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

**Z80**

Z80 mikroprocesorius turi atminties struktūra, pagrįsta 16 bitų adresų magistrale, todėl teoriškai gali tiesiogiai adresuoti iki 64 KB atminties. Skirtingai nei dauguma šiuolaikinių procesorių, Z80 neskirsto arminties į vartotojo ir priežiūros režimus. Tai suteikia galimybę pasiekti bet kurią atminties vietą pagal programos poreikį. Kiekvienas atminties adresas talpina 8 bitų duomenų žodį, o skaitymas ir rašymas vykdomi per atsitiktinę prieigą (angl. random access).

Viena stipriausių Z80 funkcijų yra Steko rodyklė (SP), kuri valdo atminties dalį naudojamą steko operacijoms. Stekas yra itin svarbus laikinas duomenų saugojimo mechanizmas, naudojamas, pavyzdžiui, grįžimo adresams saugoti vykdant subprogramas ar tvarkant pertraukimus (angl. Interruptions). Kitas svarbus komponentas yra programos skaitiklis (PC), kuris saugo adresą, kur yra kita vykdyta instrukcija. Jei sistema naudoja dinaminę RAM (DRAM), Z80 siūlo automatinį atnaujinimo mechanizmąą per R registrą, kuris padeda išlaikyti dinaminės atminties duomenis.

Z80 pertraukimų apdorojimas yra vykdomas naudojant vektorinius adresus, saugomus atmintyje. Šie vektoriai procesoriui iš karto pateikia adresą, į kurį reikia atlikti šokimo (angl. jump) komandą ir įvykdyti atitinkamą pertraukimų aptarnavimo rutiną (angl. interrupt service routines (IRS)). Be to, Z80 paliko indeksuotą adresavimą, naudodamas IX ir IY registrus, kad ženkliai paspartina operacijas su lentelėmis ar struktūrizuotais duomenimis ir padidina bendrą efektyvumą. Nors Z80 neturi vartotojo/priežiūros (angl. user/supervisor) režimų, kaip šiuolaikiniai procesoriai, jo lanksti atminties sistema leidžia „inžinieriams“ sukurti panašias funkcijas naudojant programinę įrangą.

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

MCS-48 mikroprocesorių modeliai pasižymi efektyvia ir aiškiai apibrėžta atminties struktūra, kuri leidžia lengvai valdyti tiek duomenis, tiek programas. Šiuose mikrokompiuteriuose yra du pagrindiniai atminties tipai: programų ir duomenų atmintis, kurių prieinamumo taisyklės skiriasi.

Programų atmintis 8048 ir 8748 mikrokompiuteriuose yra organizuota į 1024 8 bitų žodžius. 8748 turi ištrinamą ir programuojamą EPROM atmintį, o 8048 naudoja gamykloje užprogramuotą ROM. Tuo tarpu 8035 visiškai priklauso nuo išorinės programų atminties. Programų atmintis prasideda adresu 0, kur tam tikros reikšmingos įvykio vietos yra iš anksto nustatytos, pvz., reset vektorius – adresu 0, išorinės pertraukimo vektorius – adresu 3, o laikmačio pertraukimo vektorius – adresu 7. Toks išdėstymas palengvina pertraukimų apdorojimą ir sistemos inicijavimą. Be to, naudojant EA (angl. External Access) režimą, procesorius gali persijungti į išorines programų atminties jungtis, kas ypač naudinga derinant ar testuojant programas.

Duomenų atmintis suskirstyta į 64 baitų vidinę atmintį, iš kurių apatiniai 8 baitai yra tiesiogiai adresuojami ir veikia kaip darbo registrai. Šie registrai suskirstyti į dvi bankų grupes, kurias galima pasirinkti naudojant instrukciją SEL RB, leidžiančią palaikyti atskirus kontekstus pertraukimų tvarkymo ir pagrindinės programos vykdymo metu. Likę 56 baitai yra netiesiogiai adresuojami ir naudojami kaip bendros paskirties RAM arba stekas subprogramų ir pertraukimų iškvietimams. Glaudžiai integruotas stekas veikia kartu su programos skaitikliu ir palaiko iki 8 lygmenų subprogramų ar pertraukimų iškvietimų įdėjimą be poreikio naudoti išorinę atmintį. Be to, MCS-48 architektūra leidžia prijungti išorinę programų ir duomenų atmintį.

Svarbu paminėti, kad MCS-48 mikrokompiuteriai neturi aiškaus priežiūros režimo (supervisor mode) ir veikia vienu lygiu. Dėl to programuotojams nereikia rūpintis papildomais režimų aprašymais ar sudėtingais valdymo mechanizmais, o tai palengvina šių mikrokompiuterių taikymą įterptinėse sistemose.

# Komandų sistema (ISA)

**Z80**

Z80 komandų sistema (angl. Instruction set architecture (ISA)) yra universalaus dizaino palaikanti ne vieną operaciją, todėl Z80 procesorius yra tinkamas tiek įterptinėms sistemoms, tiek bendros paskirties kompiuteriams. Z80 instrukcijų rinkinys yra padidintas ,,Intel 8080 ISA“ rinkinys turintis daugiau papildomų instrukcijų, adresavimo rėžimų bei patobulintų funkcijų. O tai užtikrina efektyvų įvairių operacijų vykdymą.

**Instrukcijų formatas**. Z80 instrukcijos gali būti 1 baito, 2 baitų arba 3 baitų ilgio:

* 1 baito instrukcijos – Vienos operacijos komandos. Pvz. NOP (jokia operacija).
* 2 baitų instrukcijos – Apima tiesioginius duomenis arba išplėstines operacijas. Pvz. LD r, n (įkelia 8 bitų reikšmę į registrą).
* 3 baitų instrukcijos – Naudoja išplėstinį adresavimą. Pvz. JP nn (šuolis į 16 bitų adresą nn).

**Instrukcijų kategorijos:**

1. Duomenų užkrovimo ir apsikeitimo instrukcijos (angl. Load and Exchange Instructions) – Perkelia duomenis tarp registrų, atminties ir įvesties/išvesties įrenginių. Pvz.:

* LD A, (nn) – įkelia akumuliatorių reikšme iš adreso nn.
* LD (HL), n – įkelia 8 bitų reikšmę į atmintį, kurią nurodo HL

1. Aritmetinės ir loginės instrukcijos (angl. Arithmetic and Logical Instructions) – Atlieka sudėtį, atimtį, loginę AND, OR, XOR operacijas ir palyginimus. Pvz.:

* ADD A, r – prideda registro r reikšmę prie akumuliatoriaus.
* SUB n – atima tiesioginę reikšmę n iš akumuliatoriaus.

1. Rotavimo ir poslinkio instrukcijos (angl. Rotate and Shift Instructions) – Perstumia arba suka bitus registruose ar atmintyje. Pvz.:

* RLC r – suka bitus į kairę per pernešimą.
* SRL r – logiškai perstumia bitus į dešinę.

1. Bitų manipuliavimo instrukcijos (angl. Bit Manipulation Instructions) – Tikrina, nustato arba išvalo pasirinktus bitus. Pvz.:

* BIT b, r – tikrina, ar bitas b yra nustatytas registre r.
* SET b, r – nustato bitą b registre r.

1. Šuolių, iškvietimų ir grįžimo instrukcijos (angl. Jump, Call, and Return Instructions) – Valdo programos srautą naudojant sąlyginius ir besąlyginius šuolius, subprogramų iškvietimus ir grįžimus. Pvz.:

* JP nn – besąlyginis šuolis į adresą nn.
* CALL nn – iškviečia subprogramą adresu nn.
* RET – grįžta iš subprogramos.

1. Įvesties/išvesties instrukcijos (angl. Input/Output Instructions) – Sąveikauja su I/O prievadais išoriniams įrenginiams valdyti. Pvz.:

* IN A, (n) – įkelia duomenis iš prievado n į akumuliatorių.
* OUT (n), A – išveda akumuliatoriaus duomenis į prievadą n.

1. CPU valdymo instrukcijos (angl. CPU Control Instructions) – Valdo procesoriaus būseną ir operacijas, pvz., įgalina arba išjungia pertraukimus. Pvz.:

* DI – išjungia pertraukimus.
* EI – įgalina pertraukimus.
* HALT – sustabdo CPU, kol įvyks pertraukimas.

**Adresavimo režimai**

Z80 palaiko skirtingus adresavimo režimus, taip leidžiant lengviau pasiekti duomenis:

* Tiesioginis adresavimas (angl. Immediate Addressing): Duomenys yra instrukcijos dalis.

Pvz.: LD A, n.

* Tiesioginis adresų nurodymas (angl. Direct Addressing): Nurodomas konkretus adresas.

Pvz.: LD A, (nn).

* Registrų adresavimas (angl. Register Addressing): Duomenys paimami iš registrų.

Pvz.: LD A, B.

* Indeksuotas adresavimas (angl. Indexed Addressing): Bazinis adresas naudojant indeksų registrus (IX arba IY) su poslinkiu.

Pvz.: LD A, (IX+d).

* Numanytas adresavimas (angl. Implied Addressing): Operandai yra numanomi instrukcijoje.

Pvz.: NOP (jokia operacija).

**Vėliavėlės (angl. Flags)**

Z80 turi vėliavėlių registrą, kuris rodo aritmetinių ir loginių operacijų rezultatus:

* S (Sign): Nustatoma, jei rezultatas yra neigiamas.
* Z (Zero): Nustatoma, jei rezultatas yra lygus nuliui.
* P/V (Parity/Overflow): Nustatoma, jei yra lyginis bitų skaičius arba įvyko perpildymas.
* C (Carry): Nustatoma, jei įvyko pernešimas arba skolinimasis.
* H (Half Carry): Naudojama BCD aritmetikoje tarp pusės baito pernešimui.

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

MCS-48 serijos instrukcijų rinkinio architektūrą (ISA) sudaro gerai optimizuotas instrukcijų rinkinys leidžiantis efektyviai dirbti su į valdymą orientuotomis programomis. Šį rinkinį sudaro duomenų perdavimo, aritmetinės ir loginės operacijos, bitų manipuliacijos, programos valdymo bei įvesties/išvesties instrukcijų komandos. Visos instrukcijos yra 1 arba 2 baitų ilgio, taip siekiant efektyviai naudoti atmintį.

**Instrukcijų kategorijos:**

1. Duomenų perdavimo instrukcijos (angl. Data Transfer Instructions). Pvz.:

* MOV A, Rn: Perkelia duomenis iš registro Rn į akumuliatorių.
* MOVX A, @BUS: Apsikeičia duomenimis tarp akumuliatoriaus ir išorinės atminties per magistralę (BUS).
* MOVP A, @A: Užkrauna baitą iš programos atminties į akumuliatorių pagal adresą.

1. Aritmetinės instrukcijos (angl. Arithmetic Instructions). Pvz.:

* ADD A, Rn: Prideda registro Rn turinį prie akumuliatoriaus.
* ADDC A, Rn: Prideda su pernešimo vėliavėle (Carry Flag).
* INC Rn: Padidina registro reikšmę vienetu.

1. Loginės operacijos (angl. Logical Operations). Pvz.:

* ANL A, Rn: Atlieka loginį AND tarp akumuliatoriaus ir registro Rn.
* ORL A, #data: Atlieka loginį OR tarp akumuliatoriaus ir tiesioginių duomenų.
* CLR A: Išvalo akumuliatorių (nustato 0).

1. Bitų manipuliavimo instrukcijos (angl. Bit Manipulation Instructions). Pvz.:

* CLR C: Išvalo pernešimo vėliavėlę.
* CPL A: Papildo akumuliatoriaus bitus (invertuoja).

1. Šakų instrukcijos (angl. Branch Instructions). Pvz.:

* JMP addr: Besąlyginis šuolis į nurodytą adresą.
* DJNZ Rn, addr: Sumažina registro Rn reikšmę ir šoka, jei ji nėra lygi 0.
* JC addr: Šoka, jei pernešimo vėliavėlė yra nustatyta.

1. Įvesties/išvesties instrukcijos (angl. Input/Output Instructions). Pvz.:

* IN A, Pn: Įkelia duomenis iš prievado Pn į akumuliatorių.
* OUTL Pn, A: Išveda akumuliatoriaus duomenis į prievadą Pn.

1. Valdymo instrukcijos (angl. Control Instructions). Pvz.:

* NOP: Jokia operacija.
* EN I: Įgalina pertraukimus.
* DIS TCNTI: Išjungia laikmačio pertraukimus.

# Mikroarchitektūra

**Z80**

Z80 mikroarchitektūra pasižymi galingu ir lanksčiu dizainu, kuris optimizuotas veikimui ir apdorojimo lankstumui. Z80 procesorius sudarytas iš daugybės specializuotų ir bendros paskirties registrų, Aritmetinis loginis blogas (ALU) ir valdymo vieneto, kurie yra glaudžiai sujungti vidinėje duomenų magistralėje. Z80 palaiko sudėtingą instrukcijų rinkinį (ISA), kuris apima 8 ir 16 bitų aritmetines instrukcijas, logines operacijas, bitų manipuliaciją bei įvairius adresavimo būdus tokius kaip indeksuotą, tiesioginį ir santykinį. Šios instrukcijos yra glaudžiai susijusios su Z80 mikroarchitektūros elementais, užtikrinant efektyvų jų vykdymą.

Pagrindiniai ypatumai:

1. Dvigubi registrų rinkiniai (angl. Dual Register Sets). Yra itin naudingi tvarkant pertraukimus ir vykdant kelias užduotis vienu metu.
2. Indeksuotas adresavimas naudojant IX ir IY registrus. Du 16 bitų indeksų registrai (IX ir IY) leidžia efektyviai tvarkyti didelius duomenų masyvus ir struktūrizuotus duomenis. Tai pagerina programavimo lankstumą ir supaprastina darbą su lentelėmis ar struktūromis.
3. Pertraukimų valdymas (angl. Interrupt Handling). Z80 palaiko tris pertraukimų režimus, įskaitant vektorinius ir nemaskuojamus pertraukimus. Ši savybė daro Z80 tinkamą realaus laiko ir įterptinėms programoms, kuriose pertraukimai yra kritiškai svarbūs.
4. Atminties atnaujinimo mechanizmas (angl. Memory Refresh Mechanism). Autonomiškai valdo dinaminius RAM atnaujinimo ciklus. Tai leidžia lengvai integruoti Z80 su dinaminės atminties technologijomis be papildomos programinės ar aparatinės įrangos.

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

Supaprastinta, vieno lusto MCS-48 mikroarchitektūra yra optimizuota įterptinėms valdymo programoms. Ši architektūra apima: 8 bitų CPU, programų atmintį (ROM/EPROM), duomenų atmintį (RAM) ir programuojamus I/O prievadus.

Pagrindiniai ypatumai:

1. Aritmetinis-loginis blokas (ALU). ALU atlieka 8 bitų sudėtį, sudėtį su pernešimu, loginę AND, OR, Exclusive-OR (XOR), inkrementavimą/dekrementavimą, bitų komplementą, poslinkį ir rotaciją. Visi rezultatai saugomi akumuliatoriuje, kuris yra pagrindinis duomenų registras.
2. Programų atmintis (angl. Program Memory).

* Vidinė atmintis įprastai yra 1 KB (1024 žodžiai po 8 bitus).
* Adresai pasiekiami per 10 bitų programos skaitiklį (PC).
* Specialios vietos programų atmintyje (Reset vektorius: Adresas 0; Išorinės pertrauktys: Adresas 3; Laikmačio perpildymo pertrauktys: Adresas 7)

1. Atmintis ir duomenys (angl. Memory and Data).

* Vidinė 64 baitų RAM atmintis.
* Dinamiškai pasirenkami darbo registrų bankai naudojant Bank Selection Bit.
* Likusi RAM dalis naudojama stekui ir subprogramų parametrams saugoti.

1. Programos būsenos žodis (angl. Program Status Word – PSW).

* PSW apima būsenos vėliavėles, tokias kaip Carry ir Auxiliary Carry, bei vartotojo apibrėžiamą vėliavėlę F0.
* PSW taip pat valdo Steko rodyklę (SP), kuri leidžia valdyti iki 8 lygių subprogramų įdėjimą.

1. Laikmačiai ir skaitikliai (angl. Timers and Counters).

* 8 bitų laikmatis/skaitiklis. Gali veikti kaip laikmatis naudojant padalintą laikrodį (pvz., 80 µs raiška su 6 MHz laikrodžiu). Arba gali veikti kaip įvykių skaitiklis, kurio reikšmė didinama išoriniais signalais per T įvestį.

1. Įvesties/išvesties prievadai (angl. I/O Ports).

* Įrenginyje yra trys I/O prievadai. Port 1 ir Port 2 – quasi-bidirekciniai prievadai, kurie leidžia naudoti kontaktus tiek įvestims, tiek išvestims. BUS – pilnai dvipusis prievadas, skirtas išorinei atminčiai ir periferiniams įrenginiams.

# Mašinos kodo arba asemblerio pavyzdžiai

**Z80**

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

# Aukšto lygio programavimo kalbų palaikymas

**Z80**

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

# Įvesties-išvesties ir pertraukimų mechanizmai

**Z80**

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

# Išvados. Z80 ir MCS-48/8048/8035/8748 architektūrų skirtumai.

# 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>