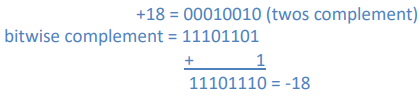
**Aritmetika celih števil**

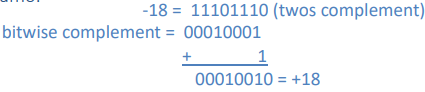
* NEGACIJA
* SEŠTEVANJE IN ODŠTEVANJE
* MNOŽENJE
* DELJENJE

**NEGACIJA**

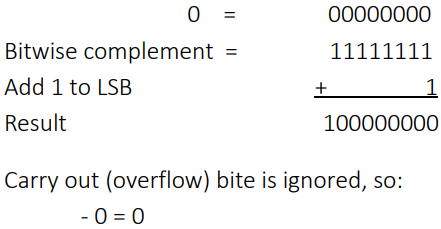
* Dvojiško komplement
* Naredimo negacijo po bitih (vključno s predznakom)
* Rezultat obravnavamo kot nepredznačeno celo število in mu dodamo 1

****

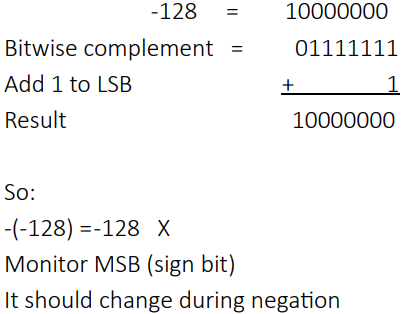
* Negativ negativnega števila je število samo:

****

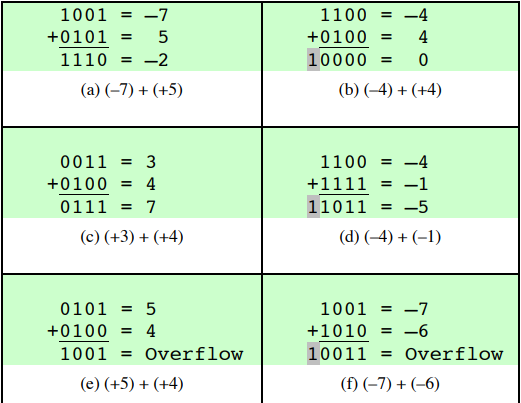
* **Primer 1**

****

* **Primer 2**

****

**Seštevanje**

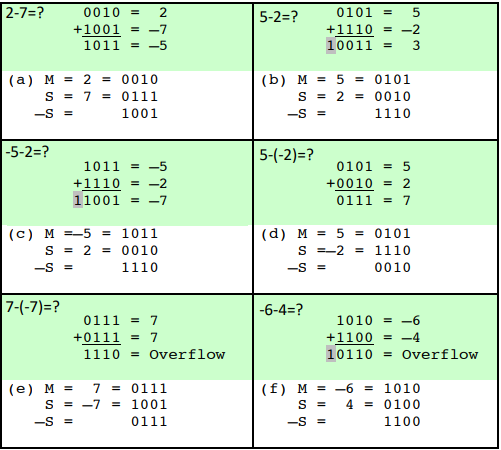
****

**Pravilo prekoračitve**

* Pri seštevanju dveh pozitivnih ali dveh negativnih števil, se prekoračitev pojavi le, če ima rezultat nasproten predznak.

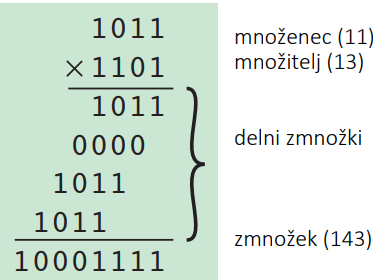
**Pravilo odštevanja**

* Če želimo odšteti eno številko (odštevanec) od druge (zmanjševanec), vzamemo dvojiški komplement odštevanca in ga prištejemo k zmanjševancu.
* Primeri: odštevanje števil v dvojiškem komplementu

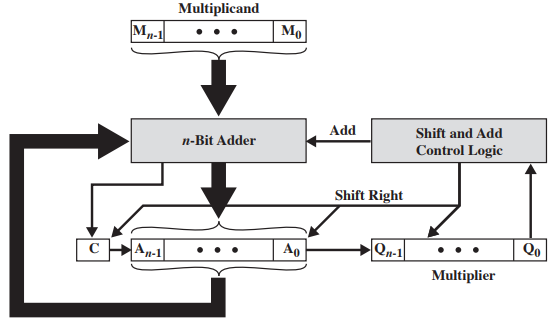
****

**Množenje nepredznačenih binarnih celih števil**

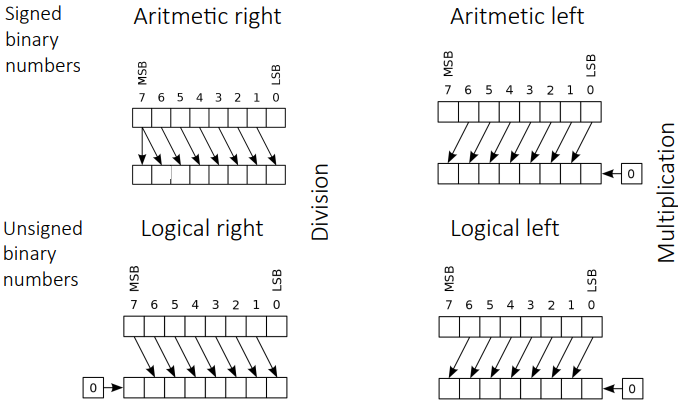
* 4-bitna cela števila, dajo 8-bitni rezultat

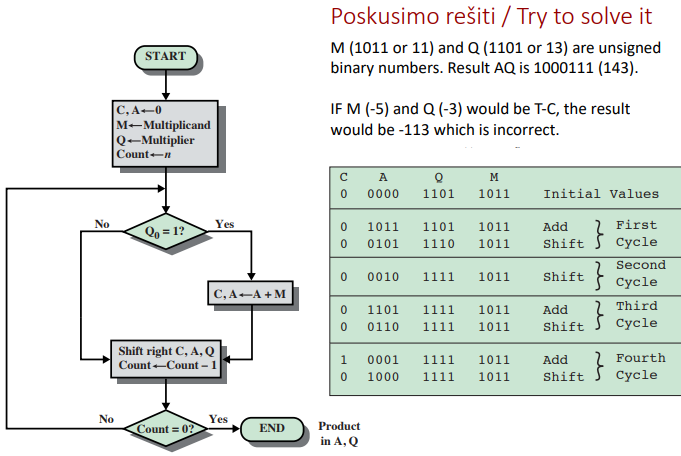
****

* Strojna izvedba nepredznačenega binarnega množenja - blokovni diagram

****

**Logični in aritmetični premik**

****

****

**Množenje negativnih števil**

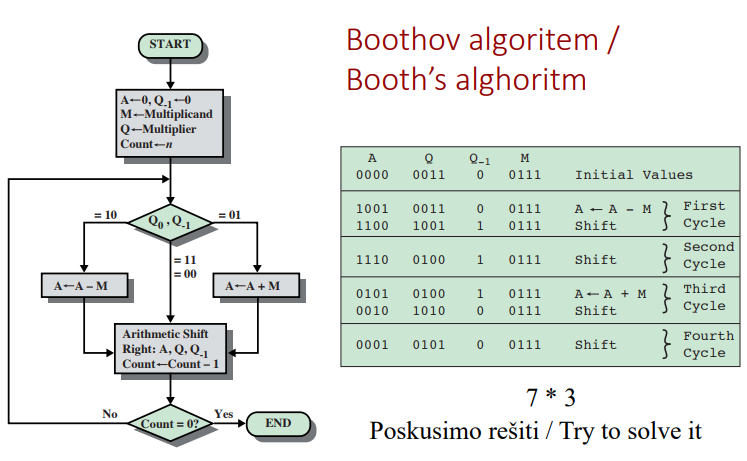
* Ne dela!
* Rešitev 1

🡪 Pretvori v pozitivno število, če je potrebno

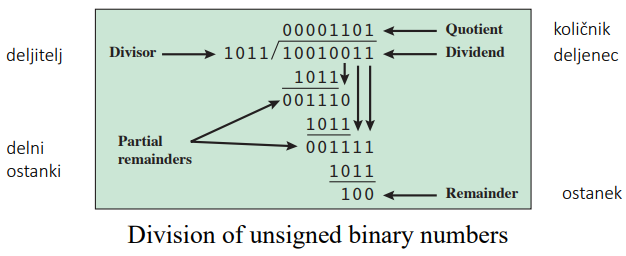
🡪 Množi kot prej

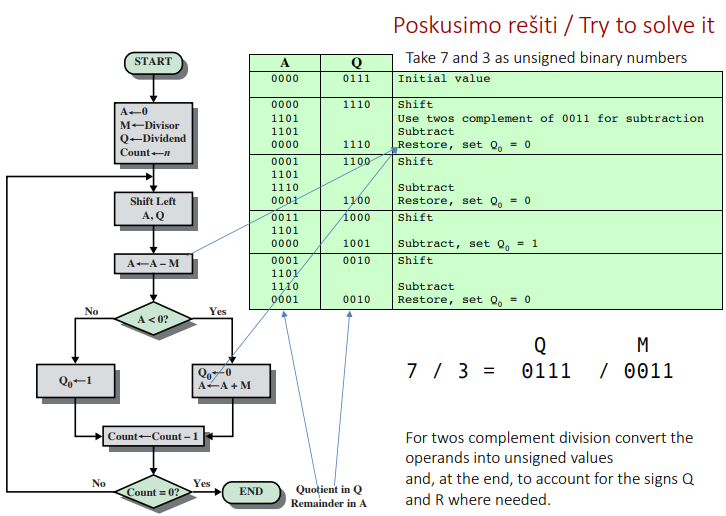
* Če sta predznaka različna, negiraj rezultat
* Rešitev 2

🡪 Boothov algoritem



* Bolj kompleksno kot množenje
* Negativna števila so problematična!

****

****

**REALNA ŠTEVILA**

* Ulomki (racionalna) + iracionalna števila
* Lahko predstavimo v čisti dvojiški obliki

1001,1010 = 23 + 20 +2-1 + 2-3 =9,625

* **Predstavitev s fiksno vejico**
* Vse do sedaj predstavljene predstavitve so s fiksno vejico
* Položaj dvojiške vejice je fiksen in se predpostavlja, da je desno od skrajne desne števke
* Programer lahko uporabi isto predstavitev za binarne ulomke z množenjem števila tako da je dvojiška vejica implicitno nameščena na kakšni drugi lokaciji
* Z zapisom s fiksno vejico je mogoče predstavljati obseg pozitivnih in negativnih celih števil, osredotočenih okoli 0
* Omejitve:
* Zelo velikega števila ni mogoče zastopat niti zelo majhnih decimalnih vrednosti (ulomek)
* Decimalni del količnika pri deljenju dveh velikih številk se lahko izgubi

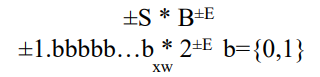
**Predstavitev s plavajočo vejico**

* Principi
* IEEE standard
* **Znanstvena predstavitev**

976,000,000,000,000 = 9.76 \* 1014

0.0000000000000976 = 9.76 \* 10-14

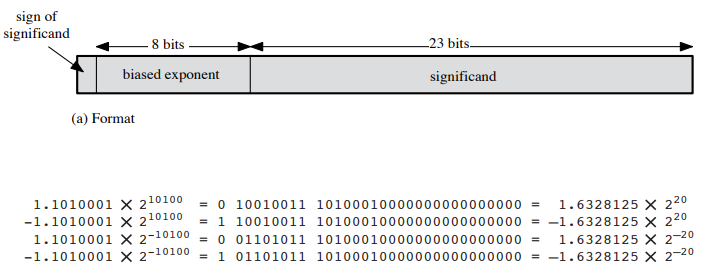
* V decimalnem sistemu pomaknemo decimalno vejico na priročno mesto in uporabimo eksponent števila 10, da vemo kje se vejica nahaja
* Enak pristop je mogoče uporabiti pri binarnih številih.
* **Števila s plavajočo vejico**

****

* Predznak
* Mantisa ali Signifikand
* Exponent

****

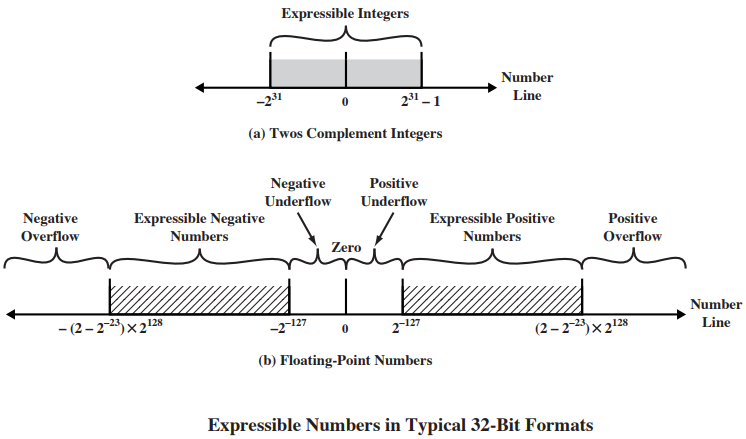
* Napačno poimenovanje
* Vejica je v resnici fiksirana med bitti eksponenta in mantiso
* Eksponent določa položaj binarne vejice

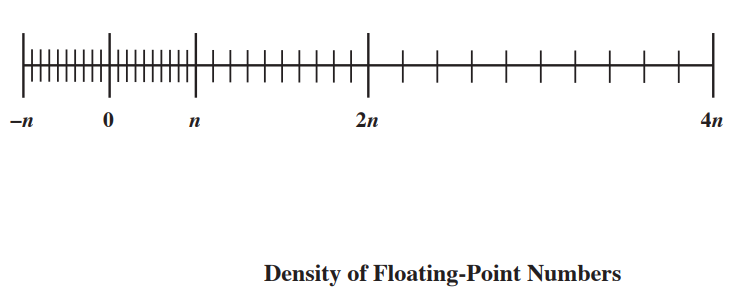


**Primeri 32 bitnih števil s plavajočo vejico**

**Deli predstavitve PV**

* Eksponent je v pristranski obliki.
* Na primer 32-bitni PV
* 8 bitno eksponentno polje
* Možne vrednosti 0-255
* Odštej 127 za pravilno vrednost
* Obseg-127 do +128
* Significand
* FP števila so običajno normalizirana
* Tj. eksponent je poravnan tako, da je vodilni bit (MSB) mantise enak 1
* Ker je vedno 1, ni potrebe, da se ga hrani

****

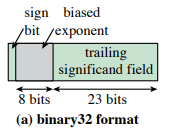
****

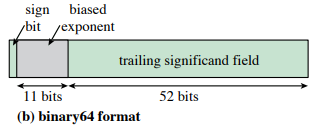
**IEEE Standard 754(-2019 ISO/IEC 60559:2020)**

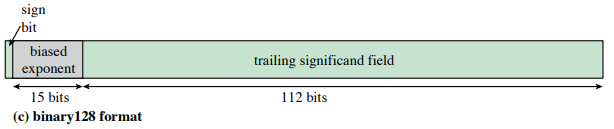
* Opredeljuje najpomembnejši prikaz s plavajočo vejico
* Standard je bil razvit za lažjo prenosljivost programov z enega procesorja na drugega in za spodbujanje razvoja prefinjenih numerično usmerjenih programov
* Standard je bil široko sprejet in se uporablja na skoraj vseh sodobnih procesorjih in aritmetičnih koprocesorjih
* IEEE 754-2019 zajema binarne in decimalne predstavitve s plavajočo vejico
* **IEEE 754-2019**

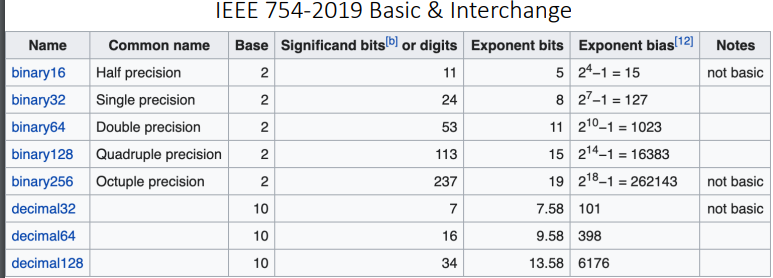
Določi naslednje različne vrste formatov s plavajočo vejico:

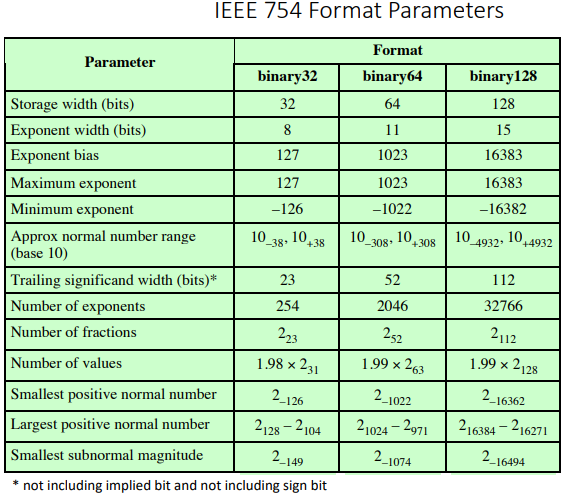
* Aritmetična oblika
* Oblika podpira vse obvezne operacije, ki jih določa standard. Oblika se lahko uporabi za prikaz operandov s plavajočo vejico ali rezultatov za operacije, opisane v standardu.
* Osnovna oblika
* Ta oblika zajema osem predstavitev s plavajočo vejico, pet binarnih in tri decimalni, katerih kodiranje določa standard in jih je mogoče uporabiti za aritmetiko. Vsaj eden od osnovnih formatov je izveden v kateri koli skladni izvedbi.
* Oblika izmenjave
* Popolnoma določeno binarno kodiranje s fiksno dolžino, ki omogoča izmenjavo podatkov med različnimi platformami in se lahko uporablja za shranjevanje.

****

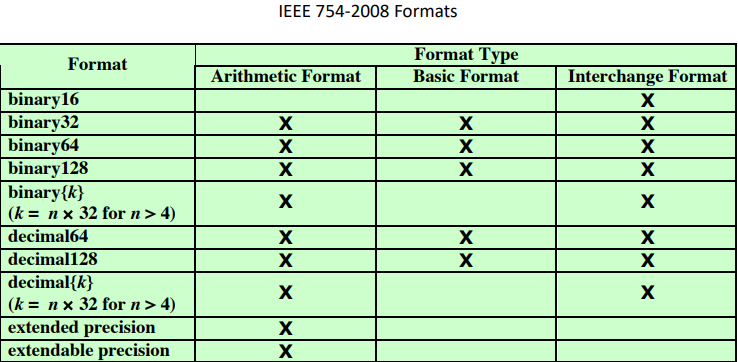
****

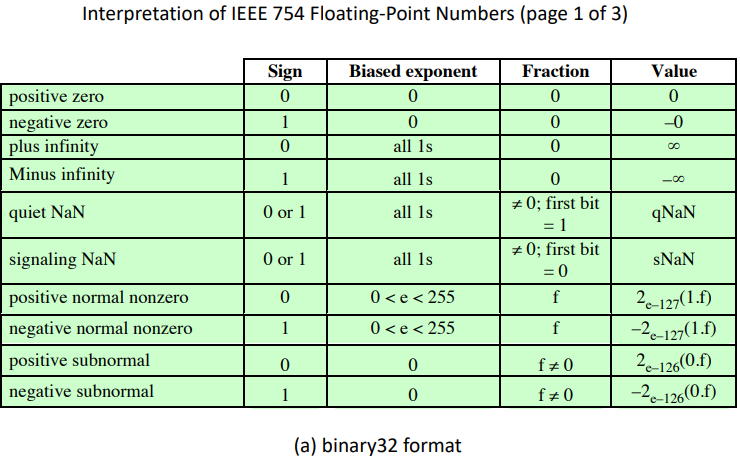
****

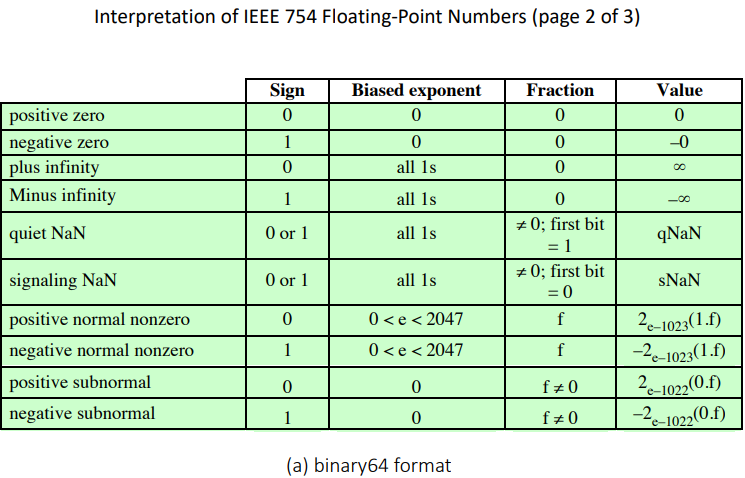
****

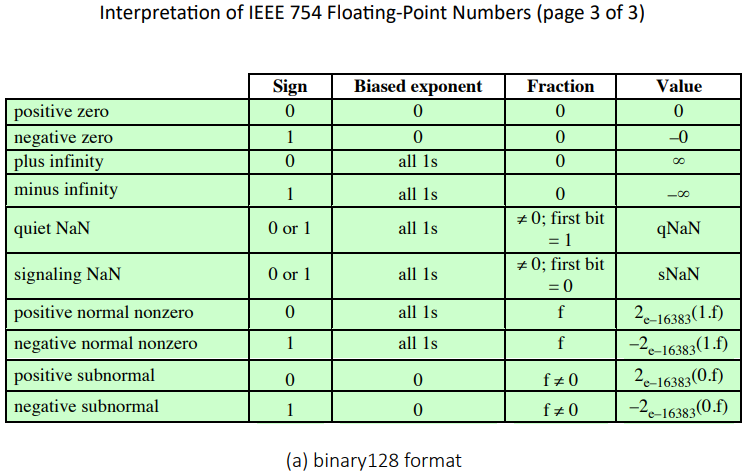
****

* **DODATNE OBLIKE**
* RAZŠIRJENI NATANČNI FORMATI
  + Omogočajo dodatne bite v eksponentu (razširjeni obseg) in v mantisi (razširjena natančnost)
  + Zmanjšujejo pretirano napako končnega rezultata, ki lahko nastane z zaokroževanjem
  + Zmanjšuje možnost vmesne prekoračitve, ki prekine izračun, katerega končni rezultat bi bil predstavljiv v osnovnem format
  + Omogoča nekatere prednosti večjega osnovnega formata, ne da bi prišlo do časovne kazni, ki je običajno povezana z večjo natančnostjo
* RAZŠIRLJIVA OBLIKA NATANČNOSTI
* Natančnost in domet sta določena od in pod nadzorom uporabnika
* Lahko se uporablja za vmesne izračune, vendar standardni ne omejujejo ali oblikujejo ali dolžine

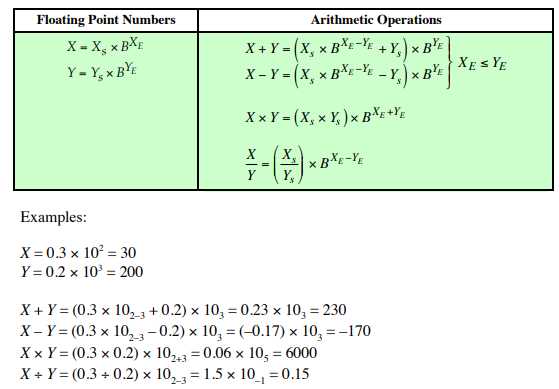
****

****

****

****

**FP Aritmetične operacije**

****

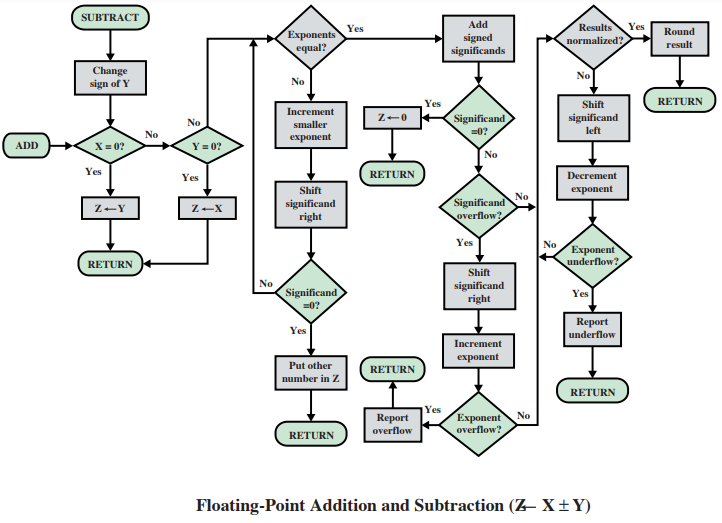
**Izjemni rezultati**

Operacija s plavajočo vejico lahko povzroči tudi enega od teh pogojev:

* Prekoračitev eksponenta: + ∞ ali - ∞.
* Podkoračitev eksponentov: npr. -200 <-127
* Prekoračitev significanda
* Se lahko reši z zaokroževanjem
* Podkoračitev significanda
* Se lahko reši s poravnavo

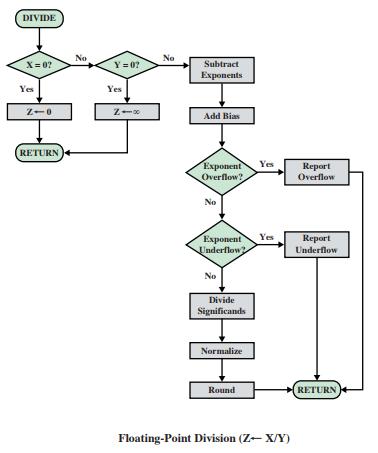
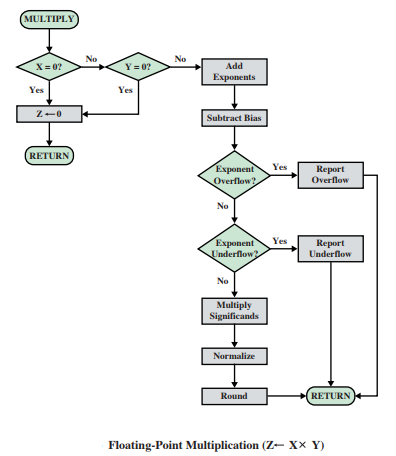
**FP aritmetika: +/-**

* Preveri se za ničlo
* Poravna se mantisi s popravljanjem eksponenta
* Mantisi se sešteje ali odšteje
* Normalizira se rezultat

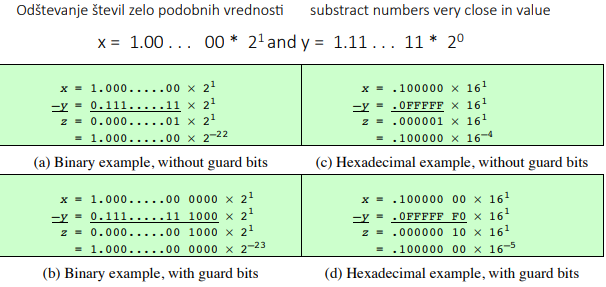


**FP aritmetika: ×/÷**

* Preveri, če je v računu nič
* Eksponenta se seštejeta ali odštejeta
* Mantisi se zmnožita ali delita
* Število se normalizira
* Nato zaokroži
* Vmesni rezultat bi moral biti shranjen v pomnilniku dvojne dolžine

****

**Uporaba varovalnih bitov**

****

**Natančnost**

Standardni pristopi IEEE:

* Zaokrožitev do najbližje:
* Rezultat je zaokrožen na najbližjo predstavljivo številko.
* Zaokrožitev proti + ∞:
* Rezultat je zaokrožen proti pozitivni neskončnosti.
* Zaokrožitev proti -∞:
* Rezultat je zaokrožen navzdol proti negativni neskončnosti.
* Zaokrožitev proti 0:
* Rezultat se zaokroži proti nič.

**Intervalna aritmetika**

Negativna neskončnost in zaokrožitev na plus sta uporabna pri izvajanju intervalne aritmetike

* Zagotavlja učinkovito metodo za spremljanje in nadzor napak pri izračunih s plavajočo vejico, tako da ustvari dve vrednosti za vsak rezultat
* Obe vrednosti ustrezata spodnji in zgornji končni točki intervala, ki vsebuje resnični rezultat
* Širina intervala označuje natančnost rezultata
* Če končne točke niso predstavljive, se intervalne končne točke zaokrožijo navzdol oziroma navzgor
* Če je razpon med zgornjim in spodnjim robom dovolj ozek, smo dobili dovolj natančen rezultat

**Odrezovanje**

* Zaokrožitev proti ničli
* Dodatni biti se ne upoštevajo
* Najenostavnejša tehnika
* Stalna pristranskost proti ničli v operaciji
* Resna pristranskost, ker vpliva na vsako operacijo, za katero obstajajo neničelni dodatni biti

**IEEE 745 – neskončnost**

* Obravnava se kot omejevalni primer prave aritmetike, pri čemer so vrednosti neskončnosti podane z naslednjo razlago:

- ∞ < (every finite number) < + ∞

