This paper proposes the design for the fastest CLA. It mentions about all the blocks and topology used in the desig
Introduction The Carry Look-Ahead Adder (CLA) is a digital circuit that is used to add two binary numbers. It is
Carry Look-Ahead Adder Design CLA design consists of three blocks: the Propagate & Generate block, Carry Lool

where G_i is the Generate signal, P_i is the Propagate signal, C_i is the Carry signal, A_i is the i^{th} bit of the first number,
Here, we are focusing on the 4-bit CLA design. Therefore, using recursive approach for carry in above equations, w

$C_1 = G_1 + P_1 \cdot C_0$
$= G_1 + P_1 \cdot (G_0 + P_0 \cdot C_{in})$
$= G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot C_{in}$

$C_2 = G_2 + P_2 \cdot C_1$
$= G_2 + P_2 \cdot (G_1 + P_1 \cdot (G_0 + P_0 \cdot C_{in}))$
$= G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0 + P_2 \cdot P_1 \cdot P_0 \cdot C_{in}$

$C_3 = G_3 + P_3 \cdot C_2$
$= G_3 + P_3 \cdot (G_2 + P_2 \cdot (G_1 + P_1 \cdot (G_0 + P_0 \cdot 0)))$
$= G_3 + P_3 \cdot G_2 + P_3 \cdot P_2 \cdot G_1 + P_3 \cdot P_2 \cdot P_1 \cdot G_0$ where C_i, G_i, and P_i are the Carry, Generate, and Propagate signals for
Improving the Basic CLA Design The above basic CLA design can be improved extensively by making the followin
Use of OR gate instead of XOR gate for Propagate block XOR gate can be replaced by OR gate which can be furth

To demonstrate that both XOR and OR gates give the same results for the Propagate block, we can see in the follo

[H] Truth Table for XOR and OR Gates

A_i	B_i	$A_i \oplus B_i$ (XOR)	$A_i + B_i$ (OR)
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	1

In case of carry generation, when both inputs are High (1), that is both $A_i = 1$ and $B_i = 1$: $G_i = A_i \cdot B_i = 1 \cdot 1 =$
$C_i = G_i + P_i \cdot C_{i-1}$
$= 1 + P_i \cdot C_{i-1}$
$= 1(regardless of P_i and C_{i-1})$ Therefore, when both inputs are 1, the carry signal is always generated ($C_i = 1$) regardles
As shown above, for the purpose of carry propagation in the CLA design, the OR gate can be used in place of XOR to

Modification of the equations for the Carry signals to reduce the number of gates
The equations for the Carry signals for basic CLA use AND and OR gates which inherently includes two extra tran

Here, we can convert AND gate between P_0, C_0 to NOR gate as shown below:

Further, we can convert OR gate to NAND gate as shown below:

Similarly, for the other carry signals, the modified equations can be written as follows: $C_1 = \overline{G'_0\left(P'_0 + \overline{C_0}\right)}$
$C_2 = \overline{G'_1\left(P'_1 + G'_0\right)} + \left(\overline{P'_1 + P'_0}\right) C_0$
$C_3 = \overline{G'_2\left(P'_2 + G'_1\right)} + \left(\overline{P'_2 + P'_1}\right) \overline{G'_0\left(P'_0 + \overline{C_0}\right)}$
$G'_{out} = \overline{\overline{G'_3\left(P'_3 + G'_2\right)} + \left(\overline{P'_3 + P'_2}\right) \overline{G'_1\left(P'_1 + G'_0\right)}}$
$P'_{out} = \overline{\overline{\left(P'_3 + P'_2\right)}\,\overline{\left(P'_1 + P'_0\right)}}$
$C_4 = \overline{G'_{out}\left(P'_{out} + \overline{C_0}\right)}$ where G'_i and P'_i are the modified Generate and Propagate signals for the i^{th} bit, respectively.
Topology and Sizing of Each Block Based on the equations and logic functions derived above, the topology of the 4
[H] [width=1]Circuit$_D iagram.png 4-bit Carry Look-Ahead Adder Design$
Propagate & Generate Block This block includes three gates for each carry bit: One NOR gate for the Propagate s
[H] [width=0.5]propgencir.png Propagate & Generate Block
Accoroding to the method of logical effort, the size of these gates is larger i.e. just double the proceedings gates of

[H] Sizing of Gates in Propagate & Generate Block

Gate	**Width (W_N/W_P) (μm)**	**Length (μm)**	**Number of Ga**
NOR	3.6 / 7.2	0.18	4
NAND	3.6 / 3.6	0.18	4
XOR	0.36 / 0.72	0.18	4

Carry Look Ahead Block This block again further divided into two more subblocks namely intermediate block and
[H] [width=0.9]clacir.png Carry Look Ahead Block for Single Carry Bit
The sizing of gates in the intermediate and AndOr block is done as per the method of logical effort. As a result, als

[H] Sizing of Gates in Intermediate and AndOr Blocks

Block	**Gate**	**Width (W_N/W_P) (μm)**	**Length (μm)**
Intermediate	NOR	0.9 / 1.8	0.18
	NAND	1.8 / 1.8	0.18
	OR	0.9 / 1.8	0.18
AndOr	AND	1.8 / 1.8	0.18
	OR	0.9 / 1.8	0.18
Inverter	-	0.9 / 1.8	0.18

Sum Block This block includes single XOR gate for the Sum signal which is implemented using Pass Transistor Log
[H] [width=0.2]sumcir.png Sum Block