Ex: Add 4 8-bit inputs → 12-bit sum

Method #0: cover with 4 8 adders

Method 1a:

Method 2:

\[ s = s = w_2 = w_6 = w_0 = w_0 = 0 \]

\[ w_7 (m_8) = 0, 1 \]

\[ 0: \hspace{1cm} 0 \hspace{1cm} 0 \hspace{1cm} 0 \hspace{1cm} 0 \]

\[ 1: \hspace{1cm} 1 \hspace{1cm} 1 \hspace{1cm} 1 \hspace{1cm} 1 \]
Carry-Propagate Adders

1) Carry-Select
2) Carry-Lookahead
3) Conditional Sum
4) Carry-Skip
5) Other parallel prefix adders
   - Kogge-Stone, 1973
   - Brent-Kung, 1982
[Parhami: Ch. 7]
Carry-Lookahead

A) Generate

\[
\begin{array}{c}
\text{GEN} = 1 \\
\end{array}
\]

B) Propagate

\[
\begin{array}{c}
\text{Ci} = 0, \text{Cont} = 0 \\
\text{Ci} = 1, \text{Cont} = 1 \\
\text{Cout} = \text{Ci} \rightarrow \text{Cont} \\
\text{PRES} = 1 \\
\end{array}
\]
\[ \text{Cont} = \text{Gen OR (PROP AND cin)} \]
- Pre-examine input bits
- Set Gen, Prop
- \( \text{cin} \rightarrow \text{Cont} \) is very fast

**Single-bit Memories**

1) Clock-less latches
2) Transparent level-sensitive latch
3) Edge-triggered FF
reg a;
always @ (posedge clk) begin
    a = a_c;
end

Use non-blocking assignment

a <= a_c

Rule 41 - Always use FAs with non-blocking assignment