USP – ICMC – SSC

SSC0603 – Estrutura de Dados – Trabalho 3: Simulador de Circuito Lógico

Implemente um simulador de circuito lógico, seguindo os critérios descritos abaixo:

1) Ser capaz de montar um circuito lógico, usando a estrutura de árvore binária e fazendo uso de ponteiro para conectar os nós.
2) Cada nó deve representar uma porta lógica ou uma entrada. As portas lógicas e entradas serão representadas por um código, sendo:
   a) AXX - AND.
   b) OXX - OR.
   c) DXX - NAND.
   d) RXX - NOR.
   e) XXX - XOR.
   f) NXX - NOT.
   g) EXX - Entrada.
   h) Com exceção da porta "NOT", todas as portas possuem apenas duas entradas.
   i) XX representa o índice do nó (e.g.: A00 é a porta AND, de índice 00), sendo SEMPRE dois dígitos.
3) Existirão duas formas de montagem do circuito:
   a) Informe de conexão nó a nó, seguindo o padrão:
      i) portaXX portaOuEntrada1XX portaOuEntrada2XX.
      ii) portaXX - porta a ter os filhos definidos pelos argumentos a seguir.
      iii) portaOuEntrada1XX - definição do nó esquerdo.
      iv) portaOuEntrada2XX - definição do nó direito (este argumento não existirá, se portaXX for uma porta NOT).
      v) Antes das informações de conexão, o programa será informado da quantidade de comandos existentes para a construção do circuito.
   b) Passagem de códigos, por meio de notação prefixa, de portas lógicas e entradas, cabendo ao programa montar o circuito por meio de recursão.
   c) A forma de montagem será escolhida por meio de uma variável, sendo:
      i) 0 - tipo de conexão 3-a.
      ii) 1 - tipo de conexão 3-b.

4) Após a montagem do circuito, o programa deverá testá-lo com n conjuntos de entradas, exibindo cada uma das saídas, separadas por uma quebra de linha.
5) Todos os circuitos serão completos, i.e., todas as portas do circuito terão outras portas lógicas e/ou entradas conectadas, não existindo “terminais soltos”.
6) Todas as regiões de memória alocadas dinamicamente DEVEM ser liberadas antes do encerramento da execução.
7) COMENTAR O CÓDIGO!!!

Exemplo 1:

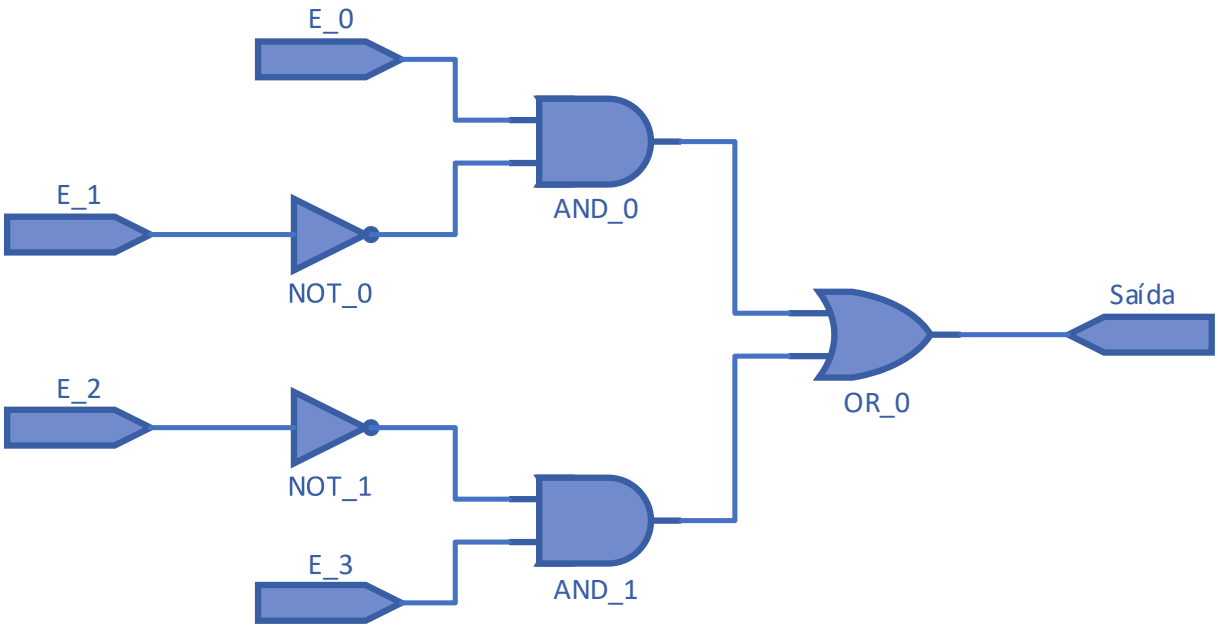
Entrada:

```
0 // A forma de montagem será do tipo descrito em 3-a.
5 // Quantidade de linhas que descrevem o circuito.
O00 A00 A01 // Porta “OR” com duas portas “AND” como entradas.
A00 E00 N00 // Definição das entradas da porta “AND” da linha anterior.
A01 N01 E03 // Definição das entradas da outra porta “AND” da terceira linha.
N00 E01 // Definição da entrada da porta “NOT” da quarta linha.
N01 E02 // Definição da entrada da porta “NOT” da quinta linha.
4 // Quantidade de entradas de teste.
0 0 0 0 // Entrada de teste 1 (E00, E01, E02, E03).
0 1 0 1 // Entrada de teste 2 (E00, E01, E02, E03).
1 0 1 0 // Entrada de teste 3 (E00, E01, E02, E03).
1 1 1 1 // Entrada de teste 4 (E00, E01, E02, E03).
```

Saída:

```
0 // Saída da entrada de teste 1.
1 // Saída da entrada de teste 2.
1 // Saída da entrada de teste 3.
0 // Saída da entrada de teste 4.
```

Exemplo do circuito lógico dado como entrada:



Exemplo 2:

Entrada:

```
1 // A forma de montagem será do tipo descrito em 3-b.
O00 A00 E00 N00 E01 A01 N01 E02 E03 // Entrada do circuito.
4 // Quantidade de entradas de teste.
0 0 0 0 // Entrada de teste 1 (E00, E01, E02, E03).
0 1 0 1 // Entrada de teste 2 (E00, E01, E02, E03).
1 0 1 0 // Entrada de teste 3 (E00, E01, E02, E03).
1 1 1 1 // Entrada de teste 4 (E00, E01, E02, E03).
```

Saída:

```
0 // Saída da entrada de teste 1.
1 // Saída da entrada de teste 2.
1 // Saída da entrada de teste 3.
0 // Saída da entrada de teste 4.
```

Exemplo do circuito lógico dado como entrada: