

## 8. Részletes tervek

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. április 4.

# Tartalomjegyzék

|          |  |          |
|----------|--|----------|
| <b>8</b> | <b>Részletes tervek</b>  | <b>4</b> |
| 8.1.     | Osztályok és metódusok tervei . . . . .                        | 4        |
| 8.1.1.   | Osztály1 . . . . .   | 4        |
| 8.1.2.   | Osztály2 . . . . .   | 4        |
| 8.2.     | A tesztek részletes tervei, leírásuk a teszt nyelvén . . . . . | 5        |
| 8.2.1.   | Alap áramkör . . . . .   | 5        |
| 8.2.2.   | MPX-es áramkör . . . . .                                       | 6        |
| 8.2.3.   | Visszacsatolt stabil áramkör . . . . .                         | 8        |
| 8.2.4.   | Visszacsatolt nem stabil áramkör . . . . .                     | 9        |
| 8.2.5.   | Flip-flop-os áramkör . . . . .                                 | 10       |
| 8.2.6.   | Kompozitos áramkör . . . . .                                   | 11       |
| 8.2.7.   | Kompoziton belüli kompozitos áramkör . . . . .                 | 14       |
| 8.3.     | A tesztelést támogató programok tervei . . . . .               | 15       |
| 8.4.     | Napló . . . . .  | 15       |

## **Ábrák jegyzéke**

## 8. Részletes tervek

### 8.1. Osztályok és metódusok tervei

#### 8.1.1. Osztály1

- Felelősség  
*[Mi az osztály felelőssége. Kb 1 bekezdés. Ha szükséges, akkor state-chart is.]*
- Ősosztályok  
*[Mely osztályokból származik (öröklési hierarchia)  
Legősebb osztály → Ősosztály2 → Ősosztály3...]*
- Interfészek  
*[Mely interfészeket valósítja meg.]*
- Attribútumok  
*[Milyen attribútumai vannak]*
  - attribútum1: attribútum jellemzése: mire való, láthatósága (UML jelöléssel), típusa
  - attribútum2: attribútum jellemzése: mire való, láthatósága (UML jelöléssel), típusa
- Metódusok  
*[Milyen publikus, protected és privát metódusokkal rendelkezik. Metódusonként precíz leírás, ha szükséges, activity diagram is a metódusban megvalósítandó algoritmusról.]*
  - int foo(Osztály3 o1, Osztály4 o2): metódus leírása, láthatósága (UML jelöléssel)
  - int bar(Osztály5 o1): metódus leírása, láthatósága (UML jelöléssel)

#### 8.1.2. Osztály2

- Felelősség  
*[Mi az osztály felelőssége. Kb 1 bekezdés. Ha szükséges, akkor state-chart is.]*
- Ősosztályok  
*[Mely osztályokból származik (öröklési hierarchia)  
Legősebb osztály → Ősosztály2 → Ősosztály3...]*
- Interfészek  
*[Mely interfészeket valósítja meg.]*
- Attribútumok  
*[Milyen attribútumai vannak]*
  - attribútum1: attribútum jellemzése: mire való, láthatósága (UML jelöléssel), típusa
  - attribútum2: attribútum jellemzése: mire való, láthatósága (UML jelöléssel), típusa
- Metódusok  
*[Milyen publikus, protected és privát metódusokkal rendelkezik. Metódusonként precíz leírás, ha szükséges, activity diagram is a metódusban megvalósítandó algoritmusról.]*
  - int foo(Osztály3 o1, Osztály4 o2): metódus leírása, láthatósága (UML jelöléssel)
  - int bar(Osztály5 o1): metódus leírása, láthatósága (UML jelöléssel)

## 8.2. A tesztek részletes tervei, leírásuk a teszt nyelvén

*[A tesztek részletes tervei alatt meg kell adni azokat a bemeneti adatsorozatokat, amelyekkel a program működése ellenőrizhető. Minden bemenő adatsorozathoz definiálni kell, hogy az adatsorozat végrehajtásától a program mely részeinek, funkcióinak ellenőrzését várjuk és konkrétan milyen eredményekre számítunk, ezek az eredmények hogyan vethetők össze a bemenetekkel.]*

### 8.2.1. Alap áramkör

- Leírás  
Olyan áramkör, melyben 2 kapcsolóval állíthatjuk egy ÉS kapu bemeneteit, melyet egy LED jelenít meg.
- Ellenőrzött funkcionalitás, várható hibahelyek  
Ellenőrizzük a kapcsoló helyes váltását, az ÉS kapu kimenetének helyes kiszámítását és a LED működését
- Áramkör létrehozása

```
kapcs1=TOGGLE()  
kapcs2=TOGGLE()  
es=AND(kapcs1,kapcs2)  
led=LED(es)
```

- Bemenet és kimenet

| <i>Bemenet</i>  | <i>Kimenet</i>  |
|---|---|
| loadCircuit test1.ovr<br>step<br>switch kapcs1<br>step<br>check -all<br>switch kapcs2<br>step | simulation successful<br>kapcs1: 0<br>kapcs2: 0<br>led: 0<br><br>kapcs1: 1<br><br>simulation successful<br>kapcs1: 1<br>kapcs2: 0<br>led: 0<br><br>led:<br>in: 0<br>out:<br>kapcs1:<br>in:<br>out: 1<br>kapcs2:<br>in:<br>out: 0<br>es:<br>in: 1, 0<br>out: 0<br><br>kapcs2: 1<br><br>simulation successful<br>kapcs1: 1<br>kapcs2: 1<br>led: 1 |

### 8.2.2. MPX-es áramkör

- **Leírás**  
Olyan áramkört hozunk létre, melyben egy 7 szegmenses kijelzőt hajtunk meg kapcsolókkal és egy MPX-xel. A 7szegmenses kijelző [2]-[7] bemeneteire kapcsolókat kötünk, a [1] bemenetét egy MPX adja, mely 4 kapcsolóból választja ki az egyiket, tehát egy 4/1-es MPX.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
Ellenőrizzük a MPX helyes működését, és a 7 szegmenses kijelzőt. Hiba a MPX kiválasztása során történhet, hogy rossz jelet juttat a kimenetére.
- **Áramkör létrehozása**

```
inmpx1=TOGGLE()
inmpx2=TOGGLE()
inmpx3=TOGGLE()
inmpx4=TOGGLE()
selmpx1=TOGGLE()
```

```
selmpx2=TOGGLE()  
mux=MPX(inmpx1,inmpx2,inmpx3,inmpx4,selmpx1,selmpx2)  
seg=TOGGLE()  
display=7SEG(mux,seg,0,0,0,0,0)
```

- Bemenet és kimenet

| <i>Bemenet</i>        | <i>Kimenet</i>               |
|-----------------------|------------------------------|
| loadCircuit test2.ovr | load successful              |
| switch inmpx1         | inmpx1: 1                    |
| switch inmpx3         | inmpx3: 1                    |
| step                  | simulation successful        |
| switch selmpx2        | inmpx1: 1                    |
| switch seg2           | inmpx2: 0                    |
| step                  | inmpx3: 1                    |
| switch selmpx2        | inmpx4: 0                    |
| switch selmpx1        | selmpx1: 0                   |
| step                  | selmpx2: 0                   |
|                       | seg: 0                       |
|                       | display: 1, 0, 0, 0, 0, 0, 0 |
|                       | selmpx2: 1                   |
|                       | seg: 1                       |
|                       | simulation successful        |
|                       | inmpx1: 1                    |
|                       | inmpx2: 0                    |
|                       | inmpx3: 1                    |
|                       | inmpx4: 0                    |
|                       | selmpx1: 0                   |
|                       | selmpx2: 1                   |
|                       | seg: 1                       |
|                       | display: 1, 1, 0, 0, 0, 0, 0 |
|                       | selmpx2: 0                   |
|                       | selmpx1: 1                   |
|                       | simulation successful        |
|                       | inmpx1: 1                    |
|                       | inmpx2: 0                    |
|                       | inmpx3: 1                    |
|                       | inmpx4: 0                    |
|                       | selmpx1: 1                   |
|                       | selmpx2: 0                   |
|                       | seg: 1                       |
|                       | display: 0, 1, 0, 0, 0, 0, 0 |

### 8.2.3. Visszacsatolt stabil áramkör

- Leírás

Egy olyan áramkört hozunk létre, melyben egy VAGY kapu szerepel, aminek egyik bemenete egy kapcsoló, kimenetét pedig visszakötjük a második bemenetére, illetve egy csomóponton keresztül egy LED-



re is eljuttatjuk.

- Ellenőrzött funkcionalitás, várható hibahelyek

Ellenőrizzük, hogy az áramkör helyesen stabilnak érzékeli-e a kapcsolást, illetve a VAGY kapu helyes működését is ellenőrizzük. Hibát a visszakötés okozhat.

- Áramkör létrehozása

```
kapcs=TOGGLE()
vagy=OR(kapcs,node[2])
node=NODE(vagy,2)
led=LED(node[1])
```

- Bemenet és kimenet

| Bemenet   | Kimenet   |
|---|---|
| loadCircuit test3.ovr<br>step<br>switch kapcs<br>step | load successful<br><br>simulation successful<br>kapcs: 0<br>led: 0<br><br>kapcs: 1<br><br>simulation successful<br>kapcs: 1<br>led: 1 |

#### 8.2.4. Visszacsatolt nem stabil áramkör

- Leírás

Egy olyan áramkört hozunk létre, melyben egy ÉS kapu szerepel, aminek egyik bemenete egy kapcsoló, kimenetét pedig visszakötjük egy inverteren keresztül a második bemenetére, illetve egy csomóponton keresztül egy LED-re is eljuttatjuk.

- Ellenőrzött funkcionalitás, várható hibahelyek

Ellenőrizzük, hogy az áramkör helyesen instabilnak érzékeli-e a kapcsolást. Hibás működést ez okozhat, tehát ha az áramkör ezt rosszul állapítja meg, és nem jelzi.

- Áramkör létrehozása

```
kapcs=TOGGLE()
inv=INV(node[2])
es=AND(kapcs,inv)
node=NODE(es,2)
led=LED(node[1])
```

- Bemenet és kimenet

| <i>Bemenet</i>                                | <i>Kimenet</i>   |
|---|--|
| loadCircuit test4.ovr<br>switch kapcs<br>step | load successful<br><br>kapcs: 1<br><br>simulation failed |

### 8.2.5. Flip-flop-os áramkör

- **Leírás**  
Egy olyan áramkört hozunk létre, melyben egy JK flipflop szerepel, J és K bemenetére kapcsolókat kötünk, órajelét egy jelgenerátorból kapja, és a kimenetét egy oszcilloszkóp kapja meg.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
Ellenőrizzük a jelgenerátort, hogy megfelelő jelet adja-e ciklikusan, ellenőrizzük a JK flipflop működését, illetve, hogy megfelelően lép-e az órajelre, továbbá ellenőrizzük, hogy az oszcilloszkóp helyesen működik-e. Hiba lehetséges a jelgenerátor működésében, a JK flipflop működésében illetve számolásában, és az oszcilloszkóp működésében.
- **Áramkör létrehozása**

```
j=TOGGLE()
k=TOGGLE()
seqgen=SEQGEN()
jk=FFJK(seqgen, j, k)
scope=SCOPE(jk, 3)
```

- **Bemenet és kimenet**

| <i>Bemenet</i>        | <i>Kimenet</i>        |
|-----------------------|-----------------------|
| loadCircuit test5.ovr | load successful       |
| switch k              |                       |
| step                  | k: 1                  |
| step                  |                       |
| switch j              | simulation successful |
| step                  | j: 0                  |
| step                  | k: 1                  |
| switch j              | seqgen: 0             |
| switch k              | scope: 0              |
| step                  |                       |
| step                  | simulation successful |
|                       | j: 0                  |
|                       | k: 1                  |
|                       | seqgen: 1             |
|                       | scope: 00             |
|                       |                       |
|                       | j: 1                  |
|                       |                       |
|                       | simulation successful |
|                       | j: 1                  |
|                       | k: 1                  |
|                       | seqgen: 0             |
|                       | scope: 000            |
|                       |                       |
|                       | simulation successful |
|                       | j: 1                  |
|                       | k: 1                  |
|                       | seqgen: 1             |
|                       | scope: 001            |
|                       |                       |
|                       | j: 0                  |
|                       |                       |
|                       | k: 0                  |
|                       |                       |
|                       | simulation successful |
|                       | j: 0                  |
|                       | k: 0                  |
|                       | seqgen: 0             |
|                       | scope: 011            |
|                       |                       |
|                       | simulation successful |
|                       | j: 0                  |
|                       | k: 0                  |
|                       | seqgen: 1             |
|                       | scope: 111            |

#### 8.2.6. Kompozitos áramkör

- Leírás

Egy olyan áramkört valósítunk meg, melyben egy kompozit szerepel. Ez a kompozit egy 2 bites balról  
2011. április 4.

tölthető shiftregisztert valósít meg. A kompozitnak két bemenete van egy kapcsoló ami a balról bejövő értéket adja, és egy jelgenerátor, amely az órajelet. Belül 2 db D flipflop található összekötve. Az első flipflop kimenetét kiadja a kompozit kimenetén is, és a második flipflop bemenetére is ráadja, ezért NODE is kell. A kompozit kimenete a 2 bit és a carry.

- Ellenőrzött funkcionalitás, várható hibahelyek  
Kompozit helyes működését ellenőrizzük.
- Áramkör létrehozása

```
input=TOGGLE()  
seqgen=SEQGEN()  
composite SHR(clk, in){  
    in2 = NODE(in, 1)  
    d1 = FFD(clk, in)  
    node1 = NODE(d1, 2)  
    d2 = FFD(clk, node1[1])  
} (in2, node1[2], d2)  
myshr = SHR(seqgen, input)  
led1=LED(myshr[1])  
led2=LED(myshr[2])  
ledcarry=LED(myshr[3])
```

- Bemenet és kimenet

| <i>Bemenet</i>  | <i>Kimenet</i>  |
|---|---|
| loadCircuit test6.ovr<br>switch input<br>step<br>step<br>switch input<br>step<br>step<br>step<br>step | load successful<br><br>input: 1<br><br>simulation successful<br>input: 1<br>seqgen: 0<br>led1: 1<br>led2: 0<br>ledcarry: 0<br><br>simulation successful<br>input: 1<br>seqgen: 1<br>led1: 1<br>led2: 0<br>ledcarry: 0<br><br>input: 0<br><br>simulation successful<br>input: 0<br>seqgen: 0<br>led1: 0<br>led2: 1<br>ledcarry: 0<br><br>simulation successful<br>input: 0<br>seqgen: 1<br>led1: 0<br>led2: 1<br>ledcarry: 0<br><br>simulation successful<br>input: 0<br>seqgen: 0<br>led1: 0<br>led2: 0<br>ledcarry: 1<br><br>simulation successful<br>input: 0<br>seqgen: 1<br>led1: 0<br>led2: 0<br>ledcarry: 1 |

## 8.2.7. Kompoziton belüli kompozitos áramkör

- Leírás

Egy olyan áramkört hozunk létre melyben egy kompozit szerepel, ami egy másik kompozitot foglal magába. A belső kompozit egyetlen invertert tartalmaz. A külső kompozit tartalmaz még egy VAGY kaput, melynek egyik bemenetére a belső kompozit kimenetét, másik bemenetére pedig a külső kompozit bemenetére érkező jelet kötjük. A külső kompozit bemenetére egy kapcsolót, kimenetére egy LED-et kötünk.

- Ellenőrzött funkcionalitás, várható hibahelyek

Leteszteljük, hogy működik-e a kompozit elem, ha belül bonyolultabb áramköri hálózat szerepel, jelen esetben egy kompozit, illetve egy VAGY kapu.

- Áramkör létrehozása

```
tog = TOGGLE()
composite innerComp(in){
  inv = INV(in)
}(inv)
composite Main(in){
  inC = innerComp(in)
  or = OR(in, inC)
}(or)
m = Main(tog)
led = LED(m)
```

- Bemenet és kimenet

| <i>Bemenet</i>        | <i>Kimenet</i>        |
|-----------------------|-----------------------|
| loadCircuit test7.ovr | load successful       |
| step                  | simulation successful |
| step                  | tog: 0                |
| switch tog            | led: 1                |
| step                  | simulation successful |
| step                  | tog: 0                |
|                       | led: 1                |
|                       | tog: 1                |
|                       | simulation successful |
|                       | tog: 1                |
|                       | led: 1                |
|                       | simulation successful |
|                       | tog: 1                |
|                       | led: 1                |

**8.3. A tesztelést támogató programok tervei**

*[A tesztadatok előállítására, a tesztek eredményeinek kiértékelésére szolgáló segédprogramok részletes terveit kell elkészíteni.]*

**8.4. Napló**

| Kezdet            | Időtartam | Résztevők | Leírás  |
|-------------------|-----------|-----------|---|
| 2011.04.01. 15:00 | 2,5 óra   | Péter T.  | Tesztesetek megtervezése, leírása, felépítésük megadása a bemeneti nyelvnek megfelelően |
| 2011.04.02. 10:00 | 3 óra     | Apagyi G. | Tesztesetek felhasználói interakciójának, illetve várt kimeneteinek megtervezése.       |
| ...               | ...       | ...       | ...   |