



KOMPYUTERDA AXBOROTNI QAYTA ISHLASHNING ARIFMETIK ASOSLARI

FARG'ONA DAVLAT UNIVERSITETI.

FIZIKA-MATEMATIKA fakulteti.

AMALIY MATEMATIKA yo'nalishi.

1-bosqich talabasi

MAHAMMADJONOVA O'G'ILOY MIRZAAKBAR QIZI

INFORMATIKANING NAZARIY ASOSLARI FANIDAN YOZGAN

ANOTATSIYA

Ushbu mavzu kompyuterda axborotni qayta ishlash jarayonining arifmetik asoslarini o'rganishga bag'ishlangan. Unda sonlar va ularning kompyuter xotirasida ifodalanishi, sanoq sistemalari (ikkilik, sakkizlik, o'nlik va o'n oltilik), arifmetik amallarni bajarish prinsiplari hamda mantiqiy va arifmetik qurilmalarning ishlash mexanizmlari yoritib beriladi. Shuningdek, axborotni kodlash, arifmetik mantiqiy qurilma (AMQ) vazifalari, kompyuterda hisoblash jarayonlarining aniqligi va tezkorligi bilan bog'liq masalalar tahlil qilinadi. Mazkur mavzu talabalarda kompyuter texnikasining ishlash asoslarini tushunish, axborotni qayta ishlash jarayonlariga nisbatan mantiqiy fikrlashni shakllantirishga xizmat qiladi.

Kalit so'zlar Sanoq tizimlari (Ikkilik, Sakkizlik, O'nlik, O'n oltilik). Ikkilik arifmetikasi (Qo'shish, ayirish, ko'paytirish, bo'lish). Bit va Bayt. To'g'ri, teskari va qo'shimcha kodlar Suzuvchi nuqtali sonlar (Floating-point). Qat'iy nuqtali sonlar (Fixed-point). Mantiqiy amallar (AND, OR, NOT, XOR). Registerlar. Summator (Qo'shuvchi qurilma). Sonlarni tasvirlash diapazoni

1. Kompyuterda axborot faqat ikilik ko'rinishda qayta ishlanadi, chunki zamonaviy elektron sxemalar faqat ikki xil kuchlanish holatini aniq ajrata



oladi: “0” va “1”. Shu sababli barcha matn, rasm, audio, video va sonlar kompyuterga kirganda ikkilik raqamlar ketma-ketligiga aylantiriladi.

Kompyuterlarning ishlash tamoyili elektron sxemalarga asoslangan bo‘lib, ular faqat ikki xil holatni – kuchlanish bor (1) va kuchlanish yo‘q (0) holatlarini aniq va ishonchli tarzda farqlay oladi. Shu sababli kompyuterlar barcha ma’lumotlarni, ya’ni matn, rasm, audio, video, sonlar va buyruqlarni **ikkilik (binary)** ko‘rinishda saqlaydi va qayta ishlaydi. Ikkilik tizim kompyuter uchun eng qulay bo‘lib, unda har bir raqam **bit** deb ataladi. Bit faqat ikki qiymatga ega bo‘lgani uchun xatolik ehtimoli minimal bo‘ladi va elektron sxemalar tomonidan tez ishlanadi. Kompyuterning barcha jarayonlari – dasturlarni bajarish, xotiraga murojaat qilish, hisob-kitoblar bajarilishi – aynan shu 0 va 1 lar ketma-ketligi asosida amalga oshadi.

2. Kompyuter arifmetikasi pozitsion sanoq tizimlariga asoslanadi, ularda raqamning qiymati o‘z pozitsiyasiga bog‘liq. Eng muhim tizim – 2-lik tizim, chunki kompyuter fizik jihatdan faqat ikkita holatni aniq qayd eta oladi.

Kompyuterda barcha arifmetik amallar va hisob-kitoblar pozitsion sanoq tizimlariga tayanadi. Pozitsion sanoq tizimi — bu raqamning qiymati uning o‘rindagi pozitsiyasiga bog‘liq bo‘lgan tizim bo‘lib, har bir belgi o‘zining o‘rniga ko‘ra ma’lum og‘irlikka ega. Masalan, o‘nlik tizimda 245 sonidagi 2 yuzlikni, 4 o‘nlikni, 5 birlikni bildiradi. Kompyuterlarda esa eng asosiy pozitsion tizim — **ikkilik (2-lik) sanoq tizimi**, chunki elektron sxemalar faqat ikki holatni farqlay oladi. Shuningdek, 8-lik (oktal), 10-lik (decimal) va 16-lik (hexadecimal) tizimlar ham ishlatiladi, chunki ular ikkilik tizim bilan bevosita bog‘liq va o‘zaro o‘tish oson.

3. Axborotni raqamli ko‘rinishga o‘tkazish jarayoni kodlash deb ataladi. Kodlash orqali real dunyodagi ma’lumotlar (harflar, signallar,



tovushlar) kompyuterga tushunarli bo'lgan ikkilik ketma-ketliklarga aylantiriladi.

Kodlashning maqsadi – real dunyo obyektlarini va belgilarni kompyuter xotirasi va protsessorida saqlash va ishlatish imkonini yaratish. Masalan:

- Harf va matnlar **ASCII** yoki **Unicode** standartlari orqali kodlanadi;
- Rasm va video fayllar **piksel va rang kodlari** yordamida raqamlashtiriladi;
- Ovoz signallari esa **raqamli namuna (digital sampling)** orqali saqlanadi.

Kodlash orqali ma'lumotlar elektron sxemalar uchun mos formatga o'tadi va kompyuter arifmetikasida ishlashga tayyor bo'ladi. Shu bilan birga, kodlash jarayoni axborotni uzatish, saqlash va qayta ishlashda aniqlik va samaradorlikni ta'minlaydi.

4. Sanoq tizimlari o'rtasida o'tish (2-lik, 8-lik, 10-lik, 16-lik) kompyuter arifmetikasida muhim, chunki bir nechta tizimlar turli vazifalarda qulaylik yaratadi: masalan, 16-lik tizim – xotira manzillarini ifodalashda qo'llaniladi.

Masalan:

- **Ikilik tizim (2-lik)** – kompyuterning ichki ishlashi uchun asosiy tizim, chunki elektron sxemalar faqat ikkita holatni – 0 va 1 ni farqlay oladi.
- **Sakkizlik tizim (8-lik)** – ba'zi eski tizimlarda xotira manzillarini qisqacha ifodalash uchun ishlatiladi.
- **O'nlik tizim (10-lik)** – insonlar uchun qulay, chunki kundalik hayotda shu tizimdan foydalanamiz.
- **O'n oltitalik tizim (16-lik)** – xotira manzillari, rang kodlari va raqamli qurilmalarda tez ishlash uchun qo'llaniladi.



Sanoq tizimlari o'rtasida o'tish **formulalar va algoritmlar** orqali amalga oshiriladi:

- Ikilik $\rightarrow$ O'nlik: har bir bitning og'irlik darajasini hisoblash orqali.
- O'nlik $\rightarrow$ Ikilik: sonni 2 ga bo'lib, qoldiqni yozish orqali.
- Ikilik $\leftrightarrow$ O'n oltitalik: har 4 bitni bir hexadecimal raqam bilan almashtirish.

5. Arifmetik-logik qurilma (ALU) barcha arifmetik (qo'shish, ayirish, ko'paytirish, bo'lish) va mantiqiy amallarni (VA, YOKI, EMAS) bajaradi. Protsessorning eng asosiy qismi aynan shu qurilmadir.

ALU ning asosiy funksiyalari:

1. **Arifmetik amallar** – qo'shish, ayirish, ko'paytirish, bo'lish, modul bo'yicha hisoblash.
2. **Mantiqiy amallar** – AND, OR, XOR, NOT kabi bit darajasidagi operatsiyalar.
3. **Solishtirish** – tenglik, katta yoki kichik bo'lishni tekshirish.
4. **Bitlarni siljitish** – ma'lumotni chapga yoki o'ngga siljitish, ko'paytirish va bo'lishni tezlashtirish uchun ishlatiladi.

ALU arifmetik amallarni bajarishda sonlarni **bitlar ketma-ketligi** sifatida qabul qiladi va natijani registrlar yoki xotiraga yuboradi. Masalan, qo'shish jarayonida ALU ikkita sonni oladi, ularni ikilik tizimda qo'shadi va natijani chiqish registriga joylashtiradi.

ALU zamonaviy kompyuterlarning tezligi va samaradorligini belgilovchi muhim blokdir. Chunki har bir dastur yoki hisoblash jarayoni aynan ALU orqali bajariladi. Shu sababli kompyuter arifmetikasini tushunish uchun ALUning ishlash tamoyili va funksiyalarini bilish zarur.



5. **Ikilik qo'shish kompyuter arifmetikasining asosi** bo'lib, ko'paytirish va bo'lish kabi murakkab jarayonlar ham shu amal asosida quriladi. Masalan, ko'paytirish takroriy qo'shish orqali bajariladi.

Ikilik qo'shishda faqat ikkita raqam – 0 va 1 ishlatiladi. Qo'shishning natijasi ham 0 yoki 1 bo'ladi, lekin agar ortiqcha qiymat yuzaga kelsa, u keyingi bitga "carry" sifatida uzatiladi. Misol:

A	B	Natija	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Bu jadvaldan ko'rinib turibdiki, $1+1=10$ natija beradi, ya'ni 0 chiqadi va 1 keyingi bitga uzatiladi. Shu prinsip asosida kompyuter **ko'p bitli sonlarni** qo'shadi va natijani hisoblaydi.

6. **Manfiy sonlar kompyuterda 2-to'ldirma kodi bilan beriladi**, bu qo'shish va ayirish amallarini soddalashtiradi. Shu sababli barcha protsessorlar ushbu usuldan foydalanadi.

Kompyuter faqat 0 va 1 qiymatlar bilan ishlay olgani sababli, **manfiy sonlarni ifodalash maxsus usulni talab qiladi**. Eng keng tarqalgan va samarali usul – **2-to'ldirma (two's complement) kodi**.

2-to'ldirma kodi quyidagicha ishlaydi:

1. Manfiy sonning modulini ikilik tizimda ifodalaymiz.
2. Har bir bitni teskari holatga o'tkazamiz ($0 \rightarrow 1$, $1 \rightarrow 0$).



3. Natijaga 1 qo'shamiz.

Masalan, -5 sonini 8-bitli tizimda ifodalash:

- 5 ni ikilikda yozamiz: 00000101
- Bitlarni teskari qilamiz: 11111010
- 1 qo'shamiz: 11111011

Natijada -5 soni 8-bitli 2-to'ldirma kodida 11111011 bo'ladi.

7. **1-to'ldirma va belgi-modul usullari** tarixda qo'llangan bo'lsa-da, ular arifmetika jarayonlarini murakkablashtirgani uchun hozir asosiy usullar sirasiga kirmaydi. 2-to'ldirma kodi eng qulay va barqaror hisoblanadi.

1-to'ldirma kodi:

- Manfiy sonni ifodalash uchun uning modulini ikilik tizimda yozib, barcha bitlarni teskari qilinadi ($0 \rightarrow 1, 1 \rightarrow 0$).
- Masalan, 5 sonini 8-bitda ifodalash: 00000101
- -5 ni 1-to'ldirma bilan: 11111010
- Ammo bu usul qo'shish va ayirish amallarini murakkablashtiradi, chunki qo'shishda ortiqcha bitlarni alohida hisoblash kerak bo'ladi.

Belgi-modul usuli:

- Eng yuqori bit sonning belgisini bildiradi (0 – musbat, 1 – manfiy), qolgan bitlar sonning modulini ifodalaydi.
- Masalan, 5 soni: 00000101, -5 soni: 10000101
- Bu usul ham qo'shish va ayirishni murakkablashtiradi, chunki belgi va modul alohida ishlanadi.



8. **Kasr sonlar floating-point shaklida saqlanadi**, bunda son mantissa (asosiy qiymat) va eksponent (daraja) ko‘rinishida ifodalanadi. Bu juda katta yoki juda kichik sonlar bilan ishlash imkonini beradi.

Floating-point son quyidagi tarkibiy qismlardan iborat:

1. **Mantissa** – sonning asosiy qiymati, ya’ni raqamlarning ketma-ketligi.
2. **Eksponent** – sonni o‘ng yoki chapga siljitish orqali kattalashtirish yoki kichraytirishni bildiradi.
3. **Belgi (sign)** – son musbat yoki manfiy ekanligini ko‘rsatadi.

Masalan, son 0.000123 floating-point ko‘rinishda quyidagicha ifodalanishi mumkin:

- Mantissa: 1.23
- Eksponent: -4

Floating-point arifmetikasining afzalliklari:

- Juda katta diapazonli sonlarni ifodalash mumkin.
- Kichik va katta sonlar bilan arifmetik amallarni tez bajarish imkonini beradi.
- Kompyuterning xotirasini samarali ishlatadi.

Ammo floating-point sonlarda **aniqlik yo‘qotilishi (precision loss)** sodir bo‘lishi mumkin, chunki sonlar cheklangan bitlar bilan saqlanadi. Shu sababli ilmiy hisob-kitoblarda natijalarni yaxshilash uchun ehtiyotkorlik talab etiladi.



9. **IEEE 754 standarti** floating-point sonlarning butun dunyoda bir xil formatda saqlanishini ta'minlaydi. Shu sababli turli kompyuter va dasturiy tizimlar orasida aniq moslik saqlanadi.

IEEE 754 standarti quyidagi asosiy jihatlarni belgilaydi:

1. **Bitlar taqsimoti:** Son uch qismdan iborat bo'ladi – belgi (sign), eksponent (exponent) va mantissa (fraction).
2. **Bit uzunligi:** Standartda 32-bit (single precision) va 64-bit (double precision) formatlar mavjud.
 - 32-bit: 1 bit belgi, 8 bit eksponent, 23 bit mantissa
 - 64-bit: 1 bit belgi, 11 bit eksponent, 52 bit mantissa
3. **Maxsus qiymatlar:** Standart NaN (Not a Number), $+\infty$, $-\infty$ kabi qiymatlarni aniqlaydi.
4. **Aniqlik va chegaralar:** Standart sonlarning aniqlik darajasini, minimal va maksimal qiymatlarni belgilaydi.

IEEE 754 standarti floating-point sonlarni saqlash va arifmetik amallarni bajarishda **bir xil natijalarni** kafolatlaydi. Shu sababli turli kompyuter tizimlari va dasturlar orasida hisob-kitoblar mos keladi.

1. **To'lish (overflow)** kompyuter xotirasida sonlarning maksimal chegarasi oshib ketganda yuzaga keladi. Bunday holatda arifmetik jarayon noto'g'ri natija qaytarishi mumkin.

To'lishning oqibatlari:

- Natijalar noto'g'ri chiqadi.
- Ba'zi dasturlarda dastur xatosi yoki tizim xatoligi yuz beradi.
- Floating-point amallarda esa katta sonlar $+\infty$ sifatida ifodalanadi.



Oldini olish usullari:

- Katta bitli registr yoki xotira formatlaridan foydalanish (32-bit → 64-bit).
- Arifmetik amallarni ehtiyotkorlik bilan rejalashtirish.
- Protsessorlar va dasturlar odatda overflow flag (bayroq) orqali to'lishni aniqlaydi.

Shuning uchun **to'lish (overflow) kompyuter arifmetikasida muhim hodisa** bo'lib, hisob-kitoblarda aniqlikni ta'minlash uchun doimo nazorat qilinadi.

2. Ostunish (underflow) son juda kichik bo'lganda mantissa 0 ga yaqinlashib, natijada son butunlay 0 ga tenglashtiriladi. Bu floating-point arifmetikasidagi tabiiy hodisa.

Masalan, floating-point formatda minimal musbat son 1.0×10^{-38} bo'lsa, bundan kichik qiymatlar ostunish holatiga tushadi. Shu holatda:

- Sonning haqiqiy qiymati yo'qoladi yoki 0 deb qabul qilinadi.
- Natijada arifmetik amallar aniqlikni yo'qotishi mumkin.

Ostunishning oqibatlari:

- Ilmiy hisob-kitoblarda xatoliklar yuzaga keladi.
- Juda kichik qiymatlar bilan ishlaydigan dasturlar noto'g'ri natija chiqaradi.

Oldini olish usullari:

- Katta diapazonli floating-point formatlardan foydalanish (32-bit → 64-bit).



- Sonlarni normalizatsiya qilish va algoritmlarda kichik qiymatlarni hisobga olish.

- Protsessorlar ostunish flagini (underflow flag) qo‘llaydi, bu dasturga xatoliklarni aniqlash imkonini beradi.

Aniqlik yo‘qotilishi (precision loss) kompyuterning haqiqiy sonlarni cheklangan bitlar yordamida saqlashi bilan bog‘liq. Masalan, 0.1 kabi sonlar ikkilikda aniq ifodalanmaydi, bu natijaga ta’sir qiladi.

3. Bit – eng kichik axborot birligi bo‘lib, kompyuter barcha ma’lumotlarni bitlar ketma-ketligi sifatida saqlaydi. 1 bayt esa 8 bitdan iborat bo‘lib, harflarni kodlashda keng qo‘llanadi.

Bitlar yordamida quyidagilar ifodalanadi:

- **Sonlar** – butun va kasr sonlar ikilik kod orqali bitlarda saqlanadi.

- **Matn va belgilar** – ASCII yoki Unicode standartlari orqali har bir harf bitlar orqali kodlanadi.

- **Rasmlar va video** – har bir piksel rang qiymati bitlar ketma-ketligi bilan ifodalanadi.

- **Ovoz signallari** – analog signal raqamli shaklga o‘tkazilib, bitlar bilan saqlanadi.

Shuningdek, **1 bayt** 8 bitdan iborat bo‘lib, kompyuter xotirasida ma’lumotni tartib bilan saqlash va ishlash uchun asosiy birlik hisoblanadi. Masalan, 1 bayt orqali 256 xil qiymatni ifodalash mumkin (0 dan 255 gacha).

4. Sonlarning uzunligi kompyuter arxitekturasiga bog‘liq: 8, 16, 32, 64 bitli formatlar mavjud. Bitlar soni qancha katta bo‘lsa, son shuncha katta yoki aniq bo‘ladi.



Kompyuterda **sonlarning uzunligi** ularni saqlash va qayta ishlashda ajratilgan **bitlar soniga** bog'liq. Har bir kompyuter arxitekturasida o'ziga xos bit kengligiga ega bo'lib, odatda quyidagi formatlar keng qo'llanadi:

- **8 bit** – kichik butun sonlar uchun (1 bayt), qiymat oralig'i: 0...255 (musbat) yoki -128...127 (2-to'ldirma bilan).

- **16 bit** – o'rta kattalikdagi butun sonlar, qiymat oralig'i: -32,768...32,767.

- **32 bit** – katta sonlar va floating-point single precision formatlari, qiymat oralig'i: -2,147,483,648...2,147,483,647.

- **64 bit** – juda katta sonlar va double precision floating-point formatlari, qiymat oralig'i: -9,223,372,036,854,775,808...9,223,372,036,854,775,807.

5. Protssessor registrari arifmetik amallarda hal qiluvchi rol o'ynaydi, chunki ular juda tezkor xotira bo'lib, ular orqali barcha hisob-kitoblar bajariladi. Registrar soni va o'lchami protssessor unumdorligiga ta'sir qiladi.

Registrlarning vazifalari:

1. **Ma'lumotlarni vaqtinchalik saqlash** – qo'shish, ayirish, ko'paytirish yoki bo'lish amallari bajarilishidan oldin operandlar registrga yuklanadi.

2. **Natijani saqlash** – ALU tomonidan hisoblangan qiymat dastlab registrda saqlanadi, so'ngra xotiraga yoki boshqa registrga uzatiladi.

3. **Tezkor hisoblash** – registrar protssessorning ishlash tezligini oshiradi, chunki ular xotira moduliga nisbatan ancha tez ishlaydi.

Masalan, ikkita sonni qo'shishda:

- Sonlar registrga yuklanadi,



- ALU qo'shadi,
- Natija yana registrga saqlanadi.

Shuning uchun arifmetik amallarni bajarishda **registrarlar hal qiluvchi vosita** hisoblanadi. Ular protsessorning samaradorligi, tezligi va hisoblash natijalarining aniqligiga bevosita ta'sir qiladi.

6. Sonlarni kodlash usullari (masalan, 2-to'ldirma, floating-point) kompyuterning qanday hisoblayotganini belgilaydi. Kodlash usuli noto'g'ri tanlansa, natijalarda xato paydo bo'lishi mumkin.

☐ **Belgi-modul (sign-magnitude)**

- Eng yuqori bit sonning belgisini (0 – musbat, 1 – manfiy) bildiradi, qolgan bitlar modulni ifodalaydi.
- Afzalligi – oddiy va tushunarli; kamchiligi – 0 sonining ikki xil shakli mavjud va arifmetik amallar murakkab bo'ladi.

☐ **1-to'ldirma (one's complement)**

- Manfiy sonning bitlarini teskari holatga o'tkazadi.
- Qo'shish va ayirish uchun ortiqcha hisoblash talab qilinadi, 0 soni ikki xil ko'rinishda bo'lishi mumkin.

☐ **2-to'ldirma (two's complement)**

- Manfiy sonni ifodalashning eng keng tarqalgan usuli.
- Qo'shish va ayirish amallari bir xil sxema orqali amalga oshiriladi.
- 0 soni yagona ko'rinishda ifodalanadi.

☐ **Floating-point (suzuvchi nuqta)**



- Kasr sonlar va juda katta yoki juda kichik sonlarni ifodalash uchun ishlatiladi.

- Son mantissa, eksponent va belgi qismidan iborat bo'ladi.

- IEEE 754 standarti asosida saqlanadi.

7. Kompyuter arifmetikasi barcha ilmiy ishlanmalar uchun poydevor bo'lib xizmat qiladi: fizik simulyatsiyalar, matematik modellashirish, statistik hisob-kitoblar aynan arifmetik amallar asosida bajariladi.

Masalan:

- Fizika va kimyo hisob-kitoblarida murakkab formulalarni tez va aniqlik bilan yechish.

- Muhandislik va konstruktsiya loyihalarida mexanik va elektron hisob-kitoblarni bajarish.

- Meteorologiya va geodeziyada katta hajmdagi ma'lumotlarni qayta ishlash.

- Sun'iy intellekt, statistik tahlil va matematik modellarda algoritmlarni ishlatish.

Kompyuter arifmetikasining afzalliklari:

- Murakkab formulalarni tezkor va xatoliksiz bajarish.

- Katta hajmdagi ma'lumotlarni samarali qayta ishlash.

- Floating-point arifmetika orqali katta va kichik sonlarni aniqlik bilan ifodalash.

Shu sababli, ilmiy ishlanmalar va hisob-kitoblar **kompyuter arifmetikasiga tayanadi**, va uning asoslarini bilish har bir ilmiy va muhandislik faoliyati uchun muhim hisoblanadi.



5. **Sun'iy intellekt, kriptografiya va signalni qayta ishlash** kabi zamonaviy sohalar kompyuter arifmetikasining to'g'ri ishlashiga bevosita bog'liq. Aniqlik, tezlik va samaradorlik ushbu sohalarning muvaffaqiyat kalitidir.

1. **Sun'iy intellekt (AI):**

- Mashina o'rganish va neyron tarmoqlarda sonlar ustida yuzlab minglab arifmetik operatsiyalar bajariladi.
- Floating-point amallar yordamida og'irliklar va biaslar hisoblanadi.
- Aniqlik va tezlik arifmetika asosida ta'minlanadi.

2. **Kriptografiya:**

- Shifrlash va de-shifrlash algoritmlari ko'pincha **modulyar arifmetika va katta sonlar bilan ishlashga** asoslanadi.
- RSA, AES kabi algoritmlar sonlarning kodlash va arifmetik operatsiyalariga tayanadi.
- Arifmetik asoslar xavfsiz va ishonchli kriptografik tizimlarni yaratishga imkon beradi.

3. **Raqamli signalni qayta ishlash (DSP):**

- Ovoz, rasm va video signallarni filtrlash, kodlash va dekodlash sonlar ustida murakkab amallarni talab qiladi.
- Fourier transformasi, konvolyutsiya va filtrlar arifmetik operatsiyalar orqali amalga oshiriladi.
- Floating-point arifmetika va yuqori aniqlik signal sifatini saqlashda muhim rol o'ynaydi.

Shunday qilib, **sun'iy intellekt, kriptografiya va raqamli signalni qayta ishlash** kompyuter arifmetikasiga bevosita tayanadi va uning asosiy samaradorligi arifmetik operatsiyalarning tezligi va aniqligiga bog'liq.



Foydalanilgan adabiyotlar

M.M. Aripov, M.X. Lutfillayev — "Informatika va axborot texnologiyalari" (Oliy o'quv yurtlari uchun darslik).

A. Ahmedov, N. Taylaqov — "Informatika"(Akademik litsey va kasb-hunar kollejlari uchun darslik, Toshkent, 2002).

T.S. Safarov — "Informatika va axborot texnologiyalari" (Tibbiyot oliy o'quv yurtlari uchun o'quv qo'llanma, 2006).

N. Taylaqov, A. Ahmedov — "Informatika. 10-sinf" (Maktab darsligi, 2017).

Mavzuga oid maxsus manbalar

"Kompyuterning arifmetik asoslari" — Informatika yo'nalishidagi maxsus o'quv qo'llanma (Unilibrary portalida mavjud).

Isroil Tojimamatov, Ruhshona Asqarova — "Kompyuterda axborotni qayta ishlashning arifmetik asoslari" (Ilmiy maqola va o'quv uslubiy materiallar, 2025).

"EHMda axborotni qayta ishlashning arifmetik asoslari" — Slaydlar va nazariy materiallar to'plami (Arxiv.uz).

Qo'shimcha (Rus va xalqaro adabiyotlar)

A.S. Kobaylo — "Логические основы цифровых вычислительных машин" (Minsk, 2010) — Raqamli qurilmalarning mantiqiy-arifmetik asoslari uchun fundamental manba.

S. Lysikov — "Арифметические и логические основы ЭВМ" — Kompyuter arifmetikasini chuqur tushunish uchun tavsiya etiladi.

"Computer Systems: A Programmer's Perspective" (R. Bryant, D. O'Hallaron) — Kompyuter arxitekturasini va sonlarning mashina darajasida tasvirlanishi bo'yicha jahon darajasidagi darslik.