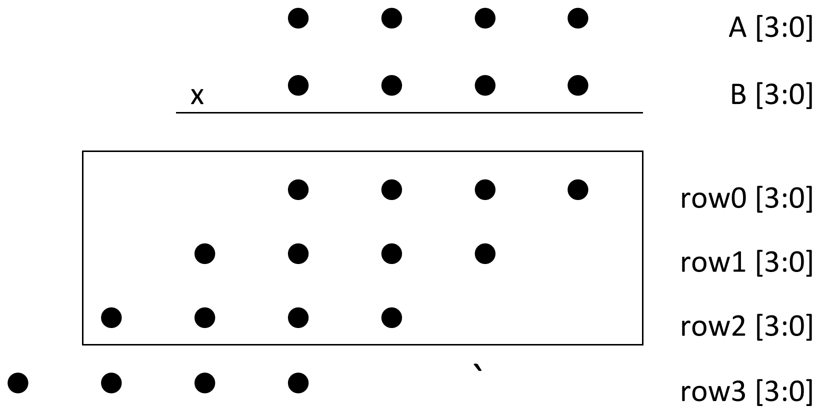
4 bits Wallace Tree Multiplier

Wallace tree is an algorithm for the efficient hardware implementation of multiplication operation. It can be constructed using half and full adders. Following is the process description of Wallace tree for 4 bits inputs A and B as coded.

1. First, bitwise multiplication is done. This is done by parallel N^2 AND gates. The result are 4 rows of partial products. This process has delay of 1 AND gate.



Then program starts reducing the layers of partial products by adders. To do so, sets of 3 rows are separated. Here a set of row0, 1 and 2 are separated, while row3 is left out for now.

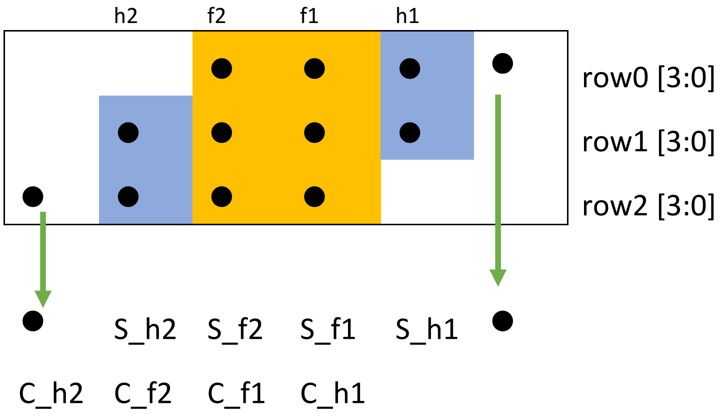
2. (Stage 1) If a column has 3 bits, they are added in the full adders (f1 and f2) and 2 bits in half adders (h1 and h2). If there is only one bit, it is brought down.

Notations

S_hx , C_hx (Sum and carry bit of half adder 'x' or hx)

S_fx , C_fx (Sum and carry bit of full adder 'x' or fx)

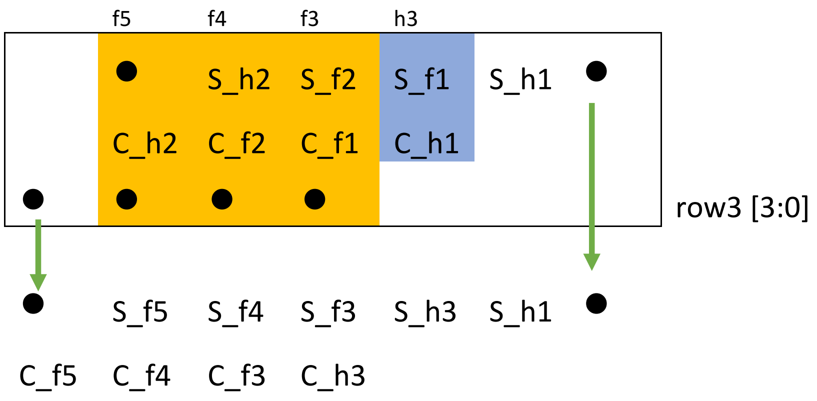
Yellow - input to full adder fx
Blue - input to half adder hx



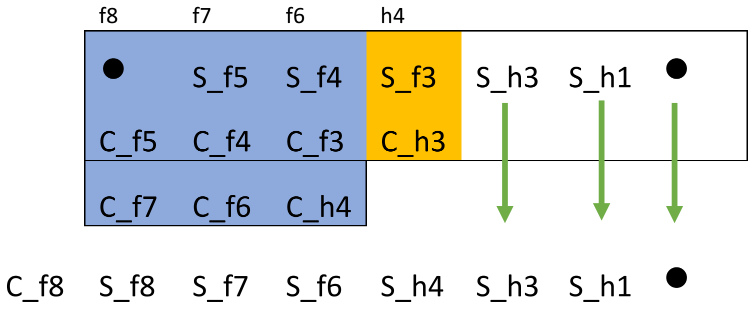
The result is 2 rows of sum and carry bits. This stage had delay of a full adder.

4 bits Wallace Tree Multiplier

1. (Stage 2) The program brings the resultant rows from Stage 1 and row3 together to create another set of 3 rows. Then the reduction process repeats once more.



2. (Final addition) In resultant 2 rows of Stage 2, the columns with 1 bits are brought down to the final results. The rest are added in **one adder** (4 bits adder here). This can be done by cascading half and full adders. Compared to the other stages, this stage has significant delay due to the propagation of carry bits.



The Final Result is 8 bits:

[MSB]							**[LSB]**
C_f8	S_f8	S_f7	S_f6	S_h4	S_h3	S_h1	row0[0]