Capítulo 2-Projeto Lógico Combinacional E

Profa. Eliete Caldeira

Quine McCluskey

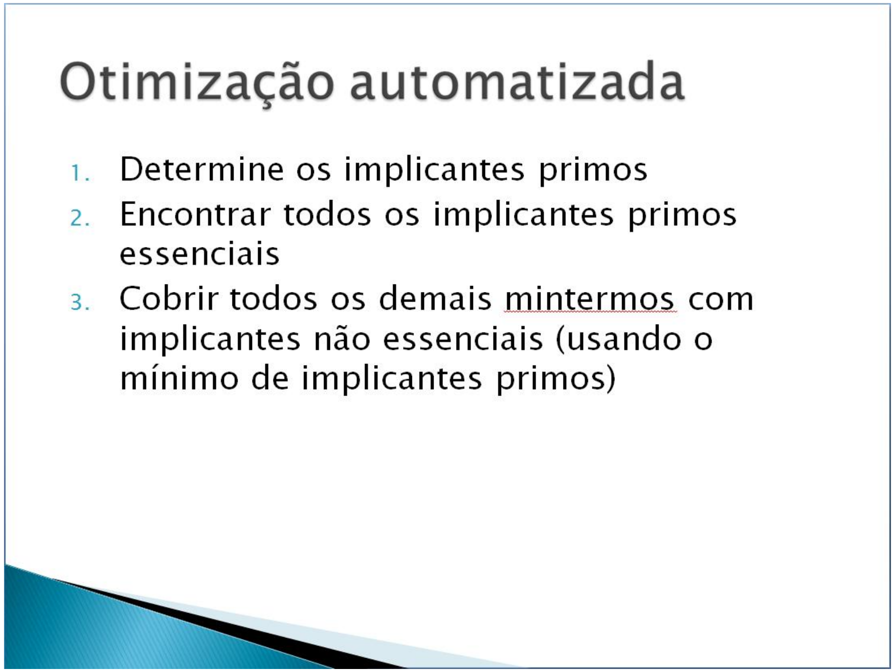
- Exercício: Para a função $F(A,B,C,D) = \Sigma m(0,1,3,7,8,12) + dc(5,10,11,13,14)$
- Minimize a função usando o método tabular de Quine McCluskey.
- Refaça usando mapas de Karnaugh para verificar a resposta obtida.

Quine McCluskey

- Exercício: Para a função $F(W,X,Y,Z) = \Sigma m(2,3,6,9) + dc(10,11,12,13,14,15)$
- Minimize a função usando o método tabular de Quine McCluskey.
- Refaça usando mapas de Karnaugh para verificar a resposta obtida.

Otimização ou melhoramento

- Primeiro e segundo passos são diretos
- Terceiro passo pode levar a muitas combinações
- Otimização: Testar todas e escolher a melhor
- Melhoramento: usar heurística para testar apenas algumas combinações iterativamente



Melhoramento iterativo

- Fazer pequenas alterações iterativamente em uma solução conhecida
- Parar quando não é possível alterar mais ou quando um determinado tempo expirou
- Exemplo:
  - Para $F = abcdefgh + abcdefgh' + jklmnop$
  - F pode ser melhorada combinando os dois primeiros termos
  - $F = abcdefg(h+h') + jklmnop$
  - $F = abcdefg + jklmnop$

Melhoramento iterativo

- Operação de expansão: remover uma literal de um termo e verificar se o novo termo é válido
- Exemplo: F=x’z+xy’z+xyz
  - Usando expansão no 1º termo
  - A) eliminando x’ – possível
  - B) eliminando z – impossível
- Para completar o passo, deve-se remover todos os termos cobertos pelo termo expandido (z cobre xyz, xy’z, x’yz, x’y’z, x’z, xz, y’z, yz, z)

  Assim F= z

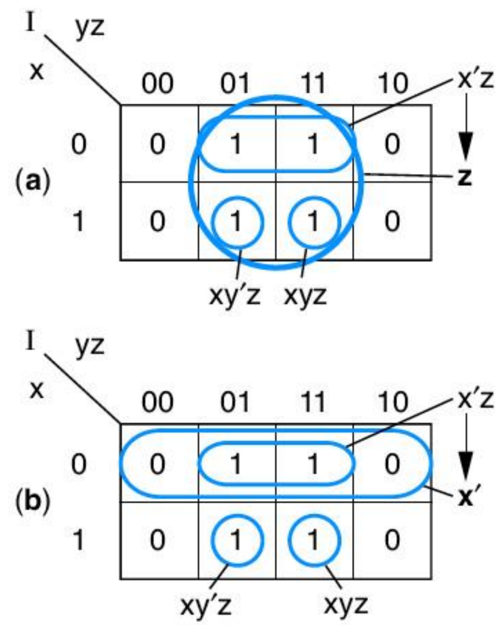


Figure 6.31 Expansions of term x'z in the function F = x'z + xy'z + xyz: (a) legal, (b) not legal (because the expanded term covers 0s).

Melhoramento iterativo

- Exercício: Use expansão para minimizar a função F=xyz+xyz’+x’y’z’+x’y’z
  - Tentando expandir o primeiro termo eliminando z
    - O termo xy cobre xyz’ e xyz que são parte da função, assim F= xy~~z+xyz’~~+x’y’z’+x’y’z e F=xy+x’y’z’+x’y’z
  - Expandindo o primeiro termo eliminando y
    - O termo x cobre xy, xy’, xz, xz’, xyz, xyz’, xy’z e xy’z’ e alguns destes termos (em vermelho) não são parte da função e F=xy+x’y’z’+x’y’z
  - Expandindo o primeiro termo eliminando x
    - O termo y cobre xy, x’y, yz, yz’, xyz, xyz’, x’yz e x’yz’ e alguns destes termos (em vermelho) não são parte da função e F=xy+x’y’z’+x’y’z
  - Expandindo o segundo termo eliminando z’
    - O termo x’y’ cobre x’y’z e x’y’z’ que são parte da função, assim F=xy+x’y’~~z’+x’y’z~~ e F=xy+x’y’
  - Expandindo o segundo termo eliminando y’
    - O termo x’ cobre x’y, x’y’, x’z, x’z’, x’yz, x’yz’, x’y’z e x’y’z’ e alguns destes termos (em vermelho) não são parte da função e F=xy+x’y’
  - Expandindo o segundo termo eliminando x’
    - O termo y’ cobre xy’, x’y’, y’z, y’z’, x’y’z’, x’y’z, xy’z’, xy’z e alguns destes termos (em vermelho) não são parte da função e F=xy+x’y’
  - Assim F=xy+x’y’ é mínima

Melhoramento iterativo

- Operações:
  - Expansão: retirar uma literal e ver se o termo ainda faz parte da função
  - Redução: aumentar uma literal e ver se o termo resultante ainda faz parte da função. É o contrário da expansão
  - Irredundância: consiste em remover completamente um termo e ver se a função continua a mesma. Se for verdade, o termo era redundante.
- Heurísticas:
  - Usar expansão, redução e irredundância em sequência
  - Usar 10 operações aleatórias de expansão, depois cinco operações aleatórias de redução e então duas de irredundância. Repita a sequencia completa até que não ocorram mais melhoramentos

Redes Multinível

- Se a velocidade for menos importante que a economia de espaço, pode-se utilizar mais níveis de lógica
- *Tradeoff*: reduzir o número de transistores em troca de aumentar o número de níveis até a saída
- Em geral, a otimização de lógicas múltiplos níveis usa a fatoração, colocando algum termo em evidência para reduzir o número de entradas de portas
- Não há forma padrão para redes multinível
- Pode-se usar lógica multinível para satisfazer critérios como o número máximo de entradas de portas e o uso de um conjunto de portas específicas disponível

Redes Multinível

- **Procedimento de otimização:**
  - Obter as somas de produtos (ou produtos de somas)
  - Transformar as expressões de forma em que os critérios sejam atendidos
  - Transformar a rede resultante em uma equivalente usando o conjunto de portas disponível

Redes Multinível

- Por exemplo, a função

  F=ab+acd+ace

  não pode ser minimizada, mas colocando termos em evidência podemos implementar usando portas com menos entradas (e menos transistores)

  F = ab+ac(d+e) = a(b+c(d+e))

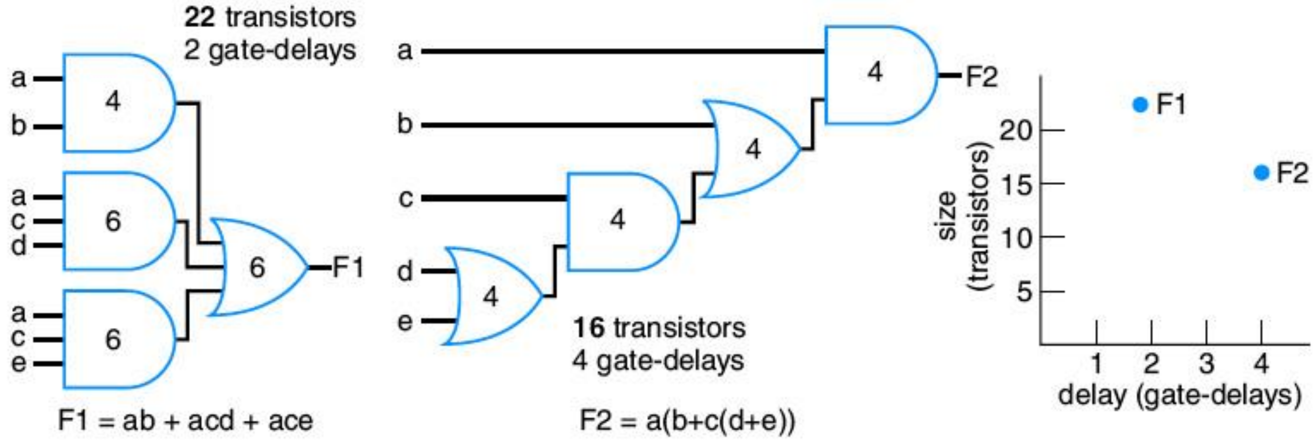


Figure 6.32 Using multilevel logic to tradeoff performance and size: (a) two-level circuit, (b) multilevel circuit with fewer transistors, (c) illustration of the size versus delay tradeoff. Numbers inside gates represent transistor counts.

Redes Multinível

- Exercício: Usando manipulação algébrica, minimize o tamanho do circuito da função

$$F = abcd+abcef$$

possivelmente às custas de menor desempenho. Faça o gráfico do *tradeoff* do circuito inicial e final em relação ao tamanho e ao atraso.

Redes Multinível

- Exercício: Usando manipulação algébrica, minimize o tamanho do circuito da função

$$F = abcd+abcef$$

possivelmente às custas de menor desempenho. Faça o gráfico do *tradeoff* do circuito inicial e final em relação ao tamanho e ao atraso.

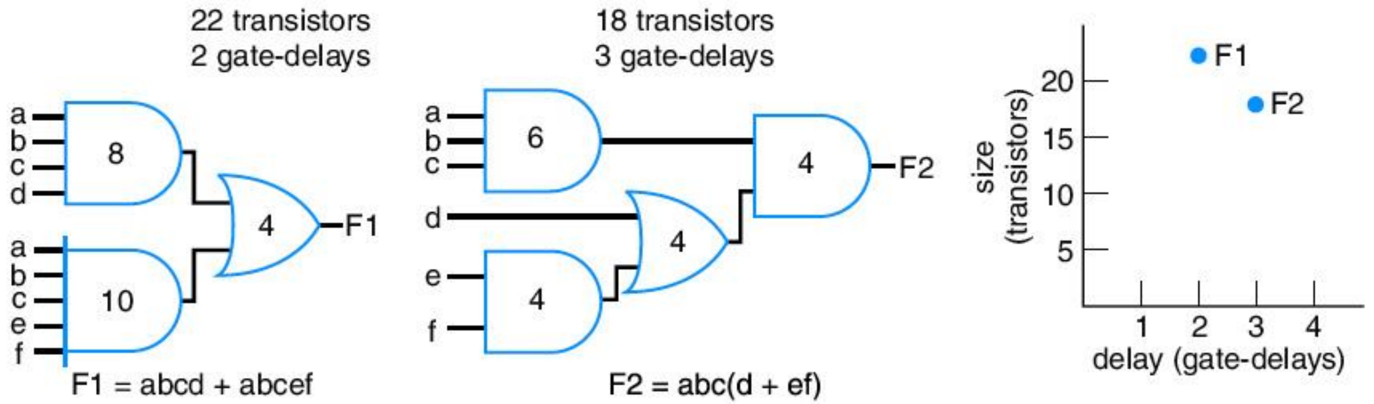
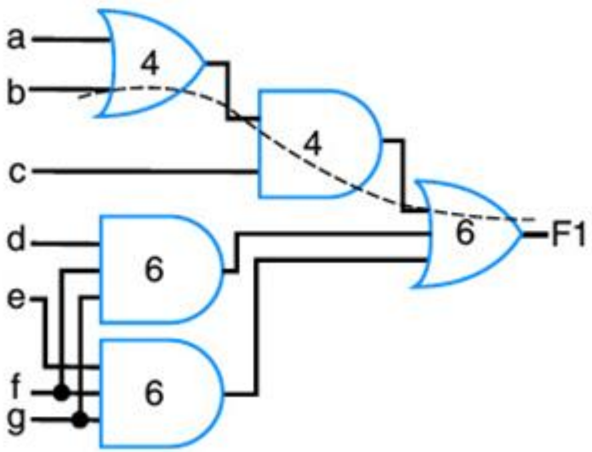


Figure 6.33 Multilevel logic to tradeoff performance and size: (a) two-level circuit, (b) multilevel circuit with fewer transistors, (c) tradeoff of size versus delay. Numbers inside gates represent transistor counts.

Redes Multinível

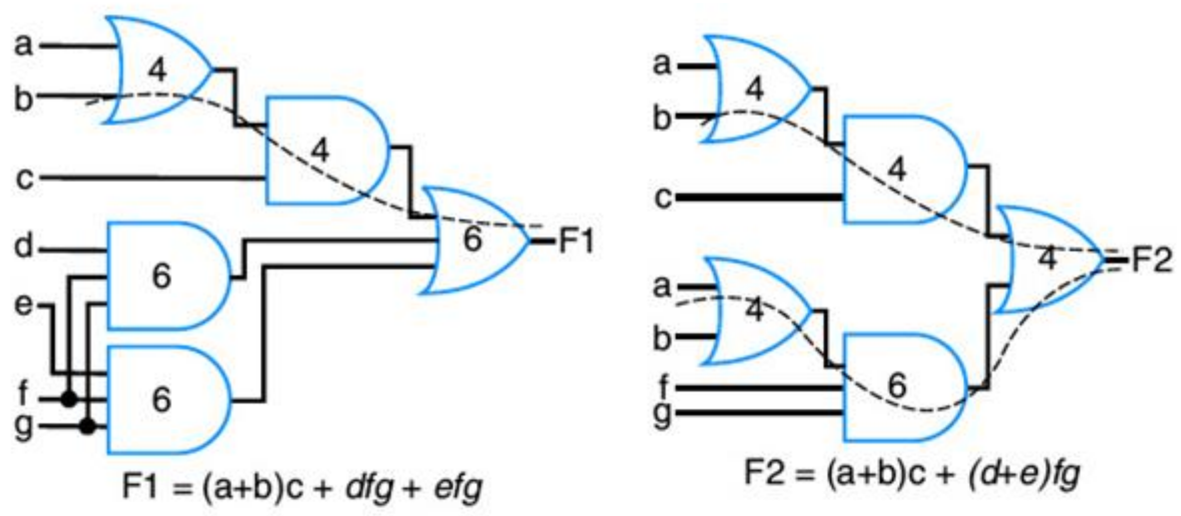
- Exercício: Use lógica de múltiplos níveis para reduzir o tamanho do circuito da figura sem aumentar o atraso.



F1 = (a+b)c + *dfg* + *efg*

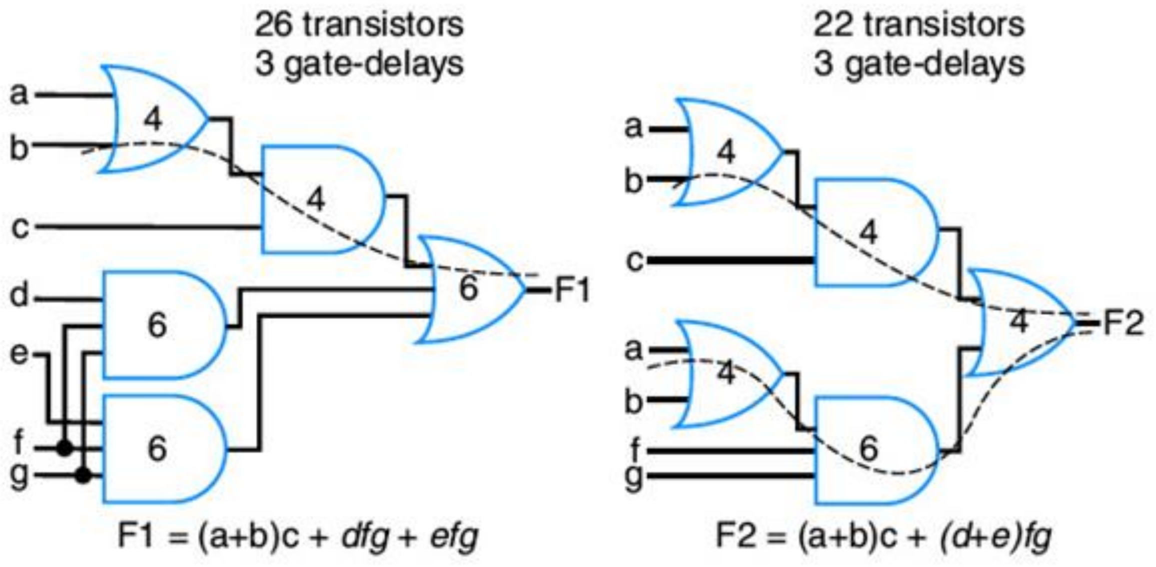
Redes Multinível

- Exercício: Use lógica de múltiplos níveis para reduzir o tamanho do circuito da figura sem aumentar o atraso.



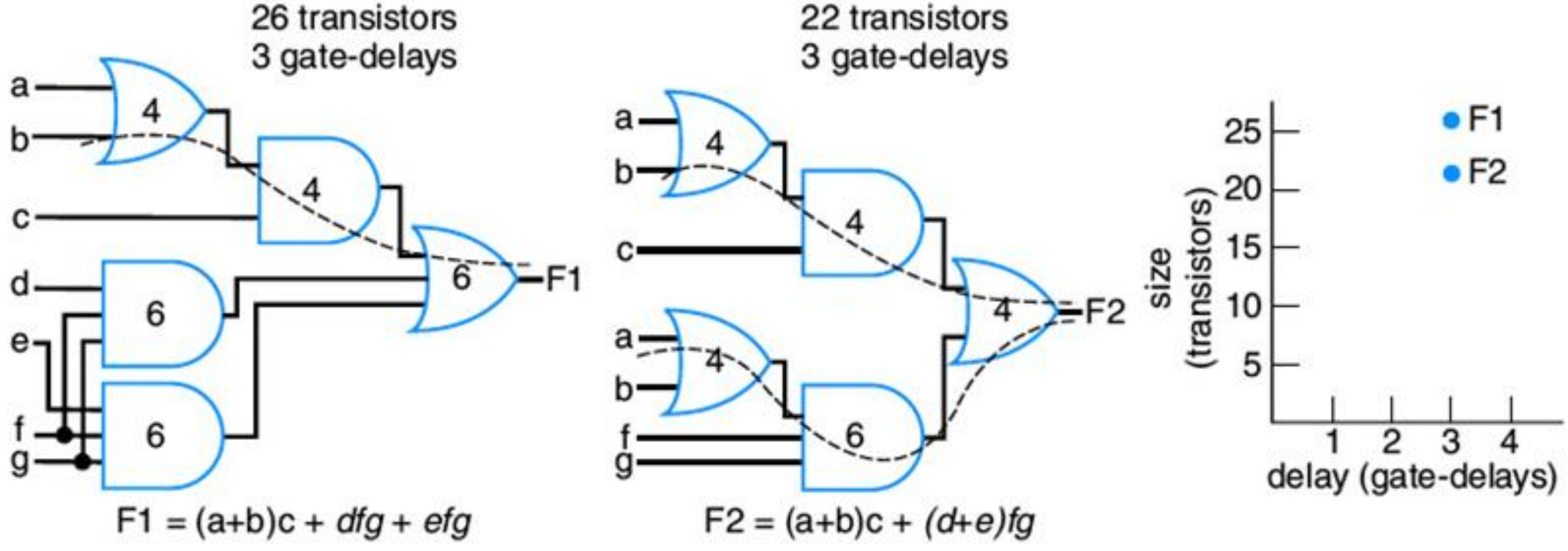
Redes Multinível

- Exercício: Use lógica de múltiplos níveis para reduzir o tamanho do circuito da figura sem aumentar o atraso.



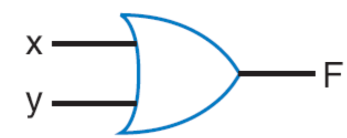
Redes Multinível

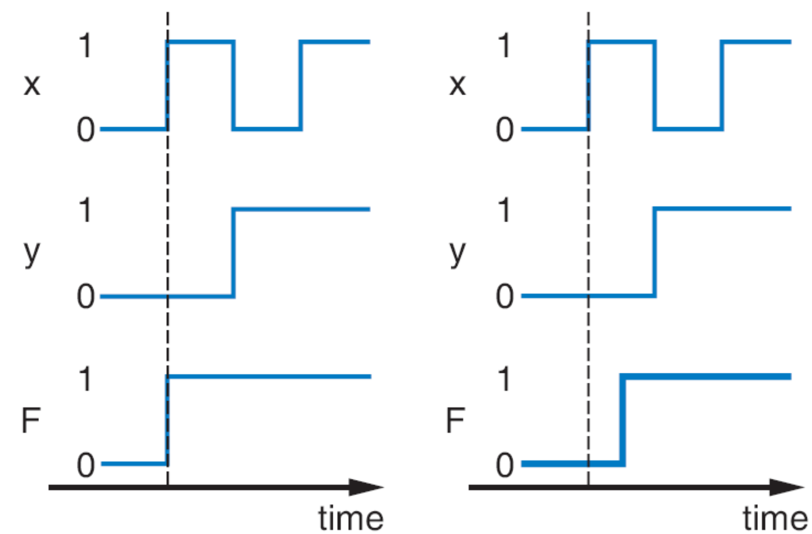
- **Exercício: Use lógica de múltiplos níveis para reduzir o tamanho do circuito da figura sem aumentar o atraso.**



Outras considerações

- Comportamento não ideal das portas → atraso no tempo
- Portas reais apresentam atrasos
- Saídas não mudam imediatamente depois da mudança das entradas





Para ser continuado...