

7. Prototípus koncepció

54 – *Override*

Konzulens:

dr. László Zoltán

Csapattagok:

Kriván Bálint	CBVOEN	balint@krivan.hu
Jákli Gábor	ONZ5G1	j_gab666@hotmail.com
Dévényi Attila	L1YRH0	devenyat@gmail.com
Apagyi Gábor	X8SG3T	apagyi.gabooo@gmail.com
Péter Tamás Pál	N5ZLEG	falconsaglevlist@gmail.com

2011. március 27.

Tartalomjegyzék

7	Prototípus koncepciója	4
7.1.	Prototípus interface-definíciója	4
7.1.1.	Az interfész általános leírása	4
7.1.2.	Bemeneti nyelv	4
7.1.3.	Kimeneti nyelv	8
7.2.	Összes részletes use-case	9
7.3.	Tesztelési terv	10
7.4.	Tesztelést támogató segéd- és fordítóprogramok specifikálása	12
7.5.	Napló	12

Ábrák jegyzéke

7. Prototípus koncepciója

7.1. Prototípus interface-definíciója

[Definiálni kell a teszteket leíró nyelvet. Külön figyelmet kell fordítani arra, hogy ha a rendszer véletlen elemeket is tartalmaz, akkor a véletlenszerűség ki-bekapcsolható legyen, és a program determinisztikusan is tesztelhető legyen.]

7.1.1. Az interfész általános leírása

A prototípus szabványos ki- és bemeneten kommunikál a felhasználóval. Az elkészített prototípus program egy saját parancsrendszert használ. A parancs kiadása után a program végrehajtja azt és kiírja az eredményt a kimenetre. Az automatikus tesztelés elősegítése érdekében lehetőség van arra, hogy a parancsokat egy előre elkészített fájlból olvassa és a kimenetet fájlba mentse. A program az áramkört is fájlból olvassa. A tesztelés elősegítése érdekében elkészítettünk néhány áramkört, azonban a felhasználó a megadott áramkört leíró fájl specifikációja alapján saját áramkört is készíthet, majd tesztelhet.

7.1.2. Bemeneti nyelv

7.1.2.1. Felhasználói parancsok

A parancsokat a standard bemenetről, illetve fájlból olvassa be a program. Minden parancsot egy sorvége karakter zár le.

Megjegyzés: minden parancs ad visszajelzést a felhasználónak a végrehajtott eseményről, ennek formátuma a Kimeneti nyelv c. fejezetben olvasható.

loadCircuit <file>

- Leírás: A megadott áramkört betölti a szimulációs program.
- Megjegyzés: A file nevét kiterjesztés nélkül kell megadni.
- Opciók: -

loadSettings <file>

- Leírás: A jelenlegi áramkörhöz a megadott konfigurációs fájl betöltése.
- Megjegyzés: A file nevét kiterjesztés nélkül kell megadni.
- Opciók: -

saveSettings <file>

- Leírás: A pillanatnyilag használt konfiguráció fájlba mentése.
- Megjegyzés: A file nevét kiterjesztés nélkül kell megadni.
- Opciók: -

switch <név>

- Leírás: A megnevezett kapcsoló átállítása.
- Megjegyzés: -
- Opciók: -

setSeqGen <név> <érték1, érték2, ...>

- Leírás: A megnevezett szekvenciagenerátor az értékparaméterek szerint beállítódik.
- Megjegyzés: -
- Opciók: -

getValue <név> | -all

- Leírás: A megadott áramköri elem kiírja az aktuális kimeneti értékét.
- Megjegyzés: -
- Opciók: a *getValue* -all parancs az összes áramköri elem kimenetének értékét kilistázza.

step

- Leírás: A parancs hatására lefut egy szimulációs ciklus, melynek két eredménye lehet:
 - véges lépésen belül stabilizálódik a rendszer, ekkor a kapcsoló(k), szekvenciagenerátor(ok) és ki-jelző(k) értéke(i) kiíródnak.
 - nem stabilizálódik az áramkör; hibaüzenet
- Megjegyzés: -
- Opciók: -

7.1.2.2. Áramkör leíró fájlok nyelvtana

A konfigurációs fájlok *.ovr kiterjesztésűek, ezekben adjuk meg a szimulálandó hálózat paramétereit. Egyszerű szövegfájl, melyben az értelmezendő parancsok soronként tagolódnak. A program feltételezi a konfigurációs fájl hibamentességét, sehol nem ellenőrizzük, hogy a bemenetnek van-e értelme! A fájl létrehozásához az alábbi parancsok, szintaxisok állnak rendelkezésre:

- X=...
 - Leírás: komponens létrehozás, illetve elnevezése (a név lesz az egyenlőség bal oldalán szereplő név). Ezzel a paranccsal az egyenlőség jel után megadott komponens hozzuk létre, melynek kimenetére ezek után az "X"-el hivatkozhatunk. Amennyiben több kimenetű komponensről beszélünk akkor az egyenlőség bal oldala egy tömböt jelent. Ebben az esetben az egyes kimenetekre a későbbiekben X[i]-vel hivatkozhatunk (a 0 és N-1 között).
 - Opciók: A lehetséges komponensek az implementált komponensek listájából választható, melyeknek paraméterül az egyes komponensekhez tartozó meghatározott paramétereket át kell adni.
 - * OR(...)
 - * AND(...)
 - * INVERTER(...)
 - * VCC(...)
 - * GND(...)
 - * MPX(...)
 - * FFJK(...)
 - * FFD(...)
 - * LED(...)
 - * 7SEG(...)
 - * TOGGLE(...)

- * SEQ(...)
- * NODE(...)
- OR(in[n])
 - Leírás: vagy kapu létrehozása.
 - Opciók:
 - * in[n]: fel kell sorolni a kapu bemenetére kötött komponensek "neveit". N bemenetű kapu esetén ide N db név. A kapu számot nem kell megadni, azt a parser automatikusan észleli a megkapott paraméterek számából. Minimum 2 bemenetet meg kell adni.
 - Példa: OR(vagy1,kapcs1,kapcs2,kapcs3) - három komponens rákapcsolása a kapura, mely így egy három bemenetes vagy kapu lesz.
- AND(in[n])
 - Leírás: "és" kapu létrehozása.
 - Opciók:
 - * in[n]: u.a. mint a vagy kapu esetén.
 - Példa: AND(es1,kapcs1,kapcs2) - két komponens rákapcsolása a kapura, mely így egy két bemenetes vagy kapu lesz.
- INV(in)
 - Leírás: inverter létrehozása.
 - Opciók:
 - * in: az inverter bemenetét kell megadni. Csak egy bemenetű inverter létezik.
 - Példa: INV(inv1,kapcs1) - egy kapcsoló rákapcsolása a kapura.
- VCC()
 - Leírás: konstans igaz jel létrehozása.
 - Opciók:
 - Példa: VCC()
- GND()
 - Leírás: konstans hamis jel létrehozása.
 - Opciók:
 - Példa: GND()
- MPX(in[4],S[2])
 - Leírás: 4:1 multiplexer létrehozása.
 - Opciók:
 - * in[4]: meg kell adni a négy bemenetet.
 - * S[2]: meg kell adni a két select bemenetet
 - Példa: MPX(mpx1,kapcs1,kapcs2,kapcs3,kapcs4,or1,and1)
- FFJK(clk,J,K)
 - Leírás: J-K flip-flop létrehozása
 - Opciók:
 - * clk: meg kell adni az órajel bemenetet

- * J: meg kell adni a J bemenetet
- * K: meg kell adni a K bemenetet
- Példa: FFJK(seqgen1,and1,and2)
- FFD(clk,D)
 - Leírás: D flip-flop létrehozása.
 - Opciók:
 - * clk: meg kell adni az órajel bemenetet.
 - * D: meg kell adni a D bemenetet
 - Példa: FFD(seqgen1,or1)
- LED(in)
 - Leírás: LED létrehozása.
 - Opciók:
 - * in: meg kell adni a LED bemenetét
 - Példa: LED(kapcs1)
- 7SEG(D[8])
 - Leírás: 7 szegmenses kijelző létrehozása.
 - Opciók:
 - * D[8]: meg kell adni a nyolc vezérlő bemenetet
 - Példa: 7SEG(7seg1,dek[7],dek[6],dek[5],dek[4],dek[3],dek[2],dek[1],dek[0])
- TOGGLE()
 - Leírás: kapcsoló létrehozása.
 - Opciók:
 - Példa: TOGGLE()
- SEQ(BITMINTA)
 - Leírás: szekvencia generátor létrehozása.
 - Opciók:
 - * BITMINTA: meg kell adni egy sorozatot, melyet a generátor egymás után kiad magából.
 - Példa: SEQ(011000110)
- NODE(in,n)
 - Leírás: csomópont létrehozása.
 - Opciók:
 - * in: meg kell adni a csomópont bemenetét
 - * n: meg kell adni, hány kimenet lesz
 - Példa: NODE(kapcs1,3) - három kimenetű elosztót hoz létre, melynek bemenete a kapcs1
- Példa áramkör konfigurációs fájl Egy olyan minta hálózatot hozunk létre melyben található két kapcsoló egy és kapura kötve és az és kapu kimenete egy inverteren keresztül egy ledre kapcsolódik.
 - t1=TOGGLE()
 - t2=TOGGLE()
 - es1=AND(t1,t2)
 - inv1=INV(es1)
 - led1=LED(inv1)

7.1.2.3. Konfigurációs fájl nyelvtana

A konfigurációs fájlban minden sorban egybeállításnak kell szerepelnie, mely a következő egységekből áll:

- az elem neve
- egyenlőségjel
- az elem értéke (szekvencia generátor esetében több érték is lehet, ezeket vesszővel elválasztva kell megadni)

példa:

```
toggle1=0  
seqGen1=0,1,1,0,1
```

7.1.3. Kimeneti nyelv

A program történései, visszajelzése a standard kimeneten jelennek meg, illetve ezek fájlba is kiíródnak. A program minden parancs után visszajelzést ad a felhasználónak a végrehajtott eseményről. A fentebb definiált parancsokra a következő jelzéseket kapja a felhasználó:

loadCircuit <file>

Lehetséges kimenetek

- `load successful`
 - Leírás: a betöltés sikeres, amennyiben az áramkört tartalmazó fájl szintaktikája megfelel a Áramkör leíró fájlok nyelvtana c. fejezetnek.
- `load failed`
 - Leírás: a betöltés sikertelen, amennyiben az áramkört tartalmazó fájl szintaktikája nem felel meg a Áramkör leíró fájlok nyelvtana c. fejezetnek.

loadSettings <file>

Lehetséges kimenetek

- `load successful`
 - Leírás: az értékek betöltése sikeres, amennyiben a konfigurációs fájlban szereplő áramköri elemek megfeleltethetők az aktuális áramkörben szereplő elemekkel, illetve a megadott értékek helyesek.
 - Megjegyzés: azon elemek, melyek beállítására nem volt információ a konfigurációs fájlban automatikusan nullázódnak.
- `load failed`
 - Leírás: az értékek betöltése sikeres, amennyiben a konfigurációs fájlban szereplő áramköri elem nem feleltethető meg az aktuális áramkörben szereplő elemek egyikével sem, illetve ha valamelyik érték helytelen.

saveSettings <file>

Lehetséges kimenetek

- `save successful`
 - Leírás: a konfigurációs értékek sikeresen fájlba mentődtek.

switch <név>

Lehetséges kimenetek

2011. március 27.

- [elem]: [érték]
 - Leírás: az [elem] megadja a módosított kapcsoló nevét, míg az [érték] megmutatja, hogy milyen értékre változott az aktuális kapcsoló kimenete.

setSeqGen <név> <érték1, érték2, ...>

Lehetséges kimenetek

- [elem]: [érték1, érték2, ...]
 - Leírás: az [elem] megadja a módosított generátor nevét, míg az [érték1, érték2, ...] megmutatja, hogy milyen értékekre változott az aktuális generátor kimenete.

getValue <név> | -all

Lehetséges kimenetek

- [elem]: [érték]
 - Leírás: az [elem] megadja a módosított kapcsoló nevét, míg az [érték] megmutatja, hogy milyen értékre változott az aktuális kapcsoló kimenete.
 - Megjegyzés: a getValue -all parancsra az összes elemet kilistázza a megadott formában új sor karakterrel elválasztva

step

Lehetséges kimenetek

- simulation successful
 - [elem1]: [érték]
 - [elem2]: [érték]
 - ...
 - Leírás: a szimuláció sikeres, amennyiben véges lépésen belül stabilizálódni tud az áramkör. Ekkor a kapcsoló(k), szekvenciagenerátor(ok) és kijelző(k) értéke(i) kiíródnak.
- simulation failed
 - Leírás: a szimuláció sikertelen, amennyiben véges lépésen belül nem tud stabilizálódni az áramkör.

7.2. Összes részletes use-case

Use-case neve	Áramkör betöltése
Rövid leírás	Az áramkört leíró fájl betöltése
Aktorok	Felhasználó
Forgatókönyv	A loadCircuit parancsot használva betöltheti az áramkört leíró fájlt, amely a program követelményeinek megfelel

Use-case neve	Konfiguráció betöltése
Rövid leírás	Egy áramkör konfigurációjának betöltése
Aktorok	Felhasználó
Forgatókönyv	A loadSettings paranccsal betölt egy egyedi a konfigurációt az áramkörhöz, amely például tartalmazhatja a szekvencia generátorok által kiadott bitsorozatot vagy a kapcsolók állását.

Use-case neve	Konfiguráció mentése
---------------	----------------------

Rövid leírás	Áramkör konfigurációjának mentése
Aktorok	Felhasználó
Forgatókönyv	A saveSettings parancs kiadásával menti az aktuális áramkör konfigurációját.

Use-case neve	Kapcsoló kapcsolása
Rövid leírás	Kapcsoló állásnak módosítás
Aktorok	Felhasználó
Forgatókönyv	Az adott áramkörben a neve alapján azonosított kapcsoló állásának módosítása a switch parancs használatával.

Use-case neve	Szekvenciagenerátor módosítás
Rövid leírás	Szekvenciagenerátor bitsorozatának megadása
Aktorok	Felhasználó
Forgatókönyv	Az adott áramkörben a neve alapján azonosított szekvenciagenerátor által kiadott bit-sorozat megadása a setSeqGen paranccsal.

Use-case neve	Érték lekérdezése
Rövid leírás	Egy, az áramkörben lévő alkatrész értékének lekérdezése
Aktorok	Felhasználó
Forgatókönyv	Az adott áramkörben a getValue parancs használatával a megadott nevű alkatrész értékének lekérdezése.

Use-case neve	Áramkör szimulálása
Rövid leírás	A betöltött áramkör szimulálása
Aktorok	Felhasználó
Forgatókönyv	A step parancs kiadásával szimulálja a betöltött áramkört.

Use-case neve	Teszt eredményének ellenőrzése
Rövid leírás	A program által generált kimenetet összehasonlítja a referencia kimenettel
Aktorok	Felhasználó
Forgatókönyv	A teszt lefutását követően egy script összehasonlítja a kapott eredményeket a várt eredményekkel.

7.3. Tesztelési terv

A tesztelés lehetőséget nyújt a program funkcióinak és menetének széleskörű vizsgálatára. Tesztelés során lehetőség nyílik a különböző tesztesetek kipróbálására. A teszt bemenetét bemeneti állományokból kapja, és a teszt eredményét kimeneti fájlban, és konzolon jeleníti meg. Ezáltal lehet összevetni a kiválasztott teszt várt, és tényleges eredményét.

Teszt-eset neve	Alap áramkör
Rövid leírás	Alap áramkör, mely kapcsolókból, és egyszerű nem visszacsatolt ÉS kapukból áll, kimeneti értékeket leddek jelzik
Teszt célja	Ez a teszteset leteszteli a kapcsolók, ÉS kapu, és a leddek működését.

Teszt-eset neve	MPX-es áramkör
Rövid leírás	Olyan áramkör, mely kapcsolókból és MPX-ből áll, a kimeneti értékeket 7 szegmenses kijelző jelzi.
Teszt célja	Ez a tesztet leteszteli a MPX és a 7szegmenses kijelző működését

Teszt-eset neve	Visszacsatolt stabil áramkör
Rövid leírás	Olyan áramkör, melyben egy visszacsatolás történik, de az áramkör stabil marad, tehát egy vagy kapu visszakötve, kimenet értékét egy led jelzi.
Teszt célja	Ez a tesztet első sorban a visszacsatolás helyes működését teszteli le, de mivel visszacsatolás is történik és egyúttal a kimeneti értéket is megjelenítjük leden, ezért csomópont is kell tehát csomópont helyes működését is teszteli

Teszt-eset neve	Visszacsatolt nem stabil áramkör
Rövid leírás	Előző áramkörhöz hasonló áramkör, csak itt úgy végezzük a visszacsatolást, hogy az áramkör ne legyen stabil: egy vagy kapu visszakötve egy inverteren keresztül
Teszt célja	Ez a tesztet az invertert és a VAGY kaput továbbá a visszacsatolást teszteli le olyan esetben, mikor az áramkör nem lesz soha stabil.

Teszt-eset neve	Flip-flop-os áramkör
Rövid leírás	Olyan áramkör melyben szerepel egy flipflop, egy jelgenerátor, és az értéket leden jelezzük ki.
Teszt célja	E tesztet során letesztelhetjük a jelgenerátor helyes működését és, hogy a flip-flop helyesen lép e felfutó élre, illetve helyesen működik e.

Teszt-eset neve	Kompozitos áramkör
Rövid leírás	Egy olyan áramkör, melyben szerepel egy kompozit elem, azaz szerepel egy áramköri hálózat egy egységben mint áramköri elem. Kompoziton belül valamilyen egyszerű áramköri hálózat szerepel.
Teszt célja	Ez a tesztet az új kompozit elem működését teszteli le, annak helyességét ellenőrizhetjük.

Teszt-eset neve	Kompoziton belüli kompozitos áramkör
Rövid leírás	Előző áramkörhöz hasonló áramkör, a különbség az, hogy a kompoziton belüli áramköri hálózat tartalmaz egy újabb kompozitot a többi elemen kívül.
Teszt célja	Ez az áramkör leteszteli, hogyan működik a program olyan esetben, mikor a kompozit további kompozitot tartalmaz, illetve a kompoziton belüli kompozit jól működik e más elemekkel összekötve.

7.4. Tesztelést támogató segéd- és fordítóprogramok specifikálása

A program által generált kimeneti fájl és az elvárt eredményeket tartalmazó fájlok összehasonlítására a DiffUtils-ban (<http://www.gnu.org/software/diffutils/>) található cmp.exe-t fogjuk használni.

7.5. Napló

Kezdet	Időtartam	Résztvevők	Leírás
2011.03.22. 12:00	2,5 óra	Apagyi G.	Prototípus áramkör leíró nyelvének definiálása
2011.03.22. 14:00	1,5 óra	Kriván B. Jákli G. Dévényi A.	Értekezlet: Specifikáció módosítása miatt szükségeszerű változtatások megbeszélése
2011.03.22. 20:00	1 óra	Jákli Gábor	Összes részletes use-case
2011.03.22. 22:00	1 óra	Dévényi A.	Felhasználói parancsok
2011.03.23. 14:00	1 óra	Dévényi A.	Konfigurációs fájl nyelvtana, kimeneti nyelv
2011.03.23. 15:00	45 perc	Jákli G.	Új use case, 7.1.1 és 7.4
2011.03.26. 16:00	1,5 óra	Péter T.	Tesztelési terv és tesztesetek