

**O'ZBEKISTON RESPUBLIKASI OLIY VA O'RTA
MAXSUS TA'LIM VAZIRLIGI**

**Buxoro Oziq – Ovqat va Yengil Sanoat
Texnologiyalari Instituti**

“KASBIY TA'LIM” fakulteti.

“Informatika va axborot texnologiyalari” kafedrası

“EHM va tizim ishini tashkil etish” fanidan

Р е ф е р а т

Mavzu: Matematik soprocessor registrlari va uning
buyruqlar tizimi.

Bajardi: 18-07 MIIT guruh talabasi
Egamberdiyev Yoqubjon.

Qabul qildi: dots.Yo'ldoshev Sh.

Buxoro – 2010.

MUNDARIJA.

I. Kirish

.....

II. Asosiy qism

.....

a) Arifmetik soprosessorning ishlatilishi.

Soprosessorning **ishlash**
prinsipi.....

[illegible]

c) Soprosessor buyruqlar tizimi.....

d) Soprosessorni dasturlash.....

III.Xulosa

.....

IV.Foydalanilgan adabiyotlar

.....

KIRISH.

Hozirgi kompyuter texnologiyalari rivojlanib borayotgan vaqtda foydalanuvchilar kompyuter qurilmalarini mukammal bilishi va qo'shimcha qurilmalaridan to'liq foydalana olishlari talab etilmoqda. Shuni hisobga olib, mutaxassislarda kompyuter qurilmalaridan foydalanish ko'nikmalarini hosil qilish va ularni kerakli amaliy dasturlar yordamida boshqarishga keng imkoniyat yaratib berish zarurdir. Buning uchun mutaxassislar albatta, kompyuter qurilmalari va ularni dastur orqali boshqarishni bilishi kerak bo'ladi.

Kompyuterning ko'pgina tashqi qurilmalari mavjud va ular turli xil amallarni bajarishga mo'ljallangan.

Qo'shimcha qurilmalarni ham shartli ravishda 2 guruhga ajratib o'rganishimiz mumkin. Ularni:

1. Tashqi qo'shimcha qurilmalar.
2. Ichki qo'shimcha qurilmalarga ajratamiz.

Soprocessor bu kompyuterning ichki qurilmasi hisoblanib, platadagi alohida razyomga o'rnatiladi.

Uning asosiy vazifasi protsessorning ishini yengillashtirib barcha arifmetik amallarni bajarish. Intel 486DX protsessorlaridan boshlab qo'zg'aluvchan nuqtali va umuman arifmetik amallar soprocessor zimmasiga yuklandi.

O'z vaqtida Weitek kompaniyasining X86 platformasi uchun soprocessorlar ishlab chiqarish keng tarqaldi. Bu kompaniya tomonidan 1167,

2167 mikrosxemalar jamlanmasi sifatida hamda 3167, 4167 mikrosxema sifatida 8086, 80286, 80386, 80486 mikroprosessorlari uchun soproprocessorlar ishlab chiqildi.

Quyida soproprocessorning ishlash prinsipi va turlari bilan yaqindan tanishib chiqamiz.

Arifmetik soproprocessorning ishlatilishi.

Arifmetik soproprocessor (FPU) qo'zg'aluvchi sonlar hamda uzun butun sonlar ustida amallar bajarish uchun xizmat qiladi (qo'llaniladi). U haqiqiy sonlar ustida bajariladigan amallar tezligini qariyb bir necha o'n baravar oshiradi (qo'shish amalini). Soproprocessor oddiy matematik amallarni (qo'shish, ayirish, ko'paytirish, bo'lish va boshqalar) va funksiyalar qiymatini (sinus, kosinus, tangens, logarifmlar va shunga o'xshash) xisoblay oladi.

Soproprocessorning qo'llanish soxasi bu – ilmiy xisob kitoblarda, mashinali grafikada va albatta real dunyoni nusxalovchi (dunyo modeli yaratiluvchi) kompyuter o'yinlarida (sababi bu o'yinlar xam mashinali grafika xam injenerlik hisob- kitoblarini o'z ichiga oladi).

Ayni vaqtda ko'pgina kompyuterlar markaziy prosessor sifatida Intel firmasining Pentium prosessorlarini qo'llamoqda. bu prosessor o'zining matematik soproprocessorlariga ega.

Soproprocessorning ishlash prinsipi.

Matematik soproprocessor markaziy prosessor tomonidan ishga tushiriladi. Ishga tushgandan so'ng u barcha xisoblash ishlarini markaziy prosessor ishi bilan parallel ravishda mustaqil bajaradi. agar markaziy prosessordan buyruq kelgan vaqtda soproprocessor band bo'lsa markaziy prosessor kutish rejimiga o'tadi. Agar soproprocessor band bo'lmasa unda markaziy prosessor buyruqni yo'naltirib javobini kutmasdan ishini

daom ettiradi. Shuni yodda tutish lozimki soprosessorning xisob – kitob ishlari markaziy prosessor bilan parallel ravishda amalgam oshiriladi.

Buyruqlar prefiksi va operandlarning manzillari.

Matematik soprosessor tomonidan bajariladigan buyruqlar dasturda markaziy prosessor buyruqlari singari yoziladi. Lekin ularning barchasi markaziy prosessorning ESC buyrug'iga mos keluvchi baytlardan boshlanadi. Bunday buyruqga duch kelgan markaziy prosessor uni soprosessorga jo'natib o'z ishini keyingi buyruqdan davom ettiradi. Soprosessor buyruqlarining assemblerli mnemonikalari F xarfi bilan boshlanadi, masalan FADD, FDIV, FSUB va boshqalar.



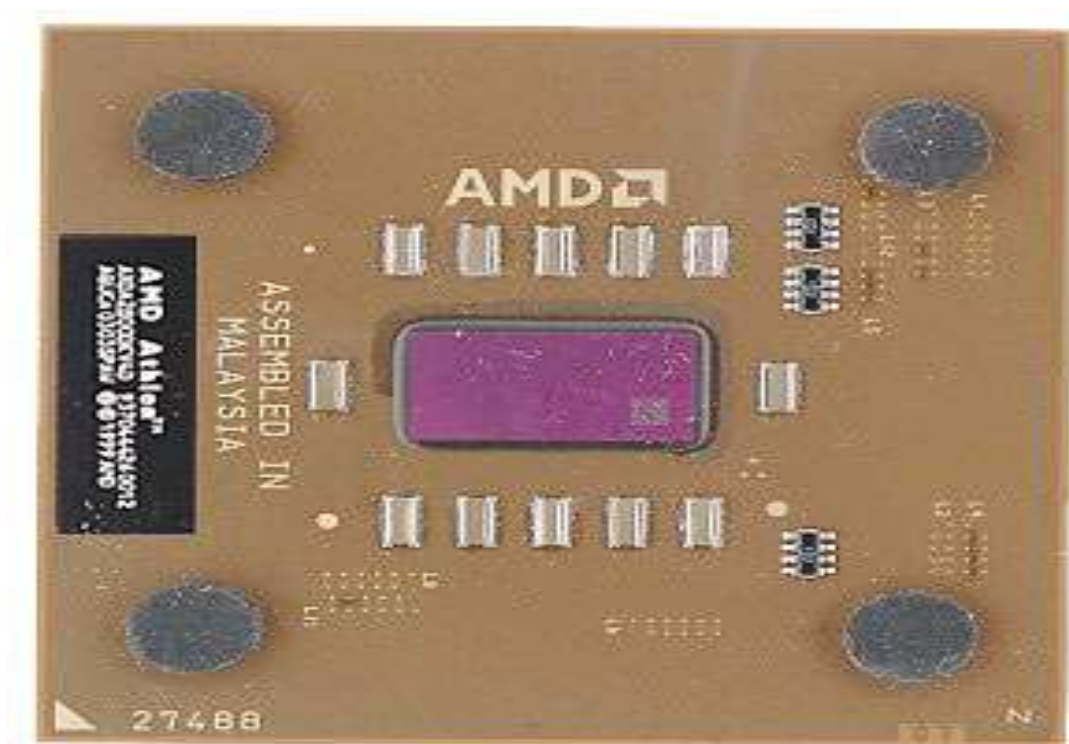
Soprosessor uyruqlari operandlarga manzillanishi mumkin, qaysiki markaziy protsessorning analogli oddiy buyruqlari. Kompyuterning asosiy xotirasidagi ma'lumotlar yoki soprosessorning ichki registrlariga operand deyiladi. Markaziy protsessor tomonidan yo'naltiriladigan xar qanday malumot soprosessor uchun buyruq vazifasini bajaradi.

Matematik soprosessor qo'zg'aluvchi nuqtali sonlar yoki butun sonlar ko'rinishidagi malumotlarni qayta ishlashi mumkin. Nuqtasi qo'g'aluvchi sonlarni umumiy ko'rinishda quyidagicha tasvirlash mumkin:

(ishora) (mantissa) *10 (ishora)(daraja)

Misol: $-1.35 \cdot 10^5$

Bu yerda: ishora – minus; mantissa – 1.35; daraja – 5; Asosiysi sonlarning ko'rinishini normallashtirish:



Agar mantissaning butun qismi nolga teng emas bitta sondan iborat bo'lsa, nuqtasi qo'zg'aluvchan son normallashtirish deyiladi. Normallashtirish sonlarni ishlatishning afzallik tomoni razryadli setkada yuqori aniqlikka ega. Bundan tashqari normallashtirish ko'rsatmalar joyidan qo'zg'alsa ma'nosi bir xil bo'lmaslikdan saqlaydi.

$$123.5678 \cdot 10^5 = 12.35678 \cdot 10^6 = 1.235678 \cdot 10^7 = 0.1235678 \cdot 10^8.$$

Yuqori darajali dasturlash tillarida nuqtasi qo'zg'aluvchi sonlarning quyidagicha ko'rinishi keltiriladi:

(ishora)(mantissa)E(ishora)(daraja)

Masalan, $-5.35 \text{ E } - 2$ bu $-5.35 * 10^{-2}$ bunday ko'rinish ilmiy natatsiya deb nomlanadi.

Arifmetik soprosessor haqiqiy sonlar bilan uch formatda ishlay oladi.

- Bir aniqlikda (4 bayt).
- Ikki aniqlikda (8 bayt).
- Kengaytirilgan aniqlikda (10 bayt).

Har qanday ko'rinishda katta bit haqiqiy sonning ishorasini bildiradi: 0 – musbat son, 1- manfiy son.

Barcha absolyut qiymatga ko'ra teng bo'lgan manfiy va musbat sonlar faqat shu bit bilan farqlanadi. Qolgan vaziyatda turli ishorali bu sonlar simmetrikdir. Manfiy sonlarni ko'rsatish uchun markaziy prosessordagidek qo'shimcha kod ishlatilmaydi.

Normallashtirilgan ikkilik sonning butun qismi doim birga teng bo'lganligi sababli bu birni saqlamasa ham bo'ladi. Matematik soprosessorni ishlab chiquvchilar bir aniqlikda va ikki aniqlikdagi formatlarda xuddi shunday yo'l tutishdi, ya'ni mantissaning butun qismi saqlanmaydi. Shunday qilib bir bit xotira hajmi tejaladi.

Aniq bo'lishi uchun quyidagi mantissani ko'rib o'tamiz:

n.nnnnnnnnnn...n

Bu yerda n simvoli yoki 0 yoki 1 orqali ifodalanadi. Normallashtirilgan sonlarning chap tomoni doimo 1 ga teng, shuning uchun sonimizni quyidagi ko'rinishda ifodalashimiz mumkin:

1.nnnnnnnnnn...n

Kengaytirilgan aniqlikdagi format soprosessor tomonidan barcha

operatsiyalarni bajarish uchun ishlatiladi. Sonlar ustidagi barcha amallarni soprocessor kengaytirilgan aniqlikda bajaradi. Bu formatda normallashtirilgan butun qismdan ortgan bit ham saqlanadi. Kengaytirilgan aniqlikning ishlatilish sababi dasturni bajarilgan amal aniqligi pasayishidan saqlash.

Daraja maydoni bu 2 ning darajalari, qaysiki mantissa unga ko'paytiriladi, va nuqtaning xarakati bir aniqlikdagi sonlar uchun 127ga teng, ikki aniqlikdagi sonlar uchun 1023ga teng va kengaytirilgan aniqlik uchun 16383ga teng.

Qo'zg'aluvchan nuqtali sonlarning absolyut qiymatini aniqlashda quyidagi formulalardan foydalansak bo'ladi:

- bir aniqlikdagi : $1.(\text{mantissa raqamlari}) * 2^{(P-127)}$

- ikki aniqlikdagi : $1.(\text{mantissa raqamlari}) * 2^{(P-1023)}$

- kengaytirilgan aniqlikdagi : $1.(\text{mantissa raqamlari}) * 2^{(P-16383)}$

Aniq bir misol keltiraylik: biz ikkilik sonda quyidagi ko'rinishga ega bo'lgan bir aniqlikdagi songa ega bo'laylik.

1 01111110 110000000000000000000000

Bu son uchun ishoraviy bit 1 ga teng (manfiy), darajasi 126 ga, mantissa 11 ga (ikkilik sanoq sistemasida) teng.

Bu sonning qiymati quyidagicha:

* 2

$(126-127) = -1.75 * 2^{-1} = -0.875$

Haqiqiy sonlar ko'rinishining alohida vaziyatlari.

- Nul – bu darajasi va mantissasi nolga teng bo'lgan son. Nul manfiy yoki musbat qiymatni qabul qilishi mumkin. Shunday qilib 2 ta: manfiy va musbat nul mavjud;

- Eng kichik musbat son bu bit darajasi nolga teng bo'lgan, daraja ko'rsatgichi birga teng, mantissa qiymati nolga teng son. Ko'rinishiga qarab eng

kichik musbat son quyidagi qiymatlarga ega: $1.17 * 10^{-38}$ (birlik aniqlikda), $2.23 * 10^{-308}$ (ikki aniqlikda), $3.37 * 10^{-4932}$ (kengaytirilgan aniqlikda);

- Eng katta manfiy son eng kichik musbat son bilan bir xil faqat bit ko'rsatgichi birga teng;

- Eng katta musbat son bu bit darajasi nolga teng bo'lgan, eng kichik bitdan tashqari barcha bitlar birga teng bo'lgan daraja maydoniga ega va mantissaning barcha razryadlarida birni o'zlashtiradigan son. Ko'rinishiga ko'ra eng katta musbat son quyidagi qiymatlarni qabul qiladi: $3.37 * 10^{38}$ (bir aniqlikda), $1.67 * 10^{308}$ (ikki aniqlikda), $1.2 * 10^{4932}$ (kengaytirilgan aniqlikda);

- Eng kichik manfiy son – eng katta musbat son bilan bir xil faqat bit ko'rsatgichi birga teng;

- Manfiy va musbat cheksizlik bu daraja ko'rsatgichida birlarni mantissa maydonida nullarni ifodalovchi sonidir. Bit holatiga ko'ra manfiy va musbat cheksizlik bo'lishi mumkin. Cheksizlik biror sonni nulgga bo'lish natijasida kelib chiqishi mumkin;

- Son emas – daraja ko'rsatkichida birlarni va mantissa maydonida istalgan qiymatni saqlovchi natija. Son emas noto'g'ri bajarilgan amallar natijasida hosil bo'ladi;

- Aniqmaslik – daraja ko'rsatgichida birlarni saqlovchi mantissa maydonida 1000 .. 0 (bir va ikki aniqlik uchun), 11000..0 (kengaytirilgan aniqlik uchun, sababi bu formatda mantissaning katta biti saqlanadi) gat eng son.

Butun sonlar.

Aritmetik soprosessor haqiqiy sonlardan tashqari butun sonlarni ham qayta ishlashi mumkin. U butun sonlarni haqiqiy va haqiqiy sonlarni butun songa

o'giruvchi buyruqlarga ega. Butun sonlarning 4 ta formati mavjud:

- Butun son;
- Qisqa butun son;
- Uzun butun son;
- Yig'ilgan o'nlik son.

Butun son 2 bayt joy egallaydi. Uning formati markaziy prossessorning formati bilan to'liq mos keladi. Manfiy sonlarni ifodalash uchun qo'shimcha kod ifodalanadi. Qisqa va uzun butun sonlar ham shu formatda lekin 4 va 8 bayt joy egallashadi.

Yig'ilgan o'nlik son 10 bayt joy egallaydi. Ushbu son 1 baytda ikkitadan joylashgan 18 ta o'nlik raqamdan iborat. Uning ishorasi eng chapki baytning katta bitida saqlanadi. Katta baytning qolgan bitlari nolga teng bo'lishi lozim.

Soprocessorning kengaytirilgan haqiqiy formatni yig'ilgan o'nlik songa aylantiruvchi buyruqlari mavjud. Agar dastur son emaslarini, cheksizliklarni va shu kabilarni yig'ilgan o'nlik formatiga o'girmoqchi bo'lsa noaniqlik yuzaga keladi.

Manfiy sonni ifodalash uchun qo'shimcha koddan foydalanamiz. Qo'shimcha kod sonning katta bitida nul saqlaydi.

0XXX XXXX XXXX XXXX

Qo'shimcha kodda musbat sondan manfiy sonni hosil qilish uchun raqamning har bir bitini teskari yozib songa birni qo'shish lozim. Masalan, +5 raqami qo'shimcha kodda quyidagi tarzda ifodalanadi:

0000 0000 0000 0101 = +5

-5 ni hosil qilish uchun har bir bitni teskarisiga o'giramiz:

1111 1111 1111 1010

Endi hosil bo'lgan raqamimizga 1 ni qo'shamiz:

$$1111\ 1111\ 1111\ 1011 = -5$$

Yig'ilgan butun son 18 razryadli. Bunday sonning har bir bayti yarim bayti (4 razryad) o'nlik sonning bitta raqamini ifodalash uchun xizmat qiladi, ya'ni 0 dan 9 gacha, shuning uchun u 0000 dan 1001 gacha qiymat qabul qilishi mumkin.

Soprosessor registrlari.

Arifmetik soprosessor 8 ta 80 bitlik raqamli registrlarga ega. Ular oraliq hisoblashlarni saqlash uchun xizmat qiladi. Registrlar boshqaruv registrlari, holat registrlari, teg registrlari, buyruqni ko'rsatuvchi va operatorlarni ko'satuvchi registrlaridan iborat.

Raqamli registrlar.

Raqamli registrlar texnik adabiyotlarda ST0 – ST7 ko'rinishida ifodalanadi. Raqamli registrlar stek sifatida ishlatiladi. Holat registri ST maydonida raqamli registr nomerini saqlaydi. Qaysiki stekning cho'qqisi hisoblanadi. Amal bajarilish vaqtida operand sifatida raqamli registr qo'llanilishi mumkin. Bu holatda buyruqda ko'rsatilgan registr nomeri holat registrning ST maydoniga qo'shiladi va shu tariqa ishlatiladigan registr aniqlanadi. Ko'pgina buyruqlar bajarilgandan so'ng holat registrning ST maydonini kengatiradi, ya'ni raqamli registrarning stekiga bajargan amallarning natijasini yozadi.

Teg registri.

Bu registr 8 ta 2 bitlik maydonga ajratilgan. Ular TAG0 dan TAG7 gacha ifodalanadi. Har bir maydon o'zining raqamli registriga ega. Teg registr maydoni o'zining raqamli registr qabul qilgan ma'lumotlarni sinflarga ajratadi:

- ❖ 00 – registr haqiqatdan nolga teng bo'lmagan ma'lumotga ega;
- ❖ 01 – registrda nul mavjud;

❖ 10 – registr haqiqiy emas sonli ma'lumotga ega – son emas, cheksizlik, noaniqlik;

❖ 11 – bo'sh registr.

Misol uchun. Faraz qiling soprosessorning barcha registrlari bo'sh edi. Keyin raqamli registrlarning stekiga haqiqiy nolga teng emas bitta qiymat kiritildi. Teg registrining ichidagi ma'lumot 16 sanoq sistemasida 3FFFH ga teng bo'ladi.

Boshqaruv registrlari.

Boshqaruv registrlari formati (bitning ifodalanishi va uning tartib raqami):

-  IM, nolinchi;
-  DM, birinchi;
-  ZM, ikkinchi;
-  OM, uchinchi;
-  UM, to'rtinchi;
-  PM, beshinchi;
-  IEM, yettinchi (Pentiumda ishlatilmaydi);
-  PC, sakkizinchi va to'qqizinchi;
-  RC, o'ninchi va o'n birinchi;
-  IC, o'n ikkinchi.

Oltinchi, o'n uchinchi, o'n to'rtinchi va o'n beshinchi bitlar ishlatilmaydi.

PC boshqaruv registrining 2 bitlik maydoni soprosessor amallarning aniq bajarishini ta'minlaydi:

- 00 – kengaytirilgan aniqlikni ishlatish. Bu rejim soprosessor yoqilgan payti o'rnatiladi;
- 10 – natijani ikki aniqlikda yaxlitlash;
- 11 – natijani bir aniqlikda yaxlitlash.

Amallar bajarilish aniqligini suni'y ravishda pasaytirish dastur ishlash tezligining oshishiga olib keladi. Aniqlik pasaytirilgan rejimlar bir va ikkilik aniqlikni qo'llovchi protsessorning emulyatsiyasi uchun ishlatiladi.

Ikki bitli RC maydoni haqiqiy sonlar ustida amallar bajarilganda ularni yaxlitlash imkonini beradi:

- 00 – yaqin songa yaxlitlash, ushbu rejim soprosessor yoqilgan payti o'rnatiladi;
- 01 – manfiy cheksizlik tomonga yaxlitlash;
- 10 – musbat cheksizlik tomonga yaxlitlash;
- 11 – nul tomonga yaxlitlash.

Xatoliklarni kamaytirish uchun yaqin songa yaxlitlash rejimini qo'llagan ma'qul.

IC boshqaruv registri maydoni cheksizlikni boshqarish uchun qo'llaniladi. U ikkita qiymatga ega: 1- afina rejimi, 0 – loyiha rejimi.

Loyiha rejimida bitta cheksizlik mavjud va u ishoraga ega emas.

Afina rejimida ikkita cheksizlik mavjud bo'lib ular manfiy va musbat. Afina rejimi cheksizliklar ustida qo'shish, ko'paytirish va shunday ko'p amallar bajarish imkonini beradi.

Holat registri.

- IE, nolinch;
- DE, birinchi;
- ZE, ikkinchi;
- OE, uchinchi;
- UE, to'rtinchi;
- PE, beshinchi;
- ES, yettinchi;

- C0, sakkizinchi;
- C1, to'qqizinchi;
- C2, o'ninchi;
- ST, on birinchi, o'n ikkinchi, o'n uchinchi;
- C3, o'n to'rtinchi;
- B o'n beshinchi.

Oltinchi bit ishlatilmaydi.

Bit C0,C1,C2,C3 bu shart kodlari. Ular buyruq bajarilishiga qarab tenglashtirish va qoldiqni topish amallarini bajaradi.

ST maydoni uch bit joyni egallaydi 11,12,13 va raqamli registr nomerini saqlaydi. Bu raqamli registr stek cho'qqisi hisoblanadi.

B – biti bandlik bitihisoblanadi. U soprocessor bror buyruqni bajarayotgan vaqtda 1ga o'rnatiladi, agar soprocessor bo'sh bo'lsa u 0 ga o'rnatiladi.

Buyruq va operandlarni ko'rstuvchi registrlar.

Bu registrlar soprocessor ish jarayonida sodir bo'ladigan aloxida xolatlar uchun mo'ljallangan.

Ximoyalangan rejimda manzil selector va qo'zg'alish nuqtasidan iborat bo'ladi. Aloxida xolat sodir bo'lgan vaqtda operativ xotiradagi operand ishlatilsa unda uning manzili operandni ko'rsatish registriga yoziladi.

Soprocessor buyruqlar tizimi.

Soprocessor buyruqlari intel firmasi markaziy processorlain buyruqlariga xo such formatda bo'lishi mumkin. Bu buyruqlar

- operativ xotiraga murojaat;
- mavjud registrlardan biriga murojaat;
- va operandsiz bo'lgan buyruq;

Xotiraga murojaat buyruqlari uzatilish manziliga qarab 4 dan 8 bayt gacha joy

egallashi mumkin. Soprosessorning barcha asseblerli mnemonikalari F xarfi bilan boshlanadi shu sababli ularni markaziy prosessorning buyruqlaridan ajratib olish juda oson.

Soprosessorning buyruqlarini bir necha guruxga bo'lish mumkin.

- ma'lumotlarni uzatish va qabul qilish buyruqlari;
- arifmetik buyruqlar;
- sonlarni tenglashtirish buyruqlari;
- transendent buyruqlar;
- boshqaruvchi buyruqlar;

Ma'lumotlarni uzatish va qabul qilish buyruqlari - operativ xotiradan ma'lumotlarni registrga ko'chirish, registrdan operativxotiraga ko'chirish, ma'lumotlarni bir registrdan boshqasiga nusxalash uchun mo'ljallangan.

Arifmetik buyruqlar - esa qo'shish, ayirish, ko'paytirish, bo'lish, kvadrat ildiz olish, qoldiqni topish, yaxlitlash va shunga o'xshash buyruqlarni bajarish uchun mo'ljallangan.

Sonlarni tenglashtirish buyrug'i - butun va xaqiqiy sonlarni tenglashtirish ularni analiz qilish uchun mo'ljallangan.

Transendent buyruqlar – turli trigonometrik funksiyalar, logarifmik, ko'rsatgichli va giperbolik funksiyalarni – sin, cos, tg, va shunga o'xshash larni hisoblash uchun mo'ljallangan.

Buyruqlarning so'ngi guruxi boshqaruvchi buyruqlar – bu buyruqlar arifmetik soprosessor ish rejimini o'rnatilishini taminlaydi. Soprosessorni ximoyalangan ish rejimiga o'tkazish uni bekor qilish va shunga o'xshash boshqa ishlarni bajaradi.

Ma'lumotlarni uzatish va qabul qilish buyruqlari

Stekga yuklash buyruqlari (Fpu Load);

- FLD xotiradan stek cho'qqisiga ST(0) xaqiqiy sonlarni yuklaydi;

- FILD xotiradan stek cho'qqisiga ST(0) butun sonlarni yuklaydi;
- FBLD xotiradan stek cho'qqisiga ST(0) ikkilik – o'nlik sonlarni yuklaydi.

Bu buyruqlar bajarilish jarayonida operand operativ xotiradan o'qilib, “kengaytirilgan aniqlik” formatiga aylantiroladi. Shundan so'ng xolat registrining ST maydoni 1 gacha qisqaradi va operand registrga yoziladi. Yani operand registrlar stekiga yoziladi, stekni ko'rsatuvchi esa (ST maydoni) 1gacha qisqaradi. Bu buyruq o'z xarakati bilan markaziy prosessorning PUSH buyrug'ini eslatadi. Registrni yuklashdan oldin ining TAG0 maydoni tekshiriladi. Agar uning qiymati 11(bo'sh registr) ga teng bo'lmasa unda xolat registrda EI flagi (mumkin bo'lmagan jarayon) o'rnatiladi, va uzilish xosil qilinadi (agar boshqaruv registrda IM- maskasi o'rnatilmagan bo'lsa).

Stekda saqlash va undan chiqarish buyruqlari (Fpu Store and Pop):

- FSTP xotira -> xaqiqiy format ;
- FISTP xotira -> butun format;
- FBSTP xotira -> o'nlik format;

Bu buyruqlar avval stek cho'qqisini xotirada saqlaydi, so'ng malumotlarni cho'qqidan siqib chiqaradi (ma'lumotlarni o'chiradi). Nomeri xolat registrining ST maydoni orqali ma'lum bo'ladigan regist ichidagi ma'lumot kerakli formatga aylantiriladi va buyruq operandi tomonidan berilgan operativ xotiraning yacheykasiga yoziladi. Bu xarakatlar markaziy protsessorning POP buyrug'iga o'xshash.

Formatning o'zgarishi buyruqga (FSTP, FISTP yoki FBSTP) bog'liq ravishda amalga oshiriladi. FSTP va FISTP buyruqlari uchun malumot formatini o'zgartirish davrida boshqaruv registrining RC maydonidagi malumotga binoan yaxlitlanadi. FBSTP buyrug'I uchun yaxlitlash doim bir xil amalgam oshiriladi ya'ni 0.5 qo'shilib so'ng kasr qismi tashlab yuboriladi.

Malumotlarni nusxalash buyruqlari (Fpu STore):

- FST xotira -> xaqiqiy format;
- FIST xotira -> butun format;
- FBST xotira -> o'nlik format;

Bu buyruqlar ma'lumotlarni stek cho'qqisidan buyruq operandi tomonidan ko'rsatilgan xotira maydoniga jo'natadi. Bunda stek ko'rsatgichi (ST maydoni o'zgarishsiz qoladi). Malumot operativ xotiraga yozilish vaqtida mos formatga almashinuv sodir bo'ladi (FST uchun xaqiqiy, FIST uchun butun, FBST uchun o'nlik).

Ma' lumot almashinuv buyruqlari (Fpu eXCHange)

- FXCH stek choqqisi ST(0) malumoti bilan buyruq operandi sifatida ko'rsatilgan registr malumotlarini almashtirish.

O'zgarmaslarni yuklash buyruqlari:

- FLDZ 0 -> ST(0)- nulni yuklash;
- FLD1 1 -> ST(0)- birni yuklash ;
- FLDP1 -> ST(0)- π sonini yuklash;
- FLDL2T loge10 -> ST(0)- loge10 ni yuklash;
- FLDL2E log2e -> ST(0)- log2e ni yuklash;
- FLDLG2 log102 -> ST(0)- log102 ni yuklash;
- FLDLN2 loge2 -> ST(0)- loge2 ni yuklash;

O'zgarmaslarni yuklash operativ xotiradan malumotlarni yuklash buyruqlaridan tezroq amalga oshiriladi.

Arifmetik buyruqlar

Soprocessor asosan olti tipdagi buyruqlarni qo'llaydi.

- Fxxx

Birinchi operand stek cho'qqisidan olinadi, ikkinchisi – stekning keyingi elementi. Buyruqning bajarilish natijasi stekga yoziladi.

- Fxxx xotira

Asos sifatida xotira olinadi, stek cho'qqisi ST(0) esa o'zlashtiruvchi. ST stek ko'rsatgichi o'zgarmaydi, buyruq faqat bir va ikki aniqlikdagi operandlar uchun o'inli.

- Fixxx xotira

Yuqoridagi buyruq tiplariga o'xshash lekin bunda operand sifatida 16 yoki 32 razryadli butun sonlar olinishi mumkin.

- Fxxx ST, ST(i)

Bu tip uchun asos sifatida ST(i) registri olinadi, o'zlashtiruvchi sifatida esa stek cho'qqisi olinadi. Stek ko'rsatgichi o'zgarmaydi.

- Fxxx ST(i), ST

Bu tip uchun asos sifatida ST(0) registri olinadi, o'zlashtiruvchi sifatida esa ST(i) registri olinadi. Stek ko'rsatgichi o'zgarmaydi.

- FxxxP ST(i), ST

ST(i) registri o'zlashtiruvchi, ST(0) registri asos. Buyruq bajarilgandan so'ng asos ST(0) stekdan chiqariladi.

“xxx” maydoni quyidagi qiymatlarni qabul qilishi mumkin:

- ADD qo'shish
- SUB ayirish
- SUBR teskari ayirish, kamayuvchi bilan ayriluvchi joylarini almashadi
- MUL ko'paytirish
- DIV bo'lish
- DIVR teskari bo'lish

Asosiy arifmetik amallardan tashqari qo'shimcha arifmetik amallar mavjud:

- FSQRT kvadrat ildiz olish
- FSCALE ikkining darajalariga oshirish
- FPREM qoldiqni xisoblash
- FRNDINT butun songacha yaxlitlash

- FXTRACT sonning darajasini va mantissasini ajratish
- FABS sonning absolyut qiymatini xisoblash
- FCHS sonning ishorasini o'zgartirish

FSQRT buyrug'iga asosan topilgan sonning kvadrat ildizi stek cho'qqisiga ST(0) ga yoziladi.

FSCALE buyrug'i ST(0) da joylashgan sonning darajasini o'zgartiradi. Bu buyruqga asosan ST(0) dagi sonning darajasining ishorasi taxminan ST(1)ga yozilgan daraja koeffitsiyenti bilan qo'shiladi. Bu buyruqning xarakterini quyidagi formula orqali ifodalash mumkin.

$$ST(0) = ST(0) * 2^n, -215 \leq n \leq +215$$

Bu formulada n- ST(1) gat eng.

FPREM buyrug'i ST(0) ni ST(1) ga bo'lganda xosil bo'ladigan qoldiqni xisoblaydi. Natijaning ishorasi ST(0) ning ishorasiga teng bo'lib, natija ST(0) stek cho'qqisida xosil bo'ladi. Buyruq bajarilgandan so'ng C2 xolat registri flagi quyidagi qiymatlarni qabul qilishi mumkin:

- 0 – bo'lish natijasida xosil bo'lgan qoldiq. ST(0) dan olingan, ST(1) bo'luvchidan kichik. Buyruq to'liq tugatildi.
- 1- ST(0) qisman qoldiqga ega. Dastur qoldiq xaqida aniq javob olish uchun buyruqni ya'na bajarishi kerak.

RNDINT buyrug'i boshqaruv registrining RC maydonidagi ma'lumotga asosan ST(0) ni yaxlitlaydi.

FABS buyrug'i ST(0) ning absolyut qiymatini xisoblaydi.

Sonlarni taqqoslash buyruqlari

Markaziy prosessorda shartli o'tish buyruqlari prosessor registrining aloxida bitli flaglari o'rnatmalari asosida amalga oshiriladi. Arifmetik soprosessorda maxsus taqqoslash buyruqlari mavjud bo'lib, ularning bajarilishi natijasida xolat registrida

shart biti kodlari o'rnatiladi:

FCOM- taqqoslash;

FICOM- butun qismini taqqoslash;

FCOMP- taqqoslash va stekdan chiqarish;

FICOMP- butun qismini taqqoslash va stekdan chiqarish;

FTST- operandni nol bilan taqqoslash;

FXAM- operand taxlili;

Taqqoslash buyrug'i operandini "x" xarfi bilan belgilab olaylik. Quyidagi jadvalda "FCOMx" buyrug'i bajarilgandan keyin shart biti kodlari qabul qilishi mumkin bo'lgan qiymatlarni keltiramiz:

$C3 = 0, C0 = 0 - ST(0) > x$

$C3 = 0, C0 = 1 - ST(0) > x$

$C3 = 1, C0 = 0 - ST(0) = x$

$C3 = 1, C0 = 1 - ST(0)$ va x larni taqqoslab bo'lmaydi.

Eng so'ngi kombinatsiya sonmaslarni, noaniqliklarni, cheksizliklarni va boshqa bazi xollarni taqqoslamoqchi bo'lingan vaziyatda yuzaga keladi.

FICOM buyrug'i operand sifatida 16 yoki 32 razryadli sonlarni qabul qiladi, qolgan xollarda u FCOM buyrug'i bilan bir xil.

FCOMP va FICOMP buyruqlari FCOM va FICOM buyruqlari bilan faqat bajarib bo'lingandan so'ng operandning stekdan siqib chiqarilishi bilan farq qiladi.

FCOMPP buyrug'i xam FCOM buyrug'i bilan bajarilishi bir, lekin buyruq bajarilgandan so'ng FCOMPP da ikkita operand xam stekdan siqib chiqariladi.

Operandni nol bilan taqqoslash uchun FTST buyrug'i qo'llaniladi. Buyruq bajarilgandan so'ng shart kodlari quyidagicha o'zgaradi:

$C3 = 0, C0 = 0 - ST(0) > 0$

$C3 = 0, C0 = 1 - ST(0) > 0$

$C3 = 1, C0 = 0 - ST(0) = 0$

$C3 = 1, C0 = 1 - ST(0)$ va x larni taqqoslab bo'lmaydi.

FMAX buyrug'i $ST(0)$ dagi ma'lumotni taxlil qiladi. U bajarilgandan so'ng shart kodlari o'rnatiladi, qaysiki ularga binoan sonning ishorasini, uning cheksizmas yoki cheksizligini, normallashtirishini va boshqalarni aytishimiz mumkin.

$C1$ biti taxlil qilinayotgan sonning ishorasini saqlaydi: 0 – musbat, 1- manfiy.

$C0$ biti orqali sonning cheksiz yoki chekli ekanini aniqlash mumkin: 0 – chekli, 1- cheksiz.

Chekli sonlarning klasifikatsiyasi $C2$ va $C3$ dagi shart kodlari orqali amalga oshirili mumkin:

$C3=0, C2=0$ - normallashtirilmagan son;

$C3=0, C2=1$ – normallashtirilgan son;

$C3=1, C2=0$ – nol

$C3=1, C2=1$ – son normal xolatdancheqarildi.

Cheksiz sonlar uchun esa $C2$ va $C3$ shart kodlari quyidagi qiymatlarga ega:

$C3=0, C2=0$ – son emas;

$C3=0, C2=1$ – cheksiz son;

$C3=1, C2=0$ – bo'sh son;

$C3=1, C2=1$ – bo'sh son;

“FSTSW AX” buyrug'i yordamida dastur soprosessorning xolat registridagi ma'lumotni markaziy prosessorning registriga ko'chirishi mumkin. Yana AH registridagi ma'lumotlarni markaziy prosessorning flag registrlariga “SAHF” buyrug'i yordamida ko'chirishimiz mumkin. Soprosessorning shart biti kodlari markaziy prosessorning flag registrlarida shunday akslanadiki, shartli o'tishlarni qo'shimcha xarakatlarsiz amalgam oshirish imkonini beradi.

Transenent buyruqlar

Transendent buyruqlar quyidagi funksiyalarni xisoblash uchun mo'ljallangan:

- trigonometrik (sin, cos, tg,...)
- teskari trigonometrik (arcsin, arccos,...)
- korsatgichli (xy , $2x$, $10x$, ex ,...)
- giperbolik (sh, ch, th,...)
- teskari giperbolik (arsh, arch, arcth,...)

Matematik soprosessorning barcha transendent buyruqlari:

- ~ FPTAN tangensni xisoblash
- ~ FPATAN arktangensni xisoblash
- ~ FYL2X $y \cdot \log_2(x)$ ni xisoblash
- ~ FYL2X1 $y \cdot \log_2(x+1)$ ni xisoblash
- ~ F2XM1 $2x-1$ ni xisoblash
- ~ FCOS $\cos(x)$ ni xisoblash
- ~ FSIN $\sin(x)$ ni xisoblash
- ~ FSINCOS $\sin(x)$ va $\cos(x)$ ni bir vaqtda xisoblash.

FPTAN buyrug'i ST(0)ning tangensini stekga ikkita son x va y ni joylashtirish orqali xisoblaydi, bu yerda $y/x = \text{tg}(\text{ST}(0))$. Buyruq bajarilgandan so'ng y soni ST(0) dan o'rin egallaydi, x esa ST(1) dan joy oladi. FPTAN buyrug'ining argumenti $0 \leq \text{ST}(0) \leq \pi/4$ oraliqda bo'lishi kerak.

Tangensning xosil bo'lgan qiymatidan foydalanib boshqa trigonometrik funksiyalarning qiymatini xisoblashimiz mumkin:

$$\sin(z) = 2 \cdot (y/x) / (1 + (y/x)^2)$$

$$\cos(z) = (1 - (y/x)^2) / (1 + (y/x)^2)$$

$$\text{tg}(z/2) = y/x$$

$$\text{ctg}(z/2) = x/y$$

$$\text{cosec}(z) = (1 + (y/x)^2) / 2 \cdot (y/x)$$

$$\sec(z) = (1+(y/x)^2) / (1-(y/x)^2)$$

Bu yerda z- FPTAN buyrug'i bajarilguncha ST(0) dagi qiymat. X va Y lar esa ST(0) hamda ST(1) registrlardagi qiymat.

FPATAN buyrug'i esa arktangensni xisoblaydi:

$$z = \arctg(ST(0)/ST(1)) = \arctg(x/y)$$

Buyruq bajarilishidan oldin x va y qiymatlar ST(0) hamda ST(1) registrlarda saqlanadi. FPATAN ning argumenti $0 < y < x$ oraliqda bo'lishi kerak. Natija ST(0) ga yoziladi.

FYL2X buyrig'i $y \cdot \log_2(x)$ ko'rinishidagi ifodalarni xisoblaydi. Operandlar ya'ni x va y ST(0) hamda ST(1) registrlarda saqlanadi. Operandlar stekdan olinadi, va unga natija yoziladi. Bu yerda x ning qiymati musbat bo'lishi kerak. Bu buyruqning natijasidan foydalanib logarifmik funksiyalarni qiyidagicha xisoblash mumkin:

Ikki asosli logarifm: $\log_2(x) = \text{FLY2}(x)$

Natural logarifm: $\log_e(x) = \log_e(2) \cdot \log_2(x) = \text{FYL2X}(\log_e(2), x) = \text{FYL2X}(\text{FLDLN2}, x)$

O'nlik logarifm: $\log_{10}(x) = \log_{10}(2) \cdot \log_2(x) = \text{FYL2X}(\log_{10}(2), x) = \text{FYL2X}(\text{FLDLG2}, x)$

Boshqaruvchi buyruqlar.

Bu buyruqlarning mnemonikalari FN va F bilan boshlanishim mumkin. Birinchi holatimiz "kutishsiz" bo'lgan buyruqlarga to'g'ri keladi. Bunday buyruqlar uchun prosessor soproessorning band yoki band emasligini tekshirib o'tirmaydi. "kutish xolatili" buyruqlar ham oddiy buyruqlar singari ishlaydi.

Soproessorning boshqarish buyruqlari ro'yxati:

- FNSRCW (FSTCW) boshqaruvchi so'zni yozish (boshqaruv registridagi ma'lumotni operativ xotiraga yozadi).
- FLDCW boshqaruvchi so'zni yuklash (boshqaruv registrini operativ xotiradagi

ma'lumot bilan to'ldiradi, va soposessor ish rejimini o'zgartirish uchun ishlatiladi).

- FNSTSW (FSTSW) xolat so'zini yozish (xolat registridagi makumotni operativ xotiraga yozadi).
- FNSTSW AX (FSTSW AX) xolat so'zini AX ga yozish (xolat registri ma'lumotini markaziy prosessorning AX registriga yozadi. U yerda ma'lumotni shartli o'tishlar orqali taxlil qilish mumkin).
- FNCLEX (FCLEX) aloxida xolatlarni tashlab yuborish (xolat registridagi aloxida xolatlar flagini tashlab yuborish).
- FNSAVE (FSAVE) to'liq xolatni yozib olish.
- FRSTORY to'liq xolatni qayta tiklash.
- FINCSTP SP steki ko'rsatgichini 1 ga oshirish
- FDECSTP SP steki ko'rsatgichini 1 taga kamaytirish
- FFREE registrni bo'shatish
- FNOP hech qanday amal bajarmaydi (холостая оператсия)
- FSETPM ximoyalangan ish rejimini jore qilish (o'rnatish). Soprosessorni ximoyalangan ish rejimiga o'tkazadi.

Soprosessorni dasturlash

Dasturlashning yuqori darajadagi tillarida tashkil qilingan turli muxitlarda ishlagan vaqtimizda (Si yoki Paskal tillarida) odatda 3 ta standart kutubxonadan birini tanlash imkoniyati bo'ladi (istalgan proyektda):

- Emulyator kutubxonisi (Pentium oilasidagi prosessorlar uchun o'zining muximligini yo'qotdi, va MMX paydo bo'lishi bilan ya'na mavqesini tikladi)
- Arifmetik soprosessorni qo'llovchi kutubxona
- Alternativ matematika kutubxonasi

Demak birinchi variantimiz (emulyator kutubxonisi) dastur yozish boshlangan vaqtda o'zi avtomatik tarzda tanlangan turadi. Bu kutubxonani qo'llovchi dasturlar tizimda soprocessor bor bo'lsa xam yo'q bo'lsa xam ishlayveradi. Xisoblashning oxirgi xolatlarida ya'ni qo'zgaluvchan nuqtali sonlar bilan ishlashda maxsus qism dasturlardan foydalaniladi qaysiki ular dasturingizga linkovka vaqtida kelib qo'shiladi (bu esa o'z navbatida dastur xajmining kengayishiga olib keladi). Bundan tashqari dastur kodiga maxsus, soprocessor bor yoki yo'qligini aniqlab qanday yo'l bilan xisoblashlarni amalga oshirishni ko'rsativchi (soprocessor yordamida, u bo'lmasa maxsus qism dasturlar yordamida) funksiya qo'shiladi.

Afsuski shunday dasturlar xam borki ularda emulatsiya kutubxonasini qo'llashning

iloji yo'q yoki qiyin masala:

- resident dasturlarda;
- drayverlarda;
- xisoblashlarning aniqligi va tezligiga qattiq shart qo'yadigan dasturlarda;

Kutubxonaning ikkinchi varianti soprocessor bor xolatda ishlashga mo'ljallangan.

Agar soprocessor bo'lmasa dastur ishlamaydi. Agar soprocessor borligi aniq bo'lsa (misol uchun PENTIUM oilasi doim arifmetika blokiga ega), bu variantdan eng tezkor variant sifatida foydalanishimiz mumkin.

Uchinchi variantimiz soprocessorni umuman ishlatmaydi (xatto u sistemada mavjud bo'lsa xam). Barcha ishlarni alternativ kutubxona tarkibidagi va dasturga linkovka etapida qo'shiluvchi qism dasturlar amalga oshiradi.

Resident dasturlarda emulyatsiya kutubxonasini qo'llab bilmasligimizning sababi shundaki bu dasturni xotirada qoldirishimiz bilan u , misol uchun _dos_keep funksiyasi, emulyatsiya modullariga ulanishni yo'qotadi.

Emulyatsiya dasturlarini chaqirish mexanizmi 34h va 3Eh uzilishlarini qo'lashga

asoslanadi. Dasturni resident tarzda qoldirishdan oldin `_dos_keep` funksiyasi ko'rsatilgan vektorli uzilishlarni tiklaydi va shu bilan resident dasturning emulyatsiya modellari bilan aloqasini uzib qo'yadi. Umuman bu bodullar xam endi xotirada yo'q – ularning o'rniga yangi dastur yuklangan bo'lishi mumkin. Shu sababli Si dasturlash tili qo'llanmasida resident dasturlar uchun alternative matematika kutubxonasini qo'llash tavsiya etiladi. Lekin ular matematik soprosessorni ishlatmaydi.

Drayver dasturlar xam shunday. Drayverlar odatga ko'ra assembler tilida yoziladi shu sababli Si emulyatsiya kutubxonasi ularga o'rinli emas.

Bu vaziyatdan chiqishning bir yo'li soprosessorni assembler tilida dasturlash bo'lishi mumkin. Shu qatorda siz soprosessorning barcha imkoniyatlaridan foydalanishingiz va xisoblashlarning yuqori aniqligiga erishishingiz mumkin. Assembler tilida dastur yozish uchun MASM va TASM kompilyatorlaridan foydalanishimiz mumkin. Yoki assembler dasturlash tilini o'zida saqlagan integrallashgan dastur muxitidan foydalanishimiz mumkin.

Soprosessor buyruqlarini biror bir soprosessor registrlarini ko'rish imkonini beruvchi otladchik (debugger) yordamida o'rganish qulayroq. Buning qulayligi birinchidan u dasturni xar bir buyriqdan so'ng to'xtab qadam-ba qadam bajarishi mumkin. Ikkinchidan xaqiqiy sonlarni istalgan vaqtda ko'rish imkoniyati mavjud. Shu maqsad bilan yuqori darajali tillarda assemblerli o'rnatmalardan foydalanishimiz mumkin. Chunki bu tillar ekranda turli ma'lumotlarni ko'rsatish uchun funksiyalar to'plamiga egaligi sababli bu funksiyalardan natijani ko'rsatish maqsadida foydalanish mumkin. Arifmetik soprosessor bilan ishlash uchun esa assemblerli koddan foydalanamiz. Bu xolatimizda otlatchikdagidek soprosessorning ishchi registrlarini ko'rish oson emas.

Aloxida xolatlarni qayta ishlash

Arifmetik soprosessorda turli buyruqlarni bajarish vaqtida yuzaga keluvchi

xatoliklarni qayta ishlovchi ikkita mexanizm bor.

Birinchi mexanizm aloxida xolat uzulishlarini qayta tiklash mexanizmiga asoslangan (INT 10h). Bu uzulish biror bir xatolik sodir bo'lgan vaqtda (masalan nolga bo'lish tanijasida) vujudga keladi, bu xolda boshqaruv registrning aloxida xolat biti maskalari o'rnatilmagan bo'ladi.

Ikkinchi xolatda esa barcha aloxida xolatlar niqoblanadi (boshqaruv registrining tegishli bitlari birga 1ga o'rnatiladi), va xatolik paytida soprosessor avvaldan ma'lum bo'lgan aloxida qiymatni qaytaradi (sonmaslarni, noaniqlik yoki cheksizlik).

Dasturchi bu qayta ishlashlardan birini tanlab ishlashi mumkin. Agar aloxida xolat uzilishi niqoblangan bo'lsa , bunday xatolikni topish usulini qiyidagicha ishlatish mumkin:

1. Xolat registridagi aloxida xolatlar registrini tashlab yuborish.
2. Soprosessorning bir yoki bir nechta buyrug'ini bajarish.
3. Xolat registridagi aloxida xolatlar flaglari xolatini tekshirish. Shu jumladan summar xatolik biti ES ni.
4. Agar biror flag o'rnatilgan bo'lsa xatoliklarniqayta ishlovchi dasturni chaqirish.
5. Xatoliklarni qayta ishlash dasturi orqali xolatlar registriga tegishli qiymatlar kirgizilgan xolda flaglar tashlab yuborish mumkin.

Soprosessor ishi natijasining aloxida qiymatlari

Soprosessor biror bir xisoblashlarni amalga oshiruvchi buyruqni bajargandan so'ng, xosil bo'lgan natijani aloxida qiymatlarga tegishlilikini tekshirish zarur bo'ladi.

Noaniq natija

Bazi bir amallar bajarilishi natijasida javobni aniq tasavvur qilib bo'lmaydigan xolatlar yuzaga kelishi mumkin. Misol uchun 1.0 ni 3.0 ga bo'lganimizda 0.01010101... ikkilik kasr (yoki o'nlik sanoq sistemasida 0.33333... cheksiz kasr) xosil bo'ladi. Bunday son biror bir xaqiqiy sonlar formatida aniq ko'rsatishi mumkin

emas. Odatda noaniq natija yaxlitlab olinadi va xatolik sifatida qabul qilinmaydi.

To'lib ketish

Agar natija juda katta va qabul qilgich formatiga o'girib bo'lmaydigan bo'lsa to'lib ketish xolati kuzatiladi. Bu aloxida xolat aniq sodir bo'ladi qachonki kengaytirilgan aniqlikdagi maksimal sonni o'zini o'ziga qo'shganda, bu formatni ikki yoki bir aniqlikdagi formatga aylantirganda. Oraliq natijalarni saqlash uchun 80 –bir ishlatilganligi sababli bir yoki ikki aniqlikdagi sonlar ustidagi amallar bajarilganda bunday xolat sodir bo'lmaydi. Kengaytirilgan aniqlikdagi sonning katta diapozoni hajm jixatdan katta operatsiya natijalarini aniq tasavvur etish imkonini beradi.

To'lib ketishni oldini olish

Bu xolat natija qabul qilgich formatida ko'rinishi uchun juda kichilik qilsa (lekin noldan farq qilsa) yuzaga keladi. Misol uchun eng kichik kengaytirilgan aniqlikagi musbat sonni ikkilik yoki bir aniqlikdagi songa aylantirmoqchi bo'lganimizda sodir bo'ladi. Agar siz faqat ikki yoki bir aniqlikdagi sonlarni, oraliq natijani saqlash uchun esa kengaytirilgan aniqlikdagi sonlardan foydalansangiz unda bu xodisa ro'y bermaydi.

NUL ga bo'lish

Bu aloxida vaziyat chekli nolli sonni nulga bo'lmoqchi bo'lganimizda sodir bo'ladi. Afina rejimida chekli sonlarni (musbat yoki manfiy) nulga bo'lganimizda natija sifatida bizga cheksizlik qaytariladi.bu cheksizlikning ishorasi bo'linuvchining va nulning ishorasiga bog'liq. Misol chun musbat chekli sonni musbat nulga bo'lsak musbat cheksizlik xosil bo'ladi. Musbat ishorali chekli sonni manfiy nolga bo'lsak – manfiy cheksizlik xosil bo'ladi. Proyekt rejimida bunday xolda xamda nolni nolga bo'lmoqchi bo'lganimizda noto'g'ri amal aloxida xolati yuzaga keladi.

Noto'g'ri amal

Bu aloxida xolat mumkin bo'lmagan amallar (nolni nolga bo'lish, manfiy sondan ildiz chiqarish, soproessorning mavjud bo'lmagan registriga murojaat, transcendent funksiyalarda, buyruq operandi sifatida sonmaslarni, noaniqlik yoki cheksizlikni ishlatmoqchi bo'lganimi)ni bajarmoqchi bo'lganimizda yuzaga keladi.

Normal xolatdan chiqarilgan operand

Yuqorida soprocessor operandlarning norma xolatini qo'llashi aytib o'tilgandi. Ammo amal bajarilgandan so'ng natijaning absolyut qiymati uni normal holatda ifodalash uchun juda kichik bo'lishi mumkin. Bunday javobni qabul qilmasligimiz xam mumkin lekin bu xisoblashlar aniqligining pasayishiga va xatto qo'pol xatolar kelib chiqishiga sabab bo'lishi mumkin.

Xulosa.

Xulosa qilib aytganda hozirgi rivojlanib kelayotgan davrda har bir mutaxassisdan kompyuterning tashqi va ichki qurilmalarini bilish talab etilmoqda. Kompyuterda ishlash jarayonimizda har bir qurilmaning

imkoniyatlari bilan chuqurroq tanishib chiqamiz. Men ham mavzuni o'rganish jarayonida soproessorning ishlash prinsipi, uning turlari, undan foydalanish usullari, soproessorning afzallik tomonlarini chuqur o'rganib chiqdim. Soproessorning registrlarini, uning buyruqlarini, registrlar vazifalarini bilib oldim.

Matematik soproessor markaziy protsessorning buyruqlarining imkoniyatlarini oshirish uchun va integral modulga ega bo'lmagan protsessor qo'zg'aluvchan nuqtali sonlar ustida amallar bajarishini ta'minlovchi qurilma ekanligini angladim.

Bu bilan men bilimlarimni mustahkamlab, bilmagan tushunmagan tushunchalarni o'rganib oldim.

Bu bilib olgan bilimlarim kelajakda albatta menga as qotishiga ishonaman. Kelajakda kim qanaqa kadr bo'lib yetishib chiqishini oldindan aniq bilmaydi, lekin shuni aniq bilamanki hozir o'rgangan bilimlarimiz bir umrga yodimizda qoladi.

Foydalanilgan adabiyotlar.

1. <http://www.google.ru/>
2. <http://www.ziyonet.uz/>
3. “EHM ning periferik qurilmalari” fanidan: ma’ruzalar matni
Asrayev Z.R. Buxoro – 2006.

