Statistically Optimized Asynchronous Barrel Shifters for Variable Length Codecs*

Peter A. Beerel, Sangyun Kim, Pei-Chuan Yeh, Kyeounsoo Kim†
{pabeerel, sangyunk, pyeh, kyskim}@eiger.usc.edu
EE-Systems Dept., University of Southern California, Los Angeles, CA 90089

Abstract: This paper presents low-power asynchronous barrel shifters for variable length encoders and decoders useful in portable applications using multimedia standards. Our approach is to create multi-level asynchronous barrel shifters optimized for the skewed shift control statistics often found in these codecs. For common shifts, data passes through one level, whereas for rare shifts, data passes though multiple levels. We compare our optimized designs with the straight-forward asynchronous and synchronous designs. Both pre- and post-layout HSPICE simulation results indicate that, compared to their synchronous counterparts, our designs provide over a 40% savings in average energy consumption for a given average performance.

1 Introduction

Asynchronous circuits sometimes can consume very low average energy for a given average performance partly because of their ability to adapt to variations in chip temperature and voltage supply level. To achieve this goal, however, the asynchronous circuit should be optimized to process common data faster than rare ones and the overhead associated with the completion sensing logic should be minimized [1]. In this paper we propose novel asynchronous shifters that achieve these two objectives.

Shifting data is required in a variety of applications, including arithmetic operations, bit indexing, and variable length coding. Barrel shifters are a common design choice because they can perform multi-bit shifts in a single operation. Many researchers have explored various barrel shifters with varying emphasis on area, power [2], and performance [3, 4, 5].

Low power barrel shifter design is particularly important in variable length codecs which are useful in battery-powered portable applications using multimedia standards such as MPEG, JPEG, H.261, etc [6, 7]. In these applications, it is possible to optimize the barrel shifter for the typical codeword statistics. In particular, as illustrated in Figure 1, more than 85% of all codewords are of length 6 or less which implies that many of the most significant output bits of the barrel shifters not needed. Thus motivated, Lin and Jen proposed a technique in which the unused bits of the barrel shifter are turned off [8].

*This research has been partially supported by NSF Grant CCR-9812164 and NSF Grant CCR-9726391.

†This work was done while the author was on leave from Korea Telecom.

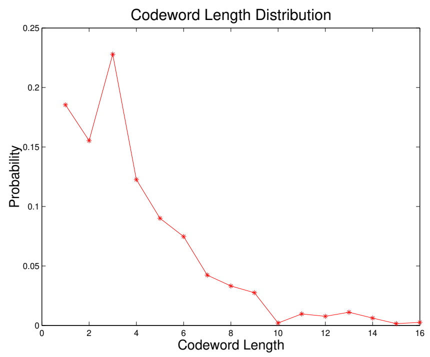

Figure 1: Codeword length distribution for MPEG video sequences (flower garden, table tennis, and mobile calendar).

This paper proposes an orthogonal approach in which the structure of the barrel shifter is optimized so that the logic executed for common (smaller) shifts is simpler and has less capacitance than that of uncommon (larger) shifts, resulting in a barrel shifter with data-dependent delay. To best take advantage of this data-dependent delay, we build completion sensing circuitry that facilitates its incorporation into an asynchronous datapath. Pre- and post-layout simulations of both dynamic and static implementations suggest that significant reductions in both average delay and average power can be achieved.

2 Asynchronous barrel shifters

In our proposed architecture, each barrel shifter output is a network of muxes arranged such that more common shifts have shorter path delays than rare shifts. Consider the simple two-level mux network depicted in Figure 2. Here, for more common shifts, $S_0 \ldots S_M$, the data is routed to the output through barrel shifter BS-1. In contrast, for less common shifts, $S_{M+1} \ldots S_{15}$, the data must pass through both barrel shifters BS-2 and BS-1.

2.1 Static design

The configuration for one output bit using static mux circuitry along with the associated completion sensing cir-

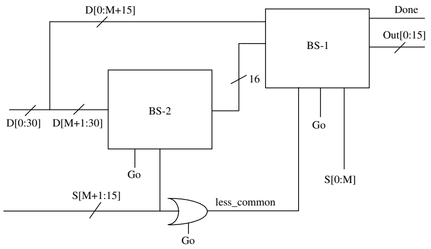

Figure 2: Two-level asynchronous barrel shifter architecture.

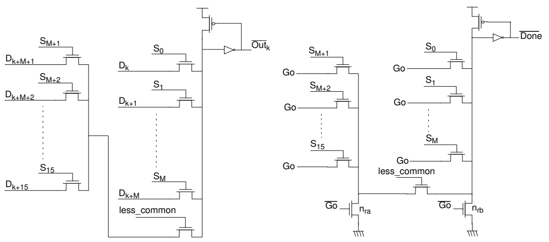

Figure 3: One shifter output designed using two-level static logic (left) along with the associated completion sensing circuitry (right).

cuitry is depicted in Figure 3.

Note that conventional completion sensing circuits either require the data inputs to be dual-rail or use the bundled data approach in which a single delay line models the worst-case component delay, both of which are not suitable for our purposes. A more recently developed idea is the speculative completion scheme [9], in which multiple delay lines are implemented and an additional mux controlled by abort circuitry is used to select one of them. Although promising, this approach seems to require the creation of the abort circuitry and an additional mux, either of which may incur significant delay overhead. In our implementation, however, we implemented the speculative completion in an efficient novel circuit that combines the necessary delay lines, mux, and abortion logic.

2.2 Dynamic design

Dynamic-logic-based shifters are faster than static-logic-based shifters because dynamic-logic-based shifters need to only pass a logic zero through the NMOS muxes. On the

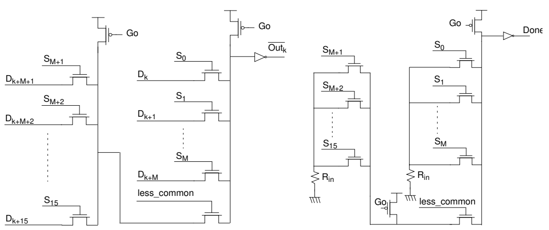

Figure 4: One shifter output designed using dynamic logic (left) along with associated completion sensing circuitry (right).

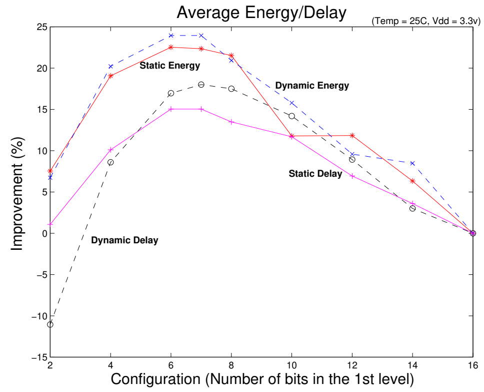

Figure 5: Average energy and delay comparison of 16 different asynchronous designs expressed as a percentage improvement over the single-level asynchronous design. The dynamic designs are simulated with all data inputs initially zero and half of them rising. The static designs are simulated with half of data inputs initially zero, half initially one, a quarter of the data inputs rising, and a quarter falling.

other hand, because outputs must be pre-charged to zero before each evaluation, if the outputs more often evaluate to one than to zero, dynamic logic implementations will consume more power than their static counterparts. Thus, the decision as to which to use depends on the speed requirement and signal statistics of the data inputs to the barrel shifter.

The implementation of our two-level barrel shifter architecture in dynamic logic is illustrated in Figure 4. As is typically done in large dynamic gates, we add an extra PMOS pull-up transistor at the output of BS-2 to avoid charge-sharing problems that might otherwise cause accidental discharge of BS-1. As in the static design, this completion sensing circuitry assumes a four-phase handshaking protocol where the delay from a shift control rising (which can occur only after `Go` rises) to rising `Done` matches the delay of the shifter and the delay from falling `Go` to falling `Done` is quick and data independent. Specifically, this latter delay is the gate's pre-charge delay.

3 HSPICE Simulation Results and Conclusions

To determine which of the 16 different possible 2-level decompositions is the best, we simulated all of them at the transistor-level using the HP 0.35μm HSPICE model. As Figure 5 indicates, the best overall design for our application is the "6-10 design", i.e., where the 6 most common shifts are handled solely by BS-1 and the other 10 shifts must pass through both BS-2 and BS-1.

As reported in Table 1, we also compare our best designs with their synchronous static and dynamic counterparts obtained by taking our one-level designs and removing the `less_common` and completion sensing circuitry. The delay improvements are approximately the same as in our earlier comparison with the one-level asynchronous design. This is because our $\overline{\texttt{Done}}$ logic models the delay of the asynchronous designs remarkably accurately.

HSPICE Simulation Results								
		Synchronous		Asynchronous (6-10 config.)				
		3.3V 25°C	3.0V 100°C	3.3V 25°C	% Imp. (D)	% Imp. (DTV)	2.5/2.1/2.3V 25°C	% Imp. (DTV)
Pre-layout	(Average Energy, pJ)	4.61	3.79	3.95	14.3%	-4.1%	2.21	41.7%
Static Muxes	(Average Delay, ns)	0.51	0.77	0.45	11.8%	41.6%	0.75	2.6%
Pre-layout	(Average Energy, pJ)	6.97	5.54	5.80	16.8%	-4.7%	2.17	60.8%
Dynamic Muxes	(Average Delay, ns)	0.31	0.45	0.26	16.1%	42.2%	0.44	2.2%
Post-layout	(Average Energy, pJ)	8.48	6.77	7.06	16.7%	-4.3%	3.24	52.2%
Dynamic Muxes	(Average Delay, ns)	0.37	0.53	0.34	5.8%	35%	5.22	1.4%

Table 1: HSPICE simulation results illustrating the improvement of the best asynchronous designs over their synchronous counterparts under various voltage and temperature assumptions. % Imp. (D) shows improvement compared to the synchronous designs when both designs are simulated at ideal voltage and typical temperature. % Imp. (DTV) shows improvement when the synchronous design is simulated at worst-case voltage and temperature.

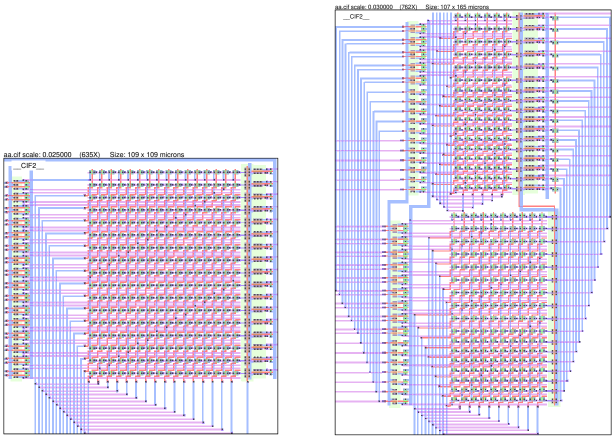

Figure 6: Layout of the (left) original (asynchronous) single-level dynamic-logic design and (right) two-level (asynchronous) 6-10 dynamic-logic design.

The improvements in average energy compared to the synchronous counterparts, however, are significantly lower than when compared to the one-level asynchronous designs due to the the energy overhead of the done logic and the generation of the `less_common` signal.

One may claim, however, that the done logic provides more than just adaptivity to the data. In particular, another advantage of the done logic is that the asynchronous design can adapt to enviromental changes in the power supply voltage level and chip temperature. Thus, it may be reasonable to compare the asynchronous design simulated at 3.3V and 25°C with the synchronous design simulated at 3.0V (worst-case voltage) and 100°C (worst-case temperature). As the table indicates, for the comparison of static (dynamic) logic designs, the improvement becomes 41% (42.2%) in average delay and -4.1% (-4.7%) in average energy (the energy loss arises because the synchronous circuit has a lower supply voltage). If we use voltage scaling, we can translate this excess speed of the asynchronous designs into savings in average energy consumption. In this case, we can run the asynchronous static (dynamic) component at 2.5V (2.1V) and achieve approximately 41.7% (60.8%) reduction in average energy consumption.

We layed out the synchronous design, the single-level asynchronous design, and the 6-10 two-level design, all using dynamic logic. The two asynchronous design layouts are shown in Figure 6. The 6-10 layout is 47% larger than the single-level asynchronous design and approximately 50% larger than the synchronous design. As reported in Table 1, the post-layout simulation results indicate that the energy improvements are not significantly altered but that the delay improvements are somewhat reduced. Considering the adaptivity to temperature and assuming architectural voltage scaling, however, the asynchronous design still yields a 52% savings in average energy consumption.

The inclusion of these designs into low power asynchronous variable length codecs (e.g., the Huffman decoder presented in [10]) is a topic of future work.

References

[1] S. Rotem, K. Stevens, R. Ginosar, P. Beerel, C. Myers, K. Yun, R. Kol, C. Dike, M. Roncken, and B. Agapiev. Rappid: An asynchronous instruction length decoder. In *ASYNC*. Apr. 1999.

[2] K. P. Acken, M. J. Irwin, and R. M. Owens. Power comparisons for barrel shifters. In *ISLPED*, pp. 209–212, 1996.

[3] R. Pereira. Fully pipelined TSPC barrel shifter for high-speed applications. *IEEE Journal of Solid-State Circuits*, pp. 686–690, June 1995.

[4] G. M. Tharakan and S. M. Kang. A new design of a fast barrel switch network. *IEEE Journal of Solid-State Circuits*, pp. 217–221, Feb. 1992.

[5] S.-J. Yih, M. Cheng, and W.-S. Feng. Multilevel barrel shifter for CORDIC design. *IEE Electronic Letters*, pp. 1178–1179, June 1996.

[6] S.-M. Lei and M.-T. Sun. An entropy coding system for digital HDTV applications. *IEEE Trans. on Video Technology*, pp. 147–155, Mar. 1991.

[7] S. Cho, T. Xanthopoulos, and A. Chandrakasan. A low power variable length decoder for MPEG-2 based on nonuniform fine-grain table partitioning. *IEEE Trans. on VLSI Systems*, vol 7., no. 2, pp. 249–257., June 1999.

[8] C.-H. Lin and C.-W. Jen. Low power parallel huffman decoding. *IEE Electronic Letters*, pp. 240–241, Feb. 1998.

[9] S. M. Nowick, K. Y. Yun, and P. A. Beerel. Speculative completion for the design of high-performance asynchronous dynamic adders. In *ASYNC*. Apr. 1997.

[10] M. Benes, S. M. Nowick, and A. Wolfe. A fast asynchronous Huffman decoder for compressed-code embedded processors. In *ASYNC*, Apr. 1998.