Design a logarithmic adder using Brent Kung architecture for adding 32 bit operands.

Assume that the delay of some common gates (inclusive of parasitic delay) is as given below:
Inverter => 100 ps
NAND gate => 150 ps
NOR gate => 150 ps
(A+B.C)' =>200 ps
Tiny XOR => 200 ps

Defining variables as below,
$C_o,0 = G_0 + P_0 . C_i,0$
where, $C_o,0$ is output carry for 0^{th} bit.
$C_i,0$ is input carry at 0^{th} bit.
G_0 is generate carry i.e. G_0= A.B, i.e. And of A_0 &B_0.
P_0 is pass carry, i.e. $P_0 = A_0$ XOR B_0.
$C_o,1 = G_1 + P_1 . G_0 + P_1 . P_0 . C_i,0$
where, $C_o,1$ is output carry for 1^{st} bit.
$= G_{1:0} + P_{1:0} . C_i,0$
Similary,

$C_o,3 = G_3 + P_3 . C_0,2$
$= G_3 + P_3 . G_2 + P_3 . P_2 . G_1 + P_3 . P_2 . G_0 .. P_1 + P_3 . P_2 . P_1 . P_0 . C_i,0$
$= (G_3 + P_3 . G_2) + P_3 . P_2 (C_0,1)$
$= G_{3:2} + P_{3:2} . C_0,1$
$= (G_{3:0} + P_{3:0}). C_i,0$

So, $G_{i:j}$ is logic 1, if group generates a carry, independent of incoming carry.
It is true when carry either is generated at bit position i or is generated at position j and propagated through position 3.

Also, $P_{i:j}$ is logic 1, if an incoming carry propagate through the composite group.
It is true when incoming carry propage through both bit positions P_i and P_j.

Dot Operation:

(G,P) . (g,p) = (G + P.g , P.p)

Example,

$(G_{3:2} , P_{3:2}) = (G_3 + P_3) . (G_2 + P_2)$
$= (G_3 + P_3 .. G_2 , P_3 . P_2)$

So,
$(G_{3:0} , P_{3:0}) = (G_{3:2} , P_{3:2}) . (G_{1:0} , P_{1:0})$

Also,

$(C_{0,3}, 0) = \{ (G_3, P_3) . (G_2, P_2) . (G_1, P_1) . (G_0, P_0) \}.(C_{i,0}, 0)$----------1

This expression above is used for calculating the carry at individual bit after the 9th stage (refer to figure1),

Gate delays:

- sum <= (P xor c), Time: 200 ps

  P and G generation at stage i,

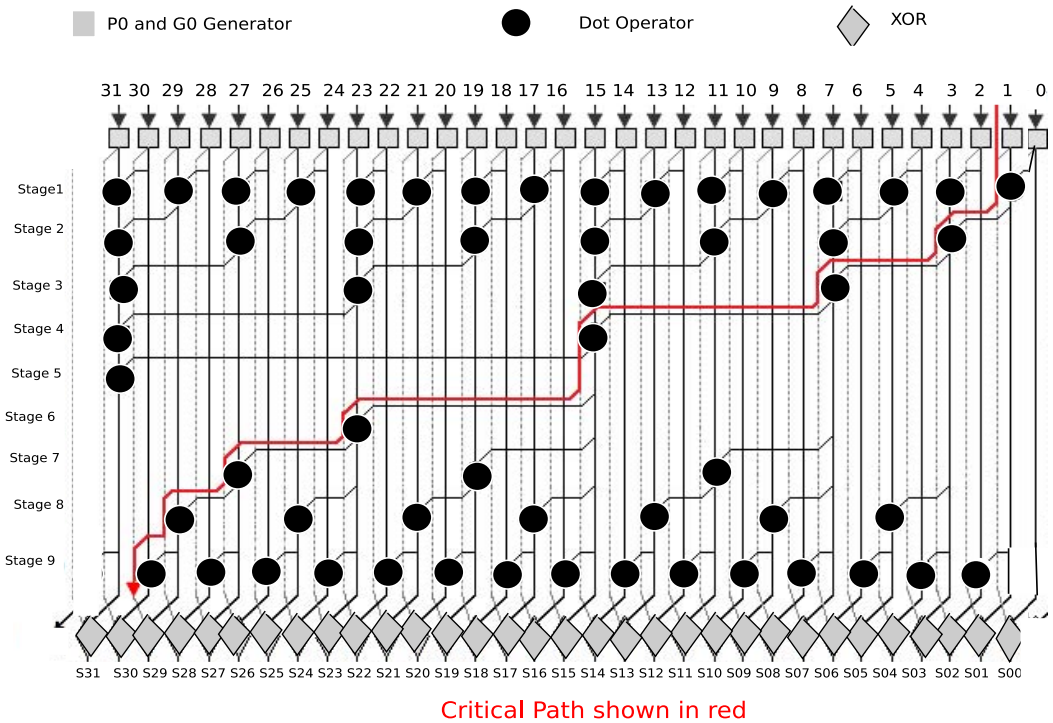
- g_i <=(a_i and b_i), Time : 250 ps
  which is combination of two gates, i.e.
  AND = NAND + Inverter
- pi <= (a_i xor b_i) , Time : 200 ps

  Dot Operation
- $G_{i,j}$ <=(g_i or(p_i and g_j)) , Time : 300 ps.
- So, its again a combination of (A+B.C)' + Inverter
- $P_{i,j}$ <= (p_i and p_j), Time : 250 ps

  Carry Generation,
- C_i <=(g_i or(p_i and c_0)) , Time : 300 ps.

As shown in figure 1, 32 bit brent kung adder has 5 stages to generate the carry out. However, to generate individual bits, we reverse traverse the tree in other 4 stages. So, there are total 9 stages for 32 bit adder.
After the 9th stage the G_9 and P_9 is genrated which is 15 bit wide. Then carry at each is calculated using the equation-1.

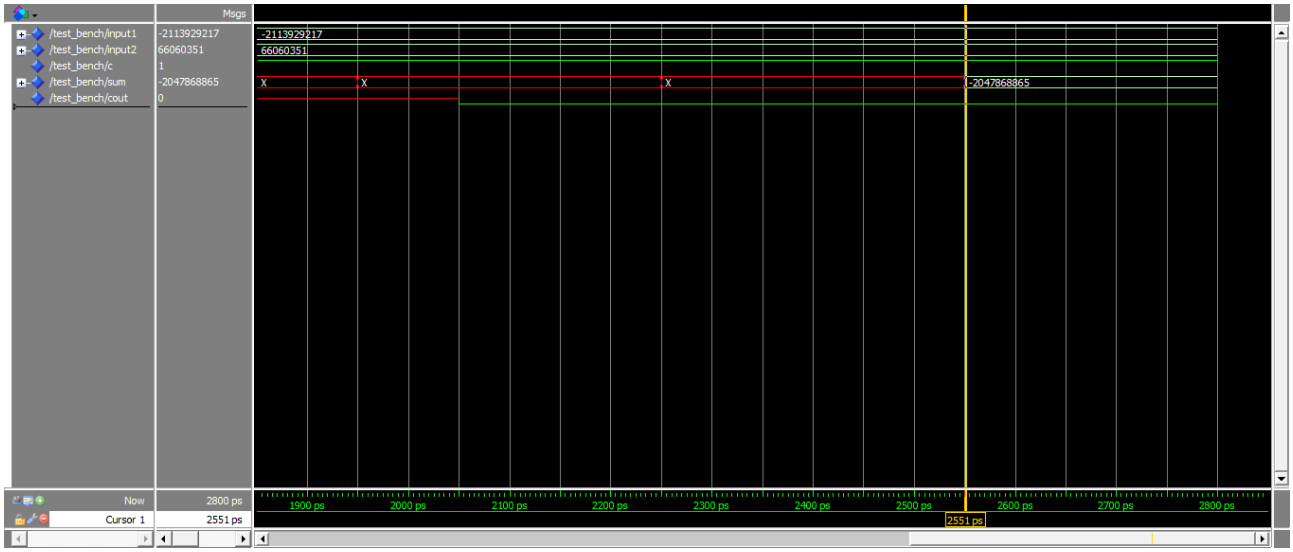
Figure-1



Critical Path shown in red

Brent Kung 32 Bit Adder

Test Cases Example,

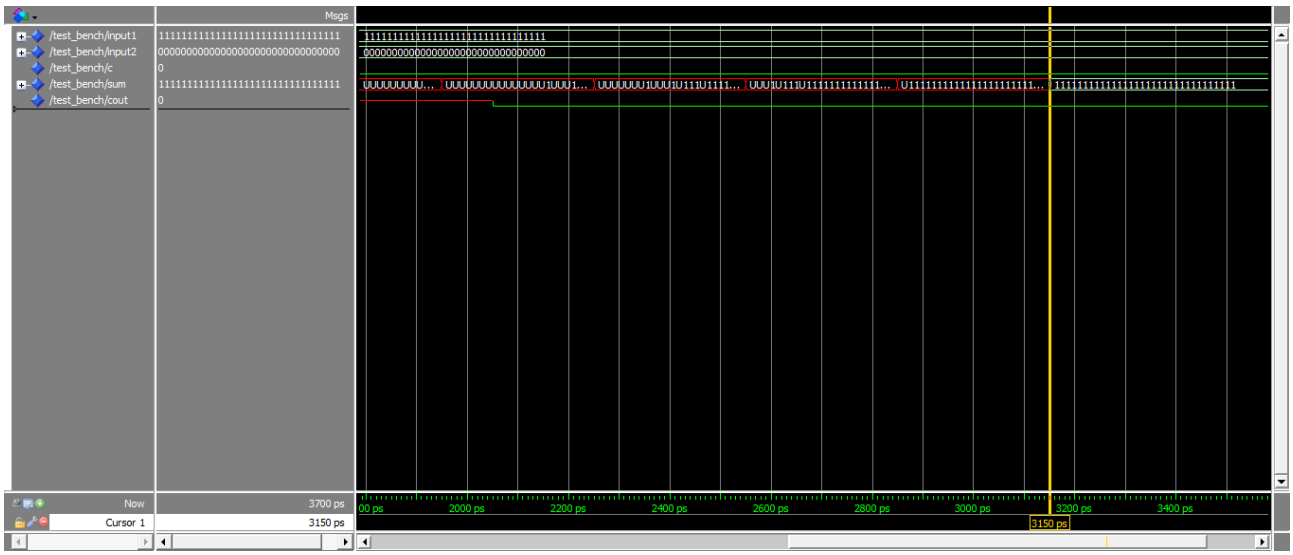


Here,
A = -2113929217
B = 66060351
C = 1
Sum = -2047868865
Carry_out = 0
Time for carry out generation = 2.050 nsec
Time for Sum out generation = 2.550 nsec

b) What is its critical path? In how much time can we guarantee that the addition will be complete?

The critical path is shown by red color line in figure 1.
The case corresponding to this path is shown below:

The worst case path is,
Generation of G_0, P_0, which takes 250psec.
Generation of C_0 which takes 300psec.
Next would be generation of C_1, C_3, C_7, C_{15}, C_{23}, C_{27}, C_{29}, C_{30}, with 300 psec each.
Next would be generation of Sum_{30} bit which takes 200psec.



A= 0b 11111111111111111111111111111111
B= 0b 00000000000000000000000000000000
C = 0b 0
So,
Total delay,
Time for output carry generation = 2.050 nsec.
Time for output sum generation = 250 + (9*300) + 200 = 3.150 nsec.