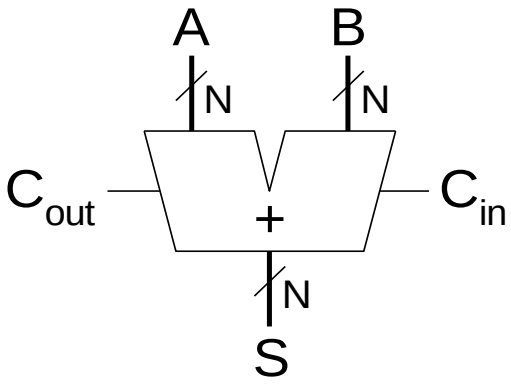
Multibit Adder, also called CPA

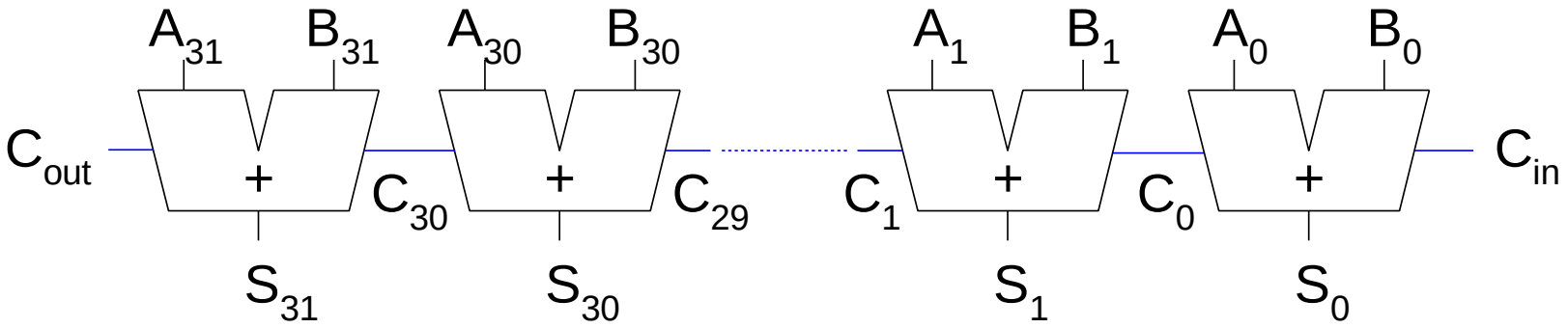
- Several types of carry propagate adders (CPAs) are:
  - Ripple-carry adders (slow)
  - Carry-lookahead adders (fast)
  - Prefix adders (faster)
- Carry-lookahead and prefix adders are faster for large adders but require more hardware.

Symbol



Ripple-Carry Adder

- Chain 1-bit adders together
- Carry ripples through entire chain
- Disadvantage: slow



Ripple-Carry Adder Delay

- The delay of an *N*-bit ripple-carry adder is:

$$t_{\text{ripple}} = Nt_{FA}$$

where t_{FA} is the delay of a full adder

Carry-Lookahead Adder

- Compute carry out (C_{out}) for k-bit blocks using *generate* and *propagate* signals
- **Some definitions:**
  - A column (bit i) produces a carry out by either *generating* a carry out or *propagating* a carry in to the carry out.
  - Generate (G_i) and propagate (P_i) signals for each column:
    - A column will generate a carry out if A_i AND B_i are both 1.

$$G_i = A_i B_i$$

    - A column will propagate a carry in to the carry out if A_i OR B_i is 1.

$$P_i = A_i + B_i$$

    - The carry out of a column (C_i) is:

$$C_i = A_i B_i + (A_i + B_i) C_{i-1} = G_i + P_i C_{i-1}$$

Carry-Lookahead Addition

- Step 1: compute *generate* (*G*) and *propagate* (*P*) signals for columns (single bits)
- Step 2: compute *G* and *P* for *k*-bit blocks
- Step 3: C_{in} propagates through each *k*-bit propagate/generate block

Carry-Lookahead Adder

- For example, we can calculate generate and propagate signals for a 4-bit block ($G_{3:0}$ and $P_{3:0}$) :
  - A 4-bit block will generate a carry out if column 3 generates a carry (G_3) or if column 3 propagates a carry (P_3) that was generated or propagated in a previous column as described by the following equation:

    $$G_{3:0} = G_3 + P_3 (G_2 + P_2 (G_1 + P_1 G_0)$$

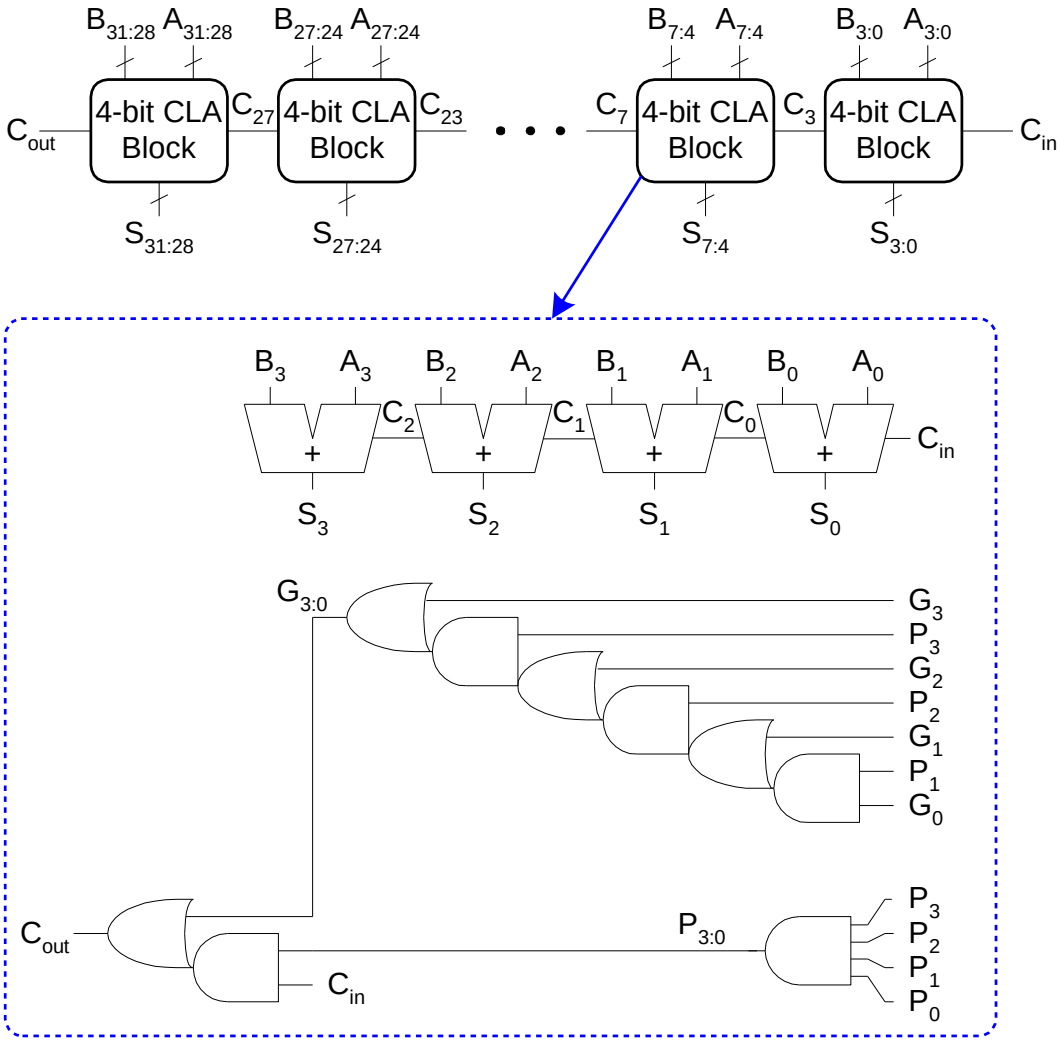
  - A 4-bit block will propagate a carry in to the carry out if all of the columns propagate the carry:

    $$P_{3:0} = P_3 P_2 P_1 P_0$$

  - The carry out of the 4-bit block (C_i) is:

    $$C_i = G_{i:j} + P_{i:j} C_{i-1}$$

32-bit CLA with 4-bit blocks



Carry-Lookahead Adder Delay

- Delay of an N-bit carry-lookahead adder with k-bit blocks:

$$t_{CLA} = t_{pg} + t_{pg\_block} + (N/k - 1)t_{AND\_OR} + kt_{FA}$$

where

  - t_{pg} : delay of the column generate and propagate gates
  - t_{pg_block} : delay of the block generate and propagate gates
  - t_{AND_OR} : delay from C_{in} to C_{out} of the final AND/OR gate in the k-bit CLA block

- An N-bit carry-lookahead adder is generally much faster than a ripple-carry adder for $N > 16$

Prefix Adder

- Computes the carry in (C_{i-1}) for each of the columns as fast as possible and then computes the sum:

$$S_i = (A_i \oplus B_i) \oplus C_i$$

- Computes G and P for 1-bit, then 2-bit blocks, then 4-bit blocks, then 8-bit blocks, etc. until the carry in (generate signal) is known for each column
- Has $\log_2 N$ stages

Prefix Adder

- A carry in is produced by being either *generated* in a column or *propagated* from a previous column.
- Define column -1 to hold C_{in}, so

  $G_{-1} = C_{in}$, $P_{-1} = 0$
- Then, the carry in to column i = the carry out of column i-1:

  $C_{i-1} = G_{i-1:-1}$

  $G_{i-1:-1}$ is the generate signal spanning columns i-1 to -1.

  There will be a carry out of column i-1 (C_{i-1}) if the block spanning columns i-1 through -1 generates a carry.
- Thus, we can rewrite the sum equation as:

  $S_i = (A_i \oplus B_i) \oplus G_{i-1:-1}$
- **Goal:** Quickly compute $G_{0:-1}$, $G_{1:-1}$, $G_{2:-1}$, $G_{3:-1}$, $G_{4:-1}$, $G_{5:-1}$, … (These are called the *prefixes)*

Prefix Adder

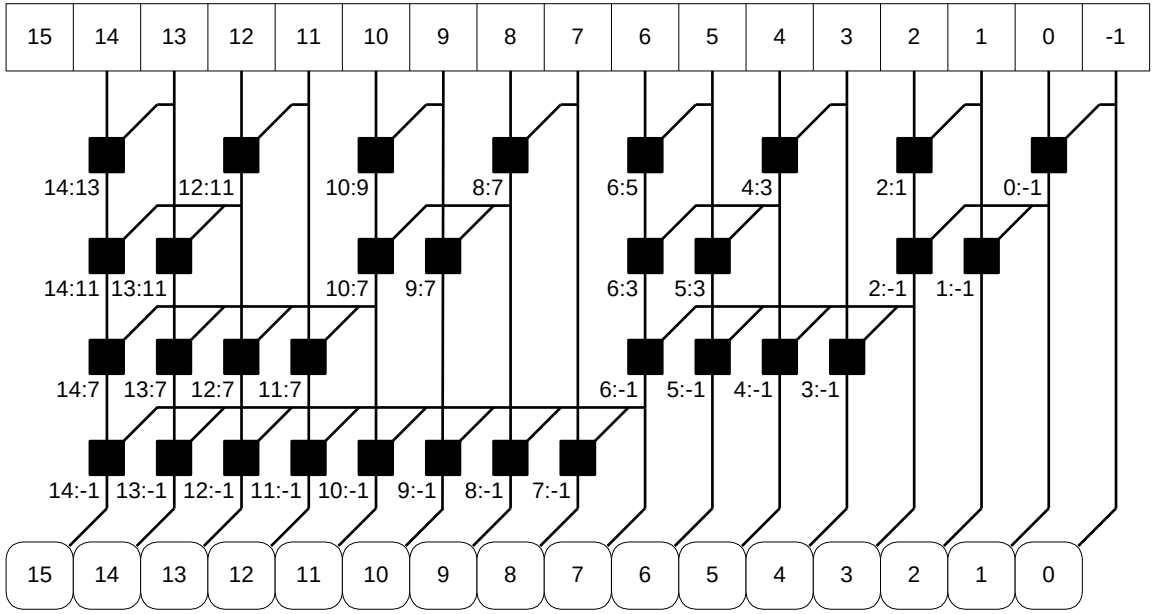
- The generate and propagate signals for a block spanning bits $i{:}j$ are:

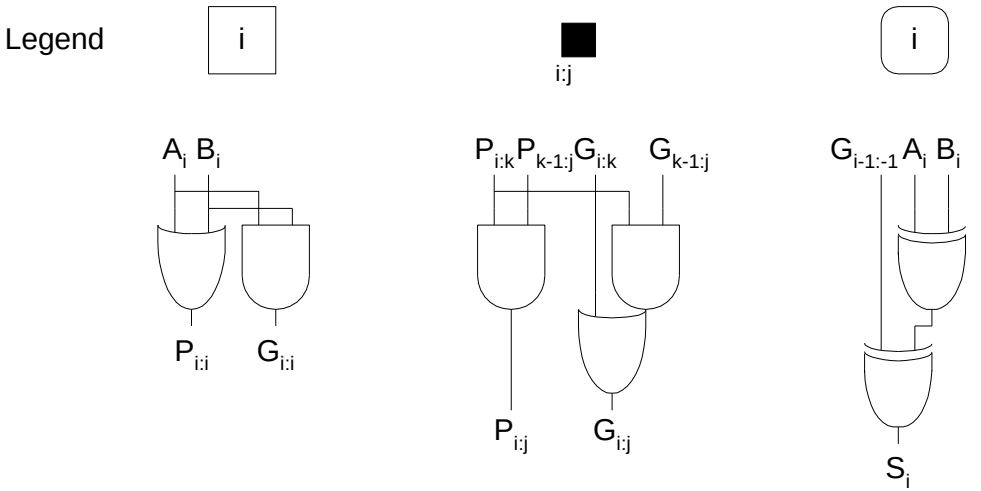
$$G_{i:j} = G_{i:k} + P_{i:k}\, G_{k-1:j}$$

$$P_{i:j} = P_{i:k} P_{k-1:j}$$

- In words, these prefixes describe that:
  - A block will generate a carry if the upper part ($i{:}k$) generates a carry or if the upper part propagates a carry generated in the lower part (k-1:j)
  - A block will propagate a carry if both the upper and lower parts propagate the carry.

Prefix Adder Schematic





Prefix Adder Delay

- The delay of an *N*-bit prefix adder is:

$$t_{PA} = t_{pg} + \log_2 N(t_{pg\_prefix}) + t_{XOR}$$

where

  - t_{pg} is the delay of the column generate and propagate gates (AND or OR gate)
  - t_{pg_prefix} is the delay of the black prefix cell (AND-OR gate)