Design of
8 x 8 Dadda Multiplier
(Self Project)

By:
Tauqeer Ahmad ,213070093
Department of Electrical Engineering, IIT Bombay

Dadda Multiplier

Dadda multipliers use 3 stages:

1. Generate all bits of the partial products in parallel.
2. Collect all partial products bits with the same place value in bunches of wires and reduce these in several layers of adders till each weight has no more than two wires.
3. For all bit positions which have two wires, take one wire at corresponding place values to form one number, and the other wire to form another number.
Add these two numbers using a fast adder of appropriate size.

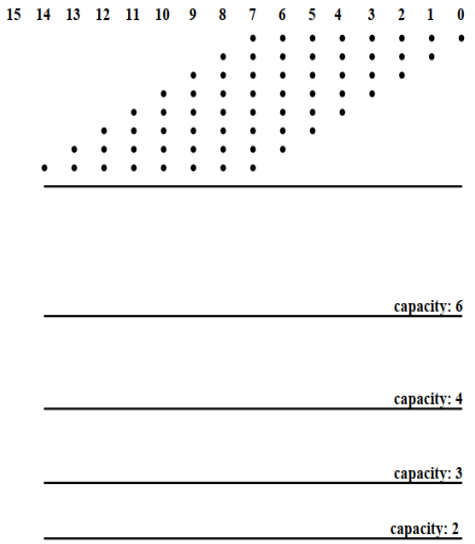
Dadda Multiplier tries to use as few adders and as small as possible (i.e prefers using HA instead of FA wherever possible). Also, it tries to reduce the number of columns as late as possible . These approaches are opposite to that of a Wallace multiplier. Dadda multipliers plan on reducing the final number of wires for any weight to 2 with as few and as small adders as possible.

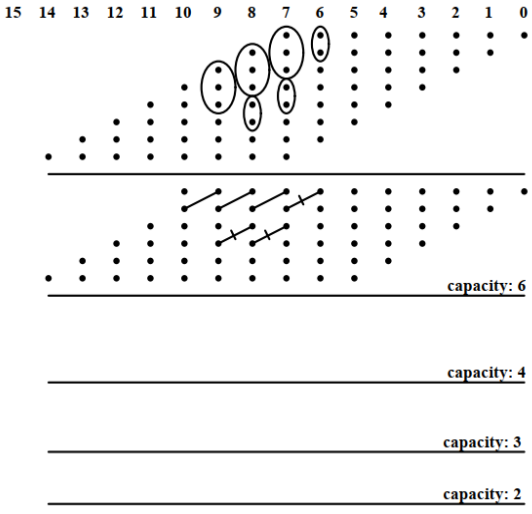
We work back from the final adder to earlier layers till we find that we can manage all wires generated by the partial product generator. The final addition will have at most 2 bits in each column. So, the capacity for the last stage should be 2.

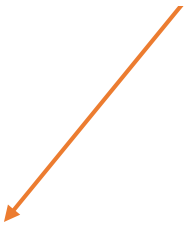
The capacity for layer i is integer of the 3/2 of the capacity of layer i-1.This implies that the capacity is 9,6,4,3,2 etc.

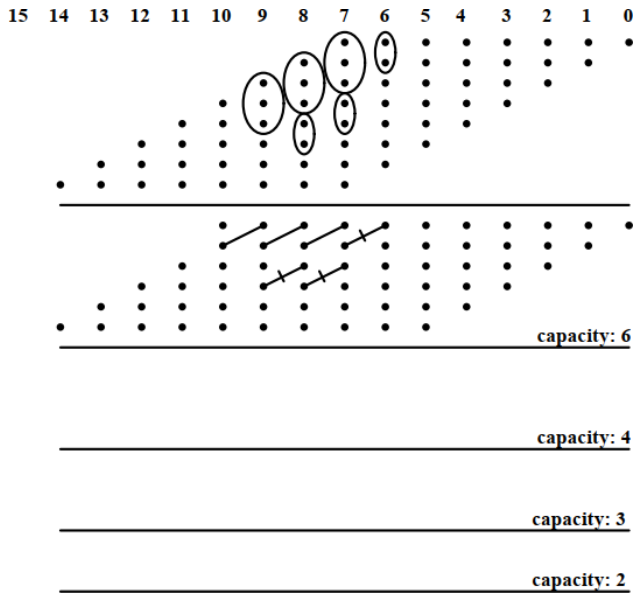
Say,We have a maximum of 8 bits in a column, then the capacity of this top layer=9. Thus, number of stages = 5 and reduction is done in 4 stages ,except the final stage where addition will be required .

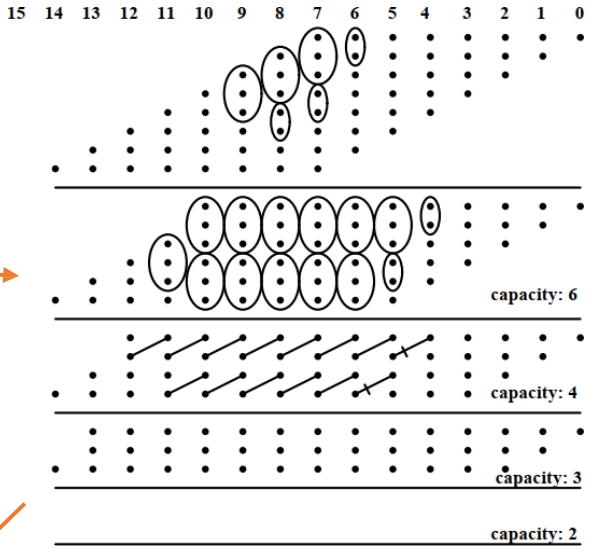
Demonstration of 8x8 multiplication:

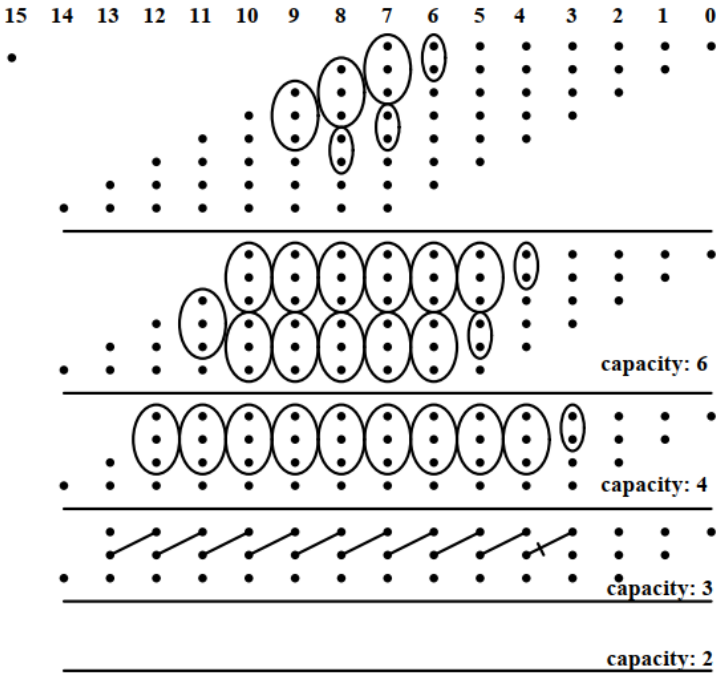


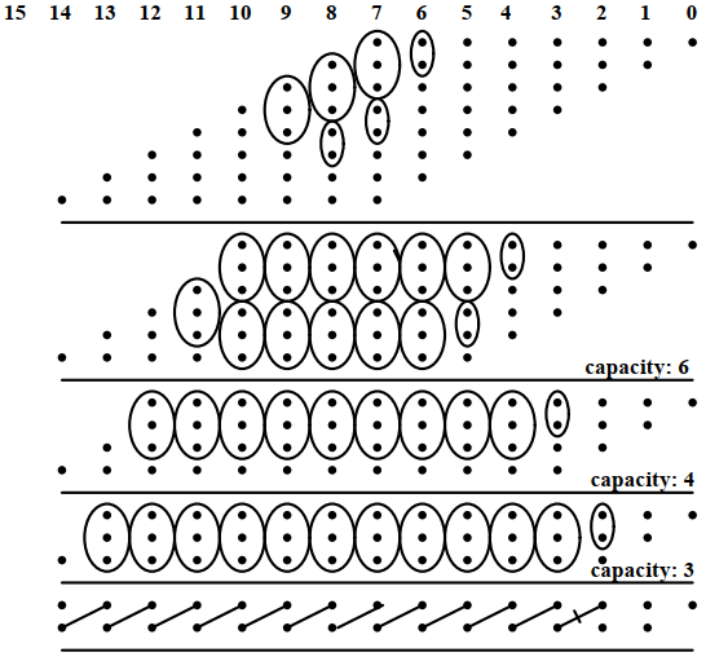





15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
capacity: 6
capacity: 4
capacity: 3
capacity: 2


15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
capacity: 6
capacity: 4
capacity: 3
capacity: 2

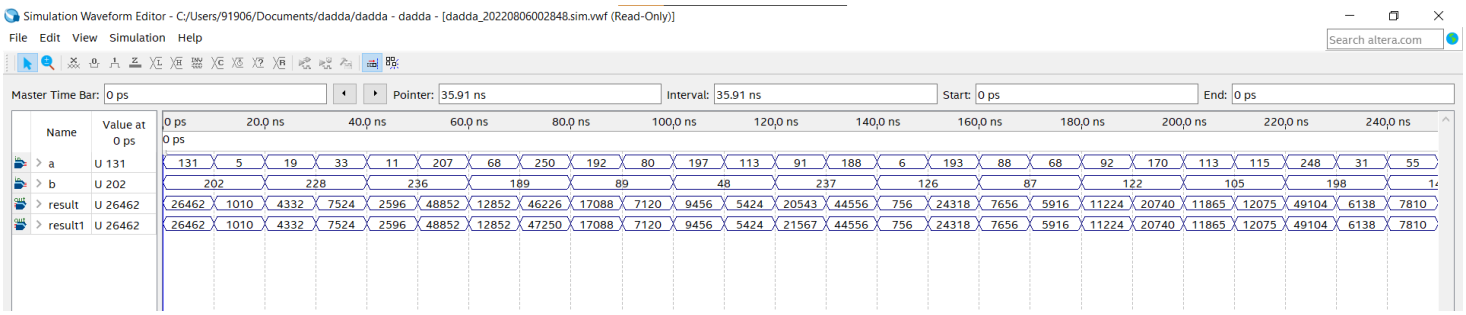

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
capacity: 6
capacity: 4
capacity: 3
capacity: 2


15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
capacity: 6
capacity: 4
capacity: 3

Simulation Results of Dadda

Synthesis and simulation done in Quartus Prime

Not considering any delay here.



a → 1st 8 bit operand (shown above in decimal format)

b→ 2nd 8 bit operand (shown above in decimal format)

result→ Obtained using Dadda's Algorithm.

Result1→ Obtained by Verilog Multiplication operation.

Can be observed that result= result1 , thus correct.

Source code shown at the end of report.