Instituto de Ciências Matemáticas e de Computação
| Universidade de São Paulo |

SSC0300 - Linguagem de Programação e Aplicações

Professor responsável: *Fernando Santos Osório*
Semestre: 2013/2
Prof. Auxiliar: *Artur Lovato Cunha*
Estagiário PAE: *Rafael L. Klaser*
Horário: Sexta Manhã 10h10

E-mail Prof. : fosorio @ icmc.usp.br
fosorio @ gmail.com
E-mail Prof.: arturlc @ icmc.usp.br
E-mail PAE: rlklaser @ gmail.com
Web: ***http://www.icmc.usp.br/~fosorio/***

TRABALHO PRÁTICO 2013 – SIMULADOR DE CIRCUITOS

Opção 01 (Versão 1.0 - 04/11/13)

WIKI http://wiki.icmc.usp.br/index.php/SSC-301

Faça um programa para "simular" um circuito lógico usando árvores de um tipo similar as árvores binárias estudadas em aula, com alocação dinâmica (usando ponteiros), de acordo com a descrição dada abaixo. O programa deve possuir um menu com as seguintes opções:

1. Ler circuito de um arquivo em disco
2. Exibir a árvore do circuito lógico
3. Exibir a função implementada pelo circuito
4. Simular o circuito lógico – Entrada de um arquivo
5. Simular o circuito lógico – Entrada do teclado
6. Salvar o circuito em um arquivo em disco
7. Terminar a execução do simulador

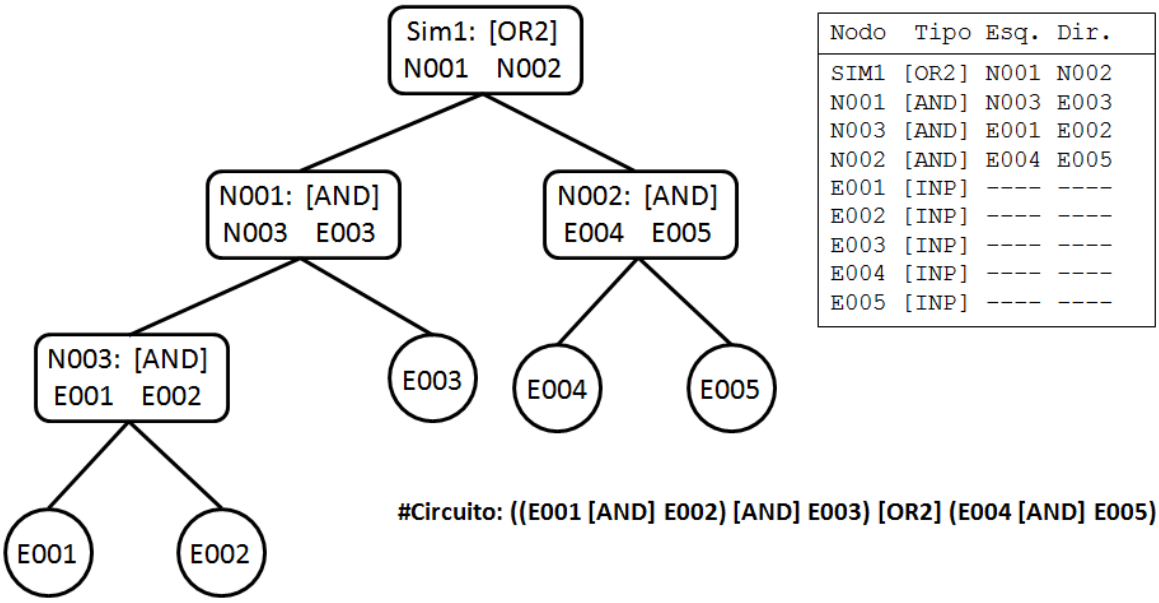
1. Ler arquivo e criar circuito: Esta opção deve permitir ao usuário ler um arquivo texto do disco, cujo nome será informado por ele. Este arquivo contém a descrição de um circuito lógico, conforme descrição informada mais abaixo (ver figura 1). O arquivo irá permitir que seja criada uma nova árvore descrevendo o circuito lido, e posteriormente simular este circuito. Exemplo:

   Entre com o nome do circuito: simul-01.txt
   >> Lendo arquivo...
   >> Circuito lido, estrutura de dados criada.

2. Exibir o circuito lógico na tela: exibir a árvore de modo a permitir que o usuário possa visualizar os diferentes níveis desta, com a descrição dos seus respectivos nodos (tipo de operador). Exemplo: (A and B) or C

3. Exibir a função lógica, conforme indicação dada no final do arquivo que descreve o circuito (circuito final informado no arquivo). Exibir em seguida, em modo pós-fixado (EDV), a função lógica do circuito carregado na memória, conforme os dados contidos na árvore. Exemplo:

   Circuito final conforme descrito no arquivo: ((A and B) and C) or (D and E)
   Circuito armazenado na árvore (pós-fixado): A B and C and D E and or



Nodo	Tipo	Esq.	Dir.
SIM1	[OR2]	N001	N002
N001	[AND]	N003	E003
N003	[AND]	E001	E002
N002	[AND]	E004	E005
E001	[INP]	----	----
E002	[INP]	----	----
E003	[INP]	----	----
E004	[INP]	----	----
E005	[INP]	----	----

Figura 1 – Esquema da árvore representando o circuito e de sua descrição textual (simul-01.txt)

4. Simular o circuito lógico baseado nos dados de entrada lidos de um arquivo em disco, obtendo o valor da saída final deste circuito. Esta opção irá ler um arquivo texto, cujo nome é informado pelo usuário, e usar os dados lidos para realizar a simulação (dados de entrada = valores dos nodos folhas da árvore). Como resultado da simulação, vamos obter um valor de saída que deve ser exibido na tela - uma única saída, com valor 0 ou 1. Exemplo, considerando o seguinte circuito:

   ((E001 and E002) and E003) or (E004 and E005)

   Entre com o nome do arquivo de entrada: input-01.txt
   Valores de entrada lidos do arquivo:
   E001: 0 E002: 1 E003: 0 E004: 0 E005: 1
   Valor de saída do circuito: 0 (zero)

5. Simular o circuito lógico baseado nos dados de entrada lidos do teclado. Esta opção realiza a mesma operação que é feita pela opção anterior, só que os dados de entrada são obtidos através da leitura destes pelo teclado. Esta opção irá ler valores 0 ou 1 correspondendo a cada uma das entradas do circuito – nodos folhas (dados de entrada). Como resultado da simulação, vamos obter um valor de saída que deve ser exibido na tela, sendo uma saída única, com um valor igual à 0 ou 1. Exemplo:

   Entre com os valores das entradas:
   E001: 0 E002: 1 E003: 0 E004: 0 E005: 1
   Valor de saída do circuito: 1 (um)

6. Salvar o circuito em um arquivo em disco com o nome especificado pelo usuário: gerar um arquivo texto contendo a descrição de todos os dados contidos na árvore do circuito. Salvar o arquivo de forma que este possa ser lido posteriormente, reconstruindo exatamente a mesma árvore que foi salva (Modo pré-fixado - VED)

7. Sair do programa.

Os arquivos usados pelo programa de simulação de circuitos possuem seus dados organizados de acordo com o formato que vamos especificar aqui. O programa deve respeitar o formato especificado para que possam ser lidos os seguintes arquivos:

1. Arquivo de Descrição de Circuitos
2. Arquivo de Entradas

⇨ 1. Arquivo de Descrição de Circuitos:

- Cada linha do arquivo descreve uma sub-função => porta lógica (operador e seus operandos – um ou dois operandos), onde a linha poderá conter os seguintes dados:

  Nome_do_Nodo Operador_Unário Operando_Esquerdo
  ou
  Nome_do_Nodo Operador_Binário Operando_Esquerdo Operando_Direito
  ou
  Nodo_Entrada Identificador_entrada ---- ----

  Exemplos:

  Operadores Unários: NOT (Not = Negação, Xor = Exclusive Or)
  Exemplo: **N001 [NOT] E001 ----**

  Operadores Binários: AND, OR2, XOR, NOR (Or2 = Or 2 entradas, NOR = Not Or)
  Exemplos: **N001 [AND] E001 E002**
  N002 [OR2] E003 E004
  N003 [XOR] E005 E006
  N004 [NOR] E007 E008

  Nodo_entrada: INP
  Exemplo: **E001 [INP] ---- ----**

  Portanto, cada linha contém o "Nome_do_Nodo" (4 caracteres), seguido de seu tipo (5 caracteres, podendo ser: [NOT], [AND], [OR2], [XOR], [NOR] ou [INP]), seguido do nome do nodo a sua esquerda "Nodo_Esq" (4 caracteres) e o nome do nodo a sua direita (4 caracteres). Se o nodo a esquerda/direita for um nodo interno da árvore, seu nome começa com 'N' seguido de um número, se o nodo a esquerda/direita for um nodo folha da árvore (que não possui mais filhos), seu nome começa com 'E' seguido de um número. Os nodos folha da árvore são os nodos usados de ENTRADA de dados ("inputs", onde seu tipo é [INP] e ele não possui nodos filhos "----").

- O arquivo termina com uma linha especial, que contém a palavra "#Circuito:", a qual indica o final das descrições do circuito (nodos e entradas), sendo esta palavra seguida de uma descrição fornecida pelo usuário do circuito lido (esta é a função lida que será apresentada na opção 3). A linha com a descrição do circuito lido é dada pronta e devemos "acreditar" que ela representa corretamente o circuito que foi descrito no arquivo. Exemplo:

```
SIM1 [OR2] N001 N002
N001 [AND] N003 E003
N003 [AND] E001 E002
N002 [AND] E004 E005
E001 [INP] ---- ----
E002 [INP] ---- ----
E003 [INP] ---- ----
E004 [INP] ---- ----
E005 [INP] ---- ----
#Circuito: ((E001 [AND] E002) [AND] E003) [OR2] (E004 [AND] E005)
```

IMPORTANTE:
A ordem das linhas que descrevem os nodos da árvore no arquivo, são colocadas obrigatoriamente de modo que a árvore será construída de cima para baixo, começando pela raiz. Isso deverá simplificar a construção da árvore do circuito.

Dica: procura o nome do nodo, e insere onde ele deve ficar, cuidando para ver se este nodo é um nodo interno (começa com 'N'), ou se é um nodo foha (começa com 'E'), ou se é o nodo raiz (não começa nem com 'N' e nem com 'E').

- Os operadores possíveis são:

  AND (E) , **OR2** (Ou) , **XOR** (Ou exclusivo), **NOR** (E negado), **NOT** (Negação)
  No caso da negação, deve ser gerado um nodo especial que terá somente a árvore esquerda, sem ter uma árvore direita associada. Este nodo deverá ser tratado de modo especial na construção e simulação do circuito (sugere-se que seu nodo a direita tenha o valor "----").

Um nodo da árvore deve portanto conter as seguintes informações:
- *Identificador*. Exemplo: E001, E002, E003, ... (nos nodos folhas) ou N001, N002, N003, ... (nas portas, ou seja, nodos intermediários);
- *Tipo do nodo*. Exemplo: Entrada ([INP]) ou Operador ([AND], [OR2], [XOR], ...);
- *Valor resultado*. Exemplo: um nodo And com folhas E001 e E002, o resultado é o valor do cálculo de E001 <and> E002, podendo resultar em 0 ou 1 (ver tabelas verdade abaixo para o cálculo da saída dos nodos de portas lógicas). Se for um nodo folha o valor é o próprio valor de entrada lido;
- *Ponteiros* para a árvore esquerda e árvore direita, junto com seu respectivo nome;
- Estas informações indicadas aqui são o mínimo aconselhado, podendo ser adicionadas outras informações se julgar necessário.

⇨ Arquivo de Entradas:

- Cada linha descreve uma entrada, contendo uma palavra (4 caracteres – nome da entrada) e um valor (0 ou 1), onde abaixo é mostrada um exemplo dos dados contidos em um arquivo de entrada:

  Entrada Valor

  Exemplo:
  E001 1
  E002 0
  E003 0
  E004 1
  E005 0

⇨ TABELAS VERDADE: (como calcular a saída dos circuitos lógicos)

AND		
In A	In B	Out
0	0	0
1	0	0
0	1	0
1	1	1

OR2		
In A	In B	Out
0	0	0
1	0	1
0	1	1
1	1	1

XOR		
In A	In B	Out
0	0	0
1	0	1
0	1	1
1	1	0

NOR		
In A	In B	Out
0	0	1
1	0	0
0	1	0
1	1	0

NOT	
Input	Output
0	1
1	0

OBSERVAÇÕES FINAIS:

- O programa deve ser **entregue até o dia 06.12.2013**.
- O Trabalho Prático Final da disciplina é um projeto individual ou em duplas.

- **Entregar** o **programa fonte por e-mail** para o professor, enviando para estes 2 endereços: fosorio@gmail.com com cópia para work2usp@yahoo.com

- Este programa deverá simular os circuitos lógicos, baseando-se em uma estrutura de dados do tipo árvore (do tipo das árvores binárias). **Esta estrutura de dados deve ser baseada (similar) ao que foi estudado em aula. As rotinas de manipulação de estruturas de dados fornecidas em aula podem ser usadas, e os programas devem ser escritos de maneira modular.** A boa estruturação e modularidade do programa irá contar na avaliação! *Evite programas "espaguetti" e o excesso de uso de variáveis globais*!

- Exemplos de arquivos de circuito usados para testar o simulador serão colocados na Internet junto a página da Wiki da disciplina:
  http://wiki.icmc.usp.br/index.php/SSC-300-2013(fosorio)

BOM TRABALHO!