Z80 vs. MCS-48/8048/8035/8048 architektūrų techninė ataskaita

Darbą parengė: Gvidas Dambrauskas

https://github.com/Diude525/computer-architecture-assignment

Turinys

[Įvadas 3](#_Toc185424349)

[Techninis ir istorinis kontekstas Z80 ir MCS-48/8048/8035/8748 architektūrose 3](#_Toc185424350)

[Registrai Z80 ir MCS-48/8048/8035/8748 architektūrose 4](#_Toc185424351)

[Adresavimo režimai Z80 ir MCS-48/8048/8035/8748 architektūrose 7](#_Toc185424352)

[Atminties struktūra Z80 ir MCS-48/8048/8035/8748 architektūrose 8](#_Toc185424353)

[Komandų sistema (ISA) Z80 ir MCS-48/8048/8035/8748 architektūrose 9](#_Toc185424354)

[Z80 ir MCS-48/8048/8035/8748 mikroarchitektūra 11](#_Toc185424355)

[Mašinos kodo arba asemblerio pavyzdžiai Z80 ir MCS-48/8048/8035/8748 architektūrose 11](#_Toc185424356)

[Aukšto lygio programavimo kalbų palaikymas Z80 ir MCS-48/8048/8035/8748 architektūrose 13](#_Toc185424357)

[Įvesties / išvesties ir pertraukimų mechanizmai Z80 ir MCS-48/8048/8035/8748 architektūrose 14](#_Toc185424358)

[Išvados, Z80 ir MCS-48/8048/8035/8748 architektūrų skirtumai. 15](#_Toc185424359)

[Cituojama literatūra: 17](#_Toc185424360)

# 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 ir MCS-48/8048/8035/8748 architektūrose

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. (Landley, n.d.)

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 (naudojęs integrinius grandynus (IC) bei pasižymėjusią LSI (angl. Low Scale Integration) technologija. Turėjo monokristalinį mikroprocesoriaus dizainą). 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. (Landley, n.d.)

,,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. (EE Journal, 2022)

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. (EE Journal, 2022)

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. (EE Journal, 2022)

# Registrai Z80 ir MCS-48/8048/8035/8748 architektūrose

**Z80**Z80 modelį sudaro akumuliatorių ir vėliavėlių registrai, bendrosios paskirties registrų masyvas, atminties rodykles kaip registrus ir specialios paskirties registrai. Detalesnis kiekvienos registrų grupės analizė: (Tallen, n.d.)

Paveikslėlis, kuriame yra tekstas, kvitas, Šriftas, skaičius

Automatiškai sugeneruotas aprašymas

pav. 1 (a) The Z80 Programming Model; (b) Expanded Flag Register with Bit Positions

1. **Bendros paskirties registrai (angl. General-Purpose Registers).** Mikroprocesorius Z80 turi šešis bendrosios paskirties 8 bitų registrus, pavadintus B, C, D, E, H ir L. Šie registrai yra būtini norint išsaugoti duomenis programos vykdymo metu. Jie taip pat gali būti suporuoti kaip BC, DE ir HL, kad būtų galima atlikti 16 bitų operacijas arba saugoti atminties adresus. Programuotojai naudoja šiuos registrus, kad galėtų efektyviai įkelti, saugoti arba perduoti duomenis.
2. **Akumuliatorius (angl. Accumulator).** Akumuliatorius (A registras) yra svarbus 8 bitų registras aritmetiniame loginiame vienete (ALU). Jame saugomi aritmetinių ir loginių operacijų duomenys. Šių operacijų rezultatai saugomi tiesiai į akumuliatorių.
3. **Vėliavos registrai (angl. Flag Register).** Vėliavos registras yra 8 bitų registras, kuriame saugoma įvairių sąlygų būsena po ALU operacijos. ALU naudoja šešias vėliavėles (angl. flip-flops), kad pavaizduotų konkrečias būsenas, pvz., ar dėl aritmetinės operacijos buvo gauta pernešimo ar nulinė reikšmė. Pavyzdžiui, kai papildymas generuoja nešimą, pernešimo vėliavėlė nustatoma ir atitinkamas bitas (D0) vėliavėlės registre rodo loginę reikšmę 1. Bitai D3 ir D5 vėliavėlių registre lieka nepanaudoti. (pav. 1 (b)).

Šeši pagrindiniai vėliavų registrai:

* S (Sign): Nurodo, ar rezultatas yra neigiamas.
* Z (nulis): nustatykite, jei operacijos rezultatas yra nulis.
* P/V (paritetas/perpildymas): nurodo lygaus pariteto arba perpildymo sąlygas.
* C (Carry): Nustatykite, kada operacija generuoja nešiojimą.
* H (Half-Carry) ir N (sudėti / atimti): daugiausia naudojama viduje atliekant dvejetainio kodo dešimtainio (BCD) operacijas.

1. **Alternatyvus registro rinkinys (angl. Alternate Register Set).** Z80 taip pat turi alternatyvų registrų rinkinį – B', C', D', E', H' ir L', kartu su alternatyviu akumuliatoriumi (A') ir vėliavėlių registru (F'). Nors šie registrai nėra tiesiogiai pasiekiami, juos galima pakeisti pagrindiniais registrais, naudojant keitimo instrukcijas, todėl greitai galima išsaugoti ir atkurti duomenis.
2. **16 bitų registrai kaip atminties rodyklės (angl. 16-Bit Registers as Memory Pointers).** Z80 turi keturis 16 bitų registrus, kurie pirmiausia naudojami kaip atminties rodyklės adresams laikyti. Atmintis tarnauja kaip instrukcijų ir duomenų saugykla, o mikroprocesorius pasiekia šias atminties vietas nurodydamas konkrečius adresus.
3. **Indekso registrai (IX ir IY) (angl. Index Registers (IX and IY)).** Z80 turi du 16 bitų indekso registrus – IX ir IY, kuriuose saugomi 16 bitų adresai. Šie registrai ypač naudingi norint pasiekti duomenis konkrečiose atminties vietose atliekant indeksuoto adresavimo operacijas.
4. **Steko rodyklės (angl. Stack Pointer (SP)).** SP yra 16 bitų registras, nukreipiantis į dabartinę dėklo viršų skaitymo / rašymo atmintyje. Stackas yra rezervuota atminties sritis, naudojama laikinai duomenims saugoti, įskaitant grąžinimo adresus paprogramių iškvietimo metu.
5. **Programų skaitiklio registras (PC) (angl. Program Counter (PC)).** PC yra dar vienas 16 bitų registras, kuris yra kitos paimamos ir vykdomos komandos atminties adreso rodyklė. Kiekvieną kartą, kai Z80 gauna nurodymą, jis padidina programos skaitiklį, kad nukreiptų į kitą adresą.
6. **Specialios paskirties registrai (angl. Special-Purpose Registers).** Z80 turi du unikalius registrus, kurių paprastai nėra kituose 8 bitų mikroprocesoriuose. Šie registrai suteikia papildomų funkcijų, ypač pažangaus atminties valdymo ir pertraukimų tvarkymo srityse.

* Pertraukimo vektorių registras (I): naudojamas pertraukimo operacijų metu, kad būtų išlaikytas pertraukimo vektoriaus adreso aukštas baitas.
* Atminties atnaujinimo registras (R): automatiškai didėja po kiekvieno nurodymo gavimo, padedant atnaujinti dinaminę RAM.

**MCS-48/8048/8035/8748.** Detalesnis kiekvienos registrų grupės analizė: (Intel Corporation, 1977)

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 ir MCS-48/8048/8035/8748 architektūrose

Z80 mikroprocesorius ir MCS-48 serijos mikrovaldikliai neturi formalių duomenų tipų kaip šiuolaikinės technologijos. Vietoj to, jie veikia su neapdorotais dvejetainiais duomenimis, daugiausiai su 8 bitų vienetais (baitais). Z80 palaiko 16 bitų duomenis per registrų poras. Visgi, bitų manipuliavimas yra įmanomas abiejose architektūrose, todėl programuotojas gali valdyti atskirus vėliavėlių (angl. Flags), būsenos indikatorius ir sąlyginių operacijų bitus. Specialios vieno bito vėliavėlės, tokiems užduotims kaip pernešimas (angl. carry), nulis (angl. null) ar paritetas (angl. parity) yra labai reikšmingi skaičiavimo rezultatams ir programos srautui. (OpenAI, 2024)

# Adresavimo režimai Z80 ir MCS-48/8048/8035/8748 architektūrose

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. (Zilog, 2016.)

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 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. (Intel Corporation, 1977) 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 ir MCS-48/8048/8035/8748 architektūrose

Z80 mikroprocesorius turi tiesiogiai adresuojamą atminties išdėstymą aplink 64 KB adresų erdvę per 16 bitų adresų magistralę. Jis turi vienodą atminties erdvę, tai reiškia, kad ji yra prieinama tiek programos instrukcijoms, tiek duomenims, todėl tai leidžia programuotojams lanksčiai paskirstyti atmintį. Z80 taip pat palaiko išorinius atminties įrenginius, tokius kaip ROM (programoms saugoti) ir RAM (duomenims ir steko operacijoms saugoti). Adresavimą galima logiškai suskirstyti į segmentus, naudojant išorinės aparatinės įrangos dekodavimo būdus. Pavyzdžiui, diapazonas 0000h–03FFh gali būti skirtas ROM programai vykdyti, o didesni diapazonai, tokie kaip 0400h–FFFFh, bus skirti RAM dinaminei atminčiai saugoti. (Zilog, 2016.)

Z80 siūlo įvairius adresavimo režimus: tiesioginį, netiesioginį adresavimą per registrus, tokius kaip HL arba IX/IY, ir tiesioginį pastovių verčių adresavimą. Kitos Z80 funkcijos yra dinaminis RAM atnaujinimas, kai R registras automatiškai perjungia atminties adresus, kad atnaujintų dinaminę atmintį, o tai užtikrina, kad duomenys būtų nepažeisti. Nors centrinis procesorius neturi aiškios prižiūrėtojo ar vartotojo apsaugos, jis palaiko pertraukimu pagrįstą daugiafunkcinį darbą ir privilegijų anuliaciją per išorinės atminties susiejimą. Todėl sistemos programuotojai vis vien apibrėžti saugomas atminties sritis. (Zilog, 2016.)

MCS-48 mikrovaldiklių šeima turi yra susiskirstyta į du pagrindinius atminties išdėstymo būdus: programos ir duomenų atmintį. Kiekvienas būdas turi skirtingas charakteristikas, kurios yra prieinamos programuotojui. Programų atmintis 8048 ir 8748 modeliuose yra 1K x 8 žodžiai, tačiau 8035 naudoja išorinę programos atmintį. Svarbu atkreipti dėmesį, kad pirmosios trys programos atminties vietos turi specialias funkcijas: adresas 0 (angl. address 0) yra atstatymo vektoriaus vieta, 3 adresas (angl. address 3) atitinka išorinio pertraukimo vektoriaus vietą, o adresu 7 (angl. address 7) yra laikmačio pertraukimo tvarkyklė. Tokios iš anksto nustatytos vietos palaiko pagrindinį sistemos inicijavimą ir pertraukų aptarnavimą. Prieiga prie išorinės atminties taip pat palaikoma naudojant išorinės prieigos (EA) režimą, kuris išjungia vidinę programos atmintį ir leidžia vykdyti programą iš išorinės saugyklos, taip palengvinant derinimą ir testavimą. (Intel Corporation, 1977)

Duomenų atmintis yra saugoma 64 baitai RAM (atmintyje). Iš tikrųjų ji yra padalinta į dvi dalis: tiesiogiai adresuojamus darbinius registrus (vietos 0–7) ir netiesiogiai adresuojamą bendrosios paskirties atmintį. Pavyzdžiui, programuotojai gali naudoti tokią instrukciją, kaip SEL RB, norėdami pasirinkti vieną iš dviejų veikiančių registrų bankų. Tai užtikrina ypač efektyvų konteksto perjungimą tarp pertraukimo ir pagrindinės programos vykdymo. Likusi duomenų atminties dalis veikia kaip bendrosios paskirties RAM arba steko vieta (angl. stack space). Šios atminties sritys atlieka dvejopą paskirtį, dar labiau padidindamos atminties naudojimą griežtai ribotoms programoms. Architektūros paprastumas užtikrina, kad visos šios funkcijos būtų matomos programuotojui be jokių papildomų trukdžių, todėl yra gerai suderintos su įterptinėms sistemoms, kurioms yra būdingos vieno lygio operacija. (Intel Corporation, 1977)

# Komandų sistema (ISA) Z80 ir MCS-48/8048/8035/8748 architektūrose

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. (Zilog, 2016.)

**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).

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. (Intel Corporation, 1977)

**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.

# Z80 ir MCS-48/8048/8035/8748 mikroarchitektūra

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. (Zilog, 2016.)

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. (Intel Corporation, 1977)

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)
3. 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.
4. Programos būsenos žodis (angl. Program Status Word – PSW). PSW apima būsenos vėliavėles, tokias kaip Carry ir Auxiliary Carry, bei vartotojo apibrėžiamą vėliavėlę F0. Taip pat valdo Steko rodyklę (SP), kuri leidžia valdyti iki 8 lygių subprogramų įdėjimą.
5. Laikmačiai ir skaitikliai (angl. Timers and Counters). - 8 bitų, gali veikti kaip laikmatis naudojant padalintą laikrodį (pvz., 80 µs raiška su 6 MHz laikrodžiu) arba kaip įvykių skaitiklis, kurio reikšmė didinama išoriniais signalais per T įvestį.
6. Įvesties/išvesties prievadai (angl. I/O Ports). 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 ir MCS-48/8048/8035/8748 architektūrose

**Z80:**

Paveikslėlis, kuriame yra tekstas, ekrano kopija, programinė įranga, Kompiuterio piktograma

Automatiškai sugeneruotas aprašymas

pav. 2 Z80 Assembler to display the message "Hello world!" (Pav nuoroda: <https://www.msx.org/wiki/Z80_Assembler_for_Dummies>)

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

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

Automatiškai sugeneruotas aprašymas

pav. 3 MCS-48 Assembly Example: BCD Addition Routine

Kodo paaiškinimas:

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

Automatiškai sugeneruotas aprašymas

pav. 4 Paveikslėlio (pav. 3) Kodo paaiškinimas

# Aukšto lygio programavimo kalbų palaikymas Z80 ir MCS-48/8048/8035/8748 architektūrose

Z80 mikroprocesorius yra sukurtas taip, kad efektyviai palaikytų aukšto lygio programavimo kalbas (angl. high-level programming languages (HLLs)), tokias kaip C, Pascal ir net FORTRAN. Vienos iš svarbiausių funkcijų yra trys poros bendros paskirties registrų (BC, DE ir HL), kurie yra tinkami 16 bitų operacijoms, todėl yra gerai yra pritaikyti adresuoti ir tvarkyti steką (frames, arrays ir pointers) Be to, IS ir IY indeksų registrai yra veiksminga vieta adresuoti atminčiai, tai leidžia tiesiogiai indeksuoti struktūrinių duomenų ir masyvų adresus, o tai būtina tokioms programavimo kalboms kaip C. Z80 palaiko SP funkcijų iškvietimą ir vietinių kintamųjų valdymą. Sąlyginės šuolio instrukcijos ir visapusiškos aritmetinės, loginės operacijos taip pat leidžia optimaliai generuoti kodą iš kompiliatorių. Todėl, dėl tokių savybių Z80 yra gana efektyvus, atsižvelgiant į tokių HLL instrukcijų (loops, braches ir funkcijų) iškvietimą ir vykdymą. (Zilog, 2016.)

MCS-48 seriją sudaro 8048, 8748, 8035 ir dar daug kitų mikrovaldiklių. Visi jie pasižymi gera architektūra ir plačiu instrukcijų rinkiniu, kuris palaiko aukšto lygio programavimo kalbas. Šie mikrovaldikliai turi kompaktišką instrukcijų rinkinį, kuriame 70% instrukcijų yra vieno baito, tuo trapu likusios – dviejų baitų. Tai užtikrina optimizuotą atminties naudojimą ir spartų vykdymo greitį. Kompaktiškas dizainas suteikia galimybę iš aukšto lygio konstrukcijų, tokių kaip kilpų (angl. loops), sąlyginių sąlygų ir funkcinių iškvietimų, šias konvertuoti į efektyvų mašininį kodą. Tokie įrankiai kaip Intet PROMPT 48 sistema ir MCS-48 Macro Assembler – dar labiau supaprastino asemblinės kalbos naudojimą, kuris dažnai tarnavo kaip tiltas palaikant aukštesnio lygio programavimo pagrindus MCS-48 mikrovaldiklikliuose. Šios programos palaiko makrokomandas, sąlyginį surinkimą ir simbolinį programavimą, o tai leidžia geriau pavaizduoti slypinčią logiką ir pakartotinai naudoti kodą įvairiuose programose. Todėl, tai suteikta vartotojams kurti struktūrizuotas, modulines programas, kurios veiksmingai palaiko aukšto lygio programavimo metodus. Be to, In-Circuit Emulators (ICE-48) – suteikia testavimo priemones, kurios mažino atotrūkį tarp žemo lygio mašininio kodo ir aukšto lygio programinės įrangos kūrimo procesų MCS-48 mikrovaldikliuose. (Intel Corporation, 1977)

# Įvesties / išvesties ir pertraukimų mechanizmai Z80 ir MCS-48/8048/8035/8748 architektūrose

Z80 iš viso valdo 256 nepriklausomus įvesties/išvesties prievadus, kurie tiesiogiai adresuojami 8 bitų adresu. Įvesties/išvesties operacijos atliekamos naudojant tokias instrukcijas kaip IN ir OUT. IN – komanda atsakinga už duomenų įvedimą į akumuliatorių, o OUT – iš akumuliatoriaus. Tuo tarpu, apatiniai 8 adresų magistralės bitai turi prievado adresą, o viršutiniai bitai yra ignoruojami. (Zilog, 2016.)

Z80 sistema tvarko asinchroninius įvykius naudodama pertraukimus (angl. interrupts), taip leidžiant mikroprocesoriui greitai reaguoti į išorines įvestis ir vidinius signalus, naudojant minimalias programinės įrangos sąnaudas. Z80 siūlo tris pertraukimo režimus: (Zilog, 2016.)

1. 0 režimas (IM 0) – CPU vykdo vieno baito instrukcijas, kurias pateikia pertraukiantis įrenginys. Tai leidžia periferiniam įrenginiui priverstinai vykdyti RST komandą (pvz. RST 38h), kuri peršoka į fiksuotą adresą, kad aptarnautų pertraukimą.
2. 1 režimas (IM 1) – CPU reaguoja į pertraukimą automatiškai vykdydama perkrovimo operaciją adresu 0038h. Tai yra fiksuota vieta atmintyje, kuri įprastai yra naudojama paprastesniam pertraukimų aptarnavimui, be jokio išorinio įsikišimo.
3. 2 režimas (IM 2) – tai pats universaliausias režimas, kai procesorius gali pereiti į bet kurią atminties vietą, kad atliktų pertraukimą. Pertraukiantis įrenginys pateikia 8 bitų vektorių, kuris sujungiamas su pertraukimo puslapio adresu (I Registru), kad sudarytų 16 bitų pertraukimo paslaugos rutinos adresą. Šis vektorinės lentelės mechanizmas leidžia dinamiškai įvykdyti kelis pertraukimus.

Z80 turi dviejų tipų pertraukimus: vieną maskuojamą (angl. Maskable Interrupt (INT)) ir vieną nemaskuojamą (angl. Non-Maskable Interrupt (NMI)). INT yra jautrus lygiui ir įjungiamas naudojant komandą EI arba išjungiamas naudojant DI komandą. Jei yra aptinkamas pertraukimas, CPU baigia vykdyti esamą komandą prieš priimdamas pertraukimą. Tuo tarpu, NMI yra krašto suaktyvinta (angl. edge-triggered) įvestis, kurios negali išjungti jokia programa. NMI priverčia centrinį procesorių nedelsiant pereiti į adreso vietą 0066h, kad būtų vykdoma neužmaskuojama pertraukimo paslauga. Taigi, Z80 pertraukimo mechanizmas kartu su plačiomis įvesties ir išvesties galimybėmis užtikrina efektyvų ir lankstų periferinių įrenginių valdymą, todėl yra tinkamas naudoti realaus laiko sudėtingoms programoms. (Zilog, 2016.)

MCS-48 mikrovaldikliai turi galingus įvesties/išvesties (I/O) ir pertraukimo mechanizmus. Ši serija turi 27 I/O linijas, kurios yra padalintos į 8 bitų dvikrypčius prievadus (prievadas 1 ir 2) ir dvikryptį BUS prievadą. Tai leidžia atlikti dinamines įvesties ir išvesties operacijas, o BUS prievadas gali būti naudojamas atmintyje susietose I/O operacijose, skaitymo ir rašymo ciklų metu naudojant RD ir WR valdymo signalus. Be to, 2 prievadas taip pat atlieka papildomas funkcijas, tokias kaip išorinės atminties adresavimas programos gavimo operacijų metu. T0, T1 ir INT yra specialūs funkcijų tikrinimo įėjimai (T0 gali būti naudojama kaip laikrodžio išvestis). (Intel Corporation, 1977)

MCS-48 palaiko du pertraukimo šaltinius:

1. Išorinis pertraukimas (angl. INT-External Interrupt). Jei yra įjungtas naudojant EN I instrukciją – tai yra sukuriamas priverstinis šuolis į atminties vietą (3). INT taip pat gali būti panaudojamas atliekant sąlyginius šuolius, tokius kaip JNI​MCS-48.
2. Laikmačio / skaitiklio pertraukimas (angl. Timer/Counter Interrupt) – tai vidinis 8 bitų laikmatis, skaitiklis, kuris gali generuoti pertraukimą, kai yra patiriamas perpildymas (angl. overflow). Laikmačio pertraukimas yra įjungiamas naudojant EN TCNTI komandą, kuri įvykdo šuolį į 7 vietą programos atmintyje.

Šiame procesoriuje yra vieno lygio pertraukimas ir joks kitas pertraukimas negali būti aptarnaujamas, kol ankstesnis nebus visiškai įvykdytas kartu su RETA vykdymu, kad būtų atkurta programos MCS-48 būsena. MCS-48 architektūros I/O ir pertraukimo mechanizmai suteikia lankstumo ir efektyvumo realaus laiko valdymo sistemose, užtikrinant patikimą periferinių įrenginių valdymą ir įvykių aptarnavimą​. (Intel Corporation, 1977)

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

Z-80 ir MCS-48 architektūros labai skiriasi funkcijomis, sudėtingumu ir dizainu. ,,Zilog“ sukurtas Z80 dizainas yra universalus ir pažangus 8 bitų procesorius skirtas įvairiems kompiuteriams. Tuo tarpu, ,,Intel“ sukurta MCS-48 serijos mikrovaldikliams pagrindinis dėmesys yra skirtas įterptoms valdymo programoms. Pagrindiniai skirtumai tarp šių dviejų architektūrų, tam tikrose kategorijose:

* **Architektūra ir sudėtingumas.** Z80 pasižymi sudėtingesne architektūra su 208 bitų vidiniu registru, įskaitant alternatyvius registrų rinkinius, bendrosios paskirties registrus (BC, DE, HL), indeksų registrus (IX, IY), steko rodyklę (angl. Stack-pointer) ir programinį skaitiklį. Jame taip pat yra keli pertraukimo režimai ir patobulintos instrukcijos. Tuo tarpu, MCS-48 yra kur kas paprastesnės struktūros. Turi vieną veikiantį registrų banką, 8 lygių techninės įrangos steką (angl. 8-level hardware stack) ir 64 baitų vidinę RAM atmintį.
* **Adresavimo režimai.** Z80 siūlo platesnį adresavimo režimų spektrą, įskaitant tiesioginį, išplėstinį, indeksuotą, santykinį ir netiesioginį adresavimą. MCS-48 turi ribotus adresavimo režimus, dažniausiai tiesioginius, netiesioginius ir santykinius (1K x 8) programų atmintyje ir mažoje vidinėje RAM atmintyje.
* **Instrukcijų rinkinys.** Z80 turi daugiau nei 150 instrukcijų (įskaitant bitų perdavimo, paieškos ir bitų manipuliavimo instrukcijas). Visos jos palaiko 16 bitų aritmetines ir išplėstines valdymo operacijas, tokias kaip DHNZ, EX ir BIT instrukcijas. Nors MCS-48 turi kompaktiškesnį instrukcijų rinkinį (apie 90), jų dauguma yra orientuotos į 8 bitų operacijas. Todėl šis procesorius yra gerai optimizuotas dirbti su paprastesnėmis užduotims.
* **Atmintis.** Z80 gali tiesiogiai tvarkyti 64 KB atminį. Tuo tarpu, MCS-48 turi itin ribotą atmintį, dalytą į programų atmintį (1 KB) ir duomenų atmintį (64 baitai) su ribotomis išplėtimo galimybėmis.
* **Pertraukimų valdymas.** Z80 palaiko daugybę pertraukimo režimų (mode 0, Mode 1 ir Mode 2), įskaitant vektorinius pertraukimus, kad būtų galima greitai ir lanksčiai reaguoti į išorinius įrenginius. MCS-48 palaiko tik pagrindinius vieno lygio pertraukimus.
* **Laikmačiai ir skaitikliai.** Z80 įmontuoto laikmačio / skaitiklio, todėl šiai funkcijai turi būti naudojami išoriniai periferiniai komponentai. MCS-48 turi 8 bitų laikmatį / skaitiklį, kuris gali veikti įvykių skaičiavimo arba laiko režimu. Taigi, jis yra gerai tinkamas valdymo programoms.
* **Laikrodis ir greitis.** Z80 palaiko 6-20 MHz dažnį, todėl geba užtikrinti našumą sudėtingiems skaičiavimams atlikti. MCS-48 greitis yra mažesnis (1-6 MHz), taip teigdamas pirmenybę efektyvumui, o ne našumui.

Apibendrinant galima teigti, kad Z80 yra labai universalus ir galingas mikroprocesorius, turintis registrų, adresavimo režimų ir instrukcijų rinkinį, skirtą bendrosios paskirties programoms. MCS-48 mikrovaldiklis yra labiau optimizuotas įterptoms programoms, nes jame yra integruoti laikmačiai, I/O įvestis. Todėl ši architektūra yra labiau pritaikyta mažesnės galios sistemoms. Dėl šių skirtumų Z80 yra labiau tinkas sudėtingoms programinės įrangos užduotims, o MCS-48 idealiai tinka į valdymą orientuotoms programoms.

# Cituojama literatūra:

**Intel Corporation (1977).** *MCS-48 Microcomputer User's Manual*. Available at: <https://vtda.org/docs/computing/Intel/98-270B_MCS-48MicrocomputerUsersManualJul77.pdf> (Accessed: 15 December 2024).

**Įrašas:** **EE Journal (2022).** *A History of Early Microcontrollers: Part 4 – The Intel 8048 and 8748*. *EE Journal*. Available at: <https://www.eejournal.com/article/a-history-of-early-microcontrollers-part-4-the-intel-8048-and-8748/> (Accessed: 15 December 2024).

**Landley, R. (n.d.).** *The Z80 Microprocessor*. Available at: <https://landley.net/history/mirror/cpm/z80.html> (Accessed: 15 December 2024).

**OpenAI (2024).** *ChatGPT: Generative Pre-trained Transformer, versija 3.5*. Available at: <https://chat.openai.com> (Accessed: 17 December 2024, 15:00). Prompt: "Duomenų tipai Z80 ir MCS-48/8048/8035/8748 architektūrose"

**Tallen, T. (n.d.).** *Chapter 3: Z80*. Available at: <https://cs.furman.edu/~tallen/csc475/materials/Chapter%203%20Z80.pdf> (Accessed: 15 June 2024).

**Zilog, Inc. (2016).** *Z80 CPU User Manual*. Available at: <https://www.zilog.com/docs/z80/um0080.pdf> (Accessed: 15 December 2024).