On the Decreasing Significance of Large Standard Cells in Technology Mapping

Jae-sun Seo, Igor L. Markov, Dennis Sylvester, and David Blaauw
Department of EECS, University of Michigan, Ann Arbor, MI 48109
{jseo,imarkov,dmcs,blaauw}@umich.edu

ABSTRACT

Technology scaling reduces gate delays while wire delays may increase. Our work studies the interaction of this phenomenon with technology mapping and its impact on modern EDA flows. In particular, we demonstrate that the use of larger standard cells increases the number of long wires and may undermine circuit delay optimization at 65nm and below. Experiments with 130nm, 90nm, 65nm, and 45nm industrial CMOS technology suggest that limiting the use of larger standard cells in technology mapping becomes more effective at 65nm and 45nm node, resulting in up to 12% improvement in critical path delay on large benchmark circuits.

1. INTRODUCTION

Over several decades, technology mapping has been extremely useful for reducing the device area of complex logic. Furthermore, recent research in Boolean matching [1-2] accomplished dramatic efficiency improvements for function matching, facilitating new technology mapping algorithms that can deal with 10-input gates. However, extending these algorithms with proper models of circuit delay and validating them with respect to recent technology nodes remains a major research challenge.

While technology mapping seeks to minimize device count, the bulk of critical path delay has shifted from gates to wires in the last 5 years. In particular, the number of repeaters required is exponentially increasing with each technology step [3-4], and 10~15% of gates in large microprocessor chips are buffers that break down long interconnects. Extensive literature exists on optimal buffering [5-7] that employs fairly accurate delay modeling, but does not attempt logic restructuring.

Our work is motivated by the apparent dichotomy between (1) the literature on buffer insertion that improves circuit performance by adding a large number of one-input one-output gates (buffers and inverters) that do not perform any logic operation, and (2) the literature on functional technology mapping, which clusters logic into 5-15 input gates, improving area, but does not evaluate overall circuit performance with respect to current technology nodes.

Previous literature [8-11] suggests that technology mapping must interact with placement of the standard cells and use accurate interconnect models for performance optimization. These works improved the critical delay through either integration of layout information in early logic synthesis stage [8-9] or iterative re-synthesis with placement information [10-11], but they have not considered the impact of technology mapping on global buffer counts and the overall circuit performance after place-and-route optimizations.

This paper proposes to, ironically, undo technology mapping for high-speed designs through reducing the wire delay components in the critical path of large circuits. Foregoing aggressive technology mapping and using a large number of standard cells (but of smaller size) will eliminate the need for excessive buffers during post-placement timing optimization. The discussions and experiments in this work also consider coupling capacitance between adjacent wires which dominates the wire capacitance in most advanced technologies, and we attempt to reduce the parallel run length of neighboring wires.

Recent work [15] points out that conglomerating small cells into a large cell may produce non-monotonic interconnects which adversely affect delay and routability as illustrated in Figure 1. By limiting the use of large standard cells, our approach inherently blocks the occurrence of this disadvantageous technology mapping, and results in a number of shorter monotonic wires.

Our considerations and conclusions are intended for ASIC/SoC designs rather than FPGA designs or microprocessor designs. In FPGA designs, programmable interconnect is uniformly buffered and linear wire delays do not significantly depend on whether long nets are broken into shorter segments. On the other hand, technology mapping into LUTs is an important and difficult task, so technology

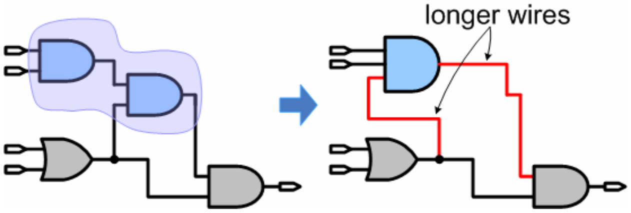


Figure 1: Indiscriminate technology mapping may produce longer wires, adversely affecting delay and routing congestion.

mapping still remains useful in FPGAs. In high-performance microprocessor designs, clock period is short and the logic between pipelines are often dominated by large fanouts. In this case, the number of inserted buffers cannot be reduced significantly by reducing the length of wires.

The remainder of this paper is organized as follows. In section 2, a “clean” experiment is analyzed with a simple combinational logic block and a long wire. Section 3 evaluates the impact of standard cell libraries on circuit delay, shows the experimental results, and explains the observations. Section 4 summarizes this paper.

2. ANALYSIS OF SINGLE PATHS

As a proof of concept, we conducted a simple experiment as depicted in Figure 2. As a baseline for comparison, 16 3-input NAND gates drive a M5 minimum-pitch 5mm wire, which is optimally repeated in 65nm technology. Traditionally, we would have the combinational logic placed in a denser cluster for minimum area as shown in Figure 2(a). Instead, however, 16 NAND gates are spread out regularly along the interconnect to implement the logic and also serve as repeaters in Figure 2(b), as proposed in [17]. These distributed NAND gates eliminate long wires and the need for repeaters, resulting in actually better performance. This effect could be exploited by decomposing the logic into more gates, i.e., undoing technology mapping. In Figure 2(c), the long wire is divided more finely with more logic gates (24 2-input NANDs) for the same functionality. Note that, in Figure 2(a)-(c), the inputs of the NAND gates which are not in the critical path are tied to Vdd for worst-case rising delays.

HSPICE simulation is done with industrial 65nm CMOS technology, where all three schemes are swept with sizing, and the optimal energy versus delay results are shown in Figure 3. Comparing to scheme (a) at iso-energy of 1.1pJ, scheme (b) achieves 13% delay reduction and scheme (c) achieves 18% delay reduction. Overall, (c) improves the energy-delay curve of (a) by a significant amount. At these delay points, Figure 4 shows a delay breakdown of the three schemes. By spreading out the NAND gates in (b), logic gate delay is increased since the load capacitance of the NAND gates is increased due to wires, but the repeater delay portion is eliminated and the overall delay is reduced by 13%. Through undoing technology mapping in (c), wire delay is further reduced due to fine chopping of wires and gate delay also slightly decreased due to reduced load capacitance. Interestingly, if long wires are present in the circuit delay, spreading out the gates and using more gates to implement a given logic could actually improve delay since they convert wire delay back to logic delay.

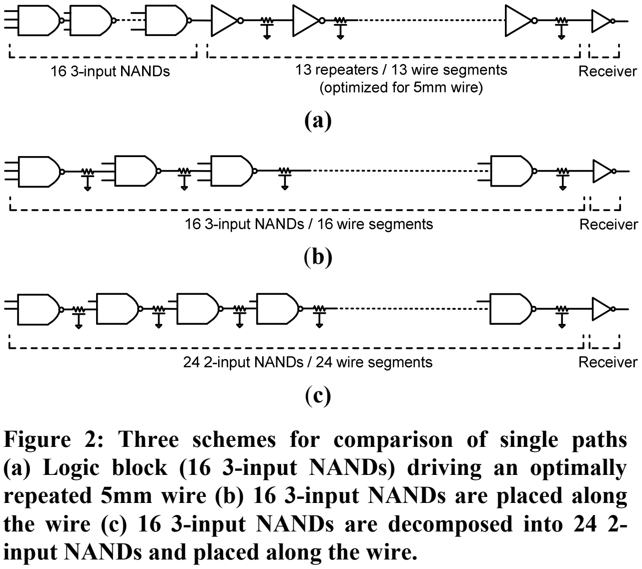

16 3-input NANDs
13 repeaters / 13 wire segments
(optimized for 5mm wire)
Receiver
(a)
16 3-input NANDs / 16 wire segments
Receiver
(b)
24 2-input NANDs / 24 wire segments
Receiver
(c)

Figure 2: Three schemes for comparison of single paths (a) Logic block (16 3-input NANDs) driving an optimally repeated 5mm wire (b) 16 3-input NANDs are placed along the wire (c) 16 3-input NANDs are decomposed into 24 2-input NANDs and placed along the wire.

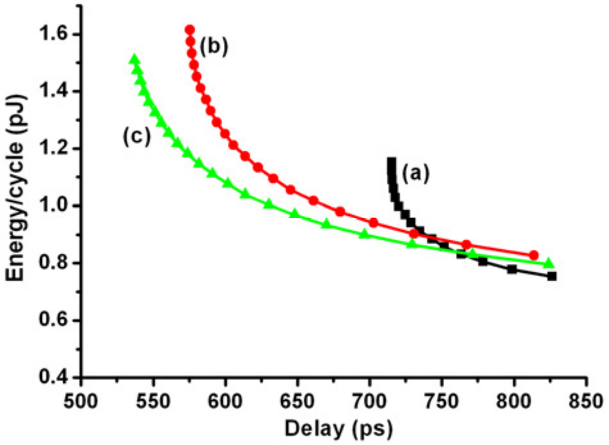


Figure 3: Energy versus delay comparison for the three different schemes in Figure 2.

One possible drawback of this approach is that the inputs to the NAND gates in the middle of the wire have to be routed to the intermediate placement locations, but the surrounding logic gates could be restructured and placed nearby the middle of the wire. We scrutinized this in the following section by performing synthesis, placement and routing on large benchmarks.

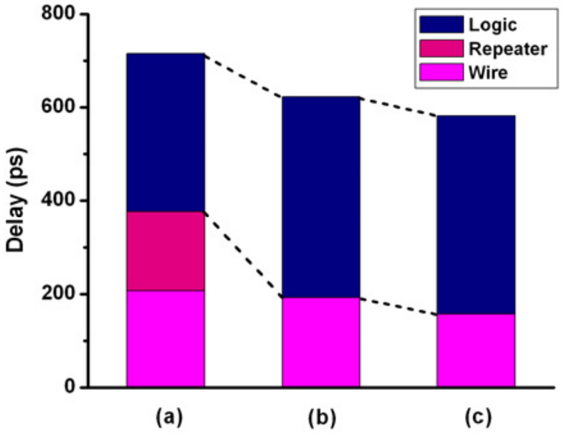


Figure 4: Delay breakdown (logic delay, repeater delay, and wire delay) of the three schemes in Figure 2 at iso-energy of 1.1pJ.

3. EVALUATING UTILITY OF LARGE CELLS IN TECHNOLOGY MAPPING

To evaluate the utility of technology mapping for general circuits in scaled technologies, we compare pairs of libraries for several benchmarks with each technology. 'Original' scheme uses original standard cell library without any restriction, while 'No Large Cells' scheme is confined to the library where there are only 1-input and 2-input gates available.

3.1 Methodology

Figure 5 shows the flow chart for both approaches. Starting from the same behavioral netlist, logic synthesis (Synopsys Design Compiler 2007.03-sp2) is applied for each scheme with a restriction on the 'No Large Cells' scheme to use only 1-input or 2-input standard cells. After logic synthesis, the structural netlist goes through timing-driven placement, physical synthesis, and timing-driven routing (Cadence SoC Encounter 6.1.2). Post-placement logic restructuring is executed if necessary, but the restriction on the number of inputs of gates still holds in the 'No Large Cells' scheme. Finally, timing analysis is performed for both approaches with all back-end parasitics including coupling capacitance. This procedure was done for industrial 130nm, 90nm, 65nm, and 45nm technologies, and benchmark circuits from IWLS 2005 [12] were used (s35932 from ISCAS family and the rest of them from OpenCores family). In the overall flow, the proposed scheme does not add any intermediate steps or iterations to the baseline. In fact, our approach seeks to reduce resource utilization (less standard cells from the library) while also improving delay.

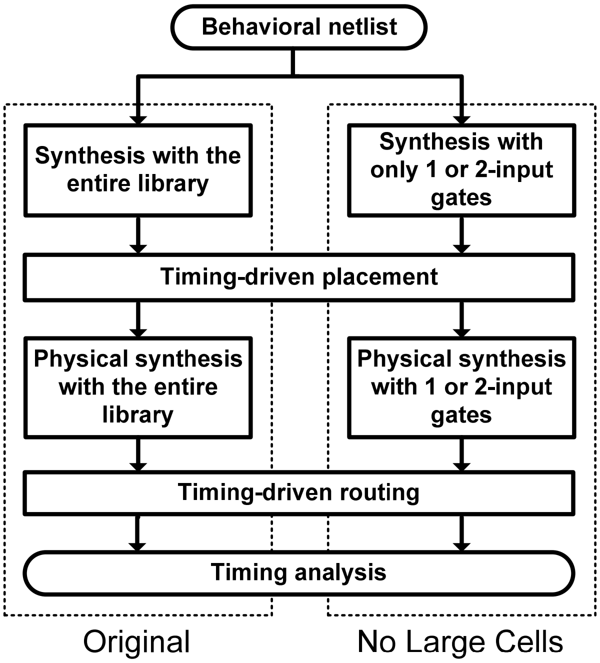

Behavioral netlist
Synthesis with the entire library
Synthesis with only 1 or 2-input gates
Timing-driven placement
Physical synthesis with the entire library
Physical synthesis with 1 or 2-input gates
Timing-driven routing
Timing analysis
Original
No Large Cells

Figure 5: Flow chart for the methodology of 'Original' and 'No Large Cells'.

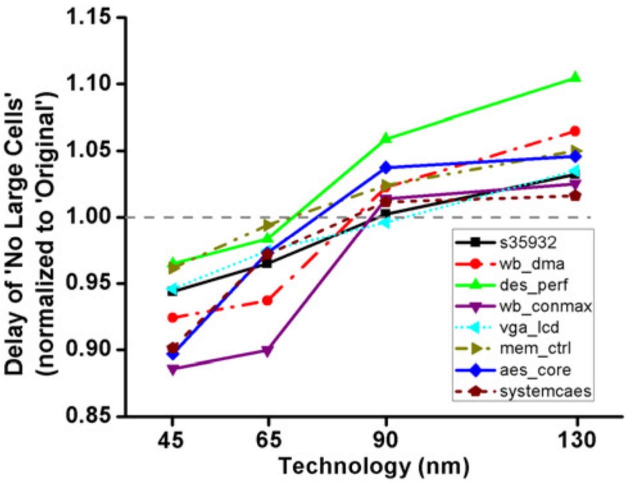


Figure 6: Critical path delay comparison of IWLS benchmarks using 'Original' and 'No Large Cells' approach in 130nm, 90nm, 65nm, and 45nm technology.

3.2 Experimental Results

One expects the 'No Large Cells' approach to increase the gate count due to a more limited standard cell library. However, the critical path could actually benefit from more gates since both the wire capacitance and the number of required buffers are reduced.

Figure 6 compares the critical path delay between 'Original' and 'No Large Cells' configurations for eight benchmarks. Delay of 'No Large Cells' scheme is normalized to that of 'Original' scheme. The monotonic trend shown in Figure 6 illustrates the decreasing utility of large standard cells in technology mapping for more advanced technologies. At 65nm and 45nm technology, discarding large standard cells (3-inputs or more) gave better results (1-12%) in critical path

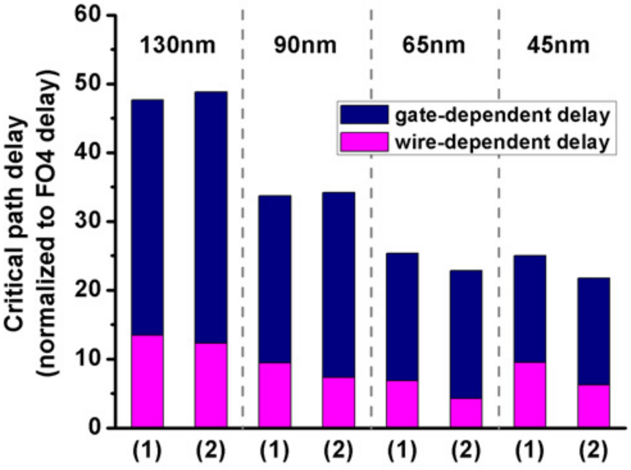


Figure 7: Critical path delay breakdown (gate-dependent delay and wire-dependent delay) of benchmark wb_conmax for (1) 'Original' and (2) 'No Large Cells' approach across four technology nodes.

Table 1: Detailed comparison of the benchmarks for 'Original' and 'No Large Cells' scheme on critical path, average wire length (=total routed wire length/wire count), inserted buffer count, total standard cell count, wire capacitance, and total standard cell area is shown for (a) 65nm and (b) 45nm technology.

(a) 65nm

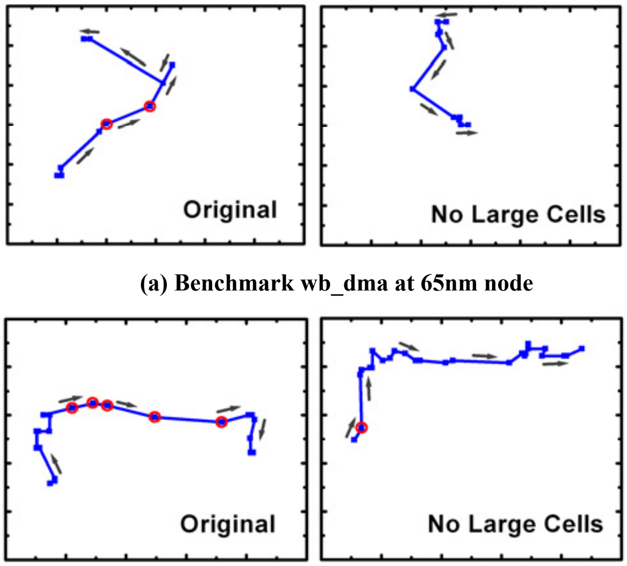
Benchmark	Original						No Large Cells (vs. Original)					
	Critical path delay (FO4)	Avg. wire length (μm)	Buffer count / total cell count	Wire capacitance (fF)		Cell area (μm^2)	Critical path delay	Avg. wire length	Buffer count / total cell count	Wire capacitance		Cell area
				total	coupling					total	coupling	
s35932	10.9	21.5	279 / 5764	19.3	9.25	35512	-3.5%	+3.4%	-10% / +35%	+14%	+12%	+10.6%
wb_dma	14.9	36.5	180 / 4968	41.0	28.5	24069	-6.3%	-15.8%	-19% / +19%	-5%	-8%	+4.0%
des_perf	20.1	23.7	511 / 69733	244.8	128.2	320170	-1.7%	-11.3%	-44% / +12%	-2%	-4%	-1.6%
wb_conmax	25.3	69.7	921 / 24720	405.5	330.8	112590	-10.0%	-45.0%	-15% / +73%	-21%	-26%	+18.3%
vga_lcd	28.2	40.7	3378 / 29778	193.3	94.6	264839	-2.5%	-23.5%	-5% / + 32%	-9%	-18%	+12.9%
mem_ctrl	21.8	28.1	217 / 5227	22.9	12.2	50224	-0.6%	-23.1%	-27% / +37%	+2%	-1%	+6.5%
aes_core	25.6	26.0	475 / 18568	84.1	54.1	108502	-2.7%	-22.9%	-11% / +19%	-15%	-22%	-9.8%
systemcaes	30.1	34.0	832 / 5678	36.9	24.1	53441	-2.8%	-19.0%	-27% / +20%	+7%	+2%	+20.9%
Average							-3.8%	-19.7%	-20% / +31%	-4%	-8%	+7.7%

(b) 45nm

Benchmark	Original						No Large Cells (vs. Original)					
	Critical path delay (FO4)	Avg. wire length (μm)	Buffer count / total cell count	Wire capacitance (fF)		Cell area (μm^2)	Critical path delay	Avg. wire length	Buffer count / total cell count	Wire capacitance		Cell area
				total	coupling					total	coupling	
s35932	12.6	14.0	283 / 7155	14.3	7.2	20131	-5.6%	-14.5%	-5.3% / +18%	+8%	+7%	+7.5%
wb_dma	15.7	27.3	268 / 6109	27.8	17.7	13600	-7.7%	-12.5%	-13% / +6%	-9%	-12%	-1.7%
des_perf	23.3	17.3	1599 / 85297	213.9	122.3	186916	-3.5%	-16.7%	-27% / +20%	-4%	-10%	+3.1%
wb_conmax	25.0	46.3	1874 / 27280	256.2	231.0	55171	-12.3%	-37.2%	-25% / +70%	-21%	-29%	+20.1%
vga_lcd	31.7	28.6	3807 / 42890	183.1	108.2	153413	-5.4%	-20.6%	-5% / +28%	-2%	-4%	+8.8%
mem_ctrl	24.3	20.3	307 / 6285	20.7	12.7	14733	-3.9%	-23.4%	-11% / +32%	-4%	-10%	+7.2%
aes_core	26.2	18.9	1329 / 17485	63.4	42.6	46640	-10.3%	-19.1%	-54% / +39%	-7%	-4%	+8.4%
systemcaes	30.4	26.4	1031 / 6557	32.6	21.1	15274	-9.9%	-22.2%	-36% / +16%	-7%	-2%	+15.7%
Average							-7.3%	-17.2%	-22% / +29%	-6%	-8%	+8.6%

delay than the original technology mapping for all benchmarks. Breaking up the wire into more segments proves to be effective at 65nm and below through reducing the wire delay components. The delay breakdown for benchmark wb_conmax is shown in Figure 7 across four technology nodes. Gate-dependent delay is defined as the sum of intrinsic gate delay and gate load delay, which is basically the circuit delay when no wire is present. Wire-dependent delay consists of inserted buffer delay and wire load delay, which are the delay elements generated due to routed wires. It can be seen that our approach increases gate-dependent delay by a minimal amount, but the wire-dependent delay component is reduced significantly (35% in 45nm), leading to an overall 12% performance improvement in 45nm node. Note that the relative portion of wire-dependent delay grew considerably at the 45nm node. This is mostly due to the sharp increase of resistance of minimum width wires in 45nm, considering that the capacitance of a unit length wire does not change significantly for each technology step.

Table 1 shows a detailed comparison on several metrics for eight benchmarks for both 65nm and 45nm technology to check whether the 'No Large Cells' approach is working as proposed. Typically a large number of buffers are inserted during timing optimization for the given benchmark circuits, and the number of buffers is reduced by 5-54% by breaking the long wires into short wires with more gates. Average wire length and wire capacitance (both total and coupling) show noticeable reduction except for the relatively small s35932 benchmark. The reduction in coupling capacitance is more than that in ground capacitance, which is due to the observed higher routing congestion in intermediate and high metal layers in the 'Original' configuration leading to increased coupling capacitance. This fact is encouraging because coupling capacitance increasingly dominates the overall wire capacitance with technology scaling. In the s35932 benchmark, the fact that the critical path delay marginally decreased despite an increase in wire length and capacitance suggests further improvement by introducing the proposed approach only on timing-critical nets. Benchmark



(a) Benchmark wb_dma at 65nm node

(b) Benchmark systemcaes at 45nm node

Figure 8: Critical path comparison between 'Original' and 'No Large Cells' configuration for benchmarks (a) wb_dma at 65nm technology node and (b) systemcaes at 45nm technology node is shown (dots with circles represent inserted buffers).

des3_perf shows a small improvement in critical path delay in spite of a large reduction in the buffer count, especially in 65nm, because the inserted buffer count is a small portion of the total standard cell count (0.7%).

It is not surprising that the standard cell count is increased by 12-54%, but the standard cell area overhead is only 8.6% on average at 45nm technology since 1-input and 2-input gates are typically smaller than complex gates. This area increase would not necessarily result in comparable die area increases in modern microprocessors or SoC designs because embedded memories and hard IP blocks consume a large portion of the total chip area, making standard cell area a relatively lesser concern [13-14]. Also, in designs with hierarchical floorplans, increasing the area of one partition does not affect the area of the entire chip, and designs requiring high I/O bandwidth (such as network processors) are pad-limited. Furthermore, by expanding this work to remove large cells only from timing critical paths, similar delay results with much smaller area increases are expected since standard cells on critical paths are responsible for only a small fraction of overall cell area.

The critical paths and signal directions of benchmarks wb_dma (65nm node) and systemcaes (45nm node) for configurations 'Original' and 'No Large Cells' are visualized in Figure 8. For benchmark wb_dma, the path is noticeably shorter, has fewer long wires and no inserted buffers in the 'No Large Cells' configuration, yielding an improvement of 6.3% in critical path delay. Benchmark systemcaes in 45nm node is also a good example of effectively converting wire inserted to send the signal to the distant location, whereas a number of small standard cells are spread out to serve as a repeater while also performing logic operation in the 'No Large Cells' case.

Table 2: Dynamic and leakage power comparison between 'Original' and 'No Large Cells' scheme for (a) 65nm and (b) 45nm technology.

(a) 65nm

Benchmark	Dynamic power		Leakage power	
	Original (mW)	No Large Cells	Original (μW)	No Large Cells
s35932	5.3	+0.6%	47.2	+14.8%
wb_dma	2.8	+1.1%	34.1	+5.9%
des_perf	26.4	-4.5%	448.4	0%
wb_conmax	11.7	-7.9%	145.3	+22.7%
vga_lcd	22.6	+7.1%	408.1	+11.5%
mem_ctrl	2.4	+0.5%	35.3	+11.3%
aes_core	9.4	-9.0%	95.6	+2.3%
systemcaes	4.0	+6.5%	37.9	+25.2%
Average		-0.7%		+11.7%

(b) 45nm

Benchmark	Dynamic power		Leakage power	
	Original (mW)	No Large Cells	Original (μW)	No Large Cells
s35932	2.7	+1.2%	67.9	+20.0%
wb_dma	1.3	+5.1%	48.1	+0.3%
des_perf	11.9	+0.4%	706.2	+6.1%
wb_conmax	10.3	-5.4%	242.8	+22.6%
vga_lcd	9.4	+4.8%	460.1	+7.0%
mem_ctrl	1.4	+1.5%	45.9	+14.6%
aes_core	3.6	+2.1%	162.7	+12.7%
systemcaes	1.8	-4.4%	59.4	+23.7%
Average		+0.5%		+13.4%

In addition to circuit performance, power consumption is considered in our analysis. Table 2 shows both dynamic and leakage power consumption of the final netlists for 'Original' and 'No Large Cells' schemes in 65nm and 45nm technology. We used randomized switching data with average activity factor of 0.2 for each benchmark, and measured power using Synopsys NanoSim. For a few benchmarks, power consumption of the 'No Large Cells' scheme is actually lower than that of the 'Original' scheme, due to the interaction of the appreciably lower buffer count and smaller wire capacitance. For the vga_lcd benchmark, buffer count is not significantly reduced by the simplified technology mapping, resulting in 7.1% and 4.8% power increase in 65nm and 45nm, respectively. The power overhead for the wb_dma, mem_ctrl, and s35932 benchmarks is insignificant. Overall, despite the increased gate count, the capacitance of 1-input and 2-input gates is small, leading to comparable overall power consumption of the 'No Large Cells' scheme as that of the 'Original' scheme.

The leakage power overhead in Table 2 is largely proportional to the standard cell area increase in Table 1. More precisely, the reason why the leakage overhead is slightly larger than the area overhead is that small standard cells have shorter stacks of transistors leading to less stack effect and more leakage power. However, the additional leakage power is relatively small (~1/100 of dynamic power in all benchmarks) and the net effect on total power as seen in Table 2 is very low for these typical high-performance designs.

Our results on full integrated circuits motivate placement-aware technology mapping and post-placement logic restructuring, which can indeed improve timing. However, commercial tools available to us only partially include this feature, and in its absence, we demonstrate that large standard cells are not particularly useful on critical paths. The arguments from Section 2 suggest that even with placement-driven technology mapping and post-placement logic restructuring, large cells will be less useful on critical paths. An additional advantage of our approach is that breaking down large cells into smaller ones improves routability by enhancing the ability to reduce routing congestion [15-16].

Throughout these benchmark experiments for critical path delay optimization, we execute synthesis, placement, and routing for the same circuit. As a result the size of the circuit and the length of long wires will also decrease for each technology step, which is why the wire-dependent delay in the 'Original' approach in Figure 7 decreases at each technology node from 130nm to 65nm. However, when technology scaling is used to double the number of on-chip transistors, the chip size and longest wires do not shrink. If technology mapping is skipped under this assumption (higher levels of integration for scaled technologies), wire delay will dominate due to inter-module communication and we suggest that the performance improvement using the proposed approach would increase.

4. CONCLUSION

Our work offers a first-of-a-kind careful analysis of technology mapping across four technology nodes. While this step has been commonly used in logic synthesis flows, we point out that the use of large standard cells in it appears unnecessary and even harmful for high-performance designs at 65nm and below (low power designs could still benefit from technology mapping through reduced leakage). This is a consequence of uneven scaling of wire and gate delay, as well as the fact that technology mapping essentially trades gate counts for an increased number of long wires (as shown in Table 1). Empirical trends observed for large benchmark circuits mapped to 130nm, 90nm, 65nm, and 45nm libraries suggest that the 65nm node is an inflection point for the utility of large cells in technology mapping.

5. REFERENCES

[1] A. Abdollahi and M. Pedram, "A new canonical form for fast boolean matching in logic synthesis and verification," *Proc. Design Automation Conference*, pp. 379-384, 2005.

[2] G. Agosta *et al.*, "A unified approach to canonical form-based boolean matching," *Proc. Design Automation Conference*, pp. 841-846, 2007.

[3] D. Sylvester and K. Keutzer, "Getting to the bottom of deep submicron II: a global wiring paradigm," *Proc. International Symposium on Physical Design*, pp. 193-200, 1999.

[4] P. Saxena *et al.*, "Repeater scaling and its impact on CAD," *IEEE Transactions of Computer-Aided Design of Integrated Circuits and Systems*, Vol. 23, No. 4, pp. 451-463, April 2004.

[5] C. Alpert *et al.*, "Buffer insertion with accurate gate and interconnect delay computation," *Proc. Design Automation Conference*, pp. 479-484, 1999.

[6] Y. Ismail and E. Friedman, "Optimum repeater insertion based on a CMOS delay model for on-chip RLC interconnect," *Proc. International ASIC Conference*, pp. 369-373, 1998

[7] A. Nalamalpu and W. Burleson, "Repeater insertion in deep sub-micron CMOS: ramp-based analytical model and placement sensitivity analysis," *Proc. International Symposium on Circuits and Systems*, pp. 766-799, 2000.

[8] R. Otten and R. Brayton, "Planning for performance," *Proc. Design Automation Conference*, pp. 122-127, 1998.

[9] M. Pedram and N. Bhat, "Layout driven technology mapping," *Proc. Design Automation Conference*, pp. 99-105, 1991.

[10] A. Lu *et al.*, "Combining technology mapping with post-placement resynthesis for performance optimization," *Proc. International Conference on Computer Design*, pp. 616-621, 1998.

[11] G. Stenz *et al.*, "Performance optimization by interacting netlist transformations and placement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 19, No. 3, pp. 350-358, March 2000.

[12] IWLS 2005 benchmarks. http://iwls.org/iwls2005/benchmarks.html.

[13] E. Wein and J. Benkoski, "Hard macros will revolutionize SoC Design," EE Design, August 2004. http://www.eetimes.com/showArticle.jhtml?articleID=26807055.

[14] T. Chen *et al.*, "MP-trees: a packaging-based macro-placement algorithm for mixed-size designs," *Proc. Design Automation Conference*, pp. 447-452, 2007.

[15] S. Plaza, I. Markov, and V. Bertacco, "Optimizing non-monotonic interconnect using functional simulation and logic restructuring," *Proc. of ISPD*, pp. 95-102, 2008.

[16] R. Shelar, P. Saxena, X. Wang, and S. Sapatnekar, "An efficient technology mapping algorithm targeting routing congestion under delay constraints," *Proc. of ISPD*, pp. 137-144, 2005.

[17] M. Moreinis *et al.*, "Logic gates as repeater (LGR) for area-efficient timing optimization," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, Vol. 14, No. 11, pp. 1276-1281, November 2006.