

TP03 - Circuitos Lógicos em Árvores Binárias NÃO ordenadas

**** Definição Geral do Trabalho: ÁRVORES BINÁRIAS NÃO ORDENADAS** - Criar rotinas, que podem ser adaptadas a partir do TAD ABO, porém, onde a inserção deve ser diferenciada para permitir que seja criada uma árvore específica, não ordenada e com a estrutura especificada de acordo com a descrição de um circuito digital (portas lógicas). Você pode adaptar TADs em C disponíveis (sugere-se os TADs do Backes e de seu livro), ou em C++ ou Python, desde que use uma estrutura de dados do tipo Arvore Binária com Alocação Dinâmica para armazenar os circuitos.

(*) Aceito em Python e C++ mas tem que usar ÁRVORES com ALOCAÇÃO DINÂMICA

Este trabalho é relacionado ao uso de **Árvores Binárias**, usadas para representar circuitos como o do diagrama abaixo: (a Raiz é a Saída “OUTPUT” e as folhas são as Entradas “INP” do Circuito Lógico – O esquema abaixo é como se a árvore estivesse “deitada” de lado)

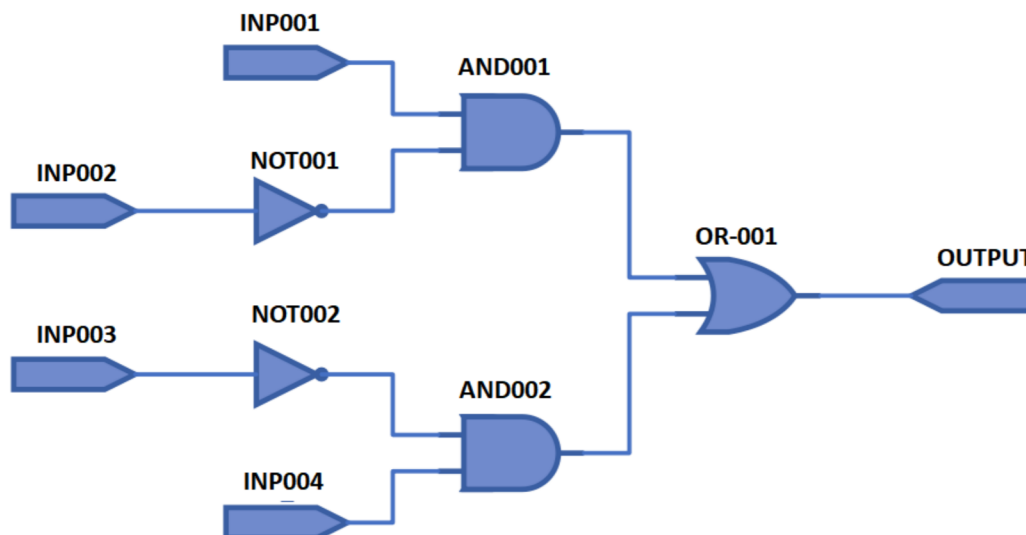


Figura 1 – Exemplo de Diagrama de um Circuito Digital com Porta Lógicas (Árvore)

Nota1: Observe que cada nodo da árvore é identificado por um IDENTIFICADOR ÚNICO e não DUPLICADO, do tipo “string”. Os nodos folhas são nodos sem filhos e especiais, pois são as entradas do circuito. O nodo raiz é um nodo especial pois representa o sinal de saída do circuito.

Nota2: Os nodos podem ser, portanto, dos seguintes tipos – Entradas (INPxxx), Portas Lógicas de 2 entradas (ANDxx, OR-xxx, XORxxx), Portas Lógicas de 1 entrada (NOTxxx com apenas 1 nodo filho) e a Saída do Circuito (OUTPUT). Podemos ter até 999 portas e nodos de cada tipo (xxx pode assumir valores de 000 a 999, não sendo obrigatório que sejam valores sequenciais ou que comece em zero, só não podem ter dois nodos com o mesmo identificador).

Nota3: A árvore será construída de cima para baixo, onde a inserção será sempre feita com uma busca, do nodo pai e pelo seu identificador, seguido de uma inserção a direita ou a esquerda, de seu nodo filho, conforme indicado na descrição do circuito, ou seja, acha nodo pai e insere filho.

**** Exemplos de descrição do Circuito da Figura 1:**

Arquivo "circuit.txt":

```
OUTPUT U .ROOT.  
OR-001 U OUTPUT  
AND001 L OR-001  
INP001 L AND001  
NOT001 R AND001  
INP002 U NOT001  
AND002 R OR-001  
NOT002 L AND002  
INP003 U NOT002  
INP004 R AND002  
$END00 X XXXXXX
```

Podemos observar que sempre é inserido ANTES o pai de um nodo, que pode ser encontrado para que seja inserido seu filho no lado Esquerdo (L - Left) ou no lado Direito (R - Right). Quando o nodo só tem um filho, ele é identificado como Único (U - Unique). Nodos folhas tem um Identificador que começa com "INP" ou "I", uma vez que os outros nodos tem identificadores que são bem diferentes.

Cada linha do arquivo, contém sempre um nodo a indicação de onde ele se liga em seu pai, e o identificador do nodo pai (deve buscar pelo pai e inserir o nodo como seu filho, de acordo com "U", "L" ou "R". Isso vale exceto para a primeira linha do arquivo, que cria o nodo raiz (Root) e que é a saída (Output) do circuito. O arquivo termina ao encontrar um nodo "\$END" (Linha: "\$END00 X XXXXXX" (se começa com "\$" já é um sinal de que não tem mais nodos para inserir na árvore).

ATENÇÃO: A descrição do circuito sempre deverá ser completa e correta para que a simulação do circuito possa ser realizada, ou seja, todos nodos terão filhos até chegar nas folhas (todas folhas são nodos INP), e a árvore deve poder ser construída corretamente. Se houver algum erro na construção (pai não encontrado, inserção de mais de um filho no mesmo lado de um determinado nodo pai, falta de um nodo folha em algum dos nodos intermediários da árvore, o programa deve ser encerrado e exibida a mensagem na tela de "CIRCUIT ERROR" (em maiúsculas). Portanto, a rotina que lê o circuito deve realizar estas verificações de erro: (1) inserção indevida de nodo filho - tentativa de inserir um nodo onde já existe um nodo inserido; (2) nodo pai não encontrado - e portanto não tem com inserir o nodo atual; (3) árvore incompleta, quando falta um nodo do tipo INP em alguma folha da árvore - verificação final ao terminar a construção da árvore.

Uma vez lido o arquivo de entrada ("circuit.txt") e criada a árvore, serão realizadas consultas da saída simulada do circuito, para um dado conjunto de entradas (valores dos nodos folhas).

Entrada no Circuito: (teclado)

```
INP001 1  
INP002 0  
INP003 1  
INP004 0
```

Saída do Circuito: (tela)

1

Nota: os valores indicados na entrada são colocados nos nodos folhas e "propagados" para a raiz, seguindo o funcionamento da simulação das portas lógicas (simulação do circuito). O valor obtido na raiz é a saída.

Mais um exemplo de Entrada e Saída para o mesmo circuito da Figura 1:

Entrada no Circuito: (teclado)

```
INP001 0
INP002 0
INP003 0
INP004 0
```

Saída do Circuito: (tela)

```
0
```

Mais exemplo de circuitos, entradas e saídas serão fornecidos em breve.

```
=====
F.Osorio
10 Dez. 2021
=====
```