

Sumário:

1. Apresentação do algoritmo de Euler.
2. Exemplo de aplicação do algoritmo de Euler a uma rede série-paralelo
3. Aplicação do algoritmo de Euler na concepção do *layout* de um circuito CMOS baseado em portas lógicas complexas (ADD 4bits, começando pelo ADD 1bit) à custa de *standard cells* optimizadas.

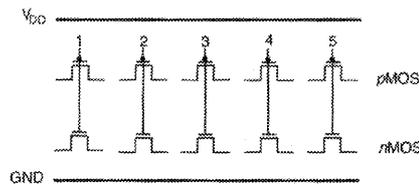
Memorando:

Parte 1:

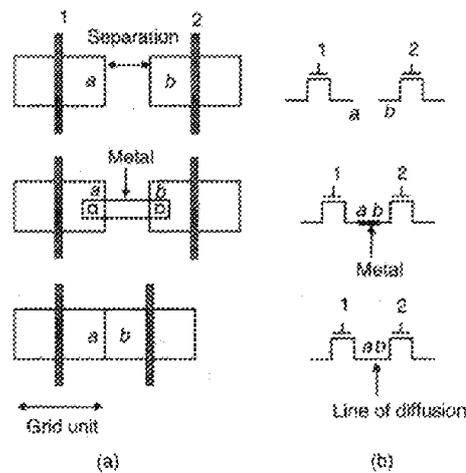
Apresentação do algoritmo de Euler

Os programas de CAD de geração automática de *layouts* de circuitos digitais produzem *layouts* VLSI de grande qualidade, a partir de descrições do tipo *netlist*, ou a partir de expressões booleanas. Os *layouts* gerados podem ser baseados em vários tipos de estruturas, tais como ROMs/RAMs, PLAs, *standard-cells*, etc. Esses programas usam métodos sistemáticos para mapear *netlists* ao nível do transístor, em *layouts* ao nível físico, com a vantagem de serem isentos de erros e de se diminuir o seu tempo de concepção.

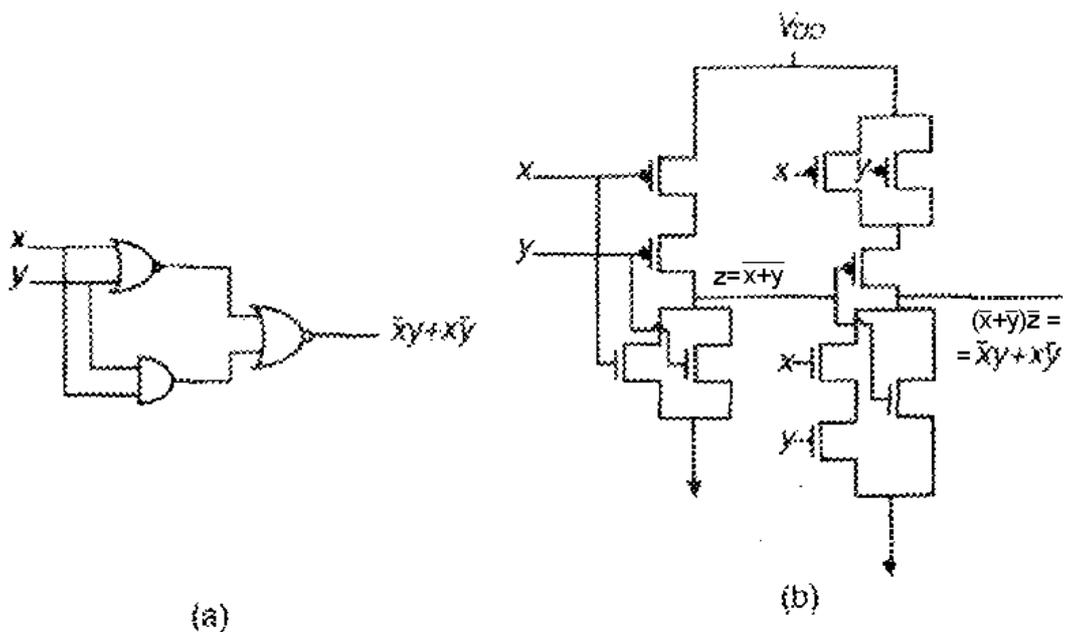
Uma estrutura baseada em *standard-cells* sintetiza uma função booleana numa estrutura regular, constituída por uma linha de transístores pMOS e uma linha de transístores nMOS. Os transístores são verticalmente alinhados, ou seja, um transístor nMOS e um transístor pMOS, que recebam o mesmo sinal de *gate*, são colocados na mesma coluna. A altura da célula é fixa. As fitas de polissilício são verticais (variáveis de entrada), e os sinais V_{DD} e V_{SS} existem na horizontal, nas extremidades superior e inferior, respectivamente.



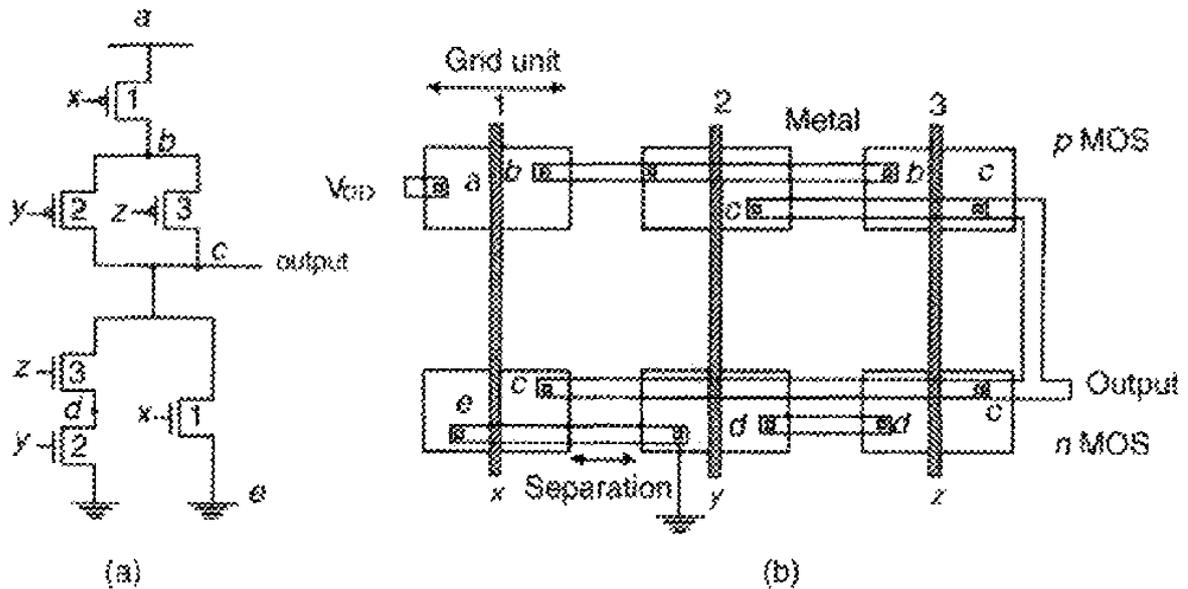
No estabelecimento de uma ligação entre dois transístores adjacentes (em série ou em paralelo) é possível introduzir a simplificação sugerida pela figura seguinte, ou seja, realizar a ligação por simples *abutment* (encosto), sem recorrer a tiras de metal.



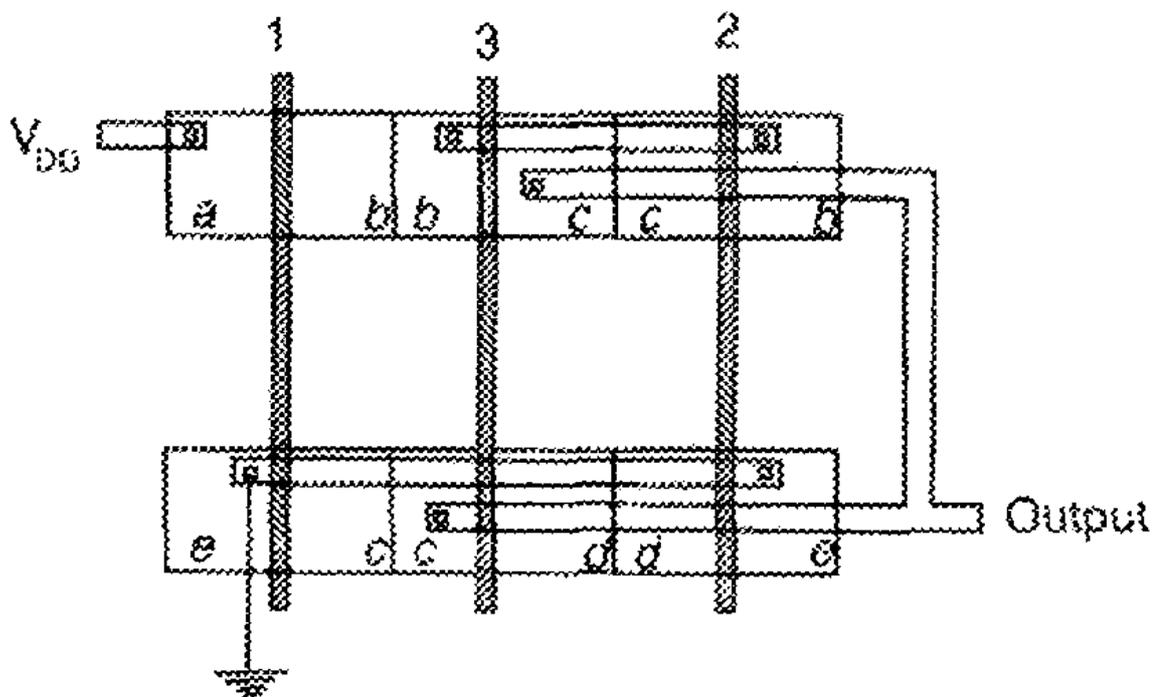
Considere-se o circuito baseado em portas lógicas complexas, representado na figura seguinte, que realiza a função XOR.



A figura seguinte é uma possível implementação do *layout* do último andar desse circuito, seguindo a abordagem *standard-cells*.



Introduzindo a otimização referida anteriormente, é possível obter o seguinte *layout*, significativamente mais compacto.



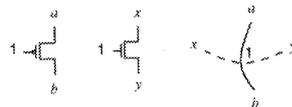
O exemplo anterior sugere que a otimização de um *layout* baseado em *standard-cells* passa por maximizar o número de vezes em que é possível realizar a ligação entre dois transístores por simples *abutment*. Para que isso

seja possível, transístores adjacentes (em série ou em paralelo) devem ser colocados em colunas adjacentes do *layout*. Seguindo este raciocínio, conclui-se que a otimização passa pela escolha de uma ordem para as tiras de polissilício, que permita interromper o menor número de vezes possível as tiras de difusão n e p.

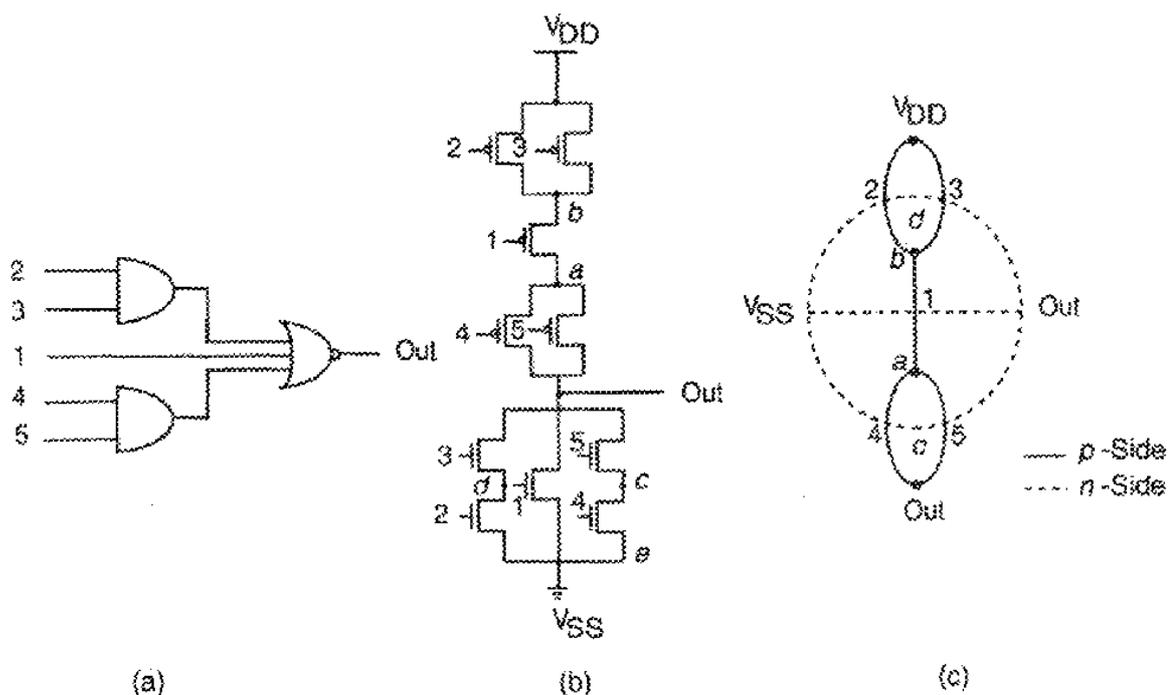
Um algoritmo que permite otimizar este aspecto é o algoritmo de Euler. A sua aplicação na otimização de *layouts* exige o mapeamento da rede CMOS num grafo para a rede p e num grafo para a rede n.

A construção de cada grafo consiste nas seguintes regras:

- todos os potenciais (ou nós) da rede de transístores são vértices do grafo;
- cada transístor é representado por um arco entre dois vértices, ligando os potenciais do *drain* e da *source*; o arco tem uma *label* correspondente ao sinal existente na *gate* do transístor;
- os grafos da rede n e da rede p são duais.



A figura seguinte mostra uma rede CMOS que implementa a função lógica indicada em a) e o respectivo grafo.

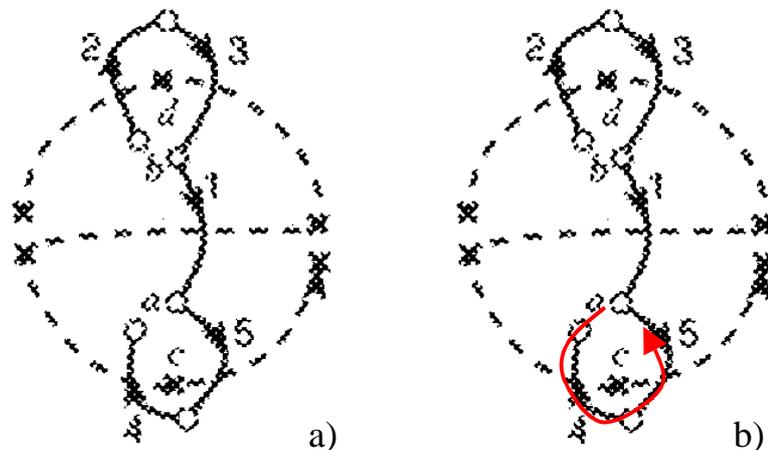


Se num grafo, os arcos 1 e 2 são adjacentes, então é possível colocar os transistores correspondentes em posições adjacentes no layout e consequentemente ligá-los através da difusão. Para minimizar o número de áreas de separação entre as zonas de difusão é necessário encontrar um caminho que passe por todos os transistores do grafo.

Ao caminho que permite percorrer todos os arcos de um grafo, passando por cada arco uma única vez, chama-se **caminho de Euler**. Se existe um caminho de Euler num grafo, então é possível ligar todos os transistores através das zonas de difusão. Dado que numa standard cell os transistores tipo-p e tipo-n, com o mesmo sinal, se encontram alinhados, é necessário que os caminhos de Euler em ambos os grafos (p e n) tenham a mesma sequência. A ordem das tiras de polissilício no *layout* é dada pela ordem das *labels* dos arcos no caminho de Euler.

A heurística usada em baixo para encontrar caminhos de Euler num grafo, baseia-se num teorema que diz que: se num grafo todas as séries e todos os paralelos de arcos tiverem um número ímpar de entidades, então existe um caminho de Euler.

A figura seguinte mostra um caminho de Euler, para o caso do circuito referido anteriormente.



Nota: A linha sobreposta na figura b) corresponde à ordenação correcta dos transistores 4 e 5 da rede p, uma vez que a sequência tem de ter a mesma ordem na rede n e na rede p. Isso não acontece na figura a).

O algoritmo de Euler consiste em:

1. construir os grafos da rede n e da rede p
2. converter o grafo de forma a assegurar que existe sempre um caminho de Euler; para isso, é necessário adicionar arcos fictícios*, de forma que todas as séries e paralelos tenham um número ímpar de entidades; ao adicionarem-se arcos fictícios, procurar intercalá-los o menos possível;
3. encontrar um caminho de Euler, procurando arranjar uma solução que permita arrumar para o início e para o fim do caminho os arcos fictícios; se isto for possível, as fitas de difusão n e p vão ser contíguas, embora o teorema de Euler não o garanta;
4. extrair a ordem das tiras de polissilício do caminho de Euler encontrado, após remoção dos arcos fictícios; as fitas de difusão terão de ser interrompidas se houver arcos fictícios fora das extremidades do caminho.

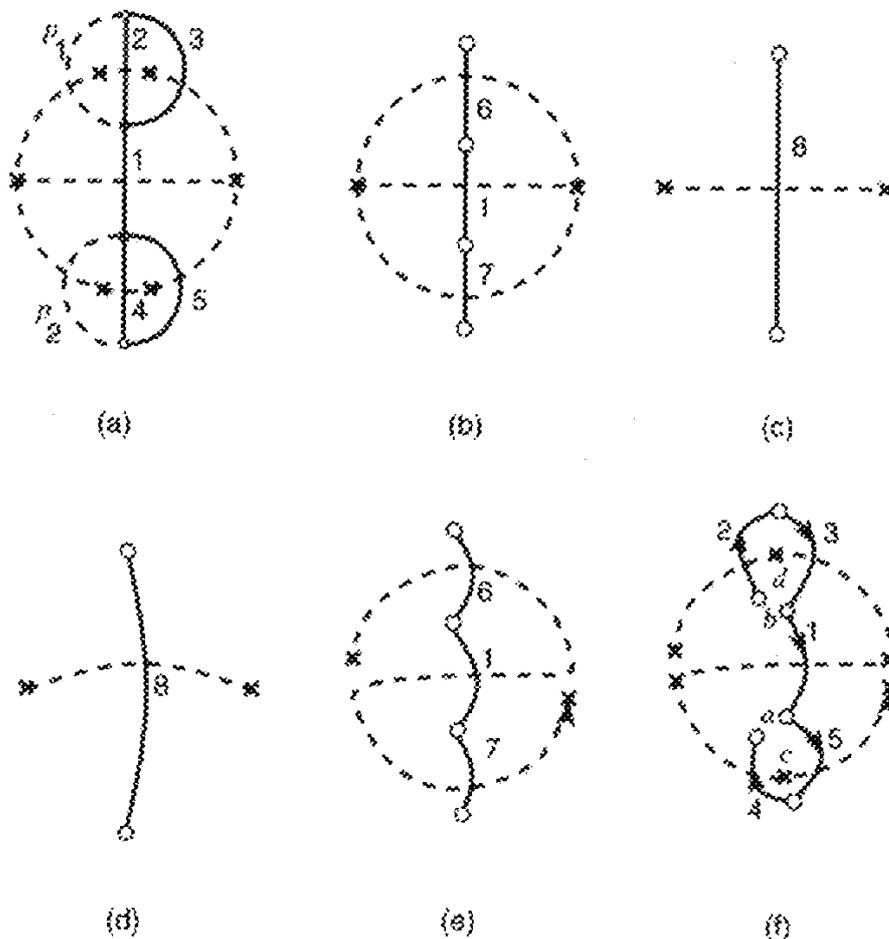
* os arcos fictícios correspondem a interrupções na fita de difusão.

A figura seguinte, resume a aplicação deste algoritmo ao circuito apresentado anteriormente, explicando como se obtém a solução referida atrás.

Na figura a) começou-se por adicionar os arcos fictícios p_1 e p_2 de modo a permitir a conversão do grafo. Adicionando o arco p_1 , obtêm-se em paralelo na rede p e em série na rede n: os arcos p_1 , 2 e 3, que podem então ser convertidos no arco 6 da figura b), uma vez que são em número ímpar. O mesmo procedimento é usado para obter o arco 7, a partir dos arcos p_2 , 4 e 5

O grafo obtido depois de efectuadas as conversões apresenta em série na rede p e em paralelo na rede n: os arcos 6, 1 e 7. Como estão em número ímpar, a conversão é imediata, obtendo-se assim o grafo da figura c).

Uma vez obtido um grafo com um só arco, inicia-se a procura do caminho de



Euler, definindo uma direcção para o caminho. Determinou-se neste caso a direcção de cima para baixo na rede p e da esquerda para a direita na rede n.

Mantendo a mesma direcção no grafo da figura e), o grafo deverá ser descrito começando no transistor 6 e acabando no 7.

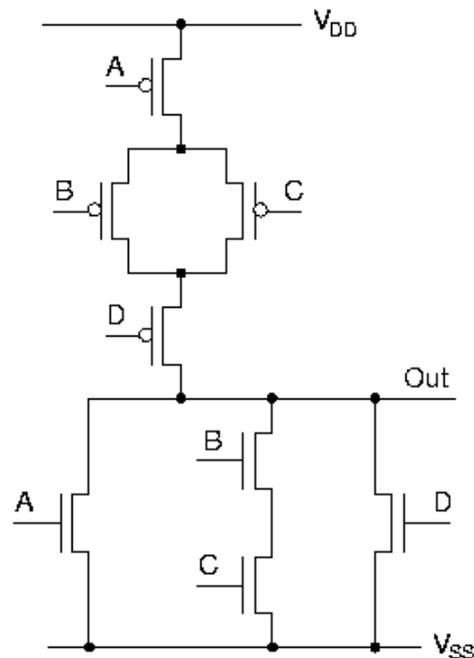
Por último, seguindo a ordem definida no grafo da figura e) e colocando os arcos fictícios p_1 e p_2 nos extremos da sequência, obtém-se a seguinte ordenação: p_1 , 2, 3, 1, 4, 5 e p_2 . Como os arcos p_1 e p_2 conseguiram ser

colocados nos extremos da sequência podem ser retirados do grafo, obtendo-se a solução apresentada na figura f) (veja-se a correcção apresentada antes).

Parte2:

Exemplo de aplicação do algoritmo de Euler a uma rede série-paralelo

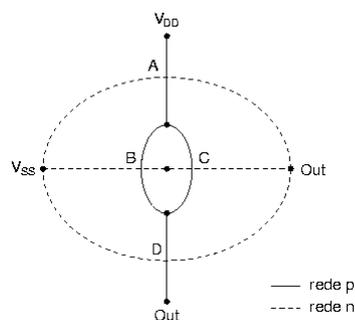
Dado o circuito da figura seguinte

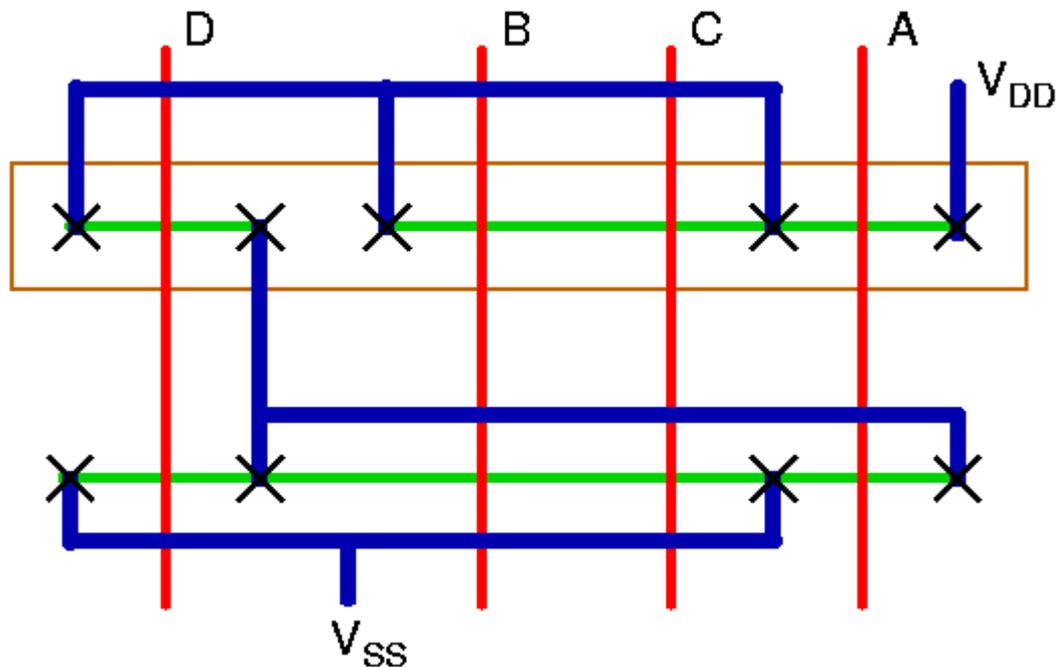


pretende-se otimizar a ordem das tiras de polissilício, usando uma implementação baseada em *standard-cells* na construção do *layout*.

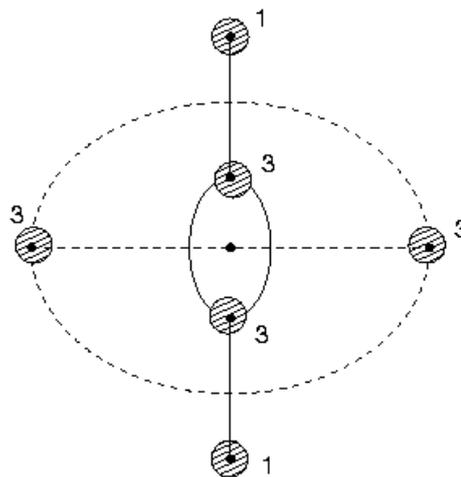
Resolução do exemplo:

Uma das heurísticas que podem ser usadas para otimizar a rede é a que resulta da aplicação do algoritmo de Euler. Assim, começa-se por construir os grafos correspondentes à rede p-MOS e à rede n-MOS do circuito apresentado.





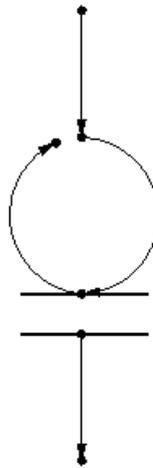
A mesma conclusão pode ser extraída da observação do número de nós ímpares que apresenta o grafo correspondente à rede p-MOS. Chamam-se nós ímpar àqueles pontos do grafo, aos quais estão ligados um número ímpar de arcos. Na figura em baixo, os nós ímpar estão indicados por um círculo, com um número adjacente que indica o número de arcos que aí estão ligados. Os restantes nós do grafo são nós par.



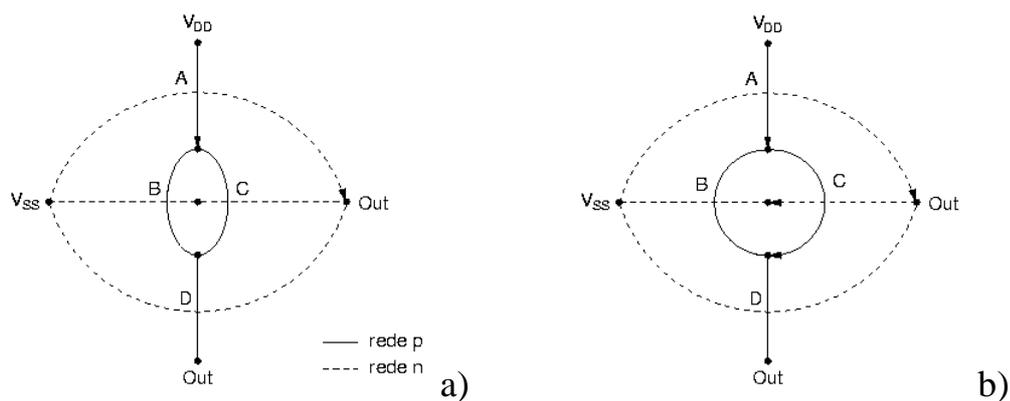
É fácil de provar que **num grafo que apresente mais de dois nós ímpar, não é possível encontrar um caminho de Euler**. Desta afirmação, resulta que no grafo vertical, correspondente à rede p-MOS, não é possível encontrar um caminho de Euler, uma vez que esse mesmo grafo tem 4 nós ímpar. Também é possível provar que **num grafo que tenha 2 nós ímpar, se existir um caminho de Euler, esse caminho tem necessariamente de começar num dos nós ímpar**

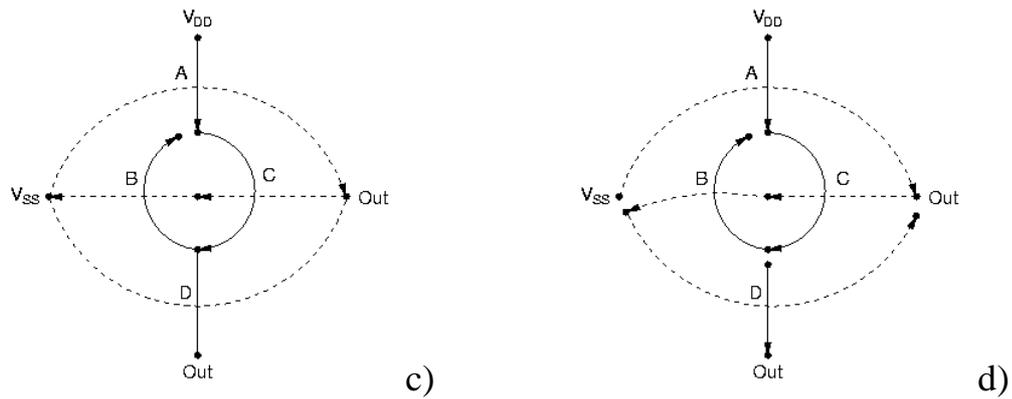
e acabar no outro. No caso do grafo apresentar apenas 1 nó ímpar, se existir um caminho de Euler, este deverá ou acabar ou começar no nó ímpar.

Dadas as afirmações anteriores, em grafos de pequenas dimensões é possível descobrir o caminho de Euler, se este existir, por simples inspeção do grafo. Assim, nos grafos correspondentes ao circuito proposto no exercício, verifica-se que é possível encontrar um caminho de Euler na grafo correspondente à rede n-MOS, embora isso não seja possível na rede p-MOS. Para otimizar o circuito, usando este método, vamos usar o caminho de Euler da rede n-MOS, procurando reduzir ao mínimo as interrupções da rede p-MOS. Inspeccionando o grafo vertical, verifica-se facilmente que o número mínimo de interrupções da rede p-MOS é 1, como mostra a figura em baixo.



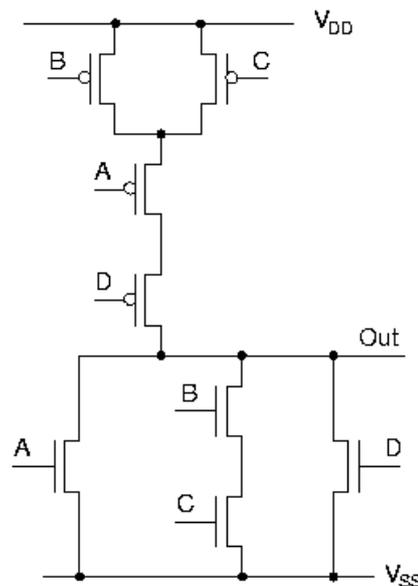
A figura seguinte mostra uma possível sequência de passos, que resulta na otimização da ordem das tiras de polissilício.





Da figura conclui-se que a ordem óptima para as tiras de polissilício deverá ser: A, C, B, D, com uma interrupção entre B e D, na rede p-MOS. É interessante notar como, apesar de terem sido definidos sentidos diferentes para o caminho de Euler, se obtém uma ordenação das tiras de polissilício idêntica (embora invertida) à que resultou do primeiro método.

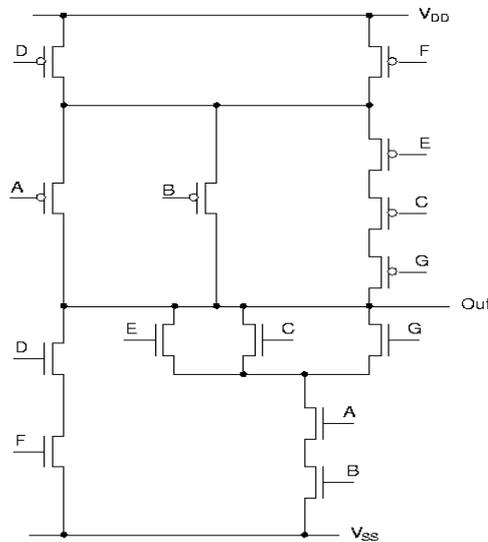
Por último, verifica-se que desenhando o circuito do exercício da maneira indicada em baixo, se conseguem encontrar caminhos de Euler nos dois grafos.



Mostre que o circuito em cima executa a mesma função que o circuito dado no exercício. Construa os respectivos grafos e descubra os caminhos de Euler que optimizam a ordem das tiras de polissilício.

Exercício proposto

Dado o circuito da figura seguinte

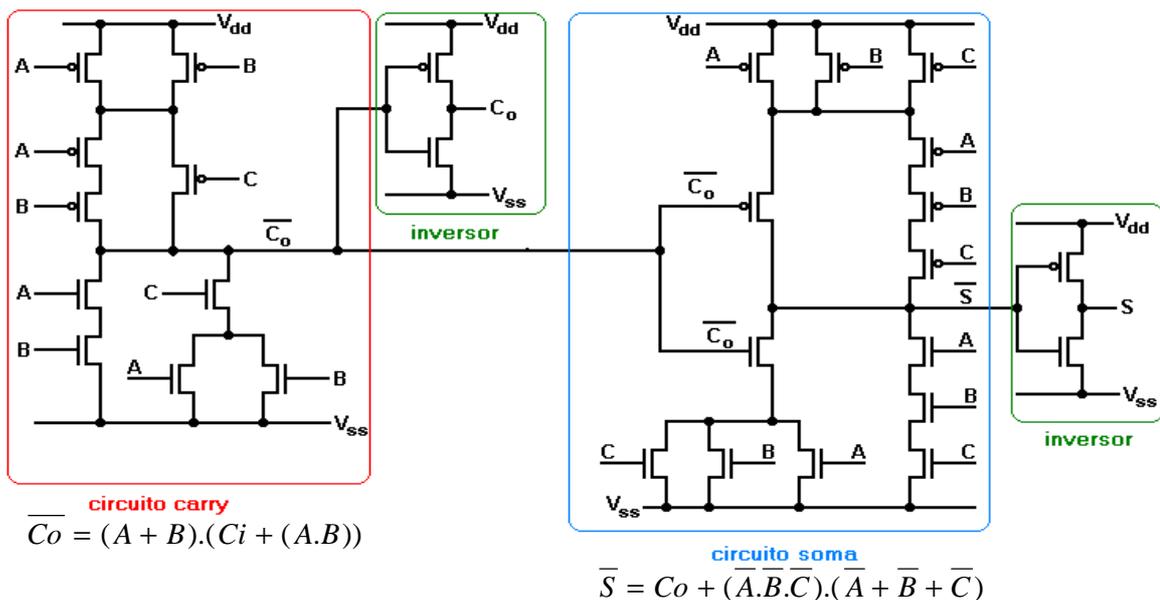


use o algoritmo de Euler para otimizar a ordem das tiras de polissilício na construção do *layout*. Compare o resultado obtido usando este método, com o que encontrou no Módulo 3 para o mesmo circuito.

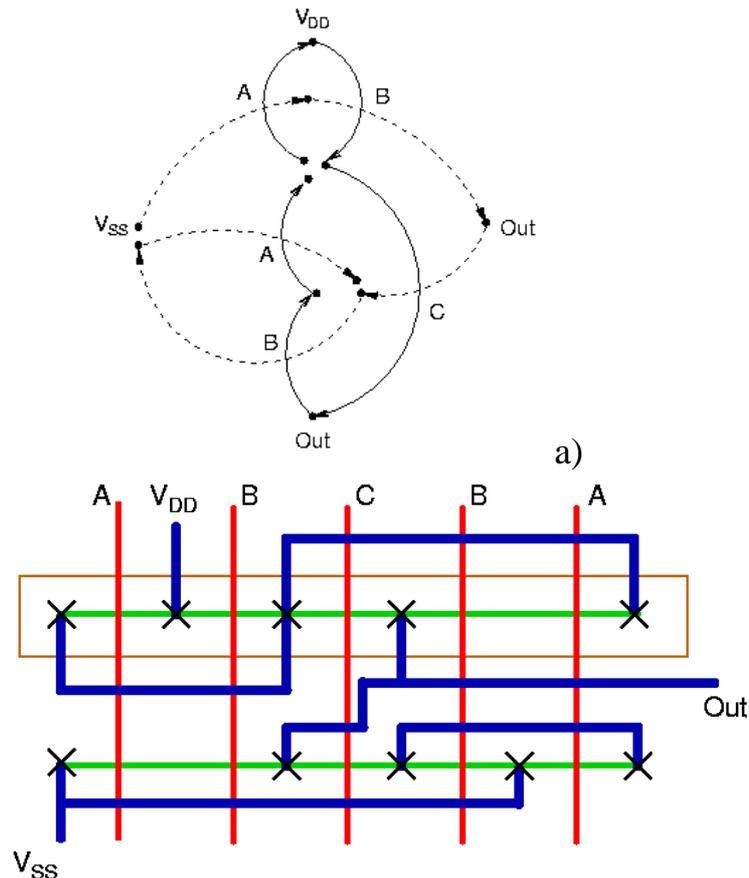
Parte 3:

Aplicação do algoritmo de Euler na concepção do *layout* de um circuito CMOS baseado em portas lógicas complexas (ADD 4bits, começando pelo ADD 1bit) à custa de *standard cells* optimizadas.

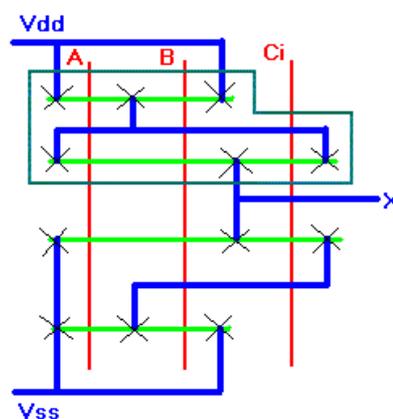
Adicionador 1bit baseado em portas complexas:



1. Voltando ao circuito de um adicionador de 1 bit, reparamos que podemos dividir em três sub-circuitos: circuito carry, soma e inversor. Portanto para efectuar o layout do Add 1bit recorre-se a três células diferentes.
2. Para cada uma delas desenhamos o layout recorrendo ao Algoritmo de Euler. Em baixo apresenta-se o grafo correspondente ao circuito carry (a) e o correspondente *stick diagram* (b).



Note-se a presença de variáveis repetidas, que permitia chegar a uma outra implementação do layout, como se indica no *stick diagram*, em baixo.



3. Usar o mesmo procedimento para desenhar o layout do circuito soma, e depois de obtidos as três diferentes células (carry, soma e inversor), passar ao desenho do layout do adicionador usando hierarquia e não esquecendo as regras DRC.
4. Depois de conseguido o layout fazer a comparação entre o layouts obtidos com, e sem a ajuda do algoritmo de Euler.

Apêndices Temáticos

- **Hr:** [Handouts Reprografia]
- **Ha:** [Handouts Aula]
- **T:** [Transparências]
- **E:** [Electrónicas]
 - ⇒ Apresentação sobre o algoritmo de Euler ('apresentacao Euler paths.ppt', 10 pág).
- **I:** [Interactivas]
- **W:** [Web]