Design of 16-bit Brent Kung adder

V Joseph Jeevan reddy-203070049

July 25, 2021

Department of Electrical Engineering

IIT-Bombay

Introduction:

Adders are a crucial part in the processors. The critical path of any circuit depends on the maximum combinational delay of any circuit. So reducing the delay of ALU circuits is very crucial. Here we mainly focusses on reducing the delay of adders. A 16-bit wide Brent Kung adder is made and is compared with the ripple carry adder.

Ripple-Carry adder:

The name itself suggests the adder type. In an N-bit ripple carry adder, N-adders are connected in cascade. The carry ripples through all the stages up-to final carry. The output carry of the i^{th} stage is connected to the input carry of the $i + 1^{th}$ stage. If we assume the delay of a single stage as 'd' then, the final stage has to wait for (N-1)d units to receive its input carry and the one unit to produce its output. So, the delay of ripple carry adder is linear with N. This is a major disadvantage of this adder as it is too slow for larger adders.

Brent Kung adder:

The Brent Kung adder is a parallel prefix adder (PPA) form of carry look ahead adder. It is a logarithmic adder. Parallel in the name defines that the process involves the execution of the operation in parallel. This is done by segmentation into smaller pieces that are then computed in parallel. Then all the bits of the sum will be processed simultaneously which leads to the faster execution of operation with reduced delay. Brent kung adder works in a tree kind of structure. Due to this the delay of n-bit adder can be performed in O(log n) and with a chip area of O(n log n).

This contains three stages. They are

1. Pre Processing stage
2. Carry generation stage and
3. Post-processing stage

Pre Processing stage:

In this stage, generate and propagate signals are generated from the inputs A and B.

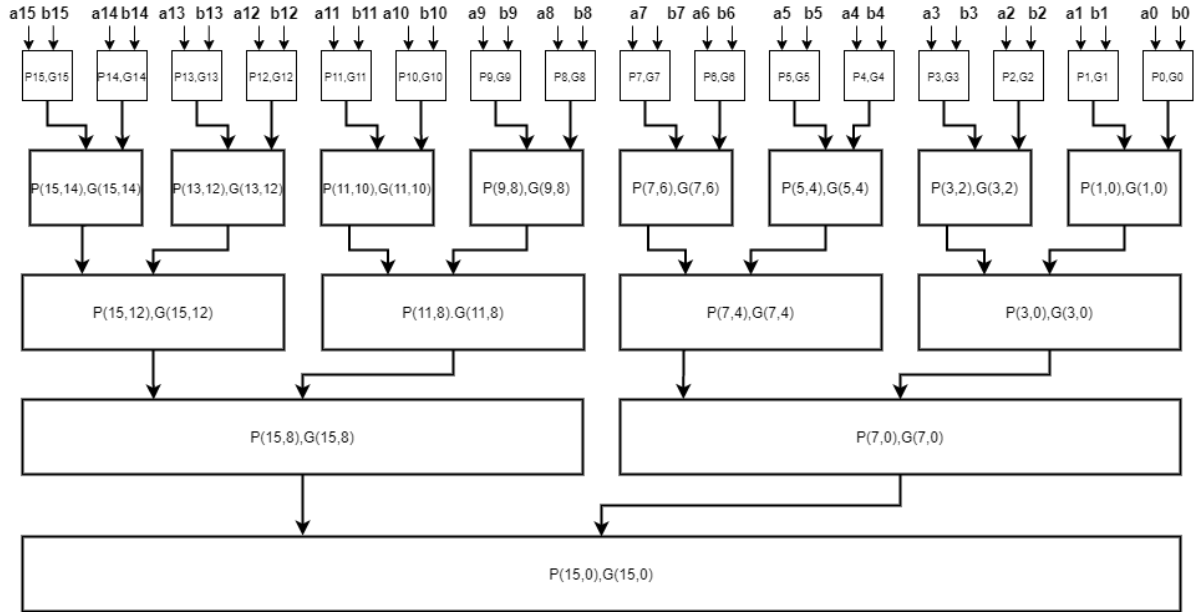
The generate and propagate signals are generated in a hierarchical fashion. Initially first order generate and propagate signals are generated and using them second order and so on and all the higher orders are calculated. Here we are designing a 16-bit adder so we can reach five orders.

Carry generation stage:

This stage follows the pre-processing stage where all the carry signals are generated using all generate and propagate signals.

Post-processing stage:

And the final stage is the post-processing stage where all the individual sum bits have been generated using propagate and the carry signals.



The flow goes as follows:

The first order generates and propagates are calculated using the equations

$G_i = A_i$ xor B_i

$P_i = A_i$ and B_i

Then the second order generates and propagates are calculated using the first order generates and propagates as

$G_{i,i-1} = G_i$ or (P_i and G_{i-1})

$P_{i,i-1} = P_i$ and P_{i-1}

The third order generates and propagates are calculated using the second order generates and propagates as

$G_{i,i-3} = G_{i,i-1}$ or ($P_{i,i-1}$ and $G_{i-2,i-3}$)

$P_{i,i-3} = P_{i,i-1}$ and $P_{i-2,i-3}$

The fourth order generates and propagates are calculated using the third order generates and propagates as

$G_{i,i-7} = G_{i,i-3}$ or ($P_{i,i-3}$ and $G_{i-4,i-7}$)

$P_{i,i-7} = P_{i,i-3}$ and $P_{i-4,i-3}$

And finally fifth order generates and propagates are calculated using the fourth order generates and propagates as

$G_{i,i-15} = G_{i,i-7}$ or ($P_{i,i-7}$ and $G_{i-8,i-15}$)

After calculation of generates and propagates, the intermediate carries are calculated. All the carries cannot be calculated at a single instant because we do not have all the intermediate values available. So, these carries are calculated step by step based on the generate and propagate values available.

And finally sums are calculated using $S_i = P_i$ xor C_i.

We can see from manual calculations that here C_{15} arrives last. So this determines the critical path.

Design:

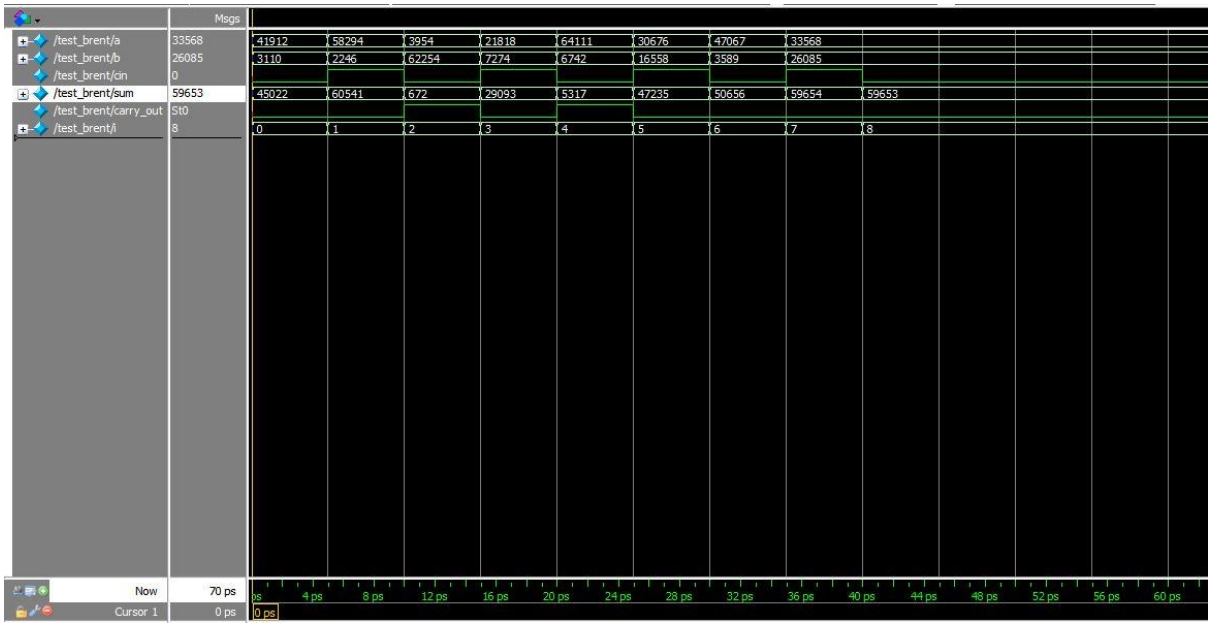
Both ripple carry adder and Brent Kung adders are designed in verilog and the appropriate waveforms have been attached.

Test-bench generation:

A random set of 8 values are generated using $random function.

Simulation results:

The waveform has been shown below.



The device used in the simulation is

Family:	Artix-7
Device:	XC7A100T
Package:	CSG324
Speed:	-3

The delays of the adders are given in the table below

	Total delay(ns)
Brent-Kung adder	4.215
Ripple-carry adder	4.467

Conclusion:

After doing the simulations, it is observed that Brent Kung adder is fast compared to the Ripple-Carry adder. The waveforms and delays are presented in the report.