Fast Adders
Ripple-Carry Adder Critical Delay

- Delay grows to $2N + 2$, where $N =$ number of bits. Problem when $N$ is large, say $N = 16$. 

![Diagram of Ripple-Carry Adder](image-url)
Carry Lookahead Adder

• Normally
  \[ Si = Xi \oplus Yi \oplus Ci \]
  \[ Ci+1 = Xi Yi + Ci (Xi \oplus Yi) \]

• We can define Propagate (P) and Generate (G) functions as follows:
  \[ Pi = Xi \oplus Yi \]
  \[ Gi = Xi Yi \]
  \[ Si = Pi \oplus Ci \]
  \[ Ci+1 = Gi + Pi Ci \]

• Then
  \[ C1 = G0 + P0 C0 \]
  \[ C2 = G1 + P1 C1 \]
  \[ C3 = G2 + P2 C2 \]
  \[ C4 = G3 + P3 C3 \]
We can “flatten” the following equations
C1 = G0 + P0 C0
C2 = G1 + P1 C1
C3 = G2 + P2 C2
C4 = G3 + P3 C3

To obtain
C1 = G0 + P0 C0
C2 = G1 + G0 P1 + C0 P0 P1
C3 = G2 + G1 P2 + G0 P1 P2 + C0 P0 P1 P2
C4 = G3 + G2 P3 + G1 P2 P3 + G0 P1 P2 P3 + C0 P0 P1 P2 P3
Carry Lookahead Adder

- Once “flattened”, all Ci’s can be computed with @3 delay.
- Since Si = Pi ⊕ Ci, all Si’s can be computed with @4 delay.
- Problem: the AND and OR gates get increasingly large w.r.t. “i”. i.e., difficult to implement larger than 5-input AND or 5-input OR.
Carry Lookahead Adder

- Cascaded carry lookahead logic
  - Four 4-bit adders with internal carry lookahead
  - Second-level carry lookahead unit extends lookahead to 16 bits
  - $S_{15@8}$ for carry-lookahead vs. $2(15) + 2 = @32$ for $C_{16}$ of ripple-carry adder

\[
G = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0
\]
\[
P = P_3 P_2 P_1 P_0
\]

\[
C_2 = G_1 + P_1 G_0 + P_1 P_0 C_0
\]

\[
C_1 = G_0 + P_0 C_0
\]
Carry Select Adder

- Redundant hardware to make carry calculation go faster
  - Compute two high-order sums in parallel while waiting for carry-in
  - One assuming carry-in = 0, another assuming carry-in = 1
  - Select correct result once carry-in is finally computed