Házi feladat

Feladatválasztás/specifikáció

Programozás alapjai 2.

Pálinkás Lőrinc Mihály - XB0SMF

Tartalom

[1. Feladat 3](#_Toc163734838)

[Digitális áramkör 3](#_Toc163734839)

[2. Feladatspecifikáció 4](#_Toc163734840)

[Feladat általános leírása 4](#_Toc163734841)

[Megvalósított áramköri elemek 4](#_Toc163734842)

[Bemenet formátuma 4](#_Toc163734843)

[Kimenet opciók 5](#_Toc163734844)

# Feladat

## Digitális áramkör

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!

# Feladatspecifikáció

## Feladat általános leírása

A program lehetőséget ad digitális áramkörök szimulálására. A felhasználó áramköröket képes betölteni szöveges file-okból, beállítani a bemeneti jelkombinációt és a kapcsolók állapotát és ez alapján kiolvasni a kimeneti jeleket.

## Megvalósított áramköri elemek

A következő elemeket képes szimulálni az áramkör:

* Forrás: állítható LOW és HIGH kimeneti jelekkel, kiolvasható az értéke
* Vezeték: két részt köt össze az áramkörben
* Csomópont: 1 bemeneti jelet több kimeneti irányba tud továbbítani
* Kapu: Több bemenetből képes pontosan 1 kimenetet produkálni. Megvalósított kapuk:
  + AND
  + OR
  + XOR
  + NOT
  + NAND
  + NOR
  + XNOR
* Lámpa: tárolja a kapott jelet, kiolvasható az értéke
* Kapcsoló: továbbítja a jelet, amennyiben zárt, egyébként LOW jelszintet ad ki

A bonyolultabb elemeket (pl. funkcionális elemek) egyelőre nem implementáljuk, mert könnyen felépíthető ezekből szimuláció során, de ha marad idő, akkor ezeket is megvalósíthatjuk.

## Bemenet formátuma

Az áramkörök felkonfigurálása szöveges file alapján történik. Ebben a felhasználó felsorolja a komponenseket, megadva, hogy hogyan kapcsolódnak. A kapcsolódás megadásához meg kell adni, hogy az adott lába az elemnek melyik csompópontra kapcsolódik. A csomópontokat számok jelölik megadáskor, azonos szám azonos csomópontot jelent. Tehát a konfigurációs file körülbelül így néz ki:

*test.txt*

SOURCE: (1) (2) (3)

AND: (1,2,4)[(…,…,…) …] <- ha több van

OR: (2,3,5) …

XNOR: (4,5,6) …

LAMP: (6)

A black line drawing of a person with a pointy object

Description automatically generatedPéldául erről az ábráról azt tudjuk leolvasni, hogy 3db forrás van jelen, ezek az 1-es, 2-es és 3-as csomópontokra küldik a jeleiket. Emellett van az 1 és 2-es csomópontra kapcsolódó ÉS kapu, mely a 4-es csomópontra küldi a jelét. Hasonlóan kell értelmezni a többit. Ez alapján az alábbi digitális áramkör szimulálható:

Fontos megjegyzés: A szimuláció során az összekötő vezetékeket is csomópontnak tekintünk, így tudjuk könnyen megadni formátumosan a kapcsolódásokat.

A példa azt is mutatja hogy milyen egy egyszerű kapu megadásának például általános formátuma:

GATE\_NAME: (IN1, IN2, OUT1) …

## Kimenet opciók

A felhasználó képes lekérdezni több információt az áramkörből:

* A lámpák státusza: minden lámpának ki tudjuk olvasni az állapotát, hogy világít-e vagy nem.
* A források státusza: minden forrásnak meg tudjuk adni és ki tudjuk olvasni a jelszintjét.
* A kapcsolók státusza: minden kapcsolónak meg tudjuk adni és ki tudjuk olvasni, hogy zárva van-e vagy sem.

Az áramkör kimenetének megadható, hogy melyik file-ba irányítjuk át a szimuláció kimenetét.

# Pontosított specifikáció

## Áramköri elemek I/O pin-ek száma

Az alábbi táblázat mutatja az egyes elemekhez tartozó ki- és bemeneti pin-ek számát, amivel létre lehet hozni:

|  |  |  |
| --- | --- | --- |
| Áramköri elem típus | Bemeneti pin-ek száma | Kimeneti pin-ek száma |
| Forrás | 0  (csak jelet ad ki) | 1  (egy jelet ad ki) |
| Csomópont | 1  (ahonnan kapja a jelet) | ≥1  (tetszőlegesen sok helyre küldhet jelet) |
| Vezeték | 1  (csomópont speciális esete) | 1  (csomópont speciális esete, amikor 1 kimenet van) |
| Kapu | ≥1  (minden kapu legalább egy bemenetből állít elő kimenetet, támogatva lesz több mint 2 bemenetű AND, OR, stb.) | 1  (minden kapu 1 logika jelet állít elő) |
| Lámpa | 1  (1 helyről fogad jelet) | 0  (nem ad ki jelet, csak eredmény tárolásra van) |
| Kapcsoló | 1  (1 helyről fogad jelet) | 1  (1 helyre továbbít jelet) |

## Felhasználói felület

A felhasználó számára lesz biztosítva egy egyszerű menü, melyben a következő műveleteket tudja elvégezni:

* Áramkör betöltése: képes megadni egy file nevét, és innen betölteni egy áramkört
* Bemeneti adatok beállítása: meg tudja adni a források bemeneti jeleit illetve a kapcsolók állását
* Kimeneti file beállítása: meg tudja adni hogy melyik file-ba irányítsa át a kimenetet, alapvetően a std::cout-ra küldi a szimuláció kimenetét
* Szimuláció: végrehajtja a szimuláció lefuttatását
* Kilépés: leállítja a programot

# Terv