Propagation delay in IIel Adder

Parallel adder carry out of Previous stage is Connected to carry in of Next stage. Hence carry is said to be propagated like ripple from LSB → MSB stage. This is called ripple carry propagation.

→ Ripple carry propagation results in time delay in addition Proces. time delay is called propagation delay

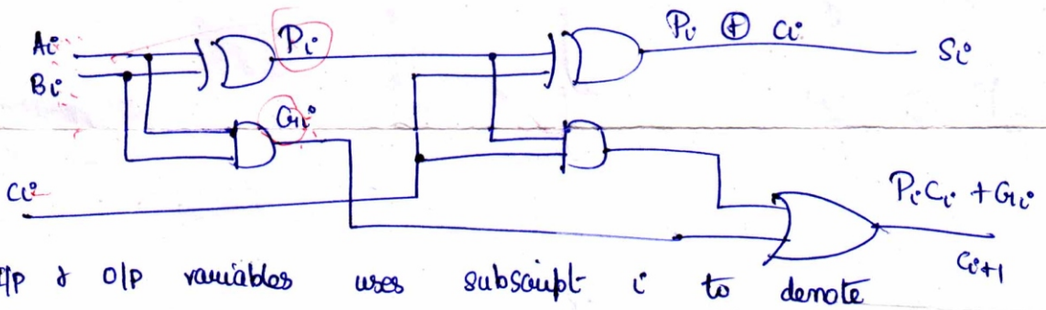
Addition of 4 bit numbers.

```
  ① ①  → carry
  0 1 1 0
  0 0 1 1
---------
  1 0 0 1
```

It is evident that addition of 1's in second Position produces a carry which is added to bits in Third Position

Signals at P_i & G_i Settle to their steady state values after they propagate Through their respective gates.

Full adder with P & G



I/P & O/P variables uses subscript i to denote typical stage of adder. P_i ⇒ Propagate G_i ⇒ Generate.

These 2 signals are Common to all full adders & depend on the I/P & O/P bits.

signal from Input carry C_i propagates Through an AND & OR gate Constitute two gate levels.

In any combinational ckt, O/P will not be Correct unless the signals are given enough time to propagate Through gates from I/P → O/P

Plm

Since all other arithmetic Operations Implemented by successive additions, time Consumed during addition Process is Critical.

Soln

TO overcome Plm, for Reducing Carry Propagation delay time is to employ faster gates with reduced delays.

However physical ckt have a limit to their Capacity Capabilities.

Other soln ⇒ Increase complexity of equipmt in such a way that carry delay time is reduced.

Some Techniques are for reducing carry propagation time in a parallel adder.

Widely used Technique is carry lookahead logic.

Consider full adder

defining by two Binary variables,

$P_i = A_i \oplus B_i$

$G_i = A_i B_i$

G_{i0} ⇒ carry generate & Produces carry of 1 when BOTH A_i & B_i are 1

o/p sum & carry ⇒ $S_i = P_i \oplus C_i$

$C_{i+1} = G_i + P_i C_i$

P_i ⇒ called as carry propagate

becz it determines whether a carry into stage i will propagate into stage i+1

Boolean fn for carry o/p of each stage:

C_0 = I/p carry

$C_{i+1} = G_i + P_i C_i$

i=0 $C_{0+1} = G_0 + P_0 C_0$

i=1 $C_2 = G_1 + P_1 C_1$

~~$G_1 + P_1 [(P_0 C_0) + G_0]$~~

$C_2 = G_1 + P_1 (G_0 + P_0 C_0)$

$G_1 + P_1 G_0 + P_1 P_0 C_0$

$C_3 = G_2 + P_2 C_2$

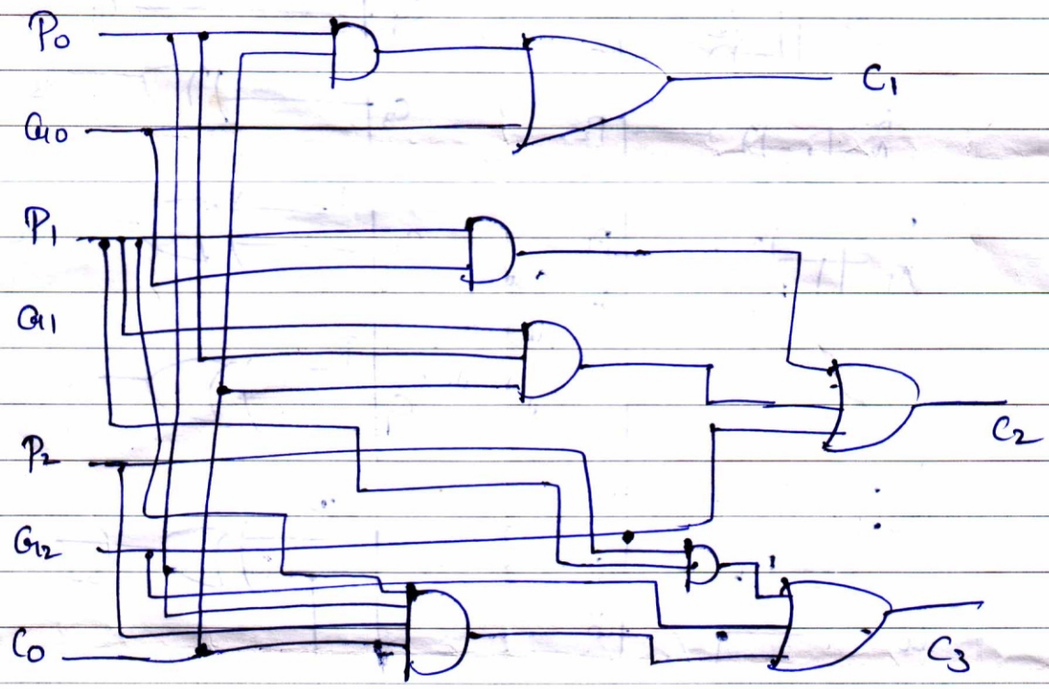
$G_2 + P_2 [G_1 + P_1 G_0 + P_1 P_0 C_0]$

$G_2 + P_2 G_1 + P_1 P_2 G_0 + P_0 P_1 P_2 C_0$

Boolean fn for each o/p carry is expressed in SOP form

Each fn can be Implemented with one level of AND gates followed by OR gate

3 Boolean fn C_0 C_1 C_2 C_3 Implemented in carry lookahead generator



4 bit IIel Adder
with look ahead carry