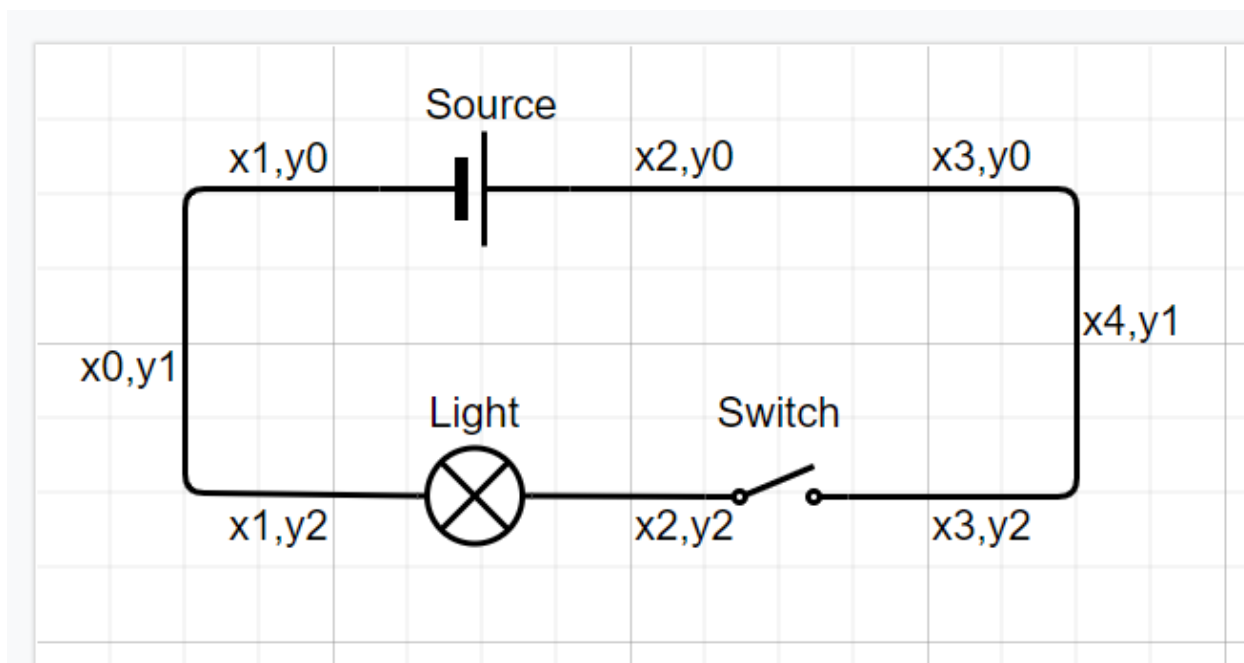


Blokové riešenie s použitím SpiceSharp

- Keďže v SpiceSharp-e sa súčiastky pripájajú jednoducho k uzlom označeným ako “nodes”, ktoré sa označujú premennou typu *String*, pre mriežku, s ktorou budeme pracovať, si môžeme vytvoriť maticu alebo dvojrozmerné pole *String*-ov. V tejto matici budú uložené názvy pre jednotlivé uzly v obvode, kde sa dotýkajú dva bloky na mriežke. Na obrázku sú tieto názvy uzlov príkladovo označené (napríklad “x1,y0”).



Obrázok 1 Jednoduchý elektrický obvod pri blokovom riešení

- Na začiatku sa vytvorí v SpiceSharp-e nový objekt obvodu, do ktorého budú postupne pri akcii vloženia alebo odobratia bloku vkladané alebo odobraté komponenty elektrického obvodu, ktoré dané bloky predstavujú.
- Pri vložení nového bloku na určené miesto v mriežke sa podľa typu el. komponentu, umiestnenia a rotácie bloku zistia uzly pre daný komponent. Ďalej sa už len tento komponent vytvorí s napojením na správne uzly a poskytnutými parametrami komponentu uložených v objekte bloku. Po vytvorení sa komponent pridá do obvodu.

- Bloky predstavujúce križovatku (iba spájajú všetky susedné bloky) vytvoria nový uzol, s ktorým sa pomocou rezistorov pripoja všetky 4 uzly bloku.
- V objekte bloku sa udržuje typ komponentu, jeho parametre a napájacie uzly, ktoré môžu byť kódované *integer* polom, pričom budú použité hodnoty 0 a 1 pre určenie tých strán bloku, na ktorých je napájací uzol. Napríklad pre rezistor by bolo takéto pole (0,1,0,1), pričom prvá hodnota by predstavovala spodnú stranu bloku a nasledujúce hodnoty ďalšie strany v smere hodinových ručičiek. Takéto kódovanie sa môže podľa potreby a implementácie upraviť.

PSEUDOKÓD PRIDANIA SÚČIASTKY DO OBVODU

```
Pridaj_blok_do_obvodu(obvod, blok, zoznam_uzlov_v_mriezke)
```

```
    uzly_komponentu = ziskaj_uzly(blok, zoznam_uzlov_v_mriezke)
```

```
    komponent = blok.vytvorKomponent(uzly_komponentu)
```

```
    obvod.AddItem(komponent)
```