A Unified Design Space for Regular Parallel Prefix Adders

Matthew M. Ziegler and Mircea R. Stan
University of Virginia, ECE Department, Charlottesville, VA 22904

Abstract

We consider sparsity, fanout, and radix as three dimensions in the design space of regular parallel prefix adders and present a unified formalism to describe such structures.

Keywords: parallel prefix adder, Kogge-Stone adder, Han-Carlson adder, Brent-Kung adder.

1. Prefix Adder Design Space

Parallel prefix addition has emerged as the fastest adder solution for high performance systems. The original ideas can be found in early works [5, 6, 1, 3], more recent papers providing nice overviews [4, 11, 2, 7].

The parallel prefix adder design space has the radix-2 Kogge-Stone adder [5] at the origin. This adder structure has minimum logic depth ($O(\log_2 N)$), and full (no "sparsity") binary (radix 2) tree with minimum (2) fanout, resulting in a fast adder but with a large area. Attempts to reduce this area penalty within the scope of prefix adders come at the expense of higher radix, increased logic depth or increased fanout as shown in Fig. 1 a), or by the use of a less regular structure.

2. Radix dimension

The radix of a node in a prefix tree is determined by the fanin (number of propagate and generate signals combined at that node). In this paper we only focus on adders where the number of bits N and the radix are powers of 2.

We use a matrix R to describe the radix structure in compact form, where each row in R corresponds to the (maximum) radix in the corresponding row (or level) (L) in the prefix tree. We define the *stride* as the horizontal distance to the closest lateral connection, and the *span* as the horizontal distance to the farthest lateral connection (for radix-2 adders the stride and span are equal), see Fig. 1 b).

Considering all possible radices (2 to N) can result in up to N / 2 possible parallel prefix adders in the radix dimension of the design space, but only radices ≤ 4 are practical in CMOS implementations. Fig. 1 e) shows the possible adders for $N = 16$ and radices ≤ 4. Applying the pseudo-carry proposed by Ling [8] can be beneficial for high radix adders and represents an important dimension in the design space but we do not consider it here for lack of space.

3. Sparsity dimension

Another option in designing a prefix adder is to trade area for logic depth. The Brent-Kung adder is the extreme case of maximum logic depth and minimum area, however, there are other intermediate degrees in which area can be traded for logic depth. The Han-Carlson adder [3] is a popular example of this compromise. We have previously explored the sparsity design space within the realm of radix-2 parallel prefix adders with minimum fanout [10, 9]. We represent the sparsity of an adder with an S matrix, where the rows correspond to the degree of sparsity in each row of the prefix tree, see Fig. 1 c).

4. Fanout dimension

The Ladner-Fischer adder [6] is the extreme case with maximum fanout at all levels and minimum logic depth. Fig. 1 d) shows the full range of 8-bit parallel prefix adders with minimum radix and logic depth. Diagrams of the full range of 16-bit adders can be found in [4]. Here we use a slightly different representation for fanout: we use an F matrix that describes the degree of fanout in a manner that is independent of the sparsity and the radix.

The *total fanout* (F_T) at each level is determined from three factors: 1) the radix of the nodes at the next level in the prefix tree which dictate the minimum fanout; we represent such fanout as forward lateral connections, 2) the stride typically requires additional fanout; we refer to this type of fanout as *voluntary fanout* (F_V) and represent it as backward lateral connections in the diagrams, 3) the sparsity typically reduces the voluntary fanout by removing nodes in the tree.

For radix-2 prefix trees the S matrix has the restrictions of $S_L \leq S_{L+1}$ and $S_L \leq 2^{L-1}$, therefore in this case we can describe the S matrix with just one number, S_{max}. In general, for adders having prefix nodes of uniform radix, S_{max} and N provide sufficient information for constructing the forward tree with minimum fanout.

5. Hybrid Design Space

The combination of the R, S, and F matrices allows the description of unique prefix trees that represent different implementation points in the prefix adder design space. We call such adders "hybrid" structures in that they combine different choices on the three dimensions of the design space that was identified in this paper.

1530-1591/04 $20.00 © 2004 **IEEE**

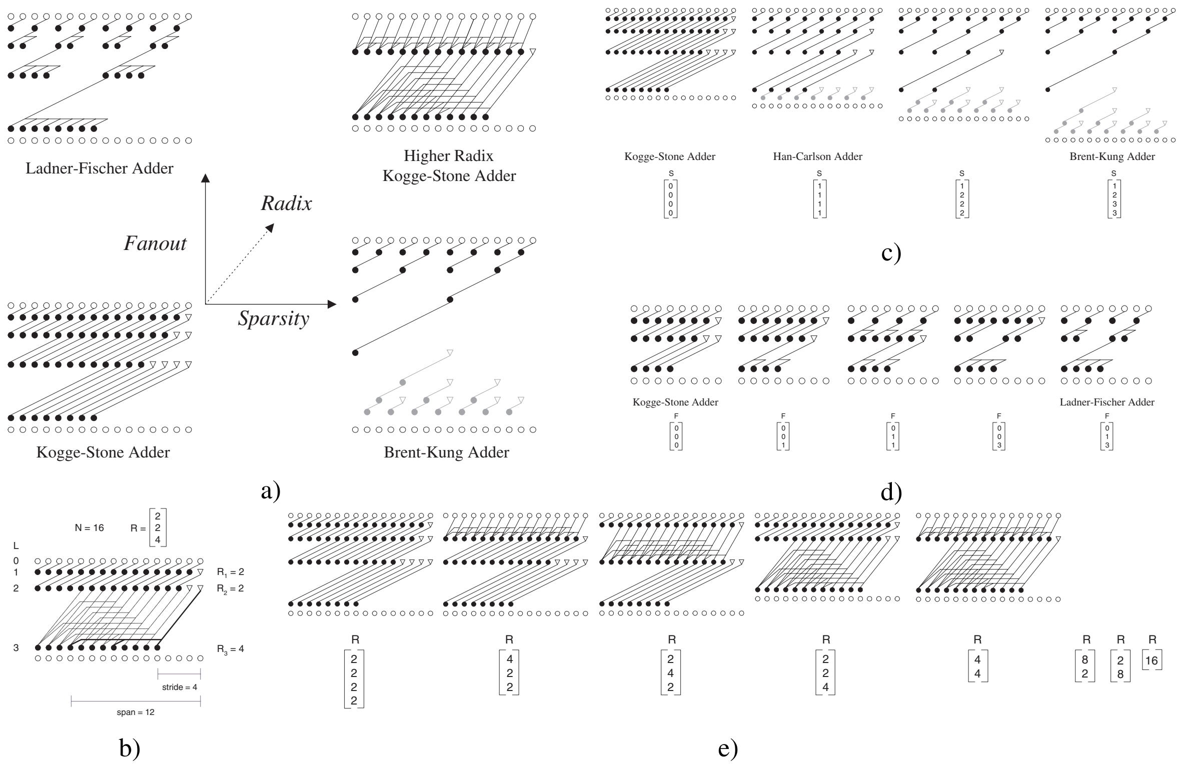

Figure 1. a) The parallel prefix adder design space along three dimensions: sparsity, fanout, and radix. b) The notation used for the parallel prefix adder design space: number of bits N, logic levels L, radix matrix R, *stride*, and *span*. c) The four possible sparsity structures for N = 16, minimum fanout, and minimum radix. f) Five structures with different fanouts for N = 8, minimum sparsity, and radix 2. e) Five possible R matrices for N = 16, with a maximum radix of 4, and minimum sparsity and fanout; the three additional combinations on the right exist for higher radices.

Once such a design space has been defined it is relatively straighforward to include it as part of a CAD tool that can explore the design space and come up with "optimal" solutions for different cost functions depending on area, delay and power dissipation. Of course good estimators for area, delay and power are also required for such an exploration of the design space.

Summary

We have presented a three dimensional design space for regular parallel prefix adders. The main objective was to develop a formalism to describe adders in the hybrid regions that incorporate variations along three dimensions of the design space: sparsity, fanout, and radix. This is still a restricted design space in that we did not consider several important possible variations: the Ling formulation for prefix adders, and all the possible non-regular tree structures that trade area and delay for lack of regularity (with the corresponding increase in design and verification effort). The regularity of the design space outlined here lends itself to adder synthesis and generation which would be natural extensions to this work.

References

[1] R. P. Brent, H. T. Kung. A regular layout for parallel adders. *IEEE Tr. Comp.*, C-31(3):260–264, Mar. 1982.

[2] F. K. Gurkaynak et al. Higher radix KS parallel prefix adder architectures. *ISCAS*, May 2000.

[3] T. Han, D. A. Carlson. Fast area-efficient VLSI adders. In *8th Symp. on Comp. Arithmetic*, May 1987.

[4] S. Knowles. A family of adders. In *Symp. on Comp. Arithmetic*, Apr. 1999.

[5] P. Kogge, H. Stone. A parallel algorithm for the efficient solution of a general class of recurrence equations. *IEEE Tr. Comp.*, C-22(8):786–793, Aug. 1973.

[6] R. E. Ladner, M. J. Fischer. Parallel prefix computation. *Journal of the ACM*, 27(4):831–838, Oct. 1980.

[7] A. Beaumont-Smith, C.-C. Lim. Parallel prefix adder design. In *Symp. on Comp. Arithmetic*, Jun. 2001.

[8] H. Ling. High speed binary adder. *IBM Journal of R&D*, 25(3):156–166, May 1981.

[9] M. M. Ziegler and M. R. Stan. Flexible IP blocks for customized synthesis. In *International ASIC Conference*, Sept. 2001.

[10] M. M. Ziegler and M. R. Stan. Optimal logarithmic adder structures with a fanout of two. In *ISCAS*, May 2001.

[11] R. Zimmermann. Efficient VLSI implementation of modulo $(2^n + 1)$ addition and multiplication. In *Symp. on Comp. Arithmetic*, Apr. 1999.