

# SCE 0110 - Elementos de Lógica Digital I

## **Representação Numérica e Circuitos Aritméticos (continuação)**

Prof. Dr. Vanderlei Bonato

# Conteúdo extra

- Subtração binária
- Circuito subtrator

# Carry-Lookahead Adder

- Apresenta uma solução para reduzir o atraso causado pela propagação dos bits de transporte (carry) existentes no “ripple-carry adder”
- Tem como estratégia antecipar se o bit “carry-in” oriundo do estágio anterior terá valor 0 ou 1
- Com essa antecipação, o desempenho de todo o somador pode ser melhorado

# Carry-out function for stage $i$

$$c_{i+1} = x_i y_i + x_i c_i + y_i c_i$$

$$c_{i+1} = g_i + p_i c_i \quad \text{[Expression 5.3]}$$

$g_i$  = generate function

$p_i$  = propagate function

Expanding the carry-out function in terms of stage  $i-1$  gives

$$\begin{aligned}c_{i+1} &= g_i + p_i(g_{i-1} + p_{i-1}c_{i-1}) \\ &= g_i + p_i g_{i-1} + p_i p_{i-1} c_{i-1}\end{aligned}$$

The same expansion for other stages, ending with stage 0, gives

$$\begin{aligned}c_{i+1} &= g_i + p_i g_{i-1} + p_i p_{i-1} g_{i-2} + \cdots + p_i p_{i-1} \cdots p_2 p_1 g_0 + \\ &\quad p_i p_{i-1} \cdots p_1 p_0 c_0\end{aligned}$$

Linha colorida é o caminho crítico ( $2n + 1 \Rightarrow n$  em bits)

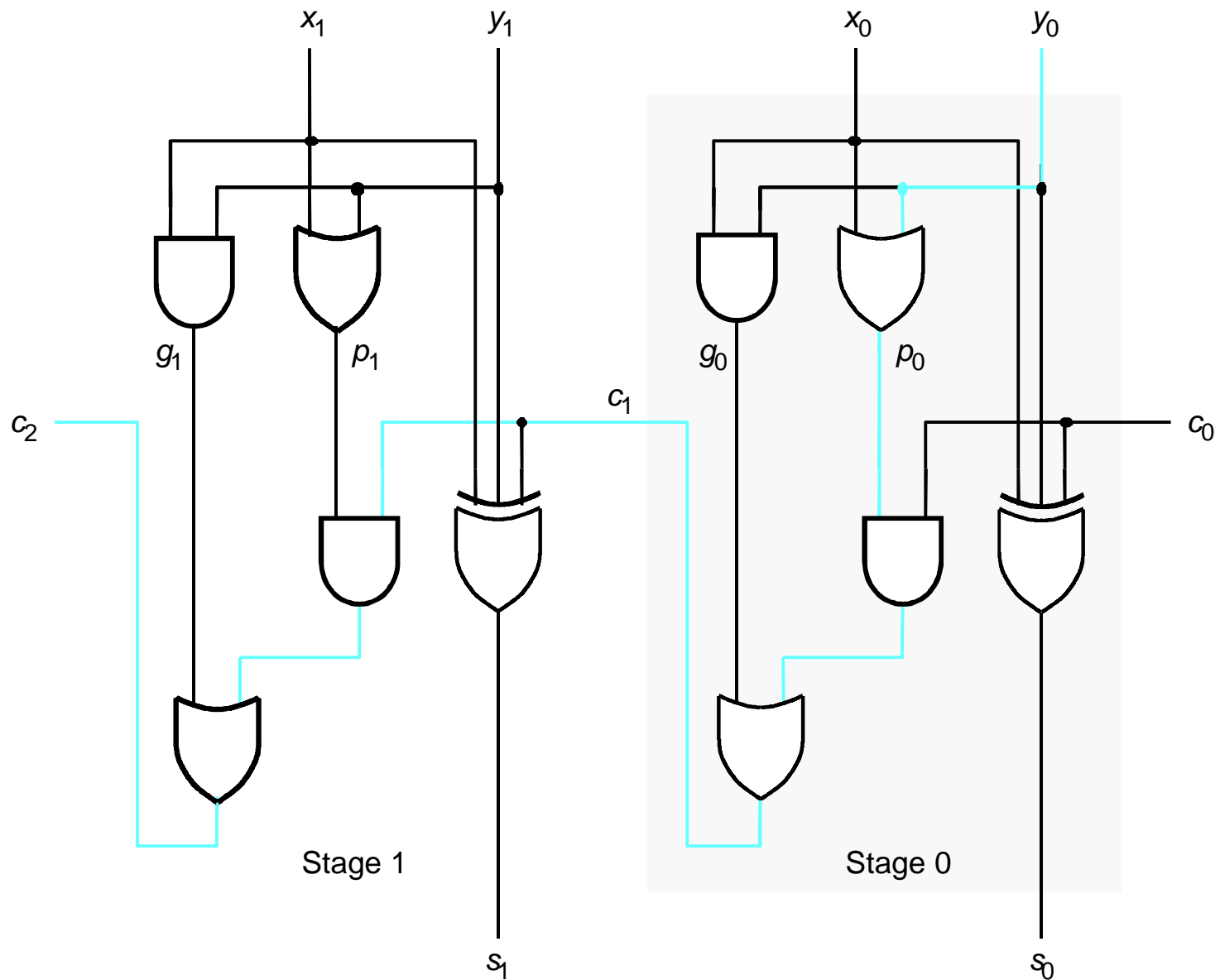


Figure 5.15. A ripple-carry adder based on expression 5.3.

Linha colorida é o caminho crítico (3 portas para qualquer  $n$  - atraso total para soma de 4 portas)

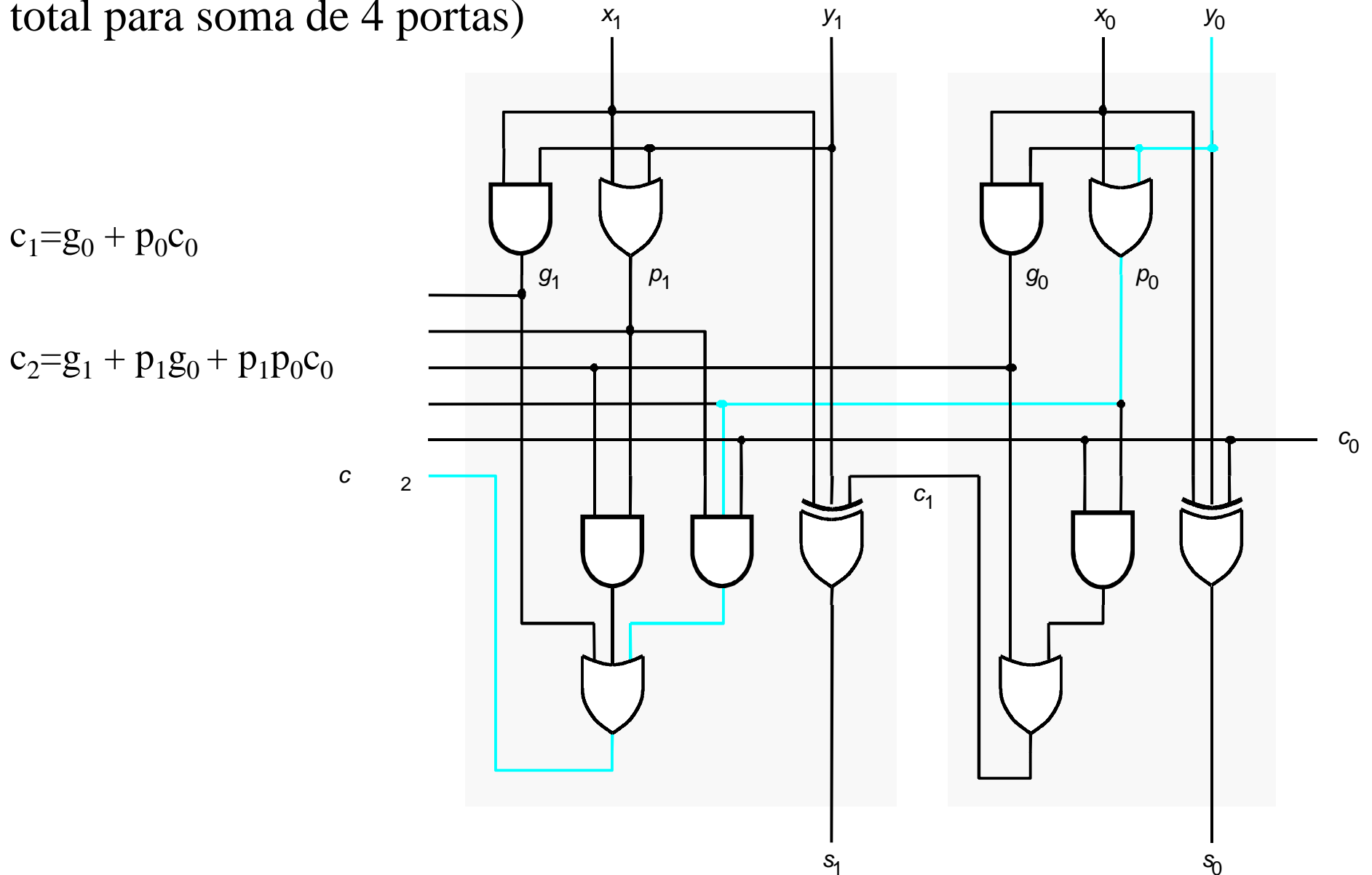


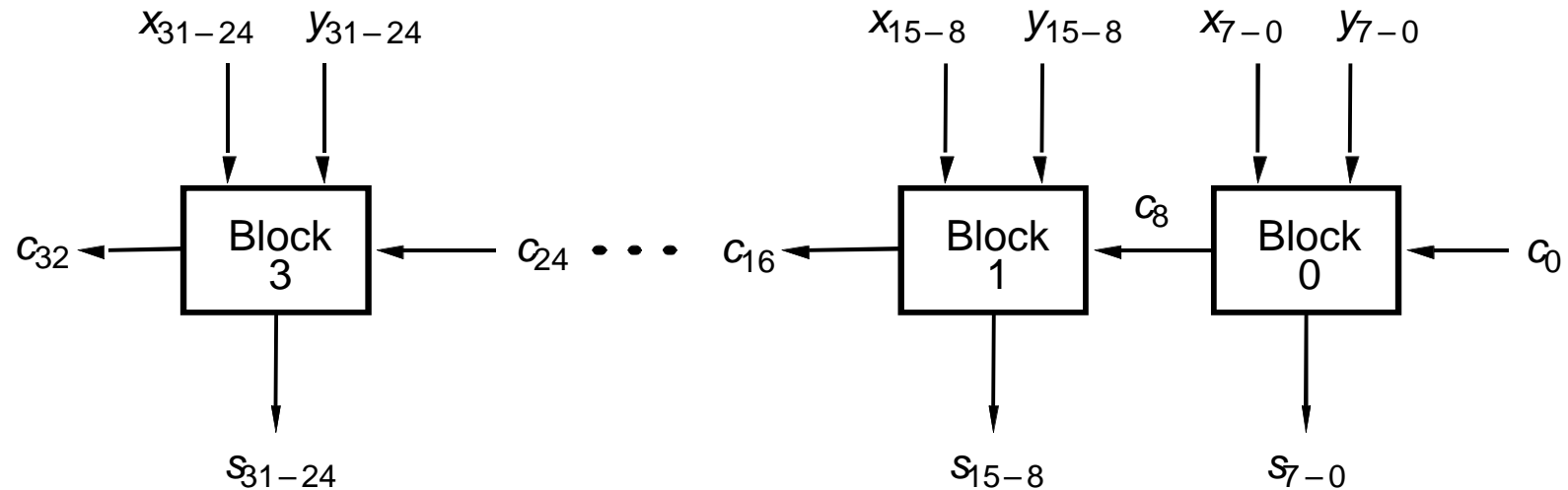
Figure 5.16. The first two stages of a carry-lookahead adder.

# Aumento da complexidade

- A complexidade do circuito aumenta significativamente em função de  $n$
- Dividir o número em blocos
  - Ex: 32 bits -> 4 blocos de 8bits cada
- Cada bloco poderia ser considerado como um estágio do somador ripple-carry
- Assim, teríamos carry-lookahead dentro do bloco e ripple-carry entre os blocos
- Mas, um circuito mais rápido poderia ser construído se um segundo nível de carry-lookahead fosse introduzido para propagar os sinais também entre os blocos



# Somador carry-lookahead com ripple-carry



$$c_8 = g_7 + p_7g_6 + p_7p_6g_5 + p_7p_6p_5g_4 + \dots + p_7p_6p_5p_4p_3p_2p_1p_0c_0$$

Figure 5.17. A hierarchical carry-lookahead adder with ripple-carry between blocks.

# Somador carry-lookahead hierárquico (delay de 8 portas para 32 bits)

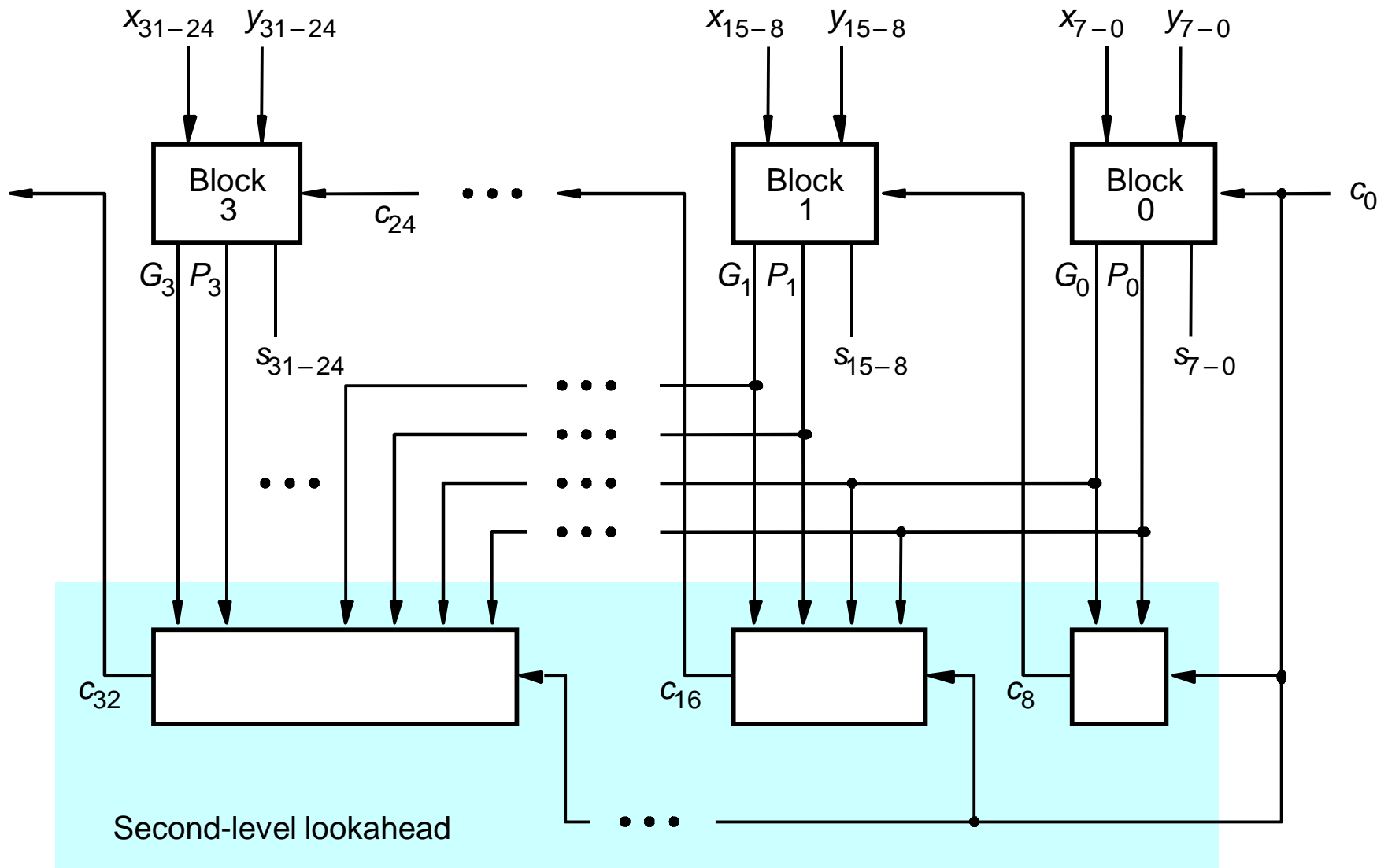


Figure 5.18. A hierarchical carry-lookahead adder.

# Restrições de *fan-in*

- Deve ser levado em consideração, pois no caso visto anteriormente, um bloco precisaria de portas AND e OR de até 9 entradas
- Realizar a fatoração da função lógica para adequar a tecnologia
- Supondo que uma dada tecnologia tenha o *fan-in* de 4, o delay de propagação do sinal de carry de um bloco do somador carry-lookahead aumentaria de 3 para 5 portas lógicas

# Ex de limitação com XOR de duas entradas

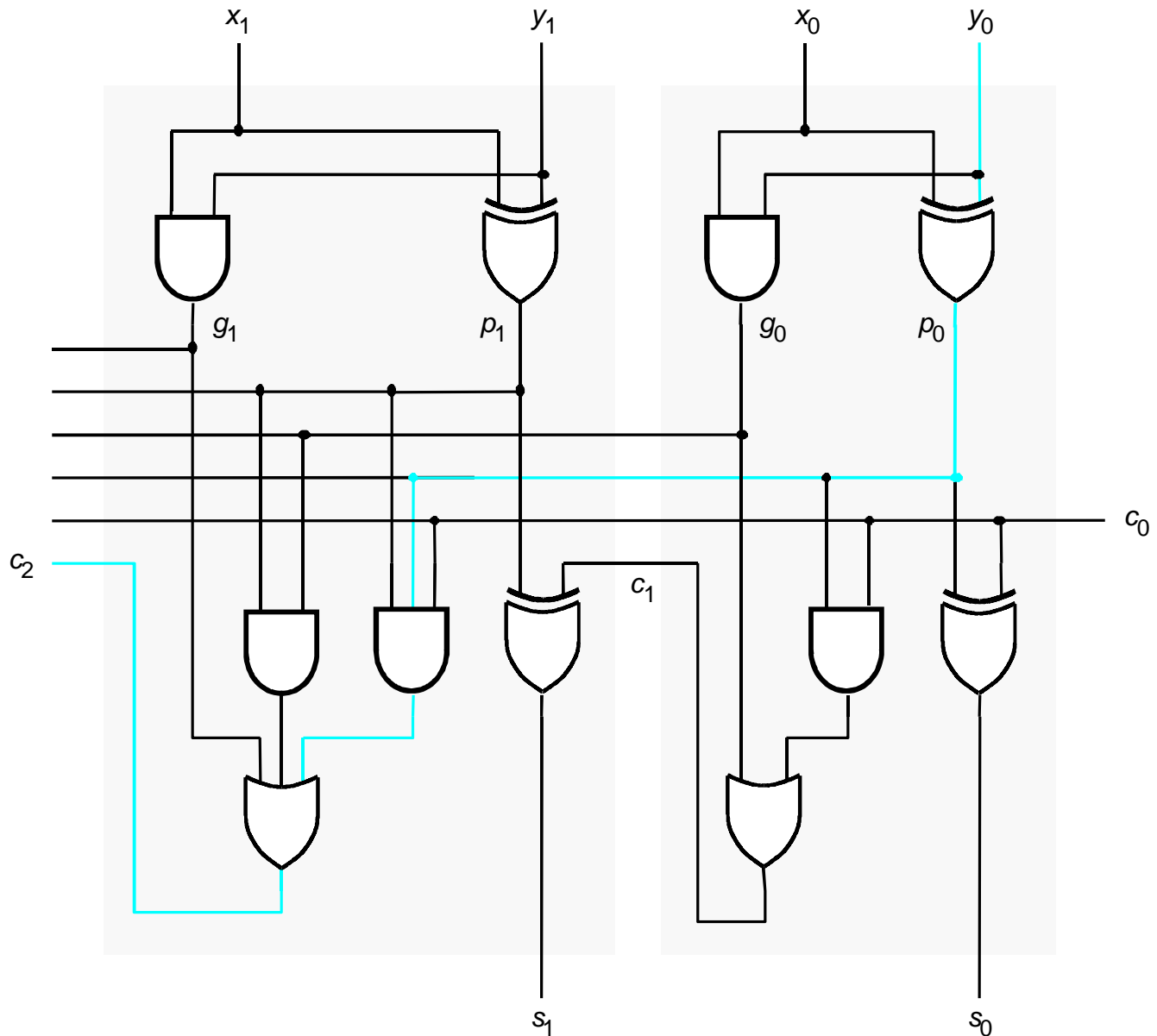


Figure 5.19. An alternative design for a carry-lookahead adder.

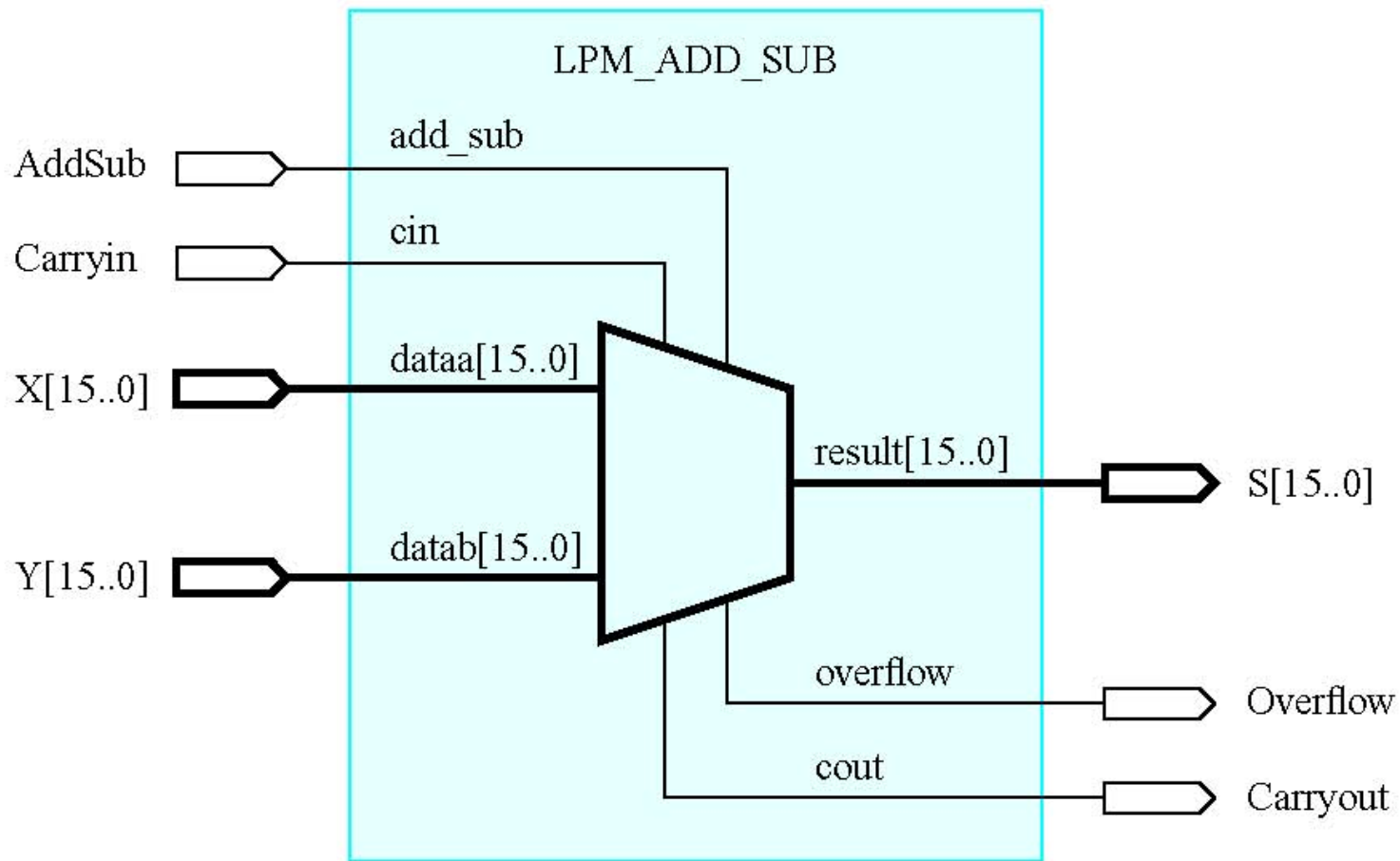


Figure 5.20. Schematic using an LPM adder/subtractor module.

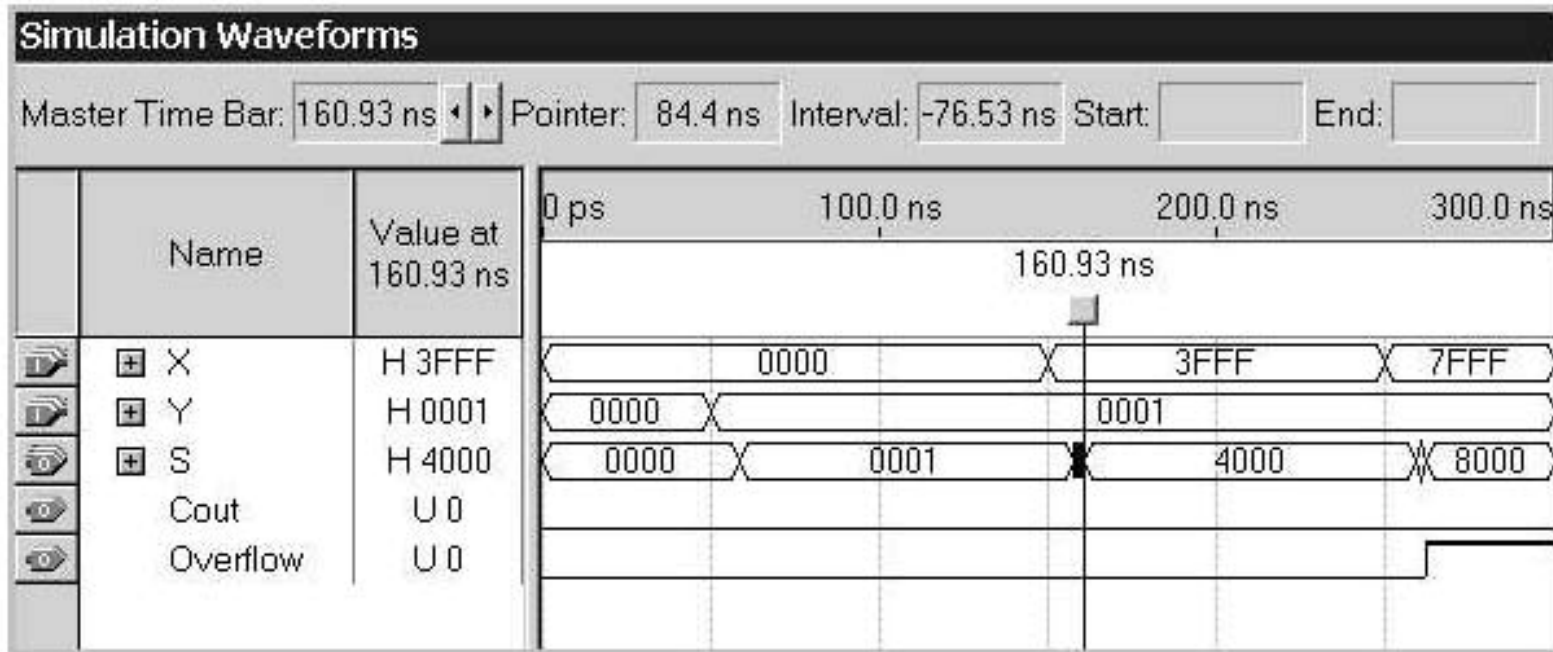


Figure 5.21. Simulation results for the LPM adder.