G and P and K, Oh My!

- Most fast adders “G”enerate, “P”ropagate, or “K”ill the carry
  - Usually only G and P are used; K only appears in some carry chains
- When does a bit Generate a carry out?
  - G_i = A_i AND B_i
  - If G_i is true, then $Cout_i = C_{i+1}$ is forced to be true
- When does a bit Propagate a carry in to the carry out?
  - P_i = A_i XOR B_i
  - If P_i is true, then $Cout_i$ ($=C_{i+1}$) follows C_i
  - Usually implemented as P_i = A_i OR B_i
    - OR is cheaper/faster than an XOR
    - If you are doing logic, Cout is still equal to $G_i + P_iC_i$
    - Just beware that Sum_i != P_i XOR C_i

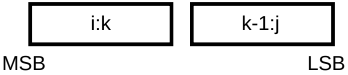
Using G and P

- We can combine G_i and P_i into larger blocks
  - Call these "group generate" and "group propagate" terms

- When does a group Generate a carry out? (*e.g.*, 4 bits)
  - $G_{3:0} = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0$

- When does a group Propagate a carry in to the carry out?
  - $P_{3:0} = P_3P_2P_1P_0$

- We can also combine groups of groups
  - $G_{i:j} = G_{i:k} + P_{i:k}G_{k-1:j}$
  - $P_{i:j} = P_{i:k}P_{k-1:j}$



Faster Carry Bypass (or Carry Skip) Adders

- We see the basic idea is to form multi-level carry chains
  - Break the bits into groups
  - Ripple the carry in each group, in parallel
  - Ripple the global carry across the groups

- How big should each group be? (N bits total, k bits per group)
  - If ripple time equals block skip time then delay **=** $2(N-1)+(N/k - 2)$

- Would groups of different sizes be faster? (yes)
  - Middle groups have longer to generate carry outs; should be larger
  - Early and late groups have ripples in critical path; should be shorter
  - Called “Variable Block Adders”