PHILIPPINE ENGINEERING JOURNAL
PEJ 2007; Vol. 28, No. 1: 21-36

A STUDY OF TRANSLATION LOOKASIDE BUFFER STRUCTURES FOR LOW POWER CONSUMPTION

Anastacia P. Ballesil* Luis M. Alarilla Jr. and Louis P. Alarcon

Department of Electrical and Electronics Engineering
University of the Philippines, Diliman, Quezon City

ABSTRACT

Modern microprocessors consume large amounts of energy, majority of which comes from the processor's clock and memory hierarchy. One particular area that can be explored for possible power reduction is the translation lookaside buffer (TLB). TLBs are small caches used to speed up virtual-to-physical address translation. The aim of this study is to design and implement different TLB design structures using VHDL. The structures are laid-out using 0.25 μm CMOS standard cells and then analyzed and characterized in terms of area, performance and power consumption. Results show that, compared to the different structures considered in this study, fully associative structures consume the least amount of power and produce the lowest miss rate. Banked associative structures, on the other hand, occupy the smallest silicon area, with a power consumption that is slightly higher than that of a fully associative structure.

1. INTRODUCTION

To provide applications the illusion of having a very large amount of memory, a technique known as virtual memory was developed. In a virtual memory system, not all parts of a process are stored in the main memory. The not-so-often used parts are stored in the disk and are only placed in the main memory upon access.[1]

Memory management is the general term to cover all techniques of translating a virtual address to physical address. The special hardware used to do the translation is the memory management unit (MMU). The MMU receives the virtual address generated by the CPU and converts it to the corresponding physical address, which can be used to access the main memory. [2]

*Correspondence to: Department of Electrical and Electronics Engineering, University of the Philippines, Diliman, Quezon City 1101 PHILIPPINES. email: anastacia.b.alvarez@gmail.com

Received: February 3, 2006
Revised: August 14, 2009
Accepted: December 18, 2009

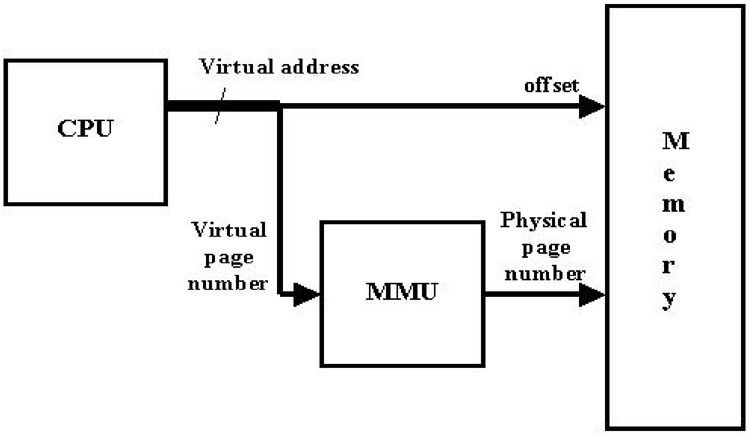

Figure 1. The Memory Management Unit

Most processors with memory management support include a translation lookaside buffer (TLB) that speeds up address translation.[3] They contain PTEs of recently used virtual-to-physical address translations. When the translation is available in the TLB, the address is translated, without having to go through the page table, thus reducing translation time.

Energy and power consumption are now becoming as important as performance due to the many advances in technology. Especially in mobile applications, power must be reduced to prolong battery life. For portable applications, power dissipation affects reliability as well as performance. One of the major contributors of this power consumption is the memory structure, which occupies a large portion of the die area.

The translation lookaside buffer (TLB) is an example of a common memory structure used in today’s microprocessors. Researches done to reduce the TLB power consumption deal with modifying the structure of the memory cell itself. This approach requires transistor-level implementation and is not applicable in high-level design entries such as hardware description languages.

2. RELATED LITERATURE

Computer designers have long predicted the need for large and fast memories. Aside from the fact that programmers constantly require increasing amounts of fast memory, the memory system also has to keep up with increasing processor speed. To achieve this, the idea of using a memory hierarchy that takes advantage of the principle of locality, was introduced. The principle of locality states that a recently accessed piece of data, and those near it, is likely to be accessed again soon. [4] To provide applications the illusion of having a very large amount of fast memory, a technique known as virtual memory was developed. In a virtual memory system, not all parts of a process are stored in the main memory. The seldom-used parts of a program are stored elsewhere, such as in disks and other storage devices, and are only placed in the main memory upon access. [1]

Memory management covers all techniques of translating virtual addresses to physical addresses. The special hardware used to do the translation is the memory management unit (MMU). The MMU receives the virtual address generated by the CPU and converts it to the corresponding physical address, which is used to access the main memory. Processors with memory management support often include a translation lookaside buffer (TLB) to speed up address translation[3]. TLBs contain recently used virtual-to-physical address translations. When the translation pair is available in the TLB, the address can be translated without having to access the page table, thus reducing translation time.

The virtual address, which his the address from the processor, is composed of the virtual page number (VPN) and the offset. The VPN is passed to the TLB for translation. The TLB outputs the corresponding physical page number (PPN). The physical address to be used by the main memory would be the PPN from the TLB together with the offset from the virtual address.

Figure 2 shows the simplified block diagram of the TLB. It is mainly composed of the content addressable memory (CAM), which stores the VPN of available translations, and the static random access memory (SRAM), which stores the corresponding PPN.

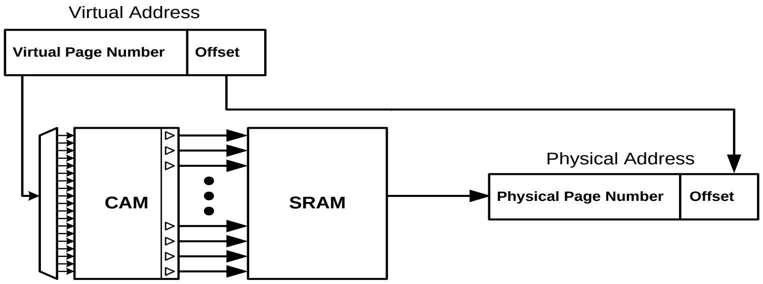

Figure 2. TLB Block Diagram

Content addressable memories (CAMs) differ from ordinary memory in that it allows access through its data rather than by the data's address or the memory location it occupies. Thus, they are able to support three different operations, namely, (1) write, (2) read, and (3) match. The write and read operations access the CAM as an ordinary memory block, where a write or read is done from a specified memory location. The match operation, on the other hand, looks for a match between the input data and data stored in the memory array. Each of the stored word has its corresponding match flag indicating a match between the data and the corresponding word, and signaling the corresponding word in the SRAM to output its data.

Figure 3 shows the block diagram of a 512-word CAM. The control logic is used for indicating the particular block to be read or written to. This is where the replacement algorithm is implemented. [5]

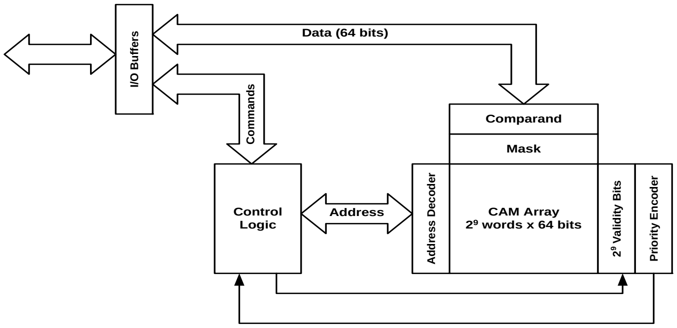

Figure 3. 512-word CAM

Manne, et. al. [6] proposed the banked associative architecture for TLB. In this architecture, the TLB is divided into several fully associative banks, where each bank has its own set of CAM and SRAM cells. Part of the virtual page number is used to select which bank contains the requested data to be read out. Figure 4 shows the organization of their proposed banked associative TLB (BA-TLB).

Their results show that the banked associative TLB performed as well as the fully associative TLB, with the advantage of having the power consumption reduced by a factor equal to approximately one over the number of banks. Power consumption is reduced because only one of the CAM banks is being accessed at any given time.

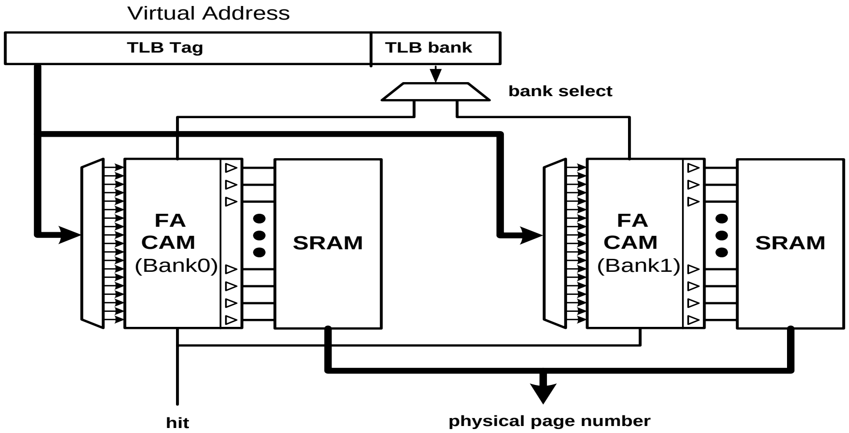

Figure 4. Organization of the BA-TLB

The power dissipated by a memory structure is proportional to its size and operating frequency, and may even vary depending on the operation being done. Power dissipation in memory structures comes from three sources: (1) the memory array, (2) the decoders and (3) the periphery. For a read cycle, for example, of a CMOS memory with m columns and n rows, power dissipation can be approximated as follows: [5]

$$\begin{aligned} P &= V_{DD} I_{DD} \\ I_{DD} &= I_{array} + I_{decode} + I_{periphery} \\ I_{DD} &= \left[m I_{act} + m(n-1) I_{hld} \right] + \left[(m+n) C_{DE} V_{\text{int}} f \right] + \left[C_{PT} V_{\text{int}} f + I_{DCP} \right] \end{aligned} \qquad \textbf{eqtn (1)}$$

The parameters used in this equation are:

I_{act} = effective current of selected (active) cells
I_{hld} = data retention current of inactive cells
C_{DE} = output node capacitance of each decoder
C_{PT} = total capacitance of logic and periphery circuits
V_{int} = internal supply voltage
I_{DCP} = static current of the periphery
f = operating frequency

For the TLB, I_{array} is the current in the memory array. The I_{decode} is the current through the decoders for the bits in the address and the match lines and the $I_{periphery}$ is the current in the external logic.

3. METHODOLOGY

The structures considered for this study were the fully associative, set associative and the two-way and four-way banked associative structures. Random and the least recently used (LRU) replacement algorithm were also used. These structures were designed for 64 entries, 128 entries and 256 entries. These structures and sizes of TLBs were designed using VHSIC Hardware Description Language (VHDL) and converted to gates, targeting a 50MHz clock frequency. The generated netlist were then converted to standard cells for a 0.25μm CMOS process. Figure 5 illustrates the methodology used in this study.

The different TLB designs consist of four main blocks, namely (1) the cam block, (2) sram_cells block, (3) addec block and (4) the missdec block. The central block of the design is the cam block, which stores the VPN and generates the output hit and match signals for the sram_cells.

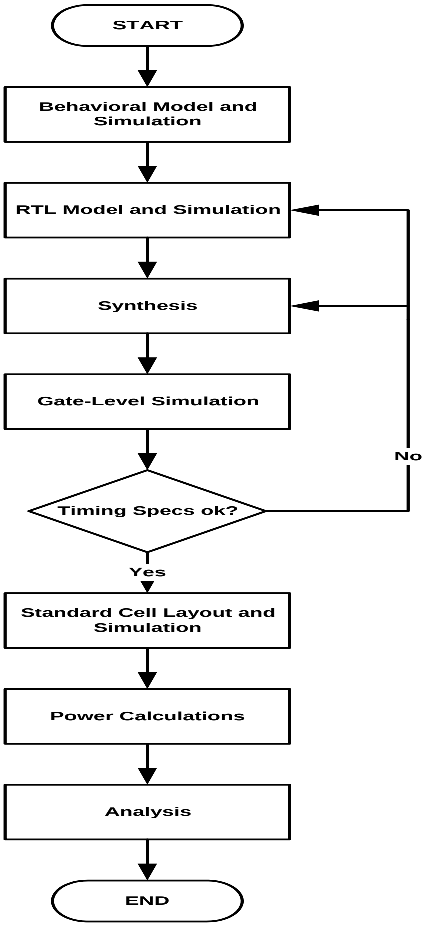

Figure 5. Methodology Flow

For set associative structures, multiple fully associative TLBs are used. Extra logic is added to produce the necessary signals to enable the particular fully associative set to be used. Banked associative structures are similar to set associative except that the least significant bits of vpn, used to enable the fully associative bank is no longer stored in the CAM array. This reduces the word length of the CAM cells.

4. RESULTS AND ANALYSIS

Three design metrics were used in comparing the different structures: (1) performance; (2) area; and (3) power consumption. Performance is measured using two parameters, namely, missrate and average access time. Three SPEC CPU2000 benchmarks were used, namely (1) 256.bzip2,

(2) 186.crafty and (3) 300.twolf. Memory traces of these benchmarks were downloaded from the trace distribution center of the Performance Evaluation Laboratory of Brigham Young University. All the traces were taken from Redhat Linux 6.0 running on a single Pentium II Xeon 450MHz processor with 6.9 GB Disk and 256 MB RAM and the caches were disabled to allow access to the virtual addresses generated by the CPU.

The miss rates of the different TLB structures are derived using the first 1 million addresses of the benchmarks described above. Figure 6 shows the effect of the number of entries on the miss rate. As can be seen from this graph, the miss rate decreases with increasing number of entries. This result is expected because more data can be stored in TLBs with larger size. All structures were found to exhibit the same trend. Comparing the three benchmarks used, crafty has the highest miss rate, followed by twolf, with bzip having the lowest miss rate.

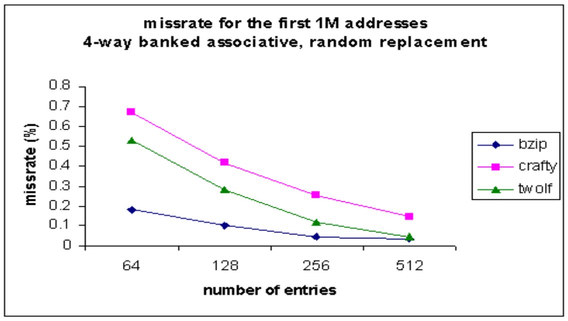

Figure 6. Miss rate vs. Number of entries

Figure 7 shows the effect of associativity on miss rate. The plot of miss rates with respect to the number of is shown in Figure 7. As can be seen from the graphs, a decrease in the number of entries per set, while keeping the total number of entries constant, results in an increase in miss rate. This can be explained by the fact that more capacity misses are expected to occur since there is only a small number of entries per set.

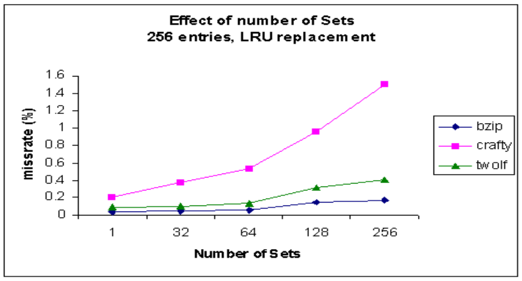

Figure 7. Miss rate vs. Number of sets

Minor differences in miss rates were seen with different number of banks. For the random replacement, no trend can be seen. For the LRU replacement we can see slight increase in miss rates with increasing number of banks. In terms of miss rates, a banked associative TLB with N entries and m banks is just the same as having a set associative TLB with N entries and m sets. Miss rate in set associative structures increases with increasing number of sets. Thus, we expect miss rate to also increase with increasing number of banks. However, since we only used 2-way and 4-way banked associative structures, this increase in miss rate is not as obvious as with the set associative structures used. For the random replacement, no trend is seen since the supposed increase in miss rate is sometimes cancelled by the randomness in replacing entries in the banks.

Figure 8 shows the effect of replacement policy on miss rate. As expected, the LRU replacement policy produces smaller miss rates as compared to the random replacement policy. This is true for all structures except for the two-way set associative structures, where the LRU miss rates are much higher compared to the miss rates for random replacement. This can be explained by the fact that loops in programs become long enough, needing more than the two entries allotted per set in a two-way set associative structure. Thus when the loop returns, it would not be accessing the item it stored recently but rather the item it recently replaced, thereby causing a miss. In terms of the average miss rate improvement brought about by using the LRU instead of the random replacement policy, bzip gives about 12%, compared to crafty's 17% and 25% in twolf.

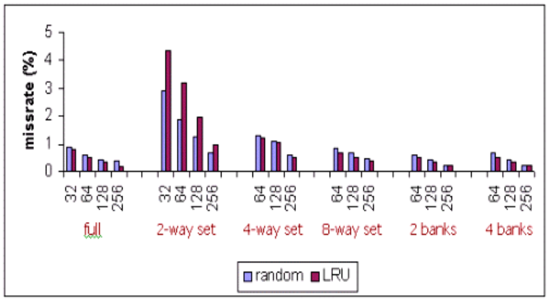

Figure 8. Effect of replacement policy on miss rate

Different designs produce different critical path delays, thereby affecting the maximum frequency of operation. These delays are due to gate and routing delays. Advances in CMOS technology allowed great improvement in microprocessor performance by integrating more transistors on a single chip. However, as the integration level increases, delays due to routing become dominant. In order for wires to be able to connect to the small transistors, their cross-sectional area has to be reduced, resulting in higher resistance per unit length. The space between wires also decreases, increasing the capacitance between them. Since the delay is proportional to the product of resistance and capacitance, wire delays tend to become comparable to gate delays. Figure 9 shows the critical path delays of the different TLB structures.

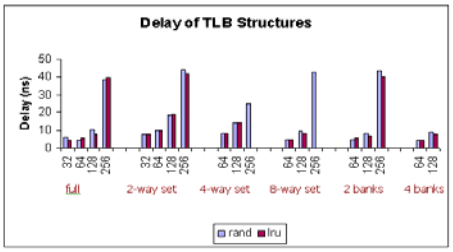

Figure 9. Critical Path Delays

The results in Figure 9 confirm the fact that an increase in the number of entries or number of sets results in an increase in delay. This is expected since an increase in the number of entries results in a longer search path and an increase in the number of sets results in an increase in routing complexity and number of comparisons done to find a miss also increases. Replacement policy, however, show almost no effect in the delay of structures.

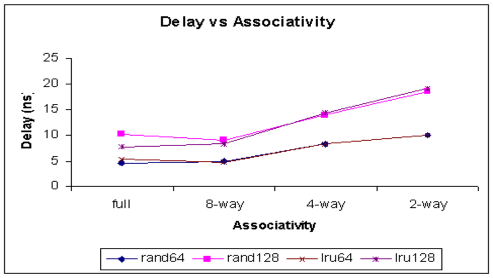

Figure 10. Delay vs. Associativity

In terms of associativity, it can be seen in Figure 10 that an increase in the number of sets results in an increase in the delay. This increase in the delay can be attributed to the gate delays brought about by the additional logic needed to select the accessed set. Banking, however, has almost no effect on the delay. Differences in values can be attributed to the implementation differences during synthesis and place and route. Optimization efforts in synthesis and place and route results in changes in the number of cells, load capacitance and wire length. An increase in the number of cells result in increased area and power, while increase in load capacitance results in increased power and delay. Wire lengths, on the other hand, increase the delay.

Figure 11 shows the effect of increasing the number of entries on the average access time. It is obvious from the graph that the average access time for all structures is proportional to the increase in the number of entries. This is because although the miss rate decreases with increasing number of entries, the delay also increases with increasing number of entries.

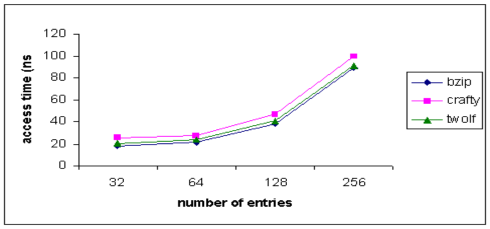

Figure 11. Access time vs. number of entries

In implementing the design, a decoder block is used to enable the required sets in a set associative TLB. Each set serves as an additional capacitive load to the decoder. Looking at the delays of the structures, we can see that the delay doubles as the number of entries doubles. Thus clock period increases by around 4 times, while the decrease in miss rate is only around half. This is because the output signal is expected to be valid after half a clock cycle and therefore the clock period becomes twice the delay. Thus, the characteristics of the access time versus number of entries plot closely follow that of the delay rather than that of miss rate.

Figure 12 shows the effect of associativity on access time. Shown are the plots of access time vs. associativity for 128 entries with LRU replacement and 256 entries with random replacement. All the structures with less than 256 entries follow a curve similar to that of the 128 entries with LRU replacement. However, as shown in Figure 12, the 256-entry TLB no longer follow this trend.

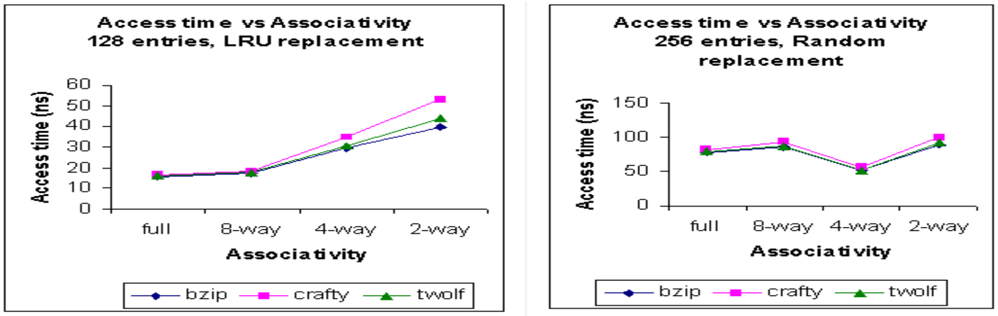

Figure 12. Access time vs associativity for (a) 128 entries with LRU replacement and (b) 256 entries with random replacement

It was shown that miss rates increase with increasing number of sets. The delay also increases with increasing number of sets. Since access time is proportional to both miss rate and delay, and increasing the number of sets results in an increase in miss rate and delay, we expect access time to increase with increasing number of sets, as shown in Figure 12a. For the 256-entry structures, however, this trend is not followed since the delays due to the routing and the memory array dominate rather than the delays due to the decoder.

Since the effect of banking on miss rates is only minimal and the delays do not show any trend, we expect banking to have minimal effect on access time. For LRU replacements, access times vary only by around 3ns. For random replacement, the difference in access times is around 3ns for 64 entries, 6ns for 128 entries and 20ns for 256 entries.

The cost of an integrated circuit is proportional to the area of the die. For this study, we use the same core to I/O distance and the same timing constraints. The Silicon Ensemble™ software is allowed to increase the number of cells as well as increase the area of a cell in order to meet the required timing specifications. Area increases with increasing number of entries, due to the increase in memory storage requirements. Figure 13 shows the plot of area versus number of entries. An almost linear increase in area and an almost exponential increase in the number of cells was observed for a linear increase in number of entries. A decrease in area, however, was observed for increasing number of banks. This is because as we increase the number of banks, the length of the stored VPN decreases since the least significant bits are used as a bank selector and no longer stored in CAM.

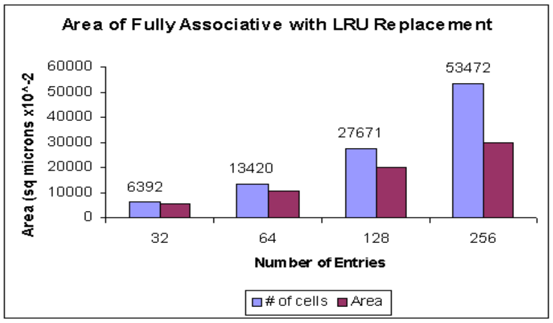

Figure 13. Area vs. Number of

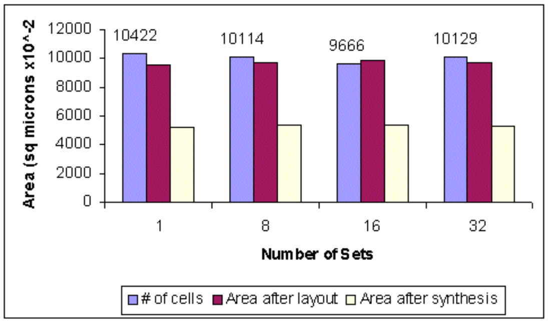

Figure 14. Area vs. Number of Sets

The effect of associativity, however, is not as straight forward as the previous two parameters. Shown in Figure 14 is a plot of area and number of cells vs. number of sets. The area report from synthesis is included in the plot to show that the area decreases with increasing number of sets. Thus, we say that the deviations in area after the layout is due to the optimization done by the software during routing since the software is allowed to add, delete and resize cells in order to meet the timing specifications.

Area is expected to increase with increasing number of sets because of the additional decoding logic. Figure 14 shows the opposite of such statement. This is because while extra logic is added for the enable circuit, the internal logic is also decreased since fewer cells are stored per set. Thus we say that the change in internal logic count dominates the change in external logic count, thus decreasing area with increasing number of sets.

The computation of the average power consumed was done by computing for the average power consumed by the different structures for a hit, a compulsory miss and a conflict miss and using these values to compute for the average power per benchmark. Since the power consumption is also dependent on the number of transition of signals, the average number of transitions for the three benchmarks is determined. Tests showed that average number of transitions for the bzip, crafty and twolf are 7.5027, 8.4509 and 8.1316, respectively.

Figure 15 shows the average power for a compulsory miss, capacity miss and a hit, for the different structures. From the figure, we can see that average power consumed during a capacity miss is just slightly higher than that of a compulsory miss, and the average power consumed during a hit is even smaller. The difference in the power consumption for the different operations is mainly due to dynamic power, which is the power dissipated during charging and discharging of node capacitances.

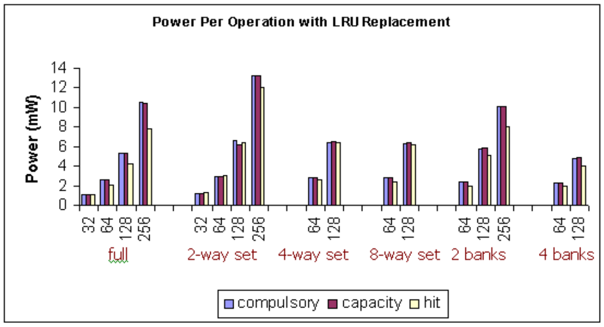

Figure 15. Average power per operation

The equation for dynamic power can be simplified as:

$$Pdyn = C_L V_{DD}^2 f_{0-1} \qquad \textbf{eqtn (2)}$$

where C_L is the effective output capacitance
V_{DD} is the supply voltage
f_{0-1} is the switching frequency

As predicted by this equation, dynamic power increases with increasing switching frequency. For the case of a miss, switching occurs when the entry changes its value to the new VPN. For the case of a read however, no change occur in the CAM entry. Thus, dynamic power, and therefore total power, is greater for misses than for hits. The difference in the power consumption during a compulsory and a capacity miss is due to the fact that, on the average, more bit changes occur during a capacity miss as compared to a compulsory miss.

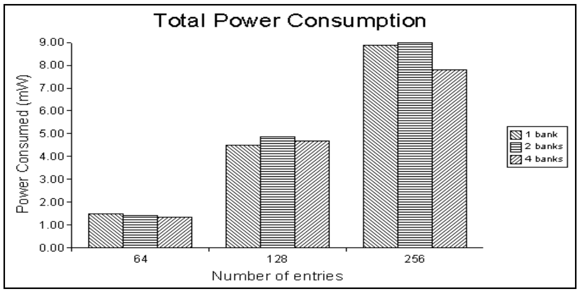

Figure 16. Power Consumption of Different Structures

It can be seen from Figure 16 that banking indeed reduces power consumption of the TLB. This is due to the fact that only one of the banks is active at a time. However, power consumption does not necessarily scale with the number of banks, as is seen from the fact that power consumption of the fully associative and the two-bank structures are comparable. This can be attributed to the fact that extra logic is needed in selecting which bank is to be activated. Comparing the four-banked structure with the fully associative, however, we see more than 12% improvement in terms of power for the four-banked structure.

Table 1 Power consumption of different structures

Structure (banks/entries)	Total Power (mW)	Dynamic (mW)	Leakage (μW)	Percent consumption (%)
1/64	1.4969	1.4959	1.0120	100
cam	0.9228	0.9224	0.4685	61.6485
sram	0.2454	0.2449	0.5165	16.3934
logic	0.3287	0.3287	0.0270	21.9581
1/128	4.4966	4.4936	3.0860	100
cam	2.8037	2.8026	1.1030	62.3501
sram	0.7146	0.7127	1.9100	15.8919
logic	0.9784	0.9783	0.0730	21.7580
1/256	8.8815	8.8778	3.7150	100
cam	6.1837	6.1818	1.8910	69.6250
sram	1.4738	1.4721	1.6990	16.5938
logic	1.2240	1.2239	0.1250	13.7812
2/64	1.4046	1.4035	1.0280	100
cam	0.8237	0.8232	0.4831	58.6422
sram	0.2694	0.2689	0.5164	19.1802
logic	0.3115	0.3115	0.0285	22.1776
2/128	4.8601	4.8573	2.8350	100
cam	3.1852	3.1842	0.9515	65.5364
sram	0.9897	0.9879	1.8303	20.3637
logic	0.6853	0.6852	0.0532	14.1000
2/256	8.9705	8.9670	3.4430	100
cam	6.1987	6.1970	1.7154	69.1009
sram	1.4000	1.3983	1.6449	15.6062
logic	1.3719	1.3718	0.0827	15.2929
4/64	1.3340	1.3330	0.9593	100
cam	0.7484	0.7480	0.4157	56.1016
sram	0.2603	0.2598	0.5164	19.5114
logic	0.3253	0.3253	0.0272	24.3870
4/128	4.6909	4.6881	2.8100	100
cam	3.1669	3.1660	0.9270	67.5123
sram	0.8451	0.8433	1.8304	18.0157
logic	0.6789	0.6788	0.0526	14.4721
4/256	7.8043	7.8008	3.5690	100
cam	5.0973	5.0955	1.8127	65.3144
sram	1.3970	1.3953	1.6462	17.9001
logic	1.3100	1.3099	0.1101	16.7856

Table 1 shows the power consumption of the different structures implemented. Structures are identified in terms of the number of banks (1 bank for fully associative structures) and the number of entries. The dynamic and leakage power contributions of the memory arrays are also included. The indicated percent consumption is the percentage consumption of the particular block indicated. The control block of the cam is included in the logic so as to isolate the memory array from the other logic involved in the design. It is clear from the data in Table 1 that leakage power is very small compared to the total power. Dynamic power still dominates at almost 100% of the total power. This, however, does not fully disregard the possibility of leakage power dominating over dynamic power. Closer investigation of the data shows that majority of the leakage power comes from the sram, followed by the cam and a negligible amount from logic. This indicates that most of the leakage power comes from the memory array rather than from combinational circuits. This, we expect, will continue to increase as technology scales to even smaller transistor sizes.

Having verified the dominance of dynamic power over leakage power, we find the need to determine the number of transitions and power dissipated per operation. For the case of a miss, switching occurs when the entry changes its value to the new VPN. For the case of a read however, no change occur in the CAM entry. Thus, dynamic power, and therefore total power, is greater for misses than for hits. The difference in the power consumption during a compulsory and a capacity miss is due to the fact that, on the average, more bit changes occur during a capacity miss as compared to a compulsory miss.

Figures 17 to 19 show the comparison of per operation power consumption of the different structures. As seen from the graphs, banking reduces power consumption due to the decreased number of active cells per operation. This however, is not the case for 128 entries, where we attribute the difference to the additional logic circuitry needed in implementing the banks.

Comparing the different operations, we see only a slight difference between the power consumption during conflict and compulsory misses. This is due to the fact that data transitions happen during both misses. Only the valid bit is constant for conflict misses, as opposed to compulsory misses where the valid bits change from logic 0 to logic 1.

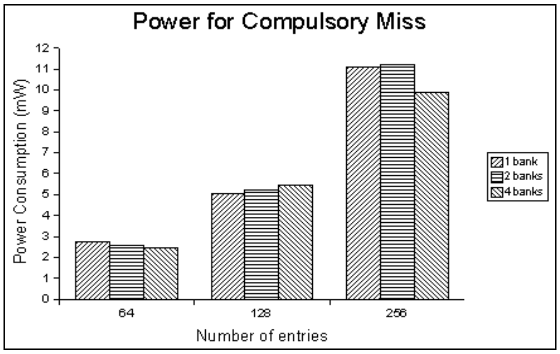

Figure 17. Power Consumed during Compulsory Miss

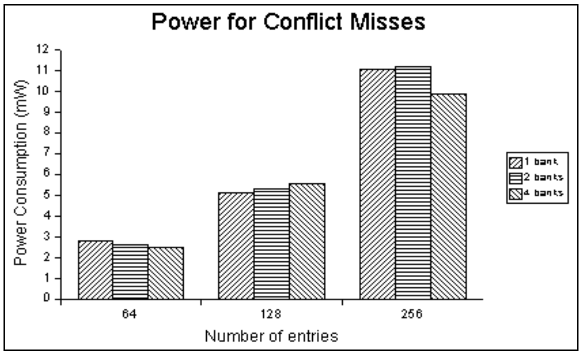

Figure 18. Power Consumed during Conflict Miss

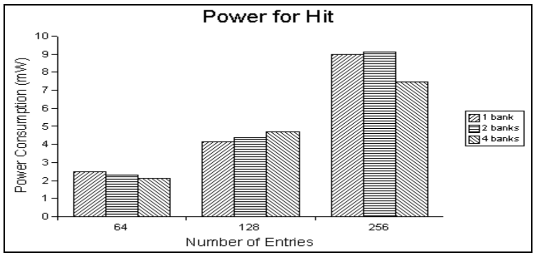

Figure 19. Power Consumed during a Hit

5. SUMMARY AND CONCLUSION

After looking at the effect of the different structures on the different metrics, it is important to draw conclusions as to when the use of such structures will give the best benefit. Thus, we need to consider the tradeoffs of the different metrics for the different TLB structures considered.

Figure 20 shows the tradeoff between miss rate and power for the different structures. It can be seen from this figure that the best structure in terms of tradeoff between power and miss rate is the banked associative TLB.

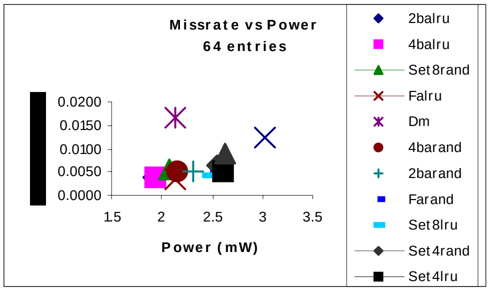

Figure 20 Miss rate and power tradeoff

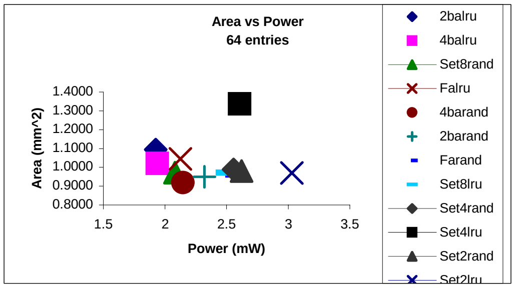

Figure 21. Area and power tradeoff

Figure 21 shows the tradeoff between area and power for the different structures. It can be seen from this figure that the structure with the smallest area and relatively low power is the 4-way banked associative structure with random replacement while the structure with the lowest power and relatively small area are the 4-way and 2-way banked associative structure with LRU replacement. Thus we say that the best structure in terms of tradeoff between power and area is still the banked associative TLB.

In summary, we have seen miss rate decreases with increasing number of entries, increasing number of entries per set, and decreasing number of banks. Results also show that LRU produces around 10% to 25% improvement over the random replacement, except in the 2-way set associative TLB, where capacity misses are magnified by the LRU replacement.

Delays also increase with increasing number of entries, due to the increase in memory array, and increasing number of sets, due to increase in external logic. Banking has minimal effects on the delay of the structure. Average access time follows the characteristics of the delay curves, except in the larger TLB sizes. For sizes less than 256 entries, the delay is dominated by the decoder logic. However, for larger sizes, the delay in the memory array starts to dominate the effect of the total delay.

Area increases with increasing number of entries due to the increase in the memory array. Increasing the number of sets also decreases the area due to the decrease in the complexity of the small memory arrays.

In terms of power, increasing the number of entries increases power consumption due to the increase the size of the memory array as well as the added logic. The power likewise increases with increasing number of sets. This is due to the fact that the power needed to drive the capacitive load of the decoder becomes larger than the power due to transitions in the memory array. In increasing the number of banks, power consumption is reduced because of the reduction in the number of active cells with only a small increase in node capacitance and external logic.

For all TLB sizes, the structure that produces the lowest miss rate is the fully associative TLB with LRU replacement. For both 64 and 128 entries, the structure with the lowest area would be the 4-way banked associative with random replacement. For the 256 entries, however, the data for a 4-way banked associative with LRU replacement policy is not available, so from the available data, the 4-way banked associative with random replacement policy has the lowest power. The structure with the lowest power would be the fully associative structure with LRU replacement policy. Table 2 summarizes these results.

Table 2 Structures Best for the Metric

Metric	64-entry	128-entry	256-entry
Power (mW)	FALRU (2.1219)	FALRU (4.3296)	FALRU (7.8343)
Miss rate	FALRU (0.1423)	FALRU (0.0819)	FALRU (0.0363)
Area (mm2)	4-way BALRU (1.0204)	4-way BALRU (2.3387)	4-way BARAND (2.7620)

REFERENCES

1. Jacob, B. L. “Virtual Memory Systems and TLB Structures”, University of Maryland at College Park (2001).
2. Baker, P. A. and G. L. Martes. “Memory Management Unit with Overlapping Control for Accessing Main Memory of a Digital Computer”. Apple Computer Patent. Unites States (1990).
3. Jacob, B. L. and T. N. Mudge. "Virtual Memory in Contemporary Microprocessors." *IEEE Micro* **18(4)** (1998).
4. Patterson, D. A. and J. L. Hennessy. “Computer Architecture: A Quantitative Approach”, Morgan Kaufmann Publishers (2003).
5. Rabaey, J. M., A. P. Chandrakasan, et al. “Digital Integrated Circuits: A Design Perspective”, Prentice Hall (2003).
6. Manne, S., A. Klauser, et al. “Low Power TLB Design for High Performance Microprocessors”, University of Colorado at Boulder (1997).