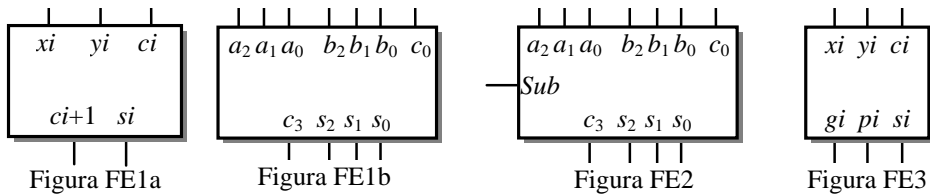


- E1. A figura FE1a mostra um somador completo de um bit. Pede-se::
 a) Construa somador completo de três bits mostrado na figura FE1b usando somadores de um bit em configuração *ripple-carry*.
 b) Usando o somador de três bits da figura FE1b e portas lógicas adicionais, construa um subtrator que efetua em **Complemento de 2** a operação $a[2:0] - b[2:0]$
 c) Converta agora o somador da figura FE1b num subtrator que efetua também em Complemento de 2 a operação $a[2:0] - b[2:0] - c_0$.

E2. Usando o somador de três bits da figura FE1b e portas lógicas adicionais, construa o somador/subtrator de três bits mostrado na FE2 em que a entrada *Sub* seleciona qual operação aritmética o circuito realiza. Com *Sub*=0, o circuito funciona como o somador original; com *Sub*=1, efetua em Complemento de 2 a subtração $a[2:0] - b[2:0] - c_0$.

- E3. A figura FE3 mostra um somador de um bit com saídas *gi* e *pi* para geração de vai-uns por *carry look-ahead* (CLA). Pede-se:
 a) Desenhe os diagramas lógicos dos circuitos de *gi*, *pi* e *si* usando portas lógicas genéricas
 b) Usando esse somador e portas lógicas adicionais, construa um somador rápido de três bits (ou seja, sem usar *ripple-carry*) como o da figura FE1b

(Fórmulas: $c_{k+1} = G_0^k + P_0^k c_0$, com $P_j^k \triangleq \prod_{i=j}^k p_i$ e $G_j^k \triangleq g_k + \sum_{i=j}^{k-1} P_{i+1}^k \cdot g_i$)

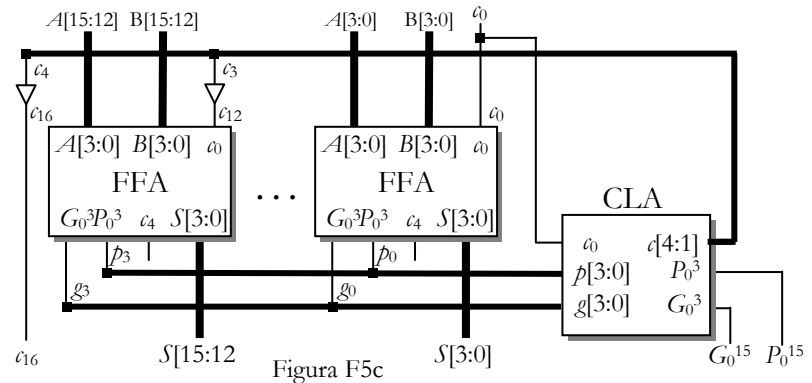
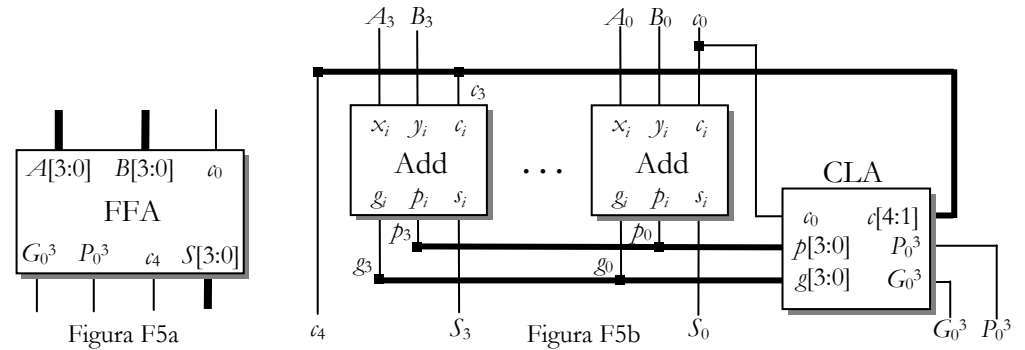


- E4. (Wakerly, 6.101) Estimate the number of product terms in a minimal sum-of-products expression for the c_{32} output of a 32-bit binary adder (nota: usando preditor de vai-um).
 E5. A figura FE5a mostra um somador rápido (FFA, *Fast Full Adder*) de quatro bits, implementado com um preditor de vai-um (CLA, *Carry Look Ahead*) e somadores de um bit (Add), como mostra a figura FE5b. Já a figura FE5c mostra como construir um somador rápido de 16 bits usando quatro blocos FFA e um bloco CLA adicional. Nota: os triângulos nas linhas c_4 e c_3 que saem do barramento de $c[4:1]$ são *seguidores* (*drivers ou buffers*) e servem para dar outros nomes a elas (c_{16} e c_{12} , neste caso); é um truque usado em CADs para melhor documentar os diagramas.
 a) Determine a expressão das saídas G_0^3 e P_0^3 do bloco CLA na forma de **soma de produtos**.
 b) Complete o diagrama da figura FE5b, desenhando os dois blocos Add omitidos.
 c) Complete o diagrama da figura FE5c, desenhando os dois blocos FFA omitidos.

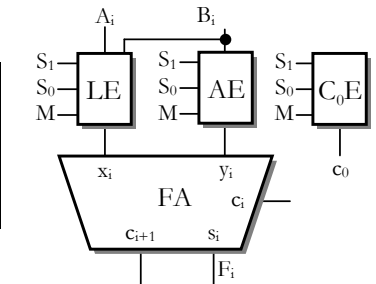
E6. Explique por que é possível usar o mesmo CLA da figura FE5b (feito para gerar vai-uns da soma bit a bit) para gerar também os vai-uns de somadores de 4 em 4 bits (figura FE5c).

E7. A figura FE7 mostra a célula básica de uma Unidade Lógica Aritmética usando somador completo (FA, *Full Adder*) e o bloco gerador do *carry-in* menos significativo (c_0).

- a) Projete os blocos AE, LE e C_0E (*Arithmetic, Logic e Carry-In Expanders*) para que a ULA efetue as operações descritas na tabela (somadas e subtrações em complemento de 2)
 b) Mostre como construir uma ULA de três bits (palavras de entrada: A[2:0] e B[2:0] usando os blocos da figura FE7.



M = 0 (modo lógico)			M = 1 (modo aritmético)		
S _i S ₀	Descrição	F _i	S _i S ₀	Descrição	F
0 0	NAND	(A _i and B _i)'	0 0	Soma	A+B
0 1	NOR	(A _i or B _i)'	0 1	Subtração	(A-B) _{c2}
1 0	se A então B	A _i → B _i	1 0	Decremento	A-1
1 1	XNOR	(A _i xor B _i)'	1 1	Subtração com decremento	(A-B-1) _{c2}



Algumas respostas

E1b) $x_i \leftarrow a_i; y_i \leftarrow b_i'; (c_i \text{ menos signif.}) \leftarrow 1; c) x_i \leftarrow a_i; y_i \leftarrow b_i'; (c_i \text{ menos signif.}) \leftarrow c_0'$

E2) $x_i \leftarrow a_i; y_i \leftarrow b_i \oplus \text{Sub}; (c_i \text{ menos signif.}) \leftarrow c_0 \oplus \text{Sub}$

E3a) $s_i = x_i \oplus y_i \oplus c_i$; $g_i = x_i y_i$; $p_i = x_i \oplus y_i$ (ou $p_i = x_i + y_i$); b) $c_1 = g_0 + p_0 c_0$; $c_2 = g_1 + p_1 g_0 + p_1 p_0 c_0$; $c_3 = g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 c_0$

E7) $x_i = M \cdot a_i + M' \cdot (a_i' \cdot b_i' + a_i' \cdot S_0) + a_i \cdot b_i \cdot S_1 + a_i \cdot b_i' \cdot S_1' \cdot S_0'$; $y_i = M \cdot (b_i \cdot S_0' + b_i' \cdot S_0 + S_1 \cdot S_0')$; $c_0 = M \cdot S_1' \cdot S_0$. Dicas: $(A_i \rightarrow B_i) = A_i' + B_i$; subtração com decremento: $y_i = B_i'$ e $c_0 = 0$.