Z80 vs. MCS-48/8048/8035/8048

Turinys

[Įvadas 2](#_Toc185170613)

[Z80 ir MCS-48 architektūrų apžvalga 2](#_Toc185170614)

[Z80 architektūra 2](#_Toc185170615)

[MCS-48/8048 architektūra 3](#_Toc185170616)

# Įvadas

Kompiuterių architektūra – tai viena iš pagrindinių informatikos ir elektronikos sričių, nagrinėjanti, kaip suprojektuoti ir valdyti kompiuterinius įrenginius siekiant efektyviai vykdyti užduotis. Šiame darbe analizuojamos dvi reikšmingos mikroprocesorių architektūros – Zilog Z80 ir Intel MCS-48 (taip pat žinoma kaip 8048 serija). Šios architektūros buvo sukurtos XX amžiaus pabaigoje ir tapo reikšmingu technologijų vystymosi etapu, turinčiu įtakos daugybei pramonės sričių.

Z80 ir MCS-48 architektūros atstovauja skirtingoms koncepcijoms ir taikymo sritims. Z80 buvo plačiai naudojamas ankstyvuosiuose asmeniniuose kompiuteriuose ir įterptosiose sistemose, o MCS-48 dažniausiai rado pritaikymą pramoniniuose valdikliuose ir paprastesnėse įterptinėse sistemose. Abi šios architektūros išsiskyrė savo novatoriškumu ir funkcionalumu, tačiau jų techniniai sprendimai, našumas ir pritaikymas smarkiai skyrėsi.

Šio darbo tikslas – išsamiai palyginti Z80 ir MCS-48 architektūras, išryškinti jų techninius skirtumus, veikimo ypatybes ir tinkamumą skirtingoms taikymo sritims. Taip pat aptarsime šių architektūrų istorinę reikšmę ir jų dabartinį statusą technologijų srityje.

Darbas suskirstytas į kelias dalis: pirmiausia apžvelgiama kiekvienos architektūros istorija ir techniniai bruožai, vėliau pateikiamas jų palyginimas pagal esminius kriterijus, tokius kaip našumas, atminties valdymas ir instrukcijų rinkiniai. Galiausiai aptariamas jų poveikis tolimesnei kompiuterių architektūrų raidai bei pritaikymas šiandieninėje technologinėje aplinkoje.

# Techninis ir istorinis kontekstas

**Z80**1969 m. viena Japonijos įmonių ,,Busicom“ kreipėsi į ,,Intel“, kad šie jiems suteiktų elektroninio stalinio stalinio skaičiuotuvo lustus. ,,Intel“ atsakė siūlydama, kad skaičiuotuvo centre galėtų būti naudojamas vieno lusto apibendrintas skaičiavimo variklis. Gautas rezultatas buvo pirmasis pasaulyje mikroprocesorius (angl. microprocessor) – 4004. Nors jis buvo pagrįstas daug didesnių pagrindinių ir mini kompiuterių idėjomis, 4004 buvo sumažintas, kad tilptų į 16 kontaktų lustą (didžiausia tuomet). Tai lėmė, jog 4004 modelio duomenų ir adresų magistralė buvo tik 4 bitų pločio. Intel toliau tobulino dizainą ir pagamino: 4040 (patobulino 4 bitų dizainą), 8008 (pirmasis 8 bitų mikroprocesorius), o 1974 m. – 8080. 8080 pasirodė esąs labai naudingas ir populiarus dizainas, todėl buvo sukurtas pirmasis namų kompiuteris Altair 880 ir CP/M operacinė sistema.

1975 m. Federico Faggin, dirbęs ,,Intel” įmonėje ir prisidėjęs prie 4004 mikroprocesoriau kūrimo su savo kolega Masatoshi Shima paliko ,,Intel“ ir įkūrė bendrą įmonę ,,Zilog“. Naujoje įmonėje Faggin ir Shima sukūrė mikroporcesorių, kuris buvo suderinimas su Intel 8080 (vykdė visas 78 instrukcijas lygiai taip pat kaip ir Intel lustas), tačiau turėjo daug daugiau galimybių (papildomas 120 intrukcijų, daugiau registrų, supaprastintas ryšys su technine įranga). Taip gimė galingasis Z80 mikroprocesorius. Originalus Z80 buvo išleistas 1976 m. liepos mėn. Naujesnės šio procesoriaus versijos išlaikė tą pačią architektūrą, bet veikė greičiau. Pavyzdžiui, originali versija veikė 2,5 MHz taktiniu dažniu, naujesnė Z80A iki 4 MHz, Z80B pasiekė 6 MHz, o galiausiai Z80H pasiekė 8 MHz taktinį dažnį. Daugelis kompanijų septintajame ir devintajame dešimtmečiuose gamino prietaisus grįstus Zilog patobulintu lustu, nes šis be papildomų problemų lengvai galėjo paleisti 8080 kodą per daug jo nekeičiant. Todėl CP/M buvo puikus operacinės sistemos pasirinkimas.

**MCS-48/8048/8035/8748**,,Intel MCS-48” šeima buvo ,,Intel“ kompanijos įėjimas į mikrovaldiklių (angl. microcontroller) rinką po ankstesnių mikroprocesorių (angl. microprocessors) 4004 ir 8080 sėkmės. Ši šeima sujungė 8 bitų procesorių su įvestį/išvestį (I / O) ir RAM, ROM (arba EPROM) prievadus (angl. ports) į vieną lustą (angl. single chip), taip supaprastinant įterptųjų sistemų dizainą bei sumažinant įvairioms programoms reikalingos techninės įrangos kiekį.

MCS-48 serija naudojo modifikuotą Harvardo architektūrą, kurioje buvo talpinamos atskiros adresų erdvės programos ir duomenų atminčiai. Tai leido vienu metu pasiekti tam tikrą komandą ir jos duomenis, o tai žymiai pagerino apdorojimo efektyvumą. Atminties pasirinkimai apėmė 1 KB ROM lustą 8048 modelyje ir 1 KB EPROM lustą 8748 modelyje. EPROM naudojimas pastarajame modelyje suteikė galimybę perprogramuoti produktą kūrimo metu, taip suteikiant galimybę pakartotinai išbandyti sistemą. Ši serija buvo sukurta sus 64 baitų vidine RAM atmintimi, kurios pakako daugumai kontrolės orientuotų programų. Be to, šie mikrovaldikliai turėjo 27 įvesties/išvesties linijas, kurios galėjo tiesiogiai sąveikauti su išoriniais įrenginiais ir periferiniais prietaisais. Turėdamos iki 11 MHz taktinį dažnį šios suteikė daugiau apdorojimo galios įvairioms įterptoms programoms.

Istoriškai, 1979-ųjų viduryje puslaidininkių pramonėje augo stipri konkurencija, nes tokios kompanijos kaip ,,Texas Instruments“ rinkai pateikė tokius mikrovaldiklių modelius kaip TMS1000, kurie viename luste integravo procesorių, RAM, ROM, ir I/O. Esant tokioms aplinkybėms, ,,Intel“ suvokė, kad turi pabandyti sukurti tam tikrą mikrovaldiklį, kad galėtų konkuruoti šios rinkos permainų sąlygomis. Pasirinkimas naudoti 8 bitų architektūr MCS-48 serijoje, užuot dar labiau supaprastinus iki 4 bitų, turėjo strateginę priežastį – suteikti galingesnį ir universalesnį sprendimą klientų poreikiams. O tai atitiko ,,Intel“ rinkodaros tikslus – pasiūlyti klientams naujovišką ir efektyvų produktą plačiai pritaikomose srityse. Vienas žymiausių pavyzdžių buvo kompiuterių klaviatūros, kuriose 8048 modelis buvo pritaikytas pirmosios IBM PC klaviatūros klavišų kodavimui atlikti. Dėl universalumo ir integracijos šis valdiklis įgijo populiarumą daugybėje įterptųjų sistemų nuo 1970-ųjų pabaigos, taip tapdamas svarbia dalimi mikrovaldiklių plėtros istorijoje.

# Registrai

**Z80**Visi Z80 procesoriaus registrai yra įdiegti naudojant statinį RAM (angl. static RAM). Procesoriuje yra du šešių bendros paskirties registrų rinkiniai, kurie gali būti naudojami kaip atskiri 8 bitų registrai arba sujungiami į 16 bitų registrus poromis. Be to, yra šeši specialiosios paskirties registrai ir du akuliatorinės (angl. accumulator) bei vėliavėlių (angl. Flag) registrų rinkiniai (pav.1).

Paveikslėlis, kuriame yra tekstas, ekrano kopija, skaičius, Šriftas

Automatiškai sugeneruotas aprašymas

pav. 1 CPU registrų konfiguracija

Specialiosios paskirties registrai:

1. **Programų skaitiklis** (angl. Program Counter (PC)) – užtikrina tinkama instrukcijų vykdymo seką ir leidžia keisti valdymo srautą, įskaitant šuolius ir iškvietimus. Taip pat atlieka svarbų vaidmenį nustatant, kur procesorius veikia, ir valdant instrukcijų nuskaitymą.
2. **Steko rodyklė** (angl. Stack Pointer) yra ypač svarbi efektyviam programos valdymui, užtikrinant, kad subprogramų iškvietimai ir pertraukimai būtų tvarkomi be klaidų. Be to, SP leidžia naudoti steką kaip dinamišką atminties sritį, kuri padeda spręsti sudėtingas programavimo užduotis.
3. **Dviejų indeksų registrai (IX ir IY)** (angl. Two Index Registers (IX and IY) yra skirti indeksuotam adresavimui palengvinti, supaprastinant programos logiką ir didinant efektyvumą dirbant su struktūros duomenimis, tokiais kaip lentelės ar masyvai. Jie itin naudingi sudėtingoms programoms, kuriose dažnai reikia apdoroti nuoseklius ar kartotinius duomenis.
4. **I registras** yra esminis 2-tojo režimo pertraukčių tvarkymui, leidžiantis netiesiogiai iškviesti dinamiškai esančias pertraukčių aptarnavimo procedūras su minimaliu vėlavimu ir efektyviu atminties panaudojimu.
5. **R registras** yra būtinas komponentas, užtikrinantis patikimą dinaminės RAM atminties veikim be papildomų programuotojo pastangų. Jo veikimo skaidrumas ir minimalus poveikis CPU našumui daro jį idealiu sprendimu automatiniam atminties atnaujinimui ,,Z80” mikroprocesoriuje.
6. **Akumuliatorinės ir vėliavėlių** (angl. accumulator and flag) **registrų pora** yra ,,Z80“ šerdis, leidžianti tiksliai atlikti aritmetines, logines operacijas ir stebėti jų būklę. Jų lankstumas bei efektyvumas padeda užtikrinti sklandų ir patikimą programos vykdymą įvairiose taikomosiose srityse

Bendros paskirties registrų pagrindinės savybės:

1. **Dvigubi registrų rinkiniai**. Procesorius turi du šešių 8 bitų registrų rinkinius: Pagrindinis rinkinys (BC, DE, HL) ir Alternatyvus (šešėlinis) rinkinys (BC`, DE`, HL`). Šie registrai gali būti naudojami kaip atskiri 8 bitų registrai (pvz. B, C, D ir t.t) ir kaip sujungti 16 bitų registrai (pvz. BC, DE, HL) adresavimui ir duomenų manipuliavimui.
2. **Registrų rinkinių keitimas**. EXX instrukcija leidžia perjungti pagrindinį ir alternatyvųjį registrų rinkinius viena komanda. Tai itin naudinga greitam konteksto perjungimui atitinkamų užduočių metu ar kai reikia greitai saugoti ir atkurti skirtingų procesų būsenas.
3. **Pertraukų tvarkymas**. Vieną registrų rinkinį galima skirti tik pertraukčių aptarnavimo rutinoms. Tai leidžia apdoroti pertraukimus be b9tinybės saugoti ar atkurti registrų reikšmes iš išorinės atminties. Toks mechanizmas sumažina pertraukimų aptarnavimo laiką ir pagerina visos sistemos veikimą.
4. **Programavimo supaprastinimas**. Registrų lankstumas palengvina programavimo logiką, ypač sistemose, kuriose naudojama tik ROM atmintis su ribotomis išorinės atminties skaitymo ir rašymo galimybėmis. Tai leidžia išvengti perteklinių duomenų perkėlimų tarp registrų ir atminties.
5. **Universalumas programose**. Šie registrai naudojami įvairioms užduotims, pavyzdžiui duomenų manipuliavimui, adreso skaičiavimams ar laikinam saugojimui.

Bendros paskirties registrai yra universalūs ir užtikrina efektyvų duomenų tvarkymą. Dvigubi registrų rinkiniai užtikrina greitą konteksto perjungimą, o tai yra ypač svarbu tvarkant pertraukimus, o jų lankstumas supaprastina programavimą apribotose atminties sistemose.

**MCS-48/8048/8035/8748**

1. **Akumuliatorinės** (angl, accumulator) yra pagrindinis registras, kurame vyksta duomenų apdorojimas. Jis laiko operandus ariminėms ir loginėms operacijoms bei saugo jų rezultatus.
2. **Registrų bankai (R0 – R7).** Teikia bendros paskirties 8 bitų registrus, naudojamas laikiniems duomenims saugoti ir skaičiavimams atlikti. Registrai suskirstyti į keturis bankus, po aštuonis registrus kiekviename. Aktyvus registrų bankas pasirenkamas naudojant programos būsenos žodį (angl. Program status word).
3. **Programos skaitiklis** (angl. Program Counter (PC)) yra 11 bitų registras saugantis kitos vykdytinos instrukcijos adresą.
4. **Steko rodyklė** (angl. Stack Pointer) – tai 8 bitų registras, nurodantis dabartinį steko viršų vidinėje RAM atmintyje. Stekas naudojamas laikinam duomenų saugojimui subprogramų iškvietimų ir pertraukčių metu.
5. **Programos būsenos žodis** (angl. Program status word (PSW)) – tai 8 bitų registras, kuriame saugomi vėliavėlių ir valdymų bitai, kurie rodo procesoriaus būseną. Svarbūs bitai:

* CY (Pernešimas) – rodo pernešimą arba pasiskolinimą aritmetinėse operacijose.
* AC (Pagalbinis pernešimas) – naudojamas BCD operacijoms.
* OV (Perpildymas) – rodo perpildymą ženklinėse operacijose.
* RS1 ir RS0 – leidžia pasirinkti aktyvų registrų banką.
* P (Paritetas) – rodo ar akumuliatorinė turi lyginį ar nelyginį bitų skaičių,

1. **Duomenų rodyklė** (angl. Data Pointer (DPTR)) yra 16 bitų registras, padalytas į aukštąjį (DPH) ir žemąjį (DPL) baitą. Naudojamas netiesioginiam išorinės atminties adresavimui.
2. **Laikmačių registrai** (T0 ir T1) – Teikia laikmačio it įvykių skaičiavimo funkcionalumą, susijusį su mikrovaldiklio aparatūros laikmačiais.
3. **Pertraukčių įjungimas** (angl. Interrupt Enable (IE)) yra 8 bitų registras, skirtas įjungti ar išjungti specifines pertrauktis. Pavyzdžiui, individualūs pertraukčių įjungimo bitai (pvz., Timer 0, Timer 1, išorinės pertrauktys) ir globalus pertraukčių įjungimo bitas.
4. **Pertraukčių prioritetas** (angl. Interrupt Priority (IE)) – priskiria prioritetus skirtingiems pertraukčių šaltiniams.
5. **Laikini ir specialios paskirties registrai**. Kai kuriuose mikrovaldiklio variantuose yra papildomi registrai specifinėms funkcijoms atlikti. Pavyzdžiui, registrai įjungimo, išjungimo (I / O) valdymui ir papildomi laikmačiai.

Duomenų tipai

**Z80**

Z80 mikroprocesorius apibrėžia įvairius duomenų tipus, kurie nustato duomenų dydį struktūrą ir paskirtį. Pagrindinis duomenų apdorojimo vienetas yra 8 bitai, tačiau procesorius leidžia derinti registrus ir naudoti sudėtingus duomenų tipus bei metodus efektyviam programavimui. Z80 daugiausiai dirba su 8 bitų duomenimis, kurie naudojami aritmetinėms, loginėms ir duomenų perdavimo operacijoms. Pavyzdžiui, akumuliatorius (A) yra pagrindinis 8 bitų registras, skirtas rezultatams saugoti, o bendros paskirties registrai (C, B, D, E, H, L) naudojami duomenų laikymui ar manipuliavimui.

Taip pat, Z80 procesorius leidžia sujungti 8 bitų registrus į 16 bitų poras (BC, DE, HL), kurios naudojamos tiek duomenų manipuliavimui, tiek adresavimui. De to, SP (stekų rodyklė) ir PC (programos skaitiklis) registrai skirti steko valdymui ir instrukcijų sekai. Tiesioginiai duomenys leidžia greitai vykdyti operacijas, kai IX ir IY registrų pagalba leidžia efektyviai pasiekti atminties vietas naudojant poslinkius. Vėliavų (angl. Flags) registras saugo būsenos indikatorius, tokius kaip Zero (Z), Carry (C), Sign (S) ir kitus, kurie naudojami operacijų rezultatų analizei ir programos srauto valdymui. Bitų lygio operacijos, pvz., SET ir RESET, leidžia tiesiogiai keisti atskirus bitus, suteikiant tiksią kontrolę. 8 or 16 bitų registrų lankstumas, tiesioginiai ir indeksuoti duomenys bei galimybė dirbti su bitais leidžia Z80 procesoriui būti universaliu ir efektyviu įrankiu žemo lygio programavimui net ribotos atminties aplinkoje.

**MCS-48/8048/8035/8748**

MCS-48 mikrovaldiklių 8 bitų šeima (8048, 8035 ir 8748) naudoja kelis pagrindinius duomenų tipus pritaikytus būtent jų architektūrai. Pagrindinė duomenų struktūra yra 8 bitų baitas kuris yra standartinis dydis aritmetinėms, loginėms ar duomenų perdavimo operacijoms, apimantis registrus, atminti ir akumuliatorių. Šie mikrovaldikliai taip pat palaiko bitų lygio duomenis, leidžiančius tiesiogiai nustatyti, išvalyti ar patikrinti atskirus bitus žodyje, kas yra ypač svarbu valdymo ir būsenos vėliavėlių (angl. Flags) nustatymui.

Be to, architektūra palaiko dvejetainius kodu užkoduotus dešimtainius skaičius (BCD), leidžiančius pirmiausia pateikti dešimtaines reikšmes dvejetainiu formatu užduotims, kuriose reikalingi skaitmeniniai ekranai. Atminties adresavimui naudojami 8 bitų arba 11 bitų adresų duomenų tipai, kurie palaikomi Įvairiose adresavimo režimų instrukcijose. Šių duomenų tipų universalumas palengvina programų ir duomenų apribojimą, užtikrindamas, kad MCS-48 mikrovaldiklis būtų efektyvus vykdant valdymui orientuotas ir įterptų sistemų užduotis.

# Adresavimo režimai

**Z80**

Z80 mikroprocesorius palaiko įvairius adresavimo režimus, kurie suteikia lankstų ir efektyvų duomenų bei instrukcijų pasiekimą. Šie režimai leidžia programuotojui pritaikyti optimalias strategijas atminties ir registrų naudojimui, atsižvelgiant į konkrečius poreikius.

1. Tiesioginis adresavimas (angl. Immediate Addressing) – duomenys nurodomi tiesiogiai pačioje instrukcijoje.

Pvz.: LD A, 0x42 – užkrauna 0x42 į akumuliatorių.

1. Išplėstinis adresavimas (angl. Extended Addressing) – Instrukcija nurodo tiesioginį 16 bitų atminties adresą, kurį reikia pasiekti.

Pvz.: LD A, (0x1234) – perskaito reikšmę iš atminties adreso 0x1234 į akumuliatorių.

1. Indeksuotas adresavimas (angl. Indexed Addressing) – Naudojamas vienas iš indeksinių registrų (IX arba IY) kaip bazinis adresas, prie kurio pridedamas 8 bitų poslinkis.

Pvz.: LD A, (IX+5) – užkrauna reikšmę iš atminties adreso, kuris yra IX turinys plius 5.

1. Santykinis adresavimas (angl. Relative Addressing) – Naudojamas šuolių instrukcijose, kur adresas apskaičiuojamas remiantis programos skaitiklio (PC) pozicija.

Pvz.: JR 0x10 – šuolis į vietą, esančią 16 baitų į priekį nuo dabartinės pozicijos.

1. Registrų adresavimas (angl. Register Addressing) – Duomenys manipuliuojami tiesiogiai tarp registrų.

Pvz.: LD B, A – nukopijuoja reikšmę iš registro A į registrą B.

1. Netiesioginis registrų adresavimas (angl. Register Indirect Addressing) – Adresas duomenims saugomas registrų poroje (pvz. HL).

Pvz.: LD A, (HL) – užkrauna reikšmę iš atminties adreso, kurį nurodo HL.

1. Numanytas adresavimas (angl. Implied Addressing) – Kai kurios instrukcijos veikia konkrečiuose registruose ar atminties vietose, jų konkrečiai nenurodant.

Pvz.: CPL – invertuoja akumuliatoriaus turinį (papildo).

1. Modifikuotas pirmojo puslapio adresavimas (angl. Modified Page Zero Addressing) – Naudojamas greitam pasiekimui pirmajame 256 baitų atminties puslapyje (adresų diapazonas 0x00–0xFF).

Pvz.: LD A, (0xF0) – tiesiogiai pasiekia atminties adresą 0xF0 pirmame puslapyje.

1. Bitų adresavimas (angl. Bit Addressing) – Veikia su atskirais registrų ar atminties bitais.

Pvz.: BIT 7, A – patikrina, ar 7-as akumuliatoriaus bitas nustatytas.

**MCS-48/8048/8035/8748**

MCS-48 mikrokompiuterių šeima siūlo įvarius adresavimo režimus, kurie leidžia efektyviai programuoti ir manipuliuoti duomenimis. Šis lankstumas daro MCS-48 mikrokompiuterių šeimą idealią naudoti įterptosiose sistemose, kur būtina efektyviai valdyti resursus ir atlikti valdymo užduotis. Pagrindiniai režimai yra tokie:

1. Tiesioginis adresavimas (angl. Direct Addressing). Šiame režime operando adresas yra tiesiogiai nurodomas instrukcijoje.

Pvz.: MOV A, 20H – perkelia turinį iš atminties vietos 20H į akumuliatorių.

1. Registrų adresavimas (angl. Register Addressing). Duomenys pasiekiami tiesiai iš darbo registrų. MCS-48 turi registrus nuo R0 iki R7, kurie gali būti adresuojami tiesiogiai.

Pvz.: MOV A, R3 – perkelia turinį iš registro R3 į akumuliatorių.

1. Netiesioginis adresavimas (angl. Indirect Addressing). Šis režimas naudoja registrą kaip rodyklę į atmintį. Registrai R0 ir R1 gali būti naudojami kaip rodyklės.

Pvz.: MOV A, @R0 – duomenys paimami iš atminties vietos, į kurią nurodo registras R0.

1. Tiesioginis duomenų adresavimas (angl. Immediate Addressing). Reikšmė yra tiesiogiai nurodoma pačioje instrukcijoje.

Pvz.: MOV A, #0AH – užkrauna akumuliatorių reikšme 0AH.

1. Numanytas adresavimas (angl. Implied Addressing). Operandas yra numanomas pačios instrukcijos ir todėl nenurodomas atskirai.

Pvz.: CLR A – išvalo akumuliatorių be papildomo operandų nurodymo.

1. Bitų manipuliacijos adresavimas (angl. Bit Manipulation Addressing). Šis režimas leidžia manipuliuoti individualiais bitais akumuliatoriuje ar atmintyje – juos tikrinti, nustatyti arba išvalyti.

Pvz.: ANL C, bit – tikrina ir manipuliuoja konkrečiu bitu, suteikdamas sąlyginę kontrolę priklausomai nuo to bito reikšmės.

# Atminties struktūra

# Mikroarchitektūra

# Mašinos kodo arba asemblerio pavyzdžiai

# Aukšto lygio programavimo kalbų palaikymas

# Įvesties-išvesties ir pertraukimų mechanizmai

# Išvados

# Cituojama literatūra: **Z80:** <https://landley.net/history/mirror/cpm/z80.html> <https://www.computinghistory.org.uk/det/12157/Zilog-Z-80-Microcomputer-System> <https://en.wikipedia.org/wiki/Zilog_eZ80> User manual: <https://www.zilog.com/docs/z80/um0080.pdf>

**MCS-48/8048/8035/8748:**  
<https://www.eejournal.com/article/a-history-of-early-microcontrollers-part-4-the-intel-8048-and-8748/> <https://devsaurus.github.io/mcs-48/mcs-48.pdf>   
User manual:  
<https://vtda.org/docs/computing/Intel/98-270B_MCS-48MicrocomputerUsersManualJul77.pdf>