

VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS

FAKULTETAS

Techninis aprašymas

Motorola 68H05 architektūra

Vasarė Pratužaitė Informatikos 2 kursas, 1 grupė, 2 pogrupis



VILNIUS 2023

Turinys

<i>Istorinis ir techninis kontekstas.....</i>	<i>3</i>
<i>Registrai.....</i>	<i>3</i>
<i>Duomenų tipai.....</i>	<i>5</i>
<i>Adresavimo režimai</i>	<i>5</i>
<i>Atminties struktūra</i>	<i>7</i>
<i>Komandų Sistema.....</i>	<i>8</i>
<i>Mikroarchitektūra</i>	<i>8</i>
<i>Mašinos kodo ar assemblerio pavyzdys.....</i>	<i>9</i>
<i>Aukšto lygio programavimo kalbų palaikymas</i>	<i>9</i>
<i>Įvesties-išvesties ir pertraukimų aprašymas.....</i>	<i>10</i>
<i>Šaltiniai</i>	<i>11</i>

Istorinis ir techninis kontekstas

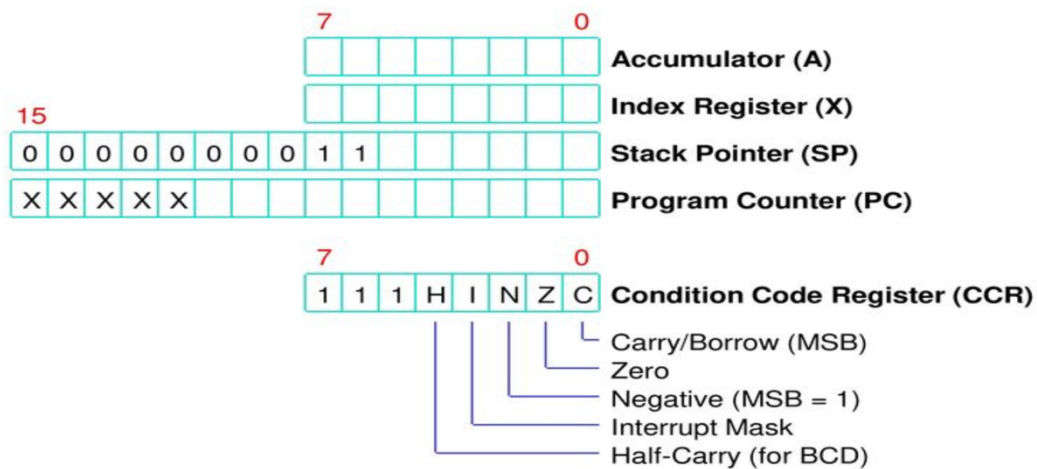
Motorola 68HC05 architektūra – tai mikrovaldiklis, kuris buvo sukurtas devintajame dešimtmetyje bei buvo plačiai naudojamas 1980-aisiais ir 1990-aisiais. Kaip ir visi „Motorola“ procesoriai, kurie dalijasi 6800 modelio linija, jie naudoja von Neumann architektūrą. Tai buvo naudojama įvairiose automobilių sistemose, pramoninėse valdikliuose ir daugelyje kitų įrenginių sistemose. Ši architektūra buvo pagrįsta „Motorola 6800“ mikroprocesoriumi ir buvo sukurta naudoti įterptosiose sistemose, kurios naudojamos valdyti konkrečias didesnio įrenginio ar sistemos funkcijas kompiuterių sistemose. Ji gerai tiko šiose sistemose, nes buvo mažo dydžio, todėl sunaudodavo mažai energijos, turėjo skirtingas įvesties / išvesties galimybes, kurios buvo atmintyje suskirstytos į vieną linijinę adresų erdvę. Jos atminties žemėlapis yra suskirstytas į baitus, kurio pradžia: 0000 USD, o pabaiga: \$xxFF, kur, atsižvelgiant į tokius veiksnius kaip ROM arba EPROM masyvo dydis, „xx“ yra 03 USD, 07 USD, 0F, 1F, 3F arba 7F USD. Adresų erdvė 68HC05 įrenginyje dažniausiai būna gan didelė, kad jame būtų integruotas ROM arba EPROM, RAM ir valdymo registrai. Viena iš pagrindinių šios architektūros ypatybių yra gebėjimas vykdyti kodą, saugomą lustiniame ROM, todėl jį buvo galima naudoti programose, kuriose išorinė atmintis nebuvo prieinama. Taip pat buvo daug integruotų periferinių įrenginių, pavyzdžiui, laikmačiai, skaitikliai ir nuosekliojo ryšio prievadai, kurie palengvino sąsają su įvairiais išoriniais įrenginiais. Vėliau ši architektūra buvo pakeista į naujesnius mikrovaldiklius, kurie gerina našumą ir papildomas funkcijas, tačiau jis vis dar yra naudojamas, nes yra patikrintas ir ištestuotas. 1994 metais „Motorola“ išleido paskutinius 68H05 procesorius. Motorolos modelis turintis pavadinimą HC05 dabar laikoma sena ir yra pakeista HC(S)08 MCU serija.

Registrai

- **Akumuliatorius (A):**
tai 8 bitų pagrindinis registras, kuriame galima saugoti duomenis iš atminties. Taip pat galima duomenis įrašyti iš atminties. Jis yra vienintelis registras, kuriame galima atlikti aritmetines ir logines operacijas. Palaiko bitų poslinkio ir pasukimo operacijas. Dauguma akumuliatoriaus operacijų turi įtakos būsenos vėliavėlėms sąlygų kodų registre.
- **Indekso registras (X):**
yra 8 bitų registras, naudojamas indeksuotai adresavimo vertei sukurti efektyvų adresą. Indekso registras taip pat gali būti naudojamas kaip laikina saugojimo vieta.
- **Steko rodyklė (SP):**
tai 16 bitų registras. Steko rodiklis (SP) saugo adresą tolesnės laisvos vietos steke. Per MCU atstatymą arba atstatymo steko rodiklio (RSP) komandą steko rodiklis nustatomas į vietą \$00FF. Steko rodiklis tada mažinamas, kai duomenys įkeliami į steką, ir didinamas, kai duomenys išimami iš steko. Kai prieinama prie atminties, aštuoni reikšmingiausi bitai yra nuolat nustatyti į 00000011. Šie aštuoni 0 bitai pridedami prie šešių mažiausios reikšmės registro bitų, kad būtų sukurtas adresas, esantis diapazone nuo \$00FF iki \$00C0. Jei viršijami 64 vietos, steko rodiklis praranda anksčiau saugomą informaciją. Čia yra išsaugojama procesoriaus būsena.
- **Programos skaitiklis (PC):**

tai 14 bitų registras, kuriame yra kito baido, kurį reikia gauti adresas. Taip pat saugomas kitos komandos adresas. Jis juda po vieną baitą į priekį, nes instrukcijos ir operandai gaunami įprasto programos vykdymo metu.

- **Sąlygų kodų registras (CCR):**
tai 5 bitų registras, kuriame saugoma dabartinė procesoriaus būseną, įskaitant nulinę vėliavėlę (Z), nešiojimo vėliavėlę (C) ir neigiamą vėliavėlę (N).
- **Laikmačio registrai:**
68HC05 turi kelis laikmačio registrus, kurie gali būti naudojami generuoti pertraukimų arba uždelimo įvykius, įskaitant laikmačio skaitiklio registrą (TCNT), laikmačio palyginimo registrą (TOC) ir laikmačio valdymo registrą (TCTL).
- **Nuosekliojo ryšio registrai:**
68HC05 turi daugybę registrų, kurie naudojami valdyti ir palaikyti ryšį per jo nuosekliojo ryšio prievadus, įskaitant nuoseklųjį duomenų registrą (SDR), nuosekliojo valdymo registrą (SCR) ir serijos būsenos registrą (SSR).
- **Prievadų registrai:**
68HC05 turi daugybę prievadų registrų, kuriuos galima naudoti norint valdyti įvairių mikrovaldiklio įvesties/išvesties kaiščių įvesties ir išvesties duomenis.
- **Pertraukimų registrai:**
Pertraukimo kaukė (Interrupt Mask) registras yra 68HC05 mikroprocesoriaus registras, naudojamas įjungti arba išjungti pertraukimus. Registre yra aštuoni bitai, po vieną kiekvienam pertraukimo šaltiniui. Kai registro bitas yra nustatytas į 1, įjungiamas atitinkamas pertraukimas. Kai bitas nustatytas į 0, atitinkamas pertraukimas išjungiamas. Pertraukimai, kuriuos galima įjungti arba išjungti, yra: NMI: neužmaskuojamas pertraukimas IRQ: maskuojamas pertraukimas TRAP: TRAP Instrukcijos pertraukimas SWI: programinės įrangos pertraukimas Neigiamas (MSB = 1) Neigiama vėliavėlė yra 68HC05 mikroprocesoriaus būsenos registro vėliavėlė, kuri nustatoma, kai reikšmingiausias skaičiaus bitas (MSB) yra 1. Ši vėliavėlė naudojama nurodyti, kad skaičius yra neigiamas.
- **Half-Carry (skirta BCD):**
68HC05 mikroprocesoriaus būsenos registro vėliavėlė, kuri nustatoma, kai pridėjami arba atimami keturi apatiniai BCD numerio bitai. Ši vėliavėlė naudojama norint nurodyti, kad apatinių keturių bitų perdavimo bitą reikia pridėti prie keturių viršutinių bitų arba atimti iš jų.
- **Zero:**
Nulio vėliavėlė yra 68HC05 mikroprocesoriaus būsenos registro vėliavėlė, kuri nustatoma, kai aritmetinės operacijos rezultatas yra 0. Ši vėliavėlė naudojama norint parodyti, kad operacija buvo sėkminga.
Nešioti / skolintis (MSB)
- **Carry/Borrow:**
Nešiojimo ir skolinimo vėliavėlė yra 68HC05 mikroprocesoriaus būsenos registro vėliavėlė, kuri nustatoma, kai yra perkeliamas arba skolinamasi iš svarbiausio skaičiaus bito (MSB). Ši vėliavėlė naudojama norint nurodyti, kad operacija galėjo būti perpildyta arba nepakankama.



1pav 68HC05 Programmer's mode

Duomenų tipai

68HC05 yra mikrovaldiklis, kurio duomenų tipai riboti iki 8 bitų ir 16 bitų sveikųjų skaičių. Įrenginys turi kelis registus, įskaitant akumuliatorių (A) ir indekso registrą (X), kiekvienas iš jų yra 8 bitų ilgio. Taip pat yra steko rodyklė (SP) ir programos skaitiklis (PC), abu 16 bitų ilgio, kurie gali būti naudojami duomenims saugoti. Architektūra palaiko įvairius duomenų tipus, įskaitant simbolių eilutes, slankiojo kabelio skaičius ir masyvus. Šie duomenų tipai gali būti naudojami su 8 bitų ir 16 bitų sveikųjų skaičių duomenų tipais naudojant specialias instrukcijas arba programinės įrangos bibliotekas. 68HC05 yra patogus mikrovaldiklis, o kelių baitų duomenų tipai saugomi su mažiausiais reikšmingais baitais žemiausiuose atminties adresuose. Pavyzdžiui, 16 bitų sveikasis skaičius būtų saugomas su mažesniu reikšminguoju baitu žemiausiu adresu, o didesniu reikšminguoju – aukštesniu adresu. Tai leidžia efektyviai valdyti ir pasiekti duomenis, taikant mažiausią atminties vietą.

Adresavimo režimai

68HC05 mikrovaldiklio architektūra ne tik siūlo galingas instrukcijas, bet ir lanksčius adresavimo režimus, kad būtų galima efektyviai pasiekti įvairių tipų atmintyje paskirstytus duomenis. Be 65 pagrindinių instrukcijų, 68HC05 pateikia aštuonis adresavimo režimus, kurie nustato šių nurodymų duomenų šaltinį ir (arba) paskirties vietą.

Inherent	Immediate	Extended	Direct	Indexed, 16-Bit Offset	Indexed, 8-Bit Offset	Indexed, No Offset	Relative	Bit Set and Clear	Bit Test and Branch
(INH)	(IMM)	(EXT)	(DIR)	(IX2)	(IX1)	(IX)	(REL)	(BS)	(BTB)

1 lentelė. Adresavimo režimai, atitinkantys skirtingų 68HC05 instrukcijų duomenų reikalavimus.

1. Įgimtas adresavimas (INH):

Operandas yra numanomas instrukcijos operacijos kodu. Pavyzdžiui, tokios instrukcijos kaip ASLA, CLRX, DECA, INCX ir NOP. Šios instrukcijos reikalauja tik vieno baido programos saugyklos.

2. Skubus adresavimas (IMM):

Operandas iš karto seka atmintyje esantį instrukcijos operacijos kodą. Naudojamas pastovioms reikšmėms nurodyti programos vykdymo metu. Simbolizuojamas # ženklu. Instrukcijų pavyzdžiai: LDA #\$40, ADD #\$01 ir AND #\$03.

3. Išplėstinis adresavimas (EXT):

Leidžia skaityti arba rašyti į bet kurią 68HC05 atminties žemėlapiu vietą. Pavyzdžiui, LDA \$ 4000, JMP \$ 1200 ir SBC \$ 01F1. Reikia trijų baitų programos saugyklos.

4. Tiesioginis adresavimas (DIR):

Instrukcijos gali skaityti arba įrašyti atminties vietas nuo \$00 iki \$FF. Palaiko skaitymo-keitimo-rašymo operacijas lustinėje RAM arba periferinio valdymo registruose. Instrukcijų pavyzdžiai: ASL \$00, INC \$FB ir ROL \$F7.

5. Indeksuotas adresavimas su 16 bitų poslinkiais (IX2):

Tiksliniai adresai, apskaičiuojami sudedant indekso registro turinį ir 16 bitų poslinkį. Palaiko tokias operacijas kaip LDA \$4000,X ir JSR \$0F4C,X. Reikia trijų baitų programos saugyklos.

6. Indeksuotas adresavimas su 8 bitų poslinkiais (IX1):

Panašus į IX2, bet naudoja 8 bitų poslinkius, pasiekiant atminties vietas nuo 0000 USD iki 01FE. Reikia vienu ciklu mažiau ir vienu baitu mažiau saugyklos nei IX2.

7. Instrukcijų pavyzdžiai:

ASL \$00,X, INC \$FB,X ir TST \$F5,X. Indeksuotas adresas (IX): Tikslinis adresas yra indekso registro turinys, išplėstas iki 16 bitų. Palaiko skaitymo, modifikavimo ir rašymo instrukcijas, tokias kaip ASL, X ir DEC, X. Vykdoma taip pat greitai, kaip tiesioginis režimas, tačiau reikia tik vieno baido saugyklos.

8. Santykinis adresavimas (REL):

Naudojamas tik pagal šakų instrukcijas tiksliniams adresams apskaičiuoti, palyginti su programos skaitikliu (PC). Reikia dviejų baitų saugyklos ir vykdoma per tris laikrodžio ciklus.

9. Instrukcijos pavyzdys:

BEQ (Branch if Equal). Bitų nustatymas ir išvalymas (BSC) adresavimas: Naudojamos BSET ir BCLR instrukcijos, skirtos keisti konkrečius bitus baite

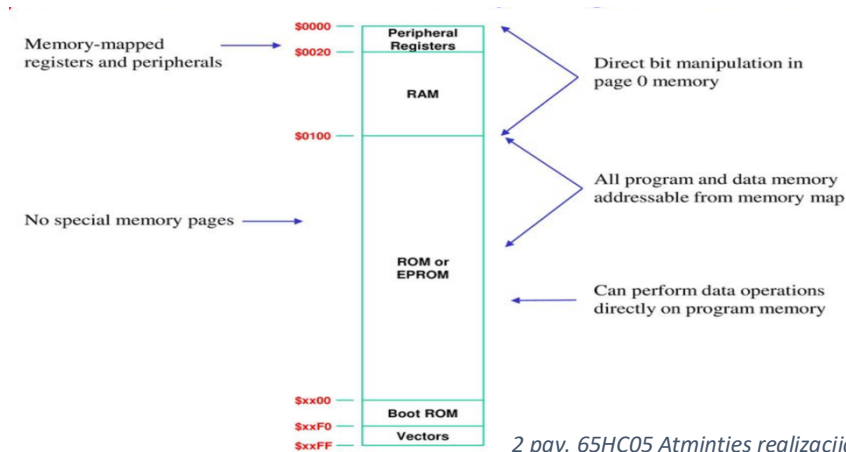
Bitų bandymas ir atšakos (BTB) adresavimas: Naudojamas BRSET ir BRCLR instrukcijose, kad patikrintų konkrečius bitus ir šakas pagal rezultatą. Panašus į BSC, bet apima santykinį išsišakojimą.

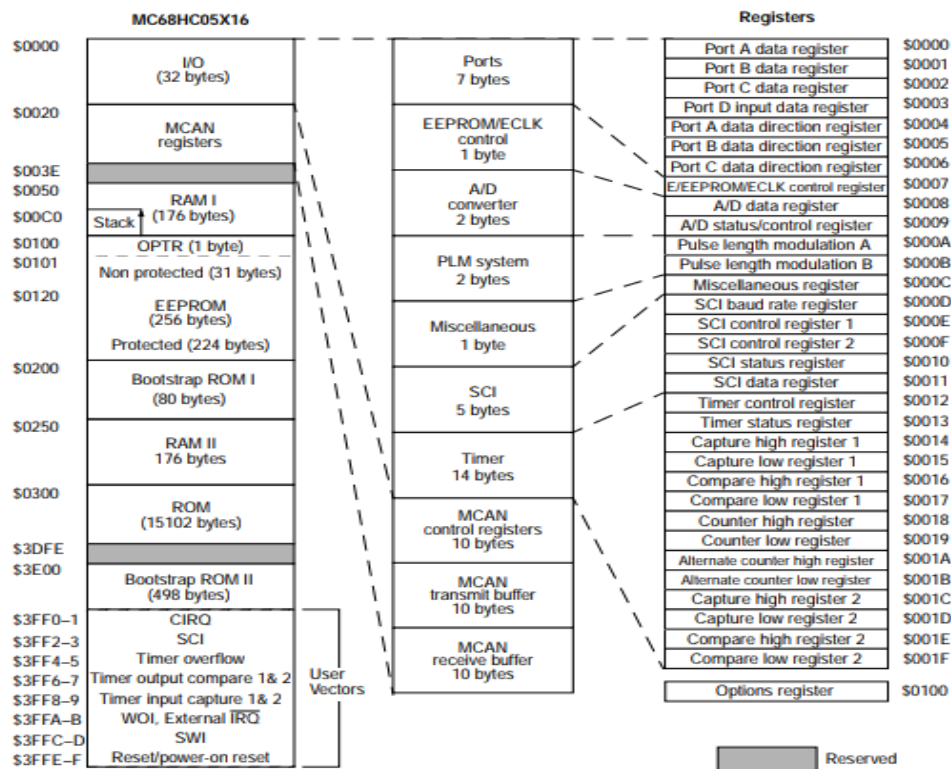
10. Šaltinio kodo formatas:

BRSET/BRCLR bitų numeris, tiesioginis_adresas, poslinkis. Adresavimo režimų įvairovė 68HC05 architektūroje leidžia efektyviai manipuluoti duomenimis įvairiuose scenarijuose, padidinant mikrovaldiklio universalumą.

Atminties struktūra

Vartotojo RAM, nuo \$0050 iki \$00FF, susideda iš 176 baitų atminties, o stekas užima vietą nuo \$00FF iki \$00C0. Taip pat yra papildoma 176 baitų vartotojo RAM dalis nuo \$0250 iki \$02FF, kuri yra visiškai prieinama vartotojui. TuometROM/EPROM dalyje visa program air atmintis adresuojama iš atminties žemėlapiu. Svarbu pabrėžti, kad tikslus „Motorola 68H05“ atminties išdėstymas gali kisti priklausomai nuo konkretaus įrenginio ar sistemos, kurioje jis naudojamas. Kai kurie įrenginiai gali turėti papildomų atminties ar kitų aparatinės įrangos išteklių, kurie nėra visuose 68H05 sistemose.





3pav. Platesnė architektūros atminties realizacija

Komandų Sistema

- ADD: prideda nurodytą reikšmę prie akumulatoriaus.
- AND: vykdomo metu atlieka bitų ir operaciją tarp akumulatoriaus ir nurodytos atminties vietos arba tiesioginės reikšmės.
- BIT: tikrina vieno bito reikšmę akumuliatoriuje ir nustato N and Z vėliavėles.
- BRANCH: atlieka sąlyginę arba besąlyginę šaką į naują atminties vietą, priklausomai nuo sąlygos.
- CMP: palygina akumulatoriaus vertę su operandu.
- DEC: sumažina atminties vietą.
- INC: padidina atminties vietą.
- JMP: vykdo besąlyginį šuolį į naują atminties vietą.
- LDA: įkelia į akumuliatorių atminties vietos nurodytą reikšmę arba tiesioginę.
- STA: išsaugo akumulatoriaus vertę atminties vietoje.
- SUB: atima atminties vietos reikšmę arba tiesioginę reikšmę iš akumulatoriaus.

Mikroarchitektūra

68HC05 nebuvo mikroprogramuojamas. Jo architektūra buvo įgyvendinta betarpiškai aparaturoje. Tai reiškia, kad procesoriaus instrukcijos buvo įmontuotos į jo paties silicio lustą. Tai leido 68HC05 veikti greičiau ir efektyviau nei mikroprogramuojamiems procesoriams. Jis taip pat nėra „minkštas“, nes nėra naudojamas simuliacijose ir mokymosi programose.

Šio mikrovaldiklio mikroarchitektūra yra sudaryta iš šių komponentų ir savybių:

- Procesorius: 68HC05 mikrovaldiklio procesorius yra 8 bitų, todėl jis gali apdoroti 8 bitų informacijos vienetą vienu metu. Procesorius vykdo įvairias instrukcijas, skirtas valdyti įrenginio veikimą.
- Registrų sistema: mikrovaldiklis turi keletą registrų, įskaitant akumuliatorių (A), indekso registrą (X), steko rodyklę (SP), programos skaitiklį (PC) ir kitus. Akumuliatorius naudojamas kaip laikinas saugojimo vienetas rezultatams, o indekso registras gali būti naudojamas adresavimo reikmėms.
- Atmintis: 68HC05 turi integruotą RAM ir ROM, kurios naudojamos laikinai saugoti duomenis ir programinę įrangą. Adresų erdvė leidžia mikrovaldikliui naudoti 64 KB atminties.
- Instrukcijų rinkinys: instrukcijų rinkinys yra įvairus ir apima veiksmus, tokius kaip duomenų perkėlimas, matematikos operacijos, šuoliai į atminties vietas ir kt. Kiekviena instrukcija yra susieta su tam tikru operaciniu kodu (opcode), kuris nurodo, ką konkreči instrukcija turėtų atlikti.
- Adresavimo režimai: 68HC05 palaiko įvairius adresavimo režimus, įskaitant inherent, immediate, extended, direct, indexed su 8 ir 16 bitų offsetais, relative ir kitus. Šie režimai leidžia lankstiai valdyti duomenų prieigą.
- Pertraukimai: mikrovaldiklis palaiko pertraukimus, kurie leidžia jam įsitraukti į kitas užduotis arba reaguoti į išorinius įvykius.

Mašinos kodo ar assemblerio pavyzdys

```

ORG $1000      ; Pradinė programos adreso vieta

START:
    LDA #$01    ; Įkrauname reikšmę į akumuliatorių
    STA $2000   ; Įrašome reikšmę į atminties vietą $2000
    INX         ; Padidiname X registrą
    STX $2001   ; Įrašome X į atminties vietą $2001

    BRA START   ; Šuolis į programos pradžią (begalinė ciklo dalis)

    END         ; Programos pabaiga

```

4pav Kodo pavyzdys

Aukšto lygio programavimo kalbų palaikymas

- C kalba: Motorola 68HC05 gali būti naudojamas C kalbos programavime.
- Pascal kalba: aplinka palaiko "Motorola 68HC05" mikrovaldiklį ir yra ypač naudojama.
- Forth kalba: Forth yra steko pagrindu sukurta programavimo kalba, kuri gali būti labai svarbi mikrovaldikliui Motorola 68HC05.

- BASIC kalba: tik kai kurios BASIC kalbos interpretuojančios aplinkos gali būti susijusios su "Motorola 68HC05".
- Ada kalba: stipri ir tipizuota programavimo kalba, kurioje gali būti naudojama "Motorola 68HC05".

Šias kalbas galima būtų panaudoti, kad jų pagalba parašytume kodą, kurį būtų galima sukompiliuoti. Taip pat aukštų programavimo kalbų padedant būtų galima naudoti kryžminį kompiliatorių.

Įvesties-išvesties ir pertraukimų aprašymas

MC68HC05X16 yra vieno lusto mikrovaldiklis, turintis 24 įvesties/išvesties linijas. Šios linijos yra suskirstytos į tris 8 bitų prievadus ir aštuonias tik įvesties linijas, išdėstytas kaip vienas 8 bitų prievadas. Kiekviena I/O linija yra programuojama kaip įvestis arba išvestis per programinę įrangą, kuri valdo duomenų krypties registrus. 8 bitų tik įvesties prievadas (D) dalijasi savo kaiščiais su A/D keitikliu, kai A/D keitiklis yra įjungtas. Norint išvengti išvesties trikdžių, duomenys privalo būti įrašyti į I/O prievado duomenų registrą prieš juos įrašant į atitinkamus duomenų krypties registro bitus. MC68HC05 palaiko kelis pertraukimų tipus, įskaitant: Laikmačio/skaitiklio pertraukimus: Įrenginyje yra du 16 bitų laikmačiai/skaitikliai, pažymėti TCNT ir TCNT2, kurie gali generuoti pertraukimo užklausas, pasiekę tam tikrą vertę arba įvykus konkretaus įvykio bei išorinius pertraukimus, kurie yra du išoriniai pertraukimo kaiščiai, INT ir INT2. Jie gali būti naudojami suaktyvinti pertraukimo užklausą, gavus konkrečią signalą iš išorinio įrenginio. Serijinio ryšio pertraukimus: MC68HC05 turi integruotą nuosekliojo ryšio sąsają, kuri gali generuoti pertraukimo užklausas, gavus arba siunčiant simbolį. Pertraukimus galima įjungti arba išjungti naudojant valdymo registrus, o pertraukimų prioritetą galima konfigūruoti naudojant prioritetų registrus.

Šaltiniai

- [1] Freescale Semiconductor, Inc., 1998. www.freescale.com.
Prieiga per internetą: <https://www.nxp.com/docs/en/data-sheet/MC68HC05X16.pdf>
[žiūrėta: 2023 m. gruodžio 16d.].
- [2] Motorola 68HC05. Prieiga per internetą: https://en.wikipedia.org/wiki/Motorola_68HC05
[žiūrėta: 2023 m. gruodžio 16d.].
- [3] Motorola, 1998. *An Introduction to Motorola's 68HC05 Family of 8-Bit Microcontrollers*.
Prieiga per internetą: <https://www.csie.ntu.edu.tw/~b6506016/asm/tutorial.pdf>
[žiūrėta: 2023 m. gruodžio 16d.].
- [4] ChatGPT. Prieiga per internetą: <https://chat.openai.com> [žiūrėta: 2023 m. gruodžio 16d.]

