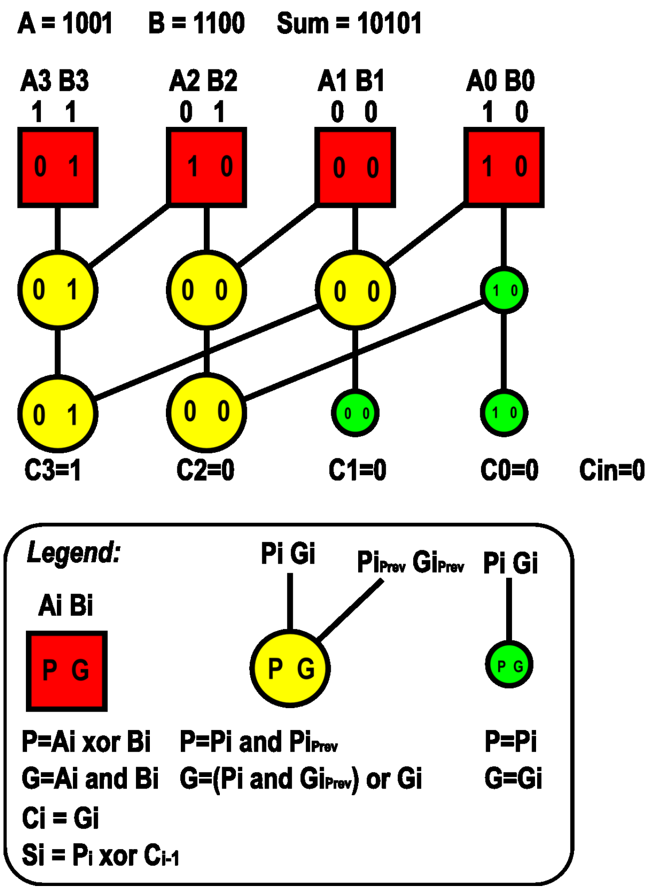
Kogge–Stone adder

The **Kogge–Stone adder** is a parallel prefix form carry look-ahead adder. It generates the carry signals in O(log n) time, and is widely considered the fastest adder design possible. It is the common design for high-performance adders in industry.

It takes more area to implement than the Brent–Kung adder, but has a lower fan-out at each stage, which increases performance. Wiring congestion is often a problem for Kogge–Stone adders as well.

An example of a 4-bit Kogge–Stone adder is shown to the right. Each vertical stage produces a "propagate" and a "generate" bit, as shown. The culminating generate bits (the carries) are produced in the last stage (vertically), and these bits are XOR'd with the initial propagate after the input (the red boxes) to produce the sum bits. E.g., the first (least-significant) sum bit is calculated by XORing the propagate in the farthest-right red box (a "1") with the carry-in (a "0"), producing a "1". The second bit is calculated by XORing the propagate in second box from the right (a "0") with C0 (a "0"), producing a "0".

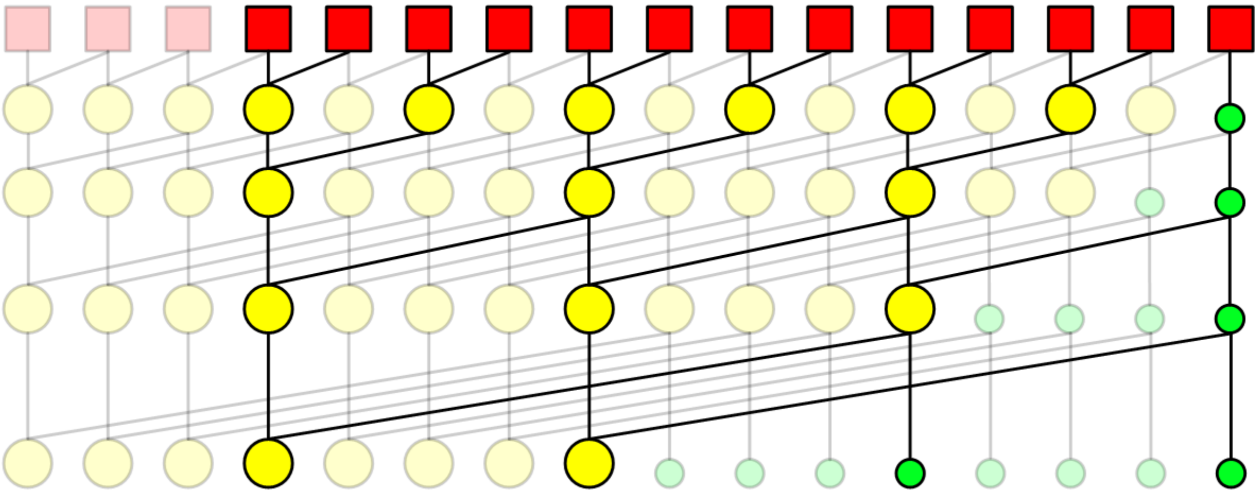


Example of a 4-bit Kogge–Stone adder with zero carry-in.

The Kogge–Stone adder concept was developed by Peter M. Kogge and Harold S. Stone, which they published in 1973 in a seminal paper titled *A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations*.[1]

Enhancements

Enhancements to the original implementation include increasing the radix and sparsity of the adder. The *radix* of the adder refers to how many results from the previous level of computation are used to generate the next one. The original implementation uses radix-2, although it's possible to create radix-4 and higher. Doing so increases the power and delay of each stage, but reduces the number of required stages. The *sparsity* of the adder refers to how many carry bits are generated by the carry-tree. Generating every carry bit is called sparsity-1, whereas generating every other is sparsity-2 and every fourth is sparsity-4. The resulting carries are then used as the carry-in inputs for much shorter ripple carry adders or some other adder design, which generates the final sum bits. Increasing sparsity reduces the total needed computation and can reduce the amount of routing congestion.



Above is an example of a Kogge–Stone adder with sparsity-4. Elements eliminated by sparsity shown marked with transparency. As shown, power and area of the carry generation is improved significantly, and routing congestion is substantially reduced. Each generated carry feeds a multiplexer for a carry select adder or the carry-in of a ripple carry adder.

Expansion

In a 4 bit adder like the one shown in the picture to the right, there are 5 outputs. Below is the expansion:

```
S0 = (A0 XOR B0) XOR Cin

S1 = (A1 XOR B1) XOR (A0 AND B0)

S2 = (A2 XOR B2) XOR (((A1 XOR B1) AND (A0 AND B0)) OR (A1 AND B1))

S3 = (A3 XOR B3) XOR ((((A2 XOR B2) AND (A1 XOR B1)) AND (A0 AND B0)) OR (((A2 XOR B2) AND (A1 AND B1)) OR (A2 AND B2)))

S4 = (A4 XOR B4) XOR ((((A3 XOR B3) AND (A2 XOR B2)) AND (A1 AND B1)) OR (((A3 XOR B3) AND (A2 AND B2)) OR (A3 AND B3)))
```

References

[1] Kogge, P. & Stone, H. "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations" (http://bwrc.eecs.berkeley.edu/Classes/ee225c/2000 225c/Papers/KoggeStone.pdf). *IEEE Transactions on Computers*, 1973, *C-22*, 783-791

External links

- Java applet to create a Kogge–Stone adder (http://people.virginia.edu/~chg5w/page4/page4.html)

Article Sources and Contributors

Kogge–Stone adder *Source*: http://en.wikipedia.org/w/index.php?oldid=473472748 *Contributors*: Cburnett, Dicklyon, Drhex, Gobelles, Goldenrowley, Greengadgetz, Oli Filth, Quanticles, Rich Farmbrough, Rilak, Tofergregg, Vadmium, Welsh, 7 anonymous edits

Image Sources, Licenses and Contributors

Image:4 bit Kogge Stone Adder Example new.png *Source*: http://en.wikipedia.org/w/index.php?title=File:4_bit_Kogge_Stone_Adder_Example_new.png *License*: Creative Commons Attribution-Sharealike 3.0 *Contributors*: 4_bit_Kogge_Stone_Adder_Example.png: Tofergregg derivative work: Greengadgetz (talk)

Image:Kogge-stone-sparsity-4.png *Source*: http://en.wikipedia.org/w/index.php?title=File:Kogge-stone-sparsity-4.png *License*: Public Domain *Contributors*: Quanticles

License