Dadda Tree (H1)

20170208

References

Some Figures from the following sites

[1] http://pages.hmc.edu/harris/cmosvlsi/4e/index.html
Weste & Harris Book Site

[2] en.wikipedia.org

The Wallace tree has three steps:

1. Multiply (that is – AND) each bit of one of the arguments, by each bit of the other, yielding n^2 results. Depending on position of the multiplied bits, the wires carry different weights, for example wire of bit carrying result of a_2b_3 is 32 (see explanation of weights below).
2. Reduce the number of partial products to two by layers of full and half adders.
3. Group the wires in two numbers, and add them with a conventional adder.[2]

The **Dadda multiplier** is a hardware multiplier design invented by computer scientist Luigi Dadda in 1965. It is similar to the Wallace multiplier, but it is slightly faster (for all operand sizes) and requires fewer gates (for all but the smallest operand sizes).[1]

In fact, Dadda and Wallace multipliers have the same 3 steps:

1. Multiply (logical AND) each bit of one of the arguments, by each bit of the other, yielding n^2 results. Depending on position of the multiplied bits, the wires carry different weights, for example wire of bit carrying result of a_2b_3 is 32.
2. Reduce the number of partial products to two by layers of full and half adders.
3. Group the wires in two numbers, and add them with a conventional adder.

https://en.wikipedia.org/wiki/Dadda_multiplier

The second phase works as follows. As long as there are three or more wires with the same weight add a following layer:

- Take any three wires with the same weights and input them into a full adder. The result will be an output wire of the same weight and an output wire with a higher weight for each three input wires.
- If there are two wires of the same weight left, input them into a half adder.
- If there is just one wire left, connect it to the next layer.

However, unlike Wallace multipliers that reduce as much as possible on each layer, Dadda multipliers do as few reductions as possible. Because of this, Dadda multipliers have a less expensive reduction phase, but the numbers may be a few bits longer, thus requiring slightly bigger adders.

To achieve this, the structure of the second step is governed by slightly more complex rules than in the Wallace tree. As in the Wallace tree, a new layer is added if any weight is carried by three or more wires. The reduction rules for the Dadda tree, however, are as follows:

https://en.wikipedia.org/wiki/Dadda_multiplier

To achieve this, the structure of the second step is governed by slightly more complex rules than in the Wallace tree. As in the Wallace tree, a new layer is added if any weight is carried by three or more wires. The reduction rules for the Dadda tree, however, are as follows:

- Take any three wires with the same weights and input them into a full adder. The result will be an output wire of the same weight and an output wire with a higher weight for each three input wires.
- If there are two wires of the same weight left, and the current number of output wires with that weight is equal to 2 (modulo 3), input them into a half adder. Otherwise, pass them through to the next layer.
- If there is just one wire left, connect it to the next layer.

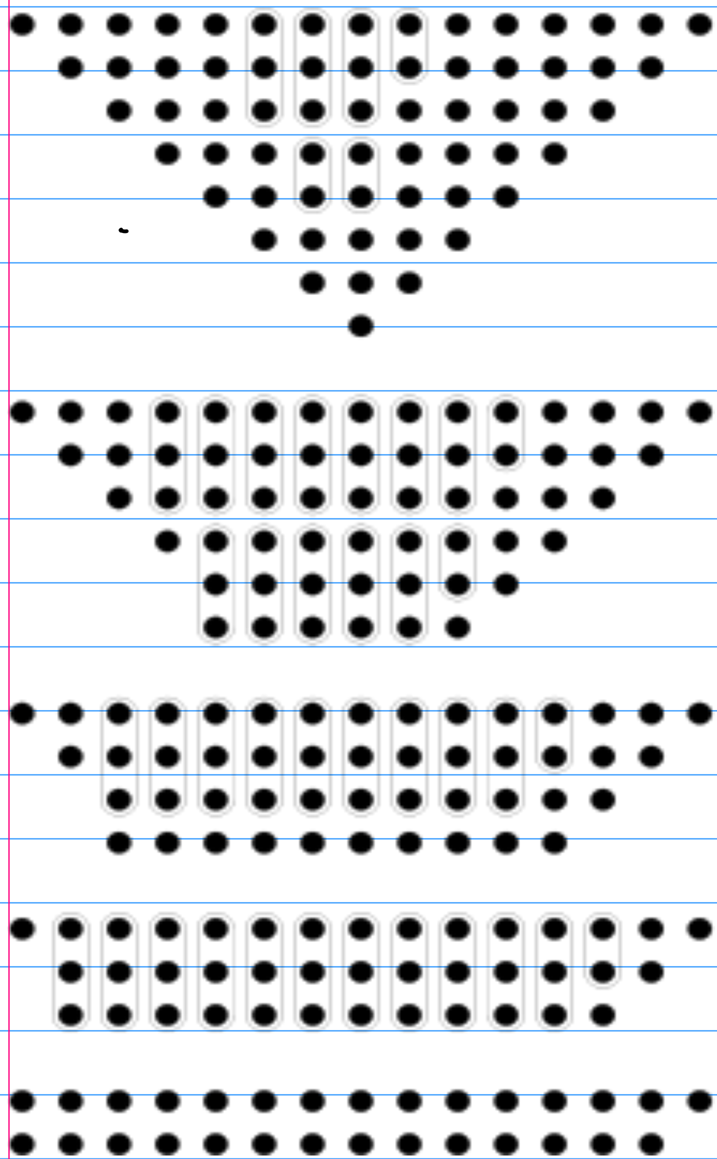
This step does only as many adds as necessary, so that the number of output weights stays close to a multiple of 3, which is the ideal number of weights when using full adders as 3:2 compressors.

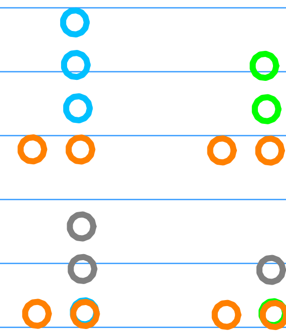
https://en.wikipedia.org/wiki/Dadda_multiplier

However, when a layer carries at most three input wires for any weight, that layer will be the last one. In this case, the Dadda tree will use half adder more aggressively (but still not as much as in a Wallace multiplier), to ensure that there are only two outputs for any weight. Then, the second rule above changes as follows:

- If there are two wires of the same weight left, and the current number of output wires with that weight is equal to 1 or 2 (modulo 3), input them into a half adder. Otherwise, pass them through to the next layer.

https://en.wikipedia.org/wiki/Dadda_multiplier





https://en.wikipedia.org/wiki/Dadda_multiplier