



2023

9. Binarni sustav

R2: SCRAPY Vodič

Broj projekta: **2021-1-FR01-KA220-SCH-000031617**



 Co-funded by
the European Union

Podrška Europske komisije za izradu ove publikacije ne znači odobravanje sadržaja, koji odražava samo stavove autora, a Komisija se ne može smatrati odgovornom za bilo kakvu upotrebu informacija sadržanih u njoj.

ECAM EPMI
30/04/2023

Sadržaj

1 Uvod	2
2. Zašto binarni?	2
3 Brojanje i pretvaranje	2
3.1 Brojanje u binarnom sistemu	3
3.2 Pretvaranje binarnog u decimalni	3
3.3 Pretvaranje iz decimalnog u binarni	5
4 Uobičajene duljine binarnih brojeva	6
5. Ispuna s vodećim nulama	7
6 Bitni operatori	7
7 Nadopuna (NOT)	7
8 OR	7
9 AND	8
10 XOR	8
11 Bitni pomaci	9
12 Zaključak	10

1 Uvod

Brojevni sustavi su metode koje koristimo za predstavljanje brojeva. Od osnovne škole svi smo uglavnom radili unutar ugodnih granica brojevnog sustava s bazom 10, ali postoje i mnogi drugi. Baza-2, baza-8, baza-16, baza-20, baza... shvatili ste. Postoji beskonačno mnoštvo sustava osnovnih brojeva, ali samo je nekoliko njih posebno važno za elektrotehniku.

Zaista popularni sustavi brojeva čak imaju i svoje ime. Baza 10, na primjer, obično se naziva decimalni brojevni sustav. Base-2, o kojoj danas govorimo, također se naziva binarnim. Drugi popularni brojčani sustav, baza 16, naziva se heksadecimalni.

Baza broja često je predstavljena cijelim brojem s indeksom iza vrijednosti. Dakle, u gornjem uvodu, prva slika bi bila 10010 nečega, dok bi druga slika bila 1002 nečega. Ovo je zgodan način za određivanje baze broja kada ikada postoji mogućnost dvosmislenosti.

2. Zašto binarni?

Pitate se zašto binarni? Pa, zašto decimalni? Oduvijek smo koristili decimale i uglavnom smo uzimali zdravo za gotovo razlog zašto smo se odlučili za brojevni sustav s bazom 10 za naše svakodnevne potrebe za brojevima. To je zato što imamo 10 prstiju, ili samo zato što su Rimljani to nametnuli svojim drevnim podređenima. Bez obzira na to što je dovelo do toga, trikovi koje smo usput naučili učvrstili su mjesto baze 10 u našim srcima; svatko zna brojati do 10. Čak i velike brojeve zaokružujemo na najbliži višekratnik od 10. Opsjednuti smo s 10!

Računala i elektronika ograničeni su na ruke i prste. Na najnižoj razini imaju samo dva načina za predstavljanje stanja bilo čega: UKLJUČENO ili ISKLJUČENO, visoko ili nisko, 1 ili 0. I tako se sva elektronika oslanja na brojevni sustav s bazom 2 za pohranjivanje, manipuliranje i matematičke brojeve.

Jako oslanjanje elektronike na binarne brojeve znači da je važno znati kako funkcionira brojevni sustav s bazom 2. Binarno ili njegove rođake, poput heksadecimalnog, često ćete susresti u svim računalnim programima. Analiza digitalnih logičkih sklopova i druge elektronike vrlo niske razine također zahtijeva veliku upotrebu binarnih podataka.

U ovoj ćete lekciji otkriti da sve što možete učiniti s decimalnim brojem možete učiniti i s binarnim brojem. Neke operacije mogu biti još lakše izvesti na binarnom broju (iako druge mogu biti bolnije). Pokrit ćemo sve to i više u ovoj lekciji.

3 Brojanje i pretvaranje

Baza svakog brojevnog sustava naziva se i radiks. Osnova decimalnog broja je deset, a osnova binarnog broja je dva. Radiks određuje koliko je različitih simbola potrebno da bi se oblikovao brojevni sustav. U našem decimalnom brojevnom sustavu imamo 10 numeričkih prikaza za vrijednosti između ništa i deset nečega: 0, 1, 2, 3, 4, 5, 6, 7, 8 i 9. Svaki od tih simbola predstavlja vrlo specifičnu, standardiziranu vrijednost.

U binarnom sustavu dopuštena su nam samo dva simbola: 0 i 1. No pomoću ta dva simbola možemo stvoriti bilo koji broj koji može decimalni sustav.

3.1 Brojanje u binarnom sistemu

Decimale možete računati beskonačno, čak i u snu, ali kako biste računali binarno? Nula i jedan u bazi dva trebali bi izgledati prilično poznato: 0 i 1. Odatle, stvari postaju izrazito binarne.

Upamtite da imamo samo te dvije znamenke, tako da kao što radimo u decimalama kada nam ponestane simbola, moramo pomaknuti jedan stupac ulijevo, dodati 1 i okrenuti sve znamenke udesno da 0. Dakle, nakon 1 dobijemo 10, zatim 11, pa 100. Počnimo brojati...

Decimal	Binarni	...	Decimal	Binarni
0	0		16	10000
1	1		17	10001
2	10		18	10010
3	11		19	10011
4	100		20	10100
5	101		21	10101
6	110		22	10110
7	111		23	10111
8	1000		24	11000
9	1001		25	11001
10	1010		26	11010
11	1011		27	11011
12	1100		28	11100
13	1101		29	11101
14	1110		30	11110
15	1111		31	11111

Počinja li to oslikavati sliku? Ispitajmo kako možemo pretvoriti te binarne brojeve u decimalne.

3.2 Pretvaranje binarnih u decimalne

Ne postoji samo jedan način pretvaranja binarnih u decimalne. U nastavku ćemo navesti dvije metode, "mathy" metodu i drugu koja je vizualnija. Pokrit ćemo oba, ali ako prvi koristi previše ružne terminologije, prijedite na drugi.

Metoda 1

Postoji praktična funkcija koju možemo koristiti za pretvaranje bilo kojeg binarnog broja u decimalni:

$$a_n 2^n + a_{n-1} 2^{n-1} + \dots + a_1 2^1 + a_0 2^0$$

Četiri su važna elementa te jednadžbe:

a_n, a_{n-1}, a_1 itd. su znamenke broja. Ovo su 0 i 1 s kojima ste upoznati, ali u binarnom obliku mogu biti samo 0 ili 1.

Također je važno promatrati položaj znamenke. Pozicija počinje s 0, krajnjom desnom znamenkom; ovaj 1 ili 0 je najmanje značajan. Svaka znamenka koju pomaknete ulijevo povećava značaj, a također povećava poziciju za 1.

Duljina binarnog broja dana je vrijednošću n , zapravo, to je $n+1$. Na primjer, binarni broj poput 101 ima duljinu 3, a nešto veći, poput 10011110 ima duljinu 8.

Svaka znamenka se množi težinom: $2^n, 2^{n-1}, 2^1$ itd. Krajnja desna težina - 20 je jednaka 1, pomaknite jednu znamenku ulijevo i težina postaje 2, zatim 4, 8, 16, 32, 64, 128, 256,... i dalje i dalje. Potencije dvojke su od velike važnosti za binarni zapis, brzo postaju vrlo poznate.

Oslobodimo se tih n -ova i eksponenata, i izvedimo našu binarnu pozicijsku jednadžbu iz osam pozicija:

$$a_7 \cdot 128 + a_6 \cdot 64 + a_5 \cdot 32 + a_4 \cdot 16 + a_3 \cdot 8 + a_2 \cdot 4 + a_1 \cdot 2 + a_0 \cdot 1$$

Uzimajući to dalje, uključimo neke vrijednosti za znamenke. Što ako imate binarni broj poput 10011011? To bi značilo (jednu) vrijednost od:

$$\begin{array}{cccccccc} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ | & | & | & | & | & | & | & | \\ a_7 & a_6 & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \end{array}$$

Metoda 2

Drugi, vizualniji način pretvaranja binarnih brojeva u decimalne je da započnete sortiranjem svake 1 i 0 u spremnik. Svaki spremnik ima uzastopnu snagu od dvije težine, 1, 2, 4, 8, 16,... na koje smo navikli. Provođenje na osam mjesta izgledalo bi otprilike ovako:

$$\begin{array}{r} 128 \quad 6432168421 \\ 1 \quad 0011011 \end{array}$$

Dakle, ako sortiramo naš binarni broj 10011011 u te spremnike, to bi izgledalo ovako:

$$\begin{array}{r} 128 \quad 6432168421 \\ 1 \quad 0011011 \end{array}$$

Za svaki spremnik koji u sebi ima binarnu vrijednost 0, samo ga prekrižite i uklonite.

$$\begin{array}{r} 128 \quad 6432168421 \\ 1 \quad 0011011 \end{array}$$

Zatim zbrojite sve preostale težine kako biste dobili svoj broj!

3.3 Pretvaranje iz decimalnog u binarni

Baš kao i prelazak iz binarnog u decimalni, postoji više od jednog načina za pretvorbu decimalnog u binarni. Prvi koristi dijeljenje i ostatke, a drugi koristi oduzimanje. Isprobajte oba i držite se onoga koji vam odgovara!

Metoda 1

Nije baš tako jednostavno pretvoriti decimalni broj u binarni. Ova pretvorba zahtijeva uzastopno dijeljenje decimalnog broja s 2, sve dok ga ne svedete na nulu. Svaki put kada podijelite ostatak dijeljenja postaje znamenka u binarnom broju koji stvarate.

Ne sjećate se kako raditi ostatke? Ako je prošlo neko vrijeme, zapamtite da, budući da dijelimo s dva, ako je dividenda paran, ostatak će biti 0; neparna dividenda znači ostatak od 1.

Na primjer, da biste pretvorili 155 u binarno, trebali biste proći kroz ovaj postupak:

$155 \div 2 = 77 R 1$ (To je krajnja desna znamenka, 1. mjesto)

$77 \div 2 = 38 R 1$ (2. pozicija)

$38 \div 2 = 19 R 0$ (3. pozicija)

$19 \div 2 = 9 R 1$

$9 \div 2 = 4 R 1$

$4 \div 2 = 2 R 0$

$2 \div 2 = 1 R 0$

$1 \div 2 = 0 R 1$ (8. pozicija)

Prvi ostatak je najmanje značajna (krajnja desna) znamenka, pa čitajte od vrha prema dolje kako biste konkretizirali naš binarni broj zdesna nalijevo: 10011011. Spojite to s gornjim primjerom...to je bingo!

Metoda 2

Ako vam se ne sviđa dijeljenje i pronalaženje ostataka, možda postoji lakša metoda za pretvorbu decimalnog u binarni. Počnite pronalaženjem najveće potencije broja dva koja je još uvijek manja od vašeg decimalnog broja i oduzmite je od decimale. Zatim nastavite oduzimati najvećom mogućom potencijom dva dok ne dođete do nule. Svaka pozicija težine koja je oduzeta, dobiva binarnu 1 znamenku; znamenke koje nisu oduzete dobivaju 0.

Nastavljajući s našim primjerom, 155 se može oduzeti sa 128, dajući 27:

$$155 - 128 = 27$$

$$128 \quad 6432168421$$

1

Naš novi broj, 27, ne može se oduzeti ni sa 64 ni sa 32. Obje te pozicije imaju 0. Možemo oduzeti 16, dajući 11.

$$27 - 16 = 11$$

$$\begin{array}{r} 128 \quad 6432168421 \\ 1 \quad 001 \end{array}$$

I 8 oduzima od 11, proizvodeći 3. Nakon toga, nema te sreće s 4.

$$11 - 8 = 3$$

$$\begin{array}{r} 128 \quad 6432168421 \\ 1 \quad 00110 \end{array}$$

Naše 3 se može oduzeti s 2, dajući 1. I konačno, 1 oduzima s 1 da bi se dobilo 0.

$$3 - 2 = 1$$

$$1 - 1 = 0$$

$$\begin{array}{r} 128 \quad 6432168421 \\ 1 \quad 0011011 \end{array}$$

Imamo binarni broj!

Bitovi, “niblovi” i bajtovi

U raspravi o izradi binarnog broja, ukratko smo obradili duljinu broja. Duljina binarnog broja je broj jedinica i nula koje ima.

4 Uobičajene duljine binarnih brojeva

Binarne vrijednosti često se grupiraju u zajedničku duljinu od 1 i 0, a taj se broj znamenki naziva i **duljina** broja. Uobičajene bitne duljine binarnih brojeva uključuju bitove, nibblove i bajtove (još ste gladni?). Svaka 1 ili 0 u binarnom broju naziva se **bit**. Od tuda se grupa od 4 bita naziva **nibble**, a 8 bita čini **bajt**.

Bajtovi su prilično uobičajena modna riječ kada radite u binarnom sistemu. Svi procesori su izgrađeni za rad s određenom duljinom bitova, koja je obično višekratnik bajta: 8, 16, 32, 64, itd.

Sažeto:

Dužina	Naziv	Primjer
1	Bit	0
4	Nibble	1011
8	Bajt	10110101

Riječ je još jedna dugačka modna riječ koja se s vremena na vrijeme izbaci. Riječ zvuči mnogo manje ukusno i mnogo je dvosmislenija. Duljina riječi obično ovisi o arhitekturi procesora. Može biti 16 bita, 32, 64 ili čak i više.

5. Ispuna s vodećim nulama

Mogli biste vidjeti binarne vrijednosti predstavljene u bajtovima (ili više), čak i ako je za izradu broja od 8 bita potrebno dodati početne nule. Vodeće nule su jedna ili više nula koje se dodaju lijevo od najvažnijeg 1-bita u broju. Obično ne vidite početne nule u decimalnom broju: 007 vam ne govori više o vrijednosti broja 7 (možda govori nešto drugo).

Vodeće nule nisu potrebne na binarnim vrijednostima, ali pomažu u predstavljanju informacija o duljini bita broja. Na primjer, možete vidjeti broj 1 ispisan kao 00000001, samo da vam kažemo da radimo unutar područja bajta. Oba broja predstavljaju istu vrijednost, međutim, broj sa sedam nula ispred dodaje informaciju o duljini bita vrijednosti.

6 Bitwise operatori

Postoji nekoliko načina za manipuliranje binarnim vrijednostima. Baš kao što možete s decimalnim brojevima, možete izvoditi standardne matematičke operacije - zbrajanje, oduzimanje, množenje i dijeljenje - na binarnim vrijednostima (o kojima ćemo govoriti na sljedećoj stranici). Također možete manipulirati pojedinačnim bitovima binarne vrijednosti koristeći bitwise operatore.

Bitwise operatori izvode funkcije bit-po-bit na jednom ili dva puna binarna broja. Oni koriste Booleovu logiku koja djeluje na skupini binarnih simbola. Ovi bitovni operatori naširoko se koriste u elektronici i programiranju.

7 Komplement (NOT)

Komplement binarne vrijednosti je kao pronalaženje točne suprotnosti svega u vezi s njom. Funkcija komplementa gleda na broj i pretvara svaku 1 u 0, a svaka 0 postaje 1. Operator komplementa također se naziva NOT.

Na primjer, da biste pronašli komplement od 10110101:

$$\begin{array}{r} \text{NOT } 10110101 \text{ (decimalno 181)} \\ \text{-----} = \\ 01001010 \text{ (decimalno 74)} \end{array}$$

NOT je jedini bitovni operator koji radi samo na jednoj binarnoj vrijednosti.

8 OR

OR uzima dva broja i proizvodi njihovu uniju. Evo postupka za OR dva binarna broja zajedno: poravnajte svaki broj tako da se bitovi podudaraju, zatim usporedite svaki od njihovih bitova koji dijele poziciju. Za svaku usporedbu bitova, ako su jedan ili oba bita 1, vrijednost rezultata na toj poziciji bita je 1. Ako obje vrijednosti imaju 0 na toj poziciji, rezultat također dobiva 0 na toj poziciji.

Četiri moguće kombinacije OR i njihov ishod su:

- 0 OR 0 = 0
- 0 OR 1 = 1

- $1 \text{ OR } 0 = 1$
- $1 \text{ OR } 1 = 1$

Na primjer, da biste pronašli $10011010 \text{ OR } 01000110$, poredajte svaki od brojeva bit po bit. Ako jedan ili oba broja imaju 1 u stupcu, vrijednost rezultata također ima 1 tamo:

$$\begin{array}{r} 10011010 \\ \text{OR } 01000110 \\ \hline 11011110 \end{array}$$

Zamislite operaciju OR kao binarno zbrajanje, bez prijenosa. 0 plus 0 je 0, ali 1 plus bilo što bit će 1.

9 AND

AND uzima dva broja i proizvodi njihovu konjunkciju. AND će proizvesti 1 samo ako su obje vrijednosti na kojima radi također 1.

Proces AND-a dviju binarnih vrijednosti zajedno sličan je OR-u. Poravnajte svaki broj tako da se bitovi podudaraju, a zatim usporedite svaki od njihovih bitova koji dijele poziciju. Za svaku usporedbu bitova, ako su jedan ili oba bita 0, vrijednost rezultata na toj poziciji bita je 0. Ako obje vrijednosti imaju 1 na toj poziciji, rezultat također dobiva 1 na toj poziciji.

Četiri moguće kombinacije 1 i njihov ishod su:

- $0 \text{ AND } 0 = 0$
- $0 \text{ AND } 1 = 0$
- $1 \text{ AND } 0 = 0$
- $1 \text{ AND } 1 = 1$

Na primjer, da biste pronašli vrijednost $10011010 \text{ AND } 01000110$, počnite poredati svaku vrijednost. Rezultat svake pozicije bita bit će samo 1 ako su oba bita u tom stupcu također 1.

$$\begin{array}{r} 10011010 \\ \text{AND } 01000110 \\ \hline 00000010 \end{array}$$

Zamislite AND kao množenje. Kad god pomnožite s 0, rezultat će također biti 0.

10 XOR

XOR je isključivi OR. XOR se ponaša kao regularni OR, osim što će proizvesti 1 samo ako jedan ili drugi brojevi imaju 1 na toj poziciji bita.

Četiri moguće XOR kombinacije i njihov ishod su:

- $0 \text{ XOR } 0 = 0$
- $0 \text{ XOR } 1 = 1$
- $1 \text{ XOR } 0 = 1$
- $1 \text{ XOR } 1 = 0$

Na primjer, da biste pronašli rezultat $10011010 \text{ XOR } 01000110$:

$$\begin{array}{r} 10011010 \\ \text{XOR } 01000110 \\ \hline 11011100 \end{array}$$

Obratite pozornost na 2. bit, 0 koja proizlazi iz dvije jedinice XOR zajedno.

11-bitni pomaci

Bitni pomaci nisu nužno bitovni operatori poput gore navedenih, ali su zgodan alat za manipuliranje jednom binarnom vrijednošću.

Postoje dvije komponente za pomak bita - smjer i količina bitova za pomak. Broj možete pomaknuti ulijevo ili udesno, a možete ga pomaknuti za jedan ili više bitova.

Prilikom pomaka udesno, jedan ili više najmanje značajnih bitova (na desnoj strani broja) jednostavno se odsijecaju i pomiču u beskonačno ništa. Vodeće nule mogu se dodati kako bi duljina bita ostala ista.

Na primjer, pomak 10011010 udesno za dva bita:

$$\begin{array}{r} \text{RIGHT-SHIFT-2 } 10011010 \text{ (decimalno 154)} \\ \hline 00100110 \text{ (decimalno 38)} \end{array}$$

Pomicanje zbrajanja ulijevo gura sve bitove prema najznačajnijoj strani (lijevoj strani) broja. Za svaki pomak dodaje se nula na poziciju najmanjeg bita.

Na primjer, pomak 10011010 ulijevo za jedan bit:

$$\begin{array}{r} \text{LEFT-SHIFT-1 } 10011010 \text{ (decimalno 154)} \\ \hline 100110100 \text{ (decimalno 308)} \end{array}$$

Taj jednostavni pomak bita obavlja kompliciranu matematičku funkciju. Pomaci ulijevo n bitova množe broj s 2^n (pogledajte kako je zadnji primjer množio unos s dva?), dok će pomak bitova udesno podijeliti cijeli broj s 2^n . Pomicanje udesno za dijeljenje može biti čudno - svi razlomci proizvedeni dijeljenjem pomakom bit će odsječeni, zbog čega je 154

dvaput pomaknuto udesno jednako 38 umjesto $154/4=38,5$. Pomaci bitova mogu biti vrlo brz način dijeljenja ili množenja s 2, 4, 8 itd.

12 Zaključak

Binarni sustav je sastavni dio svih proračuna, izračuna i operacija u elektronici. Dakle, odavde možete otići na mnogo mjesta.

Sada kada možete pretvarati između decimalnog i binarnog zapisa, to znanje možete primijeniti na razumijevanje univerzalnog kodiranja znakova: ASCII

Ili možete primijeniti svoje sjajno novo znanje na sklopove niske razine i IC-ove:

- Digitalna logika
- Registri pomaka

Također možete pogledati kako binarni igra važnu ulogu u ovim komunikacijskim protokolima:

- Serijska komunikacija
- Serijsko periferno sučelje
- I2C