XI. FIATAL MŰSZAKIAK TUDOMÁNYOS ÜLÉSSZAKA

Kolozsvár, 2006. március 24-25.

VIRTUÁLIS PROTOBOARD

Pótári Tamás

Abstract

The title of our project is Virtual Protoboard. Our aim is to make virtual software for designing logical electric circuit. By means of this software students can acquire the basis of digital networking. Furthermore, they are able to learn the routine of how the components work and are built in the electric circuit in the protoboard. Designing this software great attention is paid for external appearance particularly for the user-friendly interface. We also try to simulate realistically the work which can be performed on protoboard. Users can control their self-made logical network from the point of view of operation and they can also modify or save it to file.

Összefoglalás

A projektünk címe: Virtuális Protoboard. Célunk egy virtuális, logikai áramkör tervező szoftver készítése, melynek segítségével a tanulók könnyedén el tudják sajátítani a digitális hálózatépítés alapjait. Mindazonáltal az alkatrészek működtetésének és áramkörbe építésének rutinját is meg tudják tanulni a próbatáblán. A szoftver tervezése során nagy figyelmet fordítunk a külalakra, ezen belül is a felhasználó barát kezelőfelületre. Megpróbáljuk a szoftverünkkel a lehető legéletszerűbben szimulálni a próbatáblán elvégezhető munkálatokat. A felhasználó, az általa felépített logikai hálózatot képes működési szempontból ellenőrizni, módosítani illetve állományba elmenteni.

A protoboard

A protoboard [1] egy különleges áramkör-építési segédeszköz, egy dugaszolós kísérleti panel. A panel műanyag felületén, egymás közelében nyílások találhatók, melyekbe merev vezetékeket, alkatrész kivezetéseket lehet bedugni. A panel belsejében rugózó fémlemez bújik meg, mely az egymás melletti nyílásokba bedugaszolt vezetőket összekapcsolja. Így gyorsan, egyszerűen lehet áramköröket építeni, s nem kell forrasztani sem a kivezetéseket. (Az 1. ábra a protoboard szokásos kialakítását ábrázolja.)

1. ábra

A szoftver célja

A szoftver oktatási célokat szolgál. Ahhoz, hogy egy hallgató az otthonában is megtapasztalhassa egy egyszerű logikai áramkör összeépítését a gyakorlatban, rendelkeznie kell az ahhoz szükséges berendezésekkel, alkatrészekkel. Az ilyen tervezői környezet otthoni kialakítása jelentős anyagi követelménnyel bír. Még költségesebbé teszi a kísérletezést, ha egy-egy alkatrész, illetve berendezés tönkremegy. Más szempontból tekintve a lehetőségeket, egy bonyolult SPICE (Simulation Program with Integrated Circuit Emphasis) programra épülő tervezői szoftver megvásárolása anyagilag jóval megterhelőbb. A DesignSoft Kft. által tervezett Edison elektrotechnikai oktatóprogram már kedvezőbb tulajdonsággal bír a magáncélú alkalmazást tekintve. De ez a szoftver öreg, nem fejleszthető és csak koramúlt operációs rendszereken futtatható. Az általunk tervezett szoftver ingyenes, és minden érdeklődő számára elérhető. Nyílt forráskódú, szabadon fejleszthető, valamint Windows és Linux operációs rendszeren is futtatható. A szoftver az úgynevezett „virtuális labor” alapját is képezheti. Internet segítségével a laboratóriumi munkát a hallgató tetszőleges helyről végezheti. Az eredményeit pedig az oktatási intézmény regisztrálja és értékeli.

A szoftver lehetőségei

A szoftver segítségével lehetőségünk nyílik kezelni a tanszékünkön található mérési adatgyűjtő kártyáinkat, ami arra szolgál, hogy egy külső fizikai rendszer állapotváltozóit megjeleníthessük a programmal, illetve egy virtuális áramkör közbeiktatásával, ezen jellemzőket megváltoztassuk és a fizikai rendszerbe visszajuttassuk. (2. ábra)

2. ábra

A fejlesztési stádiumok

A szoftvert C++ programozási nyelven, OpenGL grafikai könyvtár segítségével fejlesztjük. A tervezés több stádiumból épül fel. Jelenleg az első stádiumban a hangsúlyt a grafikus megjelenítésre, az alapvető áramkörti elemek modellezésére és az egyszerűbb kombinációs hálózatok [2] tervezésére fektetjük. Szintén az első stádiumban tűzzük ki feladatul az alapvető algoritmusok tökéletesítését, melyek a hálózatanalizáló számításokat végzik a szoftver működtetése során. Ezen algoritmusok képezik a szoftver magját. A következő stádiumokban már a bonyolultabb hálózatok analíziséhez szükséges fejlesztések beépítésére kerül sor, amelyek segítségével már visszacsatolásokkal rendelkező hálózatokat is felépíthet a felhasználó.

A szoftver főbb egységei, azok feladatai

A tervezés során a szoftvert két nagy egységre bontottuk. A grafikus megjelenítést szolgáló egységre és a program-mag egységre. A grafikus megjelenítés egységében a grafikus funkciók mellett sorakoznak fel a billentyűzet és egérkezelést biztosító függvények. A program-mag végzi az összeépített hálózat működtetését és analízisét. A szoftvertervezést megelőző analízis során szükségesnek éreztük a központi szerepet játszó program-mag felosztását, feladat-végrehajtás alapján.

❖ A bejáró algoritmus feladata az áramkörben szereplő áramkörti elemek feltérképezése [3] és a köztük lévő kapcsolatok tárolása. Az eljárás a grafikus objektumokhoz kapcsolódó adatszerkezeteket vizsgálja meg. A bejáró fontossága az áramkörti elágazások vizsgálatában rejlik, hiszen minden irányt végig kell elemezni, fel kell térképezni, hogy milyen alkatrészek szerepelnek rajtuk. A műveleti sorrend a következő:

- Az áramkör tervezés során szinte minden szükséges információ tárolásra kerül a megfelelő adatszerkezetekben. Ilyenek az elágazások is, amiket változók segítségével jelzünk, pozíciójuk, sorrendbeli elhelyezkedésük és elágazás irányok alapján. Jelentősen egyszerűsíti a dolgunkat, hogy a protoboard vezetőiből (oszlopaiból) csak 5 irányú elágazás létesíthető (3. ábra). Az ellenőrző ciklikus rutin addig fut, ahány főági elágazás tárolva van.
- Az elágazások mellett, a logikai kapukat (4. ábra) csomópontként kezeljük, ami szintén feltételt szab a ciklikus ellenőrző rutinnak, mivel az elágazások így tudnak véglegesen lezárulni. Az efféle csomópontok feltérképezetlen irányba eső kimenetén szállítandó logikai érték, a két bemeneten szállítandó érték alapján módosul. De erre a következő lépés biztosít megoldást.

3. ábra

4. ábra

- ❖ Az első lépésben tárolt hálózati kapcsolatot alapján biztosítani kell az információ cserét az adattagok között. Itt érvényesülnek az IC-k típusaitól függő logikai műveletek, és ezek alapján módosulnak a szállítandó logikai jelértékek.
- ❖ Majd a lehetséges bemeneti kombinációk által meghatározott kimeneti értékeket minterm táblában tárolja az eljárás (5. ábra). A minterm tábla itt egy egyszerű tömb adatszerkezet, amelynek az értékei nem módosulnak az áramkör működtetése folyamán. Ebből az adatszerkezetből az eljárás kiolvass egy logikai függvényt, teljes diszjunktív normál alakban, amit a minterm tábla megjelenítésével együtt a felhasználó rendelkezésére bocsát.

5. ábra

A fejlesztés későbbi stádiumai

Mint ahogyan korábban megemlítettük, a szoftver fejlesztése az első stádiumban tart. Egyelőre csak alapvető alkatrészeket tud a felhasználó alkalmazni, mint például a két és három bemenetű NAND és inverter IC-k. A program működésében rejtőző hibákat a tesztelés folyamán feltárjuk, megkeressük a kifejtő okokat, és korrigáljuk. Mindemellett próbáljuk a program-magot és a grafikus motort tökéletesíteni és újabb, bonyolultabb műveletek elvégzéséhez alkalmassá tenni, a gépi erőforrásokkal való takarékoságot is figyelembe véve. Bízunk benne, hogy a végső stádiumokra, egy megfelelően működő és oktatási célokra alkalmas szoftvert állíthatunk az érdeklődők rendelkezésére.

Irodalomjegyzék

- [1] Dr. Madarász László - Praktikum a digitális áramkörökhöz
KF-GAMF Kari jegyzet, H-319, Kecskemét, 2001, 20. oldal
- [2] Thomas H. Cormen, Charles E. Leiserson, Roland L. Rivest – Algoritmusok
Műszaki Könyvkiadó, Budapest, 2003 (4. kiadás). 564. oldal, 29.1. fejezet
- [3] Thomas H. Cormen, Charles E. Leiserson, Roland L. Rivest – Algoritmusok
Műszaki Könyvkiadó, Budapest, 2003 (4. kiadás). 568. oldal, 29.1.5. fejezet

Pótári Tamás, Műszaki Informatikus hallgató

Kecskeméti Főiskola Gépipari és Automatizálási Műszaki Főiskolai Kar

6000 Kecskemét, Izsáki út 10., Pf. 91.

Telefon: 06/30-677-45-61, e-mail: trinko@freemail.hu