32-bit Prefix Adder Workflow

Input Breakdown:

- Inputs are two 32-bit binary numbers: `A[31:0]` and `B[31:0]`.
- A carry-in (`C_in`, typically 0) may also be included.

Step 1: Generate and Propagate:

- Compute the **Generate (G)** and **Propagate (P)** signals for each bit: G[i]=A[i]⋅B[i](Carry generated by bit i)G[i] = A[i] \cdot B[i] \quad (\text{Carry generated by bit } i)G[i]=A[i]⋅B[i](Carry generated by bit i) P[i]=A[i]⊕B[i](Carry propagated by bit i)P[i] = A[i] \oplus B[i] \quad (\text{Carry propagated by bit } i)P[i]=A[i]⊕B[i](Carry propagated by bit i)

Step 2: Prefix Processing (Carry Calculation):

- Use the prefix logic to compute the carry (`C[i]`) for each bit: C[i]=G[i]+(P[i]⋅C[i−1])C[i] = G[i] + (P[i] \cdot C[i-1])C[i]=G[i]+(P[i]⋅C[i−1])
- This is done in stages, where each stage combines pairs of generate and propagate signals:
    - **Stage 1**: Combine adjacent bits.
    - **Stage 2**: Combine pairs of results from Stage 1.
    - Repeat until carry for all bits is computed.

Step 3: Compute the Sum:

- Calculate the sum bit for each position using: S[i]=P[i]⊕C[i−1]S[i] = P[i] \oplus C[i-1]S[i]=P[i]⊕C[i−1]

Output Assembly:

- Combine the sum bits (`S[31:0]`) to produce the final 32-bit sum.
- The final carry (`C[32]`) indicates overflow.