HÁZI FELADAT

Szoftver Laboratórium 2.

Dokumentáció

Kiss Balázs

RWZYYD

2011. április 1.

Feladat

```
   Készítsen egyszerű objektummodellt digitális áramkör
   szimulálására! A modell minimálisan tartalmazza a következő
   elemeket:
        - NOR kapu
        - vezérelhető forrás
        - összekötő vezeték
        - csomópont
   A modell felhasználásával szimulálja egy olyan 5 bemenetű
   kombinációs hálózat működését, amely akkor ad a kimenetén
   hamis értéket, ha bementén előálló kombináció 5!

Demonstrálja a működést külön modulként fordított tesztprogrammal!
A megoldáshoz NE használjon STL tárolót vagy algoritmust!


A tesztprogramot úgy specifikálja, hogy az parancssoros batch alkalmazásként (is)
működjön, azaz a szabványos bemenetről olvasson, és a szabványos kimenetre,
és/vagy a hibakimenetre írjon!
Amennyiben a feladat teszteléséhez fájlból, vagy fájlokból kell input adatot
olvasnia, úgy a fájl neve *.dat alakú legyen!
```

Pontosított feladatspecifikáció

A feladat egy egyszerű digitális hálózatot kezelő program létrehozása, amely lehetőséget ad kapuk, vezetékek, csomópontok és vezérelhető források szimulálására. A program segítségével ezekből az építőelemekből bármekkora kombinációs hálózatot felépíthetünk. A feladatban nem szerepelt kritériumként sorrendi hálózathoz szükséges alkatrész, így ezeknek (flipflop-ok, órajelezés, stb...) a kidologzásától eltekintek.

A programot parancssoros felhasználói felülettelel valósítom meg, a felhasználó számára elérhető parancsokat dokumentálom. A program más program által is vezérelhető lesz (standard csatornákon keresztül).

A programnak a hálózat felépítése során észlelnie kell az értelmetlen kapcsolásokat (két kimenet összekötése), és a szimuláció során a versenyhelyzetek kialakulását.

A program kellően moduláris lesz ahhoz, hogy könnyen bővíthető legyen grafikus kezelőfelülettel, további alkatrésszel, sorrendi hálózat kezeléssel.

Felhasználói dokumentáció

A program letöltése

A program letölthető .zip formátumban a https://github.com/balazskiss/DigitalCircuitSimulator címről, vagy klónozható a git tárolóból az alábbi parancsok segítségével:

```
git clone git://github.com/balazskiss/DigitalCircuitSimulator.git
```

A program fordítása és futtatása

A program fordítása, majd futtatása az alábbi parancsokkal lehetséges:

```
make
./dcs
```

A program használata

A program elindítása után létrejön egy üres hálózat, melybe elkezdhetjük felvenni az alkatrészeket, vezetékeket, stb...

A program az alábbi parancsokat tudja értelmezni:

- help
- exit
- new
- close
- open
- save
- add
- print
- mod
- del
- wire
- unwire
- run

A help parancs

A help parancs segítségével egy listát kaphatunk az elérhető parancsokról és azok használatáról.

Az exit parancs

A programból az exit parancs segítségével léphetünk ki.

A new parancs

A new paranccsal új digitális hálózatot kezdhetünk. Ha már meg van nyitva hálózat, azt előbb be

kell zárnunk.

A close parancs

A pillanatnyilag megnyitott digitális hálózatot zárhatjuk be vele.

Az open parancs

Az open paranccsal egy fájlban tárolt digitális hálózatot tölthetünk be.

Használata:

```
open megnyitando_fajl_neve
```

A save parancs

A save paranccsal az aktuális munkánkat egy fájlba menthetjük, így később az open paranccsal bármikor megnyitható és folytatható lesz.

Használata:

```
save kimeneti_fajl_neve
```

Az add parancs

Az add paranccsal új alkatrészt adhatunk a hálózathoz.

Hazsnálata:

```
add alkatresz_neve
```

A program jelen verziójában az alábbi alkatrészek érhetők el:

- AND (ÉS kapu)
- NAND (NAND kapu
- OR (VAGY kapu)
- NOR (NOR kapu)
- LED
- Node (Csomópont)
- Inverter (Invertáló)
- Switch (Vezérelhető forrás)
- Positive (Pozitív)
- Negative (Negatív)

A print parancs

A print parancs a hozzáadott alkatrészek kilistázására szolgál. Kilistázza minden alkatrész nevét, bemeneteit és kimeneteit. Minden alkatrésznek kiírja az egyedi azonosítóját, amellyel a konkrét alkatrészre más parancsok pramaétereiben hivatkozhatunk.

A mod parancs

A mod parancs már előzőleg hozzáadott alkatrész beállítására/módosítására szolgál.

Használata:

```
mod alkatresz_azonosito beallitas
```

A program jelen verziójában csak a vezérelhető forrás az egyetlen módosítható alkatrész. Más alkatrész módosítása esetén hibaüzenetet kapunk.

Egy vezérelhető forrás bekapcsolása és kikapcsolása az alábbi parancsokkal lehetséges:

```
mod vezerelheto_forras_azonositoja on
mod vezerelheto_forras_azonositoja off
```

A del parancs

A del parancs előzőleg hozzáadott alkatrészek törlésére szolgál.

Hazsnálata:

```
del alkatresz_azonosito
```

A wire parancs

A wire parancs két alkatrész összekötésére szolgál. Az első paraméterként megadott alkatrész kimentére köti rá a második paraméterként megadott alkatrész bemenetét.

Használata:

```
wire alkatresz_azonosito1 alkatresz_azonosito2
```

Az unwire parancs

Az unwire parancs két alkatrész közti vezeték törlésére szolgál. Az első paraméterként megadott alkatrsz kimenetéről leálasztja amásodik paraméterként megadott alkatrész bemenetét.

Használata:

```
unwire alkatresz_azonosito1 alkatresz_azonosito2
```

A run parancs

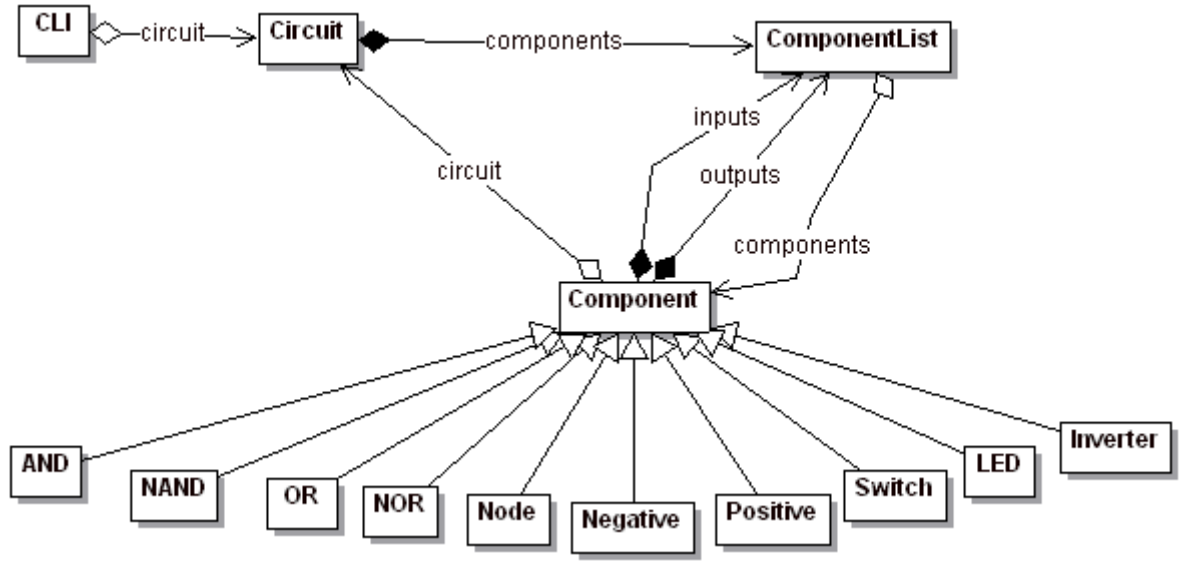
A run parancs a digitális hálózat szimulációjára szolgál. Mivel a LED az egyetlen alkatrész amin látható az értéke, a parancs kilistázza az összes LED-et a hálózatban és kiírja az értéküket (on vagy off).

Fejlesztői dokumentáció

Verziókezelés

A program verziókezelését a github.com ingyenes git tároló szolgáltatás biztosította. A projekt elérhető a https://github.com/balazskiss/DigitalCircuitSimulator cím alól.

Objektum hierarchia



Objektumok feladata

CLI

A parancssoros felhasználói felület megvalósítója. Feladata a felahsználó által begépelt parancsok értelmezése, feldolgozása és ezek alapján a Circuit osztály vezérlése. Lehetőséget ad fájl betöltésére, hálózat elmentésére, kilépésre, stb...

Circuit

A digitális hálózatot kezelő osztály. Ez az osztály tartalmazza az alkatrészeket és tárolja a kezeli a köztük levő összeköttetéseket (vezetékek).

ComponentList

Alkatrészeket tároló dinamikus méretű tömb. Component mutatókat tárolhatunk benne. Megvalósított műveletek: hozzáadás, törlés, lekérdezés (operator[]), méretlekérdezés.

Component

Az alkatrész alaposztály. Absztrakt osztály, ebből származik minden Circuit-hoz adható alkatrész. Tárolja a kapcsolatokat a bemeneteivel és a kimeneteivel, kimeneti értéke lekérdezhető. A bemeneteinke és kimeneteinek száma korlátozható. Két virtuális függvénye van, melyet az ebből származó osztályoknak meg kell valósítani a getValue() és a getName() (ez a két függvénye minden alkatrésznek egyedi).

AND, NAND, OR, NOR, Inverter, stb...

Component osztályból származó alkatrészek. A nevükben, a maximális bemenetükben és kimenetükben, valamint a kimeneti értékük számolásában térnek el.

Switch

A Switch annyiban különbözik a többi származtatott alkatrésztől, hogy az alaposztály egy value mezővel kibővíti. Ez az érték teszi lehetővé a vezérelhetőségét.

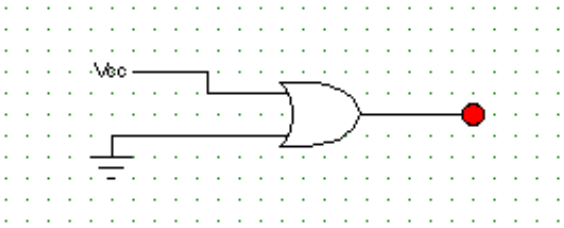
Részletes fejlesztői dokumentáció

A program forráskódját Javadoc stílusban kommenteztem és a DoxyGen programmal generáltam hozzá minden objektumra kiterjedő, részletes fejlesztői dokumentációt.

A dokumentáció a projekt doc/html könyvtárában található.

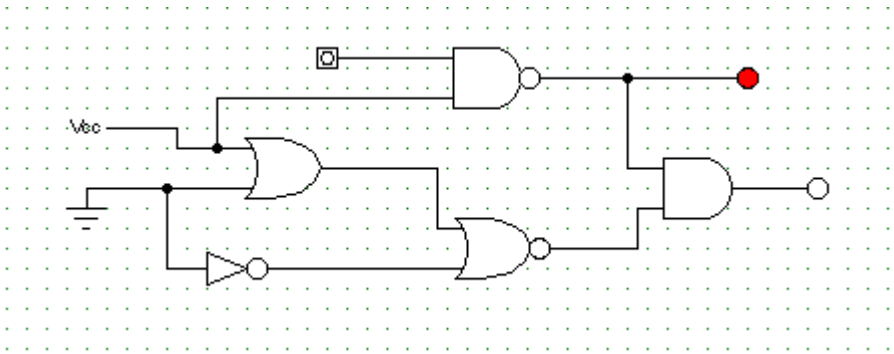
Tesztelési dokumentáció

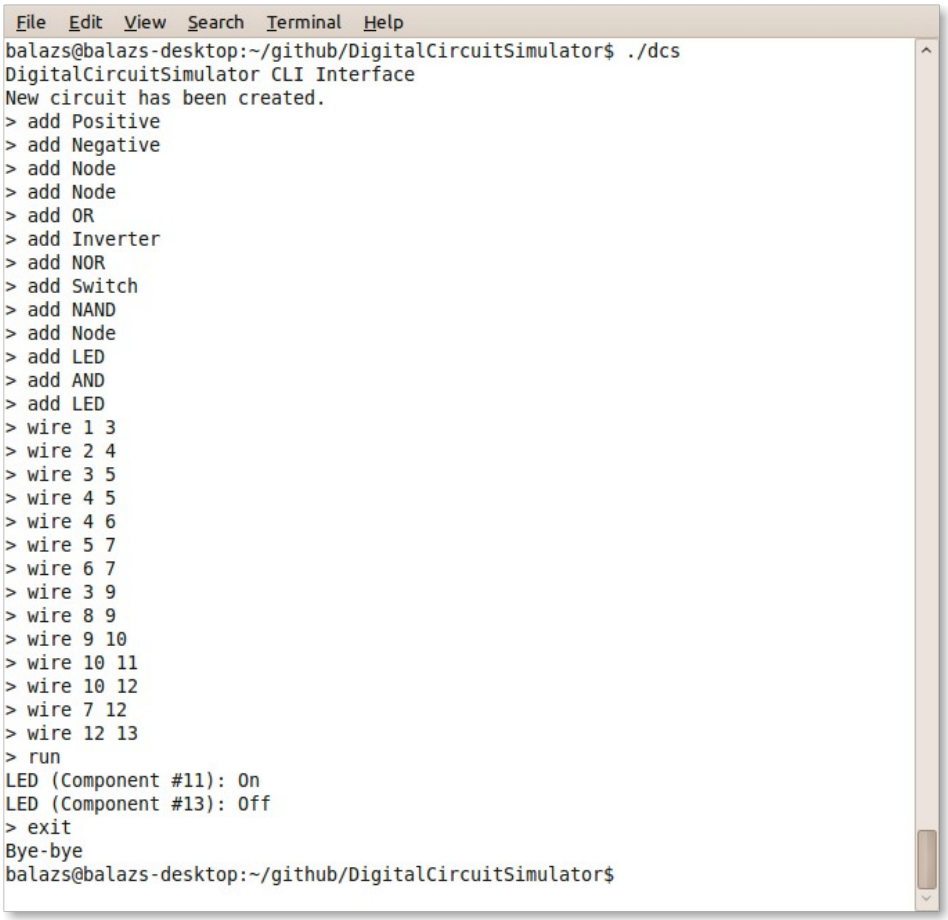
1. teszteset




```
File  Edit  View  Search  Terminal  Help
balazs@balazs-desktop:~/github/DigitalCircuitSimulator$ ./dcs
DigitalCircuitSimulator CLI Interface
New circuit has been created.
> add Positive
> add Negative
> add OR
> add LED
> wire 1 3
> wire 2 3
> wire 3 4
> run
LED (Component #4): On
> exit
Bye-bye
balazs@balazs-desktop:~/github/DigitalCircuitSimulator$
```

2. teszteset





3. teszteset

A 3. teszteset csak a program elhanyagolhatóan kevés részét nem futtatta.

A tesztprogram a következő:

new
close
new
mitirki
add Positive
add LED
wire 1 2
run
add Switch
mod 3 off
wire 2 3
print
wire 1953 17456
add Negative
add AND
add NAND
add OR
add NOR
add Switch
mod 6 on
mod 7 illegal_setting
mod 7 on
add Node
add LED
add Inverter
add Szupertehen

```
help
print
run
del 2
wire 5 6
unwire 5 6
wire kutya macska
unwire eger sas
save kimenet.dc
open kimenet.dc
close
add NAND
open
open nincsilyenfajl.txt
open kimenet.dc
exit
```

Memóriaszivárgás tesztelése

A memóriaszivárgást a Valgrind programmal ellenőriztem.