

Robert Meolic

Reverzibilno računalništvo

POVZETEK

Današnji računalniki delujejo tako, da so števila in podatki predstavljeni z dvojiškim sistemom, računanje pa je izvedeno kot zaporedje Boolovih operacij. Ta model je mogoče zelo dobro udejaniti s pomočjo električnih signalov. Ker želimo vedno manjše in hitrejša računalnika pa danes raziskovalci raziskujejo tudi radikalno drugačne tehnologije kot je npr. kvantni računalnik. Pomembna teorija na poti do novih temeljev računalništva je teorija reverzibilnega računalništva. Reverzibilna vezja imajo lastnost, da delujejo v obeh smereh, torej lahko nastavimo vrednosti na izhodni strani, rezultat pa dobimo na vhodni strani. V tem prispevku podamo matematične osnove potrebne za načrtovanje takšnih vezij.

Ključne besede: elektronika, Boolova algebra, logična vrata, digitalno načrtovanje vezij

1. Uvod

Moderna izvedba Boolove algebre je stara dobrih 100 let (Edward V. Huntington je leta 1904 prvi podal ustrezne matematične aksiome). V ozadju te pomembne teorije je ideja, da lahko vsako logično funkcijo podamo z operatorji AND, OR in NOT. To je predlagal George Boole leta 1854. Lani smo praznovali 200 let od rojstva tega genialnega angleškega znanstvenika (glej npr. <http://www.georgeboole.com/>). V poznejši zgodovini je sodelovalo veliko pomembnih raziskovalcev, omenimo samo Henry M. Sheffer-ja, ki je leta 1913 pokazal, da sta operatorja NAND in NOR še bolj priročna kot AND in OR.

Boolova algebra je teoretični model vseh današnjih elektronskih vezij, vse od preprostega daljinca do zapletenega računalnika oz. pametnega telefona. Vzrok je predvsem v tem, da jo lahko zelo dobro udejanjimo s pomočjo električnih signalov. Računalnik, ki je leta 1969 pomagal voditi raketo Apollo na luno in nazaj, je bil zgrajen iz 5600 tri-vhodnih vrat NOR. A izvedba logike z električnimi signali ima svoje meje zmogljivosti, največji problem je toplota, ki se sprošča ob pošiljanju signalov. Nismo več daleč od skrajne meje majhnosti in hitrosti vezij, zato je vedno več raziskav namenjeno radikalno drugačnim tehnologijam. Širši javnosti najbolj znan pojem s tega področja je kvantni računalnik. Pomembna teorija na poti do njega pa je reverzibilno računalništvo.

Reverzibilno računalništvo je splošen pojem, ki obsega področje strojne in programske opreme. V tem prispevku se bomo omejili na matematične osnove delovanja reverzibilnih vezij. Vsako digitalno vezje preslika množico vhodnih vrednosti v množico izhodnih vrednosti. Reverzibilna vezja imajo dodatno lastnost, da delujejo tudi v obratni smeri, torej lahko nastavimo vrednosti na izhodni strani, rezultat pa dobimo na vhodni strani. Povezava med reverzibilnim računalništvom in kvantnim računalnikom je v tem, da kvantni računalnik lahko izvaja le reverzibilne operacije. Če znamo problem razbiti na zaporedje reverzibilnih operacij, je to pomemben korak k izvedbi s kvantnim računalnikom. Seveda bo potrebno rešiti še fizikalni problem izvedbe posameznih reverzibilnih operacij a to je čisto drugačen problem.

Članek je matematično obarvan, poskuša pa biti razumljiv širši skupnosti. Še posebej upamo, da ga bodo opazili v šolah.

2. Nekaj matematike v dvojiškem sistemu

V nadaljevanju najprej ponovimo, kako deluje seštevanje v dvojiškem sistemu. Na sliki 1 je prikazan primer računa. Seštevanje izvedemo korak po korak z desne na levo, popolnoma enako kot smo se naučili v desetiškem sistemu. Razlika je le v tem, da imamo na voljo le številki 0 in 1. Računamo takole: $0 + 0 = 0$, $0 + 1 = 1$, $1 + 0 = 1$, paziti pa moramo pri $1 + 1$, rezultat je enak 0 hkrati pa 1 štejemo dalje. Računalniški gradnik, ki izvede seštevanje dveh števk (1 številko v dvojiškem sistemu lahko imenujemo tudi 1 bit) se imenuje enobitni popolni seštevalnik (angl. one-bit full adder) in kot vidimo na sliki 1 ima tri vhode (poleg obeh števk, ki ju seštevamo še prenos od prejšnjega seštevanja) ter dva izhoda (poleg rezultata seštevanja še prenos naprej).

The image shows a handwritten binary addition on a grid background. The numbers 54 and 45 are converted to binary: 54 is 00110110 and 45 is 00101101. The addition is performed from right to left, with carry bits (1s) shown below the lines. The result is 01100011, which is equal to 99 in decimal. Below the calculation is a hand-drawn logic symbol for a full adder, represented as a box with a plus sign inside. It has three inputs on the left: X, Y, and C-IN. It has two outputs on the right: S (Sum) and C-OUT.

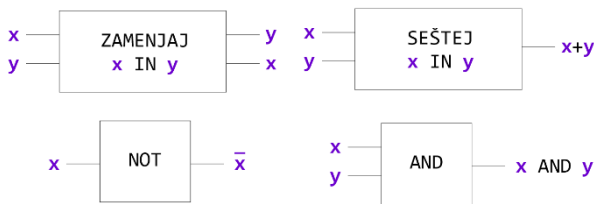
Slika1: Seštevanje v dvojiškem sistemu in simbol za enobitni popolni seštevalnik.

Ko izdelamo enobitni popolni seštevalnik lahko seštevanje poljubno velikih števil izvedemo z zaporedno vezavo enakih gradnikov (resnici na ljubo današnji računalniki uporabljajo nekoliko kompleksnejšo izvedbo, ki omogoča hitrejša računanje, za več informacij uporabite ključno besedo »carry-lookahead adder«).

3. Osnovni gradniki reverzibilnih vezij

Digitalno vezje preslika množico vhodov v množico izhodov. Preprosti primeri digitalnih vezij so prikazani na sliki 2. Hitro lahko opazimo, da sta operaciji ZAMENJAJ in NOT

sami po sebi reverzibilni, za operaciji SEŠTEJ in AND pa ni takoj jasno, kako bi ju izvedli v obratni smeri.



Slika 2: Primeri digitalnih vezij

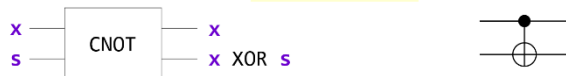
Postopek, kako poljubno operacijo naredimo reverzibilno, se imenuje »vgradnja funkcije«. Ideja je preprosta in temelji na dodajanju dodatnih vhodov in izhodov. Na primer, seštevanje 2 števil pretvorimo v prištevanje enega števila k drugemu ter na izhod pošljemo dovolj informacij, da je mogoče postopek izvesti tudi v obratni smeri (glej sliko 3).



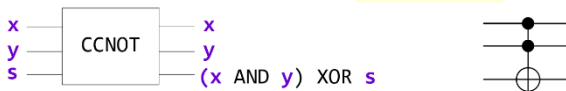
Slike 3: Operacija PRIŠTEJ je reverzibilna

Gradnja večjih reverzibilnih vezij temelji na preprosti a pomembni ugotovitvi, da z zaporedno vezavo reverzibilnih vezij vedno dobimo reverzibilno vezje. Tako potrebujemo le še čimmanjše a dovolj zmogljive gradnike, ki jih bomo zaporedno povezovali. Raziskovalci se danes vrtijo okrog gradnikov poimenovanih vrata CNOT (Feynmanova vrata), vrata CCNOT (Toffolijeva vrata), vrata CSWAP (Fredkinova vrata) in še nekaterih drugih (slika 4). Zaradi omejitve dolžine prispevka na kratko razložimo le delovanje Toffolijevih vrat. Na vohodu so 3 biti, prva dva se neposredno preslikata na izhod, medtem ko se tretji bit negira če in samo če sta prva dva enaka 1.

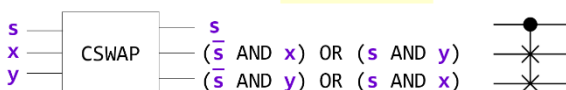
Vrata CNOT (controlled not) - Feynmanova vrata



Vrata CCNOT (controlled controlled not) - Toffolijeva vrata



Vrata CSWAP (controlled swap) - Fredkinova vrata

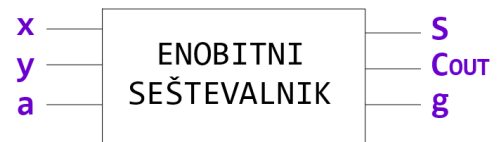


Slika 4: Pogosto uporabljani gradniki reverzibilnih vezij

4. Algoritem za gradnjo reverzibilnih vezij

Opisali bomo preprost algoritem za gradnjo reverzibilnih vezij. Danes obstaja veliko zmogljivejših algoritmov, ki so hevristični ali pa izkoriščajo posebne podatkovne strukture (za

več informacij uporabite ključno besedo »binary decision diagram«). Algoritem bomo predstavili na primeru poenostavljenega seštevalnika, v katerem na vohodu ne upoštevamo prenosa. Tak gradnik se imenuje polovični seštevalnik (ang. half-adder). Prvi korak je, da dodamo dodatne vhode in izhode (dodatni vhod označimo z a , dodatni izhod pa z g) ter tvorimo pravilnostno tabelo, ki bo reverzibilna (vsaka kombinacija bitov se mora na izhodu pojaviti natanko enkrat). Na sliki 5 je prikazana ena od možnih rešitev, sam postopek pa je natančno opisan npr. v [1] in [2].



x	y	a	S	Cout	g
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	1	0	0
0	1	1	0	1	1
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	0	1	0
1	1	1	1	1	1

Slika 5: Primer rešitve, kako delovanje enobitnega seštevalnika vgradimo v reverzibilno vezje

Načrtovanje vezja potem poteka z desne na levo. Po pravilnosti tabeli se pomikamo od zgoraj navzdol ter primerjamo kombinacijo na vohodu (vrednosti x , y in a) ter kombinacijo na izhodu (vrednosti S , $Cout$ in g). Najprej obdelamo prvo vrstico. Če izhodna kombinacija ni sestavljena iz samih ničel, potem s pomočjo operacije NOT (Toffolijeva vrata brez kontrolnih povezav) pretvorimo izhod tako, da bodo v njem samo ničle. Ta korak v našem primeru ni potreben. Nato od zgoraj navzdol poiščemo prvo vrstico, ki se razlikuje od ciljne. S pomočjo Toffolijevih vrat najprej spremenimo vse ničle, ki bi morale biti enice (za kontrolne linije izberemo tiste, ki imajo 1 v trenutnem vzorcu) nato pa spremenimo vse enice, ki bi morale biti ničle (za kontrolne linije izberemo tiste, ki imajo 1 v ciljnem vzorcu). Ta korak ponavljamo dokler ne pridemo do konca tabele. Začetek postopka je ilustriran na sliki 6, končna rešitev, ki jo dobimo po 7 korakih, pa je prikazana na sliki 7.

Avtorji algoritma so Miller, Maslov in Dueck, vsi iz Kanade. Predstavili so ga leta 2003 [3] iz česar lahko vidimo, da je to področje računalništva zelo mlado. Algoritem zagotavlja pravilno, ne pa tudi optimalno rešitev in je preveč potraten za načrtovanje večjih vezij (potreben je prehod čez celotno tabelo). Navedeni avtorji so takoj predlagali

