This protocol will be responsible for keeping the caches within each cluster coherent, and requests it can not handle will be passed up to the next level coherency protocol.

6.2 Inter-Cluster Coherence

To maintain coherence between clusters we use a MESI directory protocol. From the point of view of the directory, all caches in a cluster are lumped together and treated as one. We position a directory by every memory controller so it can intercept requests to memory and take the appropriate protocol actions. A directory is only responsible for the memory locations its associated memory controller provides. To make the directory small enough to fit on chip rather than off-chip DRAM, we use a reverse tagged directory implemented with Content Addressable Memory (CAM). For every cache line it is responsible for, the directory contains a duplicate of the cache tag and a few bits of protocol state. We reduce the associativity required for the directory by implementing it with many small CAM's where each one corresponds to a cache set. When a request is being looked up, only the CAM corresponding to the request's set needs to be examined. A cache tag's location in the reverse directory implicitly identifies the location of its owner. Because all the caches in the system are set associative, this puts a limit on the number of number of possible cache lines that could hold a block, namely Nk if the system has N clusters and each one is k-way set associative. If this associativity is still too high, multiple passes could be used which will still be faster than going to

a direct mapped directory implemented by off-chip DRAM.

To support a variable number of populated sockets the way memory addresses are interleaved can be leveraged. For a given die size, if the number of populated sockets is doubled, the number of cache lines double, however the number of sets per cache that can address a particular memory controller get halved, so the number of possible locations a directory needs to be concerned with stays the same. The only thing that changes is the implicit addressing of clusters to tags in the reverse directory.

Although photonics provides great bandwidth which might tempt one to snoop, the energy cost at the endpoints to do associative lookups for every message at every cluster in the system will be prohibitive, especially as it scales. This will also require a broadcast mechanism, which our current network topology does not provide. It could be possible to design it, but our topology was designed to minimally meet our goals and our coherency protocol works well without it. The bandwidth savings a directory protocol provides will also help the system scale to higher core counts and conserve energy.

7 Related Work

The work of Batten et al. [3], identified the potential for monolithic silicon photonics for making an interconnection network to connect DRAM to processing cores. We used their technology assumptions and baseline machine as a starting point for our work. Our work differs in that it adds the contributions of using multi-socket systems as a way to reduce cost and considering coherence much more closely.

Kirman et al. presented a photonic on-chip interconnect in [8]. Their architecture attempted to utilize each interconnection topology for the range of distances it was best at. They subdivided a CMP into four blocks and those four blocks were connected by a photonic ring topology. Within a block electrical interconnects were used at a distance where they were advantageous to optical. Our network topologies were influenced by this but we have made a more ambitious design that uses a more optimistic photonic technology.

In [10] a photonic NoC is presented for a multiprocessor system uses photonic switches built

from crossings and resonant rings. To set up a link, an electric control signal must travel in parallel to the path to set up the switches. This enables them to get higher bandwidth utilization on their links than a point to point system like what was presented in this paper, but at the cost of path set up latency and the possibility of network contention, so they get the best performance from lightly contested bulk transfers.

Proximity interconnect [4] is an interesting technology that is trying to solve many of the same problems our photonic socket-level interconnect is. It places dies very close together and uses capacitive coupling to transmit data without actual wire contacts. By doing so it is able to obtain pitches and bandwidths comparable to wires on chip. They have similar aspirations for its use whether it be making small dies to reduce cost or combining large dies to approach wafer scale integration. Photonics, especially with DWDM should be able to achieve higher bandwidths and is a little more robust as a technology since the exact relative alignment of two dies does not matter as much.

Three dimensional die stacking is another technology with the same motivation, but it could be used in conjunction with a photonic interconnect like in Corona [11]. They place their photonic network on its own die to give them more area and let them use better photonic materials which allows them to build more complicated networks. They use a large serpentine crossbar which has orders of magnitude more components than our networks and would be infeasible with our monolithically integrated photonics technology. As such they burn significantly more laser power than our design for comparable bandwidth, but it is hard to make this comparison since they are using a different photonic technology.

8 Conclusion

Previous work has shown the potential of silicon photonics for improved bandwidth and reduced energy, but here we have shown how to exploit the other important properties of distance insensitivty and bandwidth density to build cheaper, reusable, and scalable multi-socket systems. Through our analysis we have presented a general architecture and shown its feasibility with respect to bandwidth, power, area, reusability, and coherence. Our proposed architecture can scale reasonably

well up to 1024 cores with a high and uniform memory bandwidth. This is infeasible in a purely electronic system, as comparable performance would require an unbuildable monolithic 1600 mm^2 die. Throughout our study we sometimes found smaller dies to have performance advantages, but what is most crucial is that they never performed significantly worse than the larger dies.

Currently, high-performance systems use dies as large as is reasonable to manufacture because of the interconnect penalties of traversing socket boundaries. This sometimes results in paying a significant premium to fabricate larger monolithic dies. Silicon photonics could reduce the barrier to multi-socket designs, enabling a new system design methodology of picking a die size that is cheapest to manufacture, and then use as many dies as needed to build the desired system. Although tiled many-core systems provide a convenient way to split up the system, this does not necessarily mean other applications could not also obtain benefits from using multiple smaller die connected photonically.

References

[1] K. Asanovic, R. Bodik, B. C. Catanzaro, J. J. Gebis, P. Husbands, K. Keutzer, D. A. Patterson, W. L. Plishker, J. Shalf, S. W. Williams, and K. A. Yelick. The landscape of parallel computing research: A view from berkeley. Technical report, U.C. Berkeley, 2006.

[2] L. Barroso, K. Gharachorloo, R. McNamara, and A. N. et al. Piranha: A scalable architecture based on single-chip multiprocessing. *ISCA*, Jan 2000.

[3] C. Batten, A. Joshi, J. Orcutt, A. Khilo, B. Moss, C. Holzwarth, M. s Popovic, H. Li, H. Smith, J. Hoyt, F. Kartner, R. Ram, V. Stojanović, and K. Asanović. Building manycore processor-to-dram networks with monolithic silicon photonics. *High Performance Interconnects*, Jan 2008.

[4] R. Drost, R. Hopkins, R. Ho, and I. Sutherland. Proximity communication. *IEEE Journal of Solid-State Circuits*, 39(9):1529 – 1535, Sep 2004.

[5] C. H. et al. Localized substrate removal technique enabling strong-connement microphotonics in bulk si cmos processes. *Conf. on Lasers and Electro-Optics*, 2008.

[6] J. O. et al. Demonstration of an electronic photonic integrated circuit in a commercial scaled bulk cmos process. *Conf. on Lasers and Electro-Optics*, 2008.

[7] J. Hennessy and D. Patterson. *Computer Architecture: A Quantitative Approach.* Morgan Kaufmann, 4th edition, 2007.

[8] N. Kirman, M. Kirman, R. Dokania, and J. Martinez. Leveraging optical technology in future bus-based chip multiprocessors. *IEEE Micro*, 27(6), Jan 2006.

[9] P. Kongetira, K. Aingaran, and K. Olukotun. Niagara: A 32-way multithreaded sparc processor. *IEEE Micro*, page 9, Apr 2005.

[10] A. Shacham, B. Lee, A. Biberman, and K. Bergman. Photonic noc for dma communications in chip multiprocessors. *IEEE Symposium High-Performance Interconnects*, 15, Jan 2007.

[11] D. Vantrease, R. Schreiber, M. Monchiero, and M. M. et al. Corona: System implications of emerging nanophotonic technology. *ISCA*, Jan 2008.