

COP8 ir ATmega168 architektūrų palyginimas

Mikrovaldikliai yra vieni svarbiausių šiuolaikinių įterptinių sistemų komponentų – jie naudojami būtinėje technikoje, automobilinėje elektronikoje, matavimo prietaisuse, jutiklių mazguose ir įvairiose valdymo sistemoje. Skirtingų gamintojų ir kartų mikrovaldikliai sprendžia panašias užduotis, tačiau remiasi skirtingomis architektūrinėmis idėjomis: vieni projektai labiau orientuoti į paprastą akumulatorinę CISC architektūrą, kiti – į registrinę RISC architektūrą su dideliu bendros paskirties registrų skaičiumi. Šiame darbe lyginamos dvi 8 bitų mikrovaldiklių architektūros - „National Semiconductor“ COP8 ir „Atmel“ ATmega168, kurios buvo sukurtos panašiu laikotarpiu ir turėjo panašius technologinius apribojimus, tačiau pasirinko skirtingus sprendimus komandų rinkinio, atminties, registrų bei periferijos srityje. Pagrindinis darbo tikslas – apibūdinti ir palyginti šių dviejų architektūrų struktūrus, galimybes ir taikymą įterptinėse sistemoje.

Elementinė bazė ir fizinės savybės (2)

Tiek COP8, tiek ATmega168 yra monokristaliniai vieno lusto 8 bitų mikrovaldikliai, pagaminti CMOS technologija ir priskiriami labai didelio integracijos masto (VLSI) integriniams grandynams. COP8 šeimoje tame pačiame kristale sujungtas procesoriaus branduolys, programų ROM/Flash atmintis, vidinė RAM, laikmačiai, A/D keitiklis, nuoseklios sąsajos ir kita periferija, todėl tai tipinis „system-on-chip“ sprendimas. Šie mikrovaldikliai buvo tiekiami įvairiuose plastikiniuose korpusuose – 20, 28 ir 40 išvadų DIP, 16-28 išvadų SOIC, 44 išvadų PLCC ir panašiuose paketuose - tai kelių centimetrų ilgio korpusai, sveriantys vos kelis gramus, maitinami maždaug 2,7-5,5 V įtampa, o darbo režime suvartojantys nuo kelių iki kelių dešimčių miliampерų, su galimybe budėjimo režimuose sumažinti srovę iki mikroampерų.

ATmega168t taip pat yra vieno lusto VLSI mikrovaldiklis, kuriame integroruotas AVR branduolys, 16 KB Flash programų atmintis, 1 KB SRAM, 512 baitų EEPROM, keli laikmačiai, 10 bitų ADC, pertraukimų sistema ir kiti periferiniai moduliai. Šis mikrovaldiklis gaminamas 28 arba 2 išvadų PDIP, TQFP ir QFN/MLF korpusuose, kurių fiziniai matmenys ir svoris yra panašios eilės kaip ir COP8. ATmega168 veikia maždaug 1,8-5,5nV įtampos diapozone, o žemos galios režimuose gali dirbti labai mažais dažnais ir suvartoti tik keletą dešimtyjų mikroampero, todėl abu sprendimai yra aiškiai orientuoti į taupias įterptines sistemas.

Architektūros tipas ir adresų modelis (3, 4)

COP8 architektūra yra akumulatorinio tipo: pagrindinis aritmetinių ir loginių operacijų registras yra vienas 8 bitų akumulatorius A, o dauguma komandų naudoja būtent jį kaip centrā visiems skaičiavimams (3). Daug „registru“ iš tikrujų yra specialūs RAM baitai, todėl COP8 labiau primena klasikinę vieno adreso mašiną, kur operacija aprašo akumulatorių ir vieną operandą iš atminties arba konstantą (4). Dėl šios priežasties COP8 galima laikyti vieno adreso akumulatorine architektūra: viena komanda tiesiogiai manipuliuoja akumulatoriaus reikšme ir vienu papildomu operandu, o sudėtingesniems skaičiavimams reikia kelių instrukcijų.

ATmega168 atstovauja registrinei RISC architektūrai: procesorius turi 32 bendros paskirties 8 bitų registrus R0-R31, kurie sujungti su ALU ir leidžia vykdyti daugelį aritmetikos ir logikos komandų forma „registras-registras“ (3). Tai dviejų adresų mašina – instrukcijos paprastai turi du registrų operandus. O rezultatas įrašomas į vieną iš jų, todėl kompiiliatorius gali laikyti daug kintamųjų registrų failo ir retai kreiptis į RAM(4). Toks sprendimas aiškiai skirtia ATmega168 nuo akumulatorinio COP8 varianto ir rodo modernios 8 bitų RISC architektūros privalumus (3, 4).

Duomenų plotis ir atminties organizavimas (7, 8, 9, 16)

Abi architektūros yra 8bitų: COP8 akumulatorius, darbo baitai ir ALU dirba 8 bitų masteliu, o platesni skaičiai sudaromi iš kelių baitų, panašiai ir ATmega168 ALU ir registrai yra 8 bitų, nors programų atmintyje instrukcijos koduojamos 16 bitų žodžiais (7). Tieki COP8, tiek ATmega168 naudoja Harvardo architektūrą – programų ir duomenų atmintys turi atskiras adresų erdves, todėl instrukcijos ir duomenys gali būti nuskaitomi lygiagrečiai (8).

COP8 programų atminties adresų erdvė siekia iki maždaug 32 KB, todėl architektūriškai leidžia realizuoti iki 32 KB ROM/Flash lustus, kuriuose instruktorių adresai išdėstyti ištisai, nors komandų rinkinyje naudojamos ir trumpesnės šakos puslapiai ribose (8). Duomenų atmintis COP8 turi 8 bitų adresų erdvę viename banke (256 baitai), iš kurios apatinė dalis skirta RAM ir stekui, o viršutinė – periferijai ir valdymo registrams. Pažangesniuose modeliuose naudojamas bankavimas, leidžiantis pasiekti iki maždaug 1 KB vidinės RAM, pasirenkant aktyvų banką per specialų registrą (8). ATmega168 turi labai aiškų ir plokščią išdėstytmą:

16 KB Flash programų atmintis, ištisinė SRAM erdvė, sudaryta iš 32 registru srities, I/O registrų ir 1 KB duomenų SRAM, be bankavimo, ir atskira 512 EEPROM atmintis, pasiekiamą per specialius registrus (8).

Nei COP8, nei ATmega168 neturi atminties valdymo bloko ir nepalaiko virtualiosios atminties: adresai atitinka fizinius ROM, RAM ir periferijos adresus. Nėra nei puslapiaivimo, nei segmentavimo mechanizmų, kurie būdingi stalinėms ir servinėms architektūroms (9, 16). Taip pat nei viena iš architektūrų neturi spartinančios atminties – instrukcijos skaitomos tiesiai iš Flash/ROM, duomenys iš RAM, o vienintelė optimizacija yra paprastas instrukcijų „prefetch“/pipeline ATmega168 branduolyje (16).

Komandų sistema ir adresavimo būdai (10, 11)

COP8 komandų sistema yra CISC tipo: maždaug 60-70 instrukcijų, iš kurių dauguma yra vieno baito, o komandas su konstantomis ar adresais – 2-3 baitų (10). Dauguma aritmetinių ir loginių operacijų naudoja akumulatorių A, o antras operandas yra konstanta, RAM adresas ar netiesioginis adresas per B ar X registrą. Tipiniai pavyzdžiai: LD A,#10, LD A,addr, ADD A,#1, X A,addr, JMPL label, IFEQ A,#0, RET, RETI (10). ATmega168 komandų sistema yra registrinė RISC, turi apie 130 instrukcijų, dauguma jų koduojamos vienu 16 bitų žodžiu ir dirba su registrų failu R0-R31. Tipinės komandos: LDI, R16,10, ADD R16,R17, DUBI R18,1, MOV R20,R16, RJMP label, BRNE label, SBI PORTB,3, CBI PORTB,3, RETI (10). Abiejose architektūrose yra logikos, aritmetikos, šakų, duomenų perdaivimo ir pertraukimų valdymo instrukcijos, tačiau ATmega168 turi daug daugiau reguliarių registrų-registrų formų ir atskiras instrukcija I/O operacijoms, o COP8 stipriai remiasi akumulatoriumi ir „skip“ tipo sąlyginėmis komandomis (10).

Adresavimo būdai taip pat rodo architektūrinius skirtumus (11). COP8 palaiko tiesioginį adresavimą į RAM, netiesioginį adresavimą per B ir X registrus su autoinkrementu ir autodekrementu (pvz., [B+], [B-]), immediate operandus, santykinį adresavimą šakoms ir „skip“ komandoms, bei specialių programų ROM lentelių adresavimą per programos skaitiklį (11). ATmega168 palaiko registrų adresavimą (R0-R31), immediate operandus, tiesioginį adresavimą SRAM ir I/O registrus (IN/OUT, SBI/CBI), netiesioginį adreavimą per 16 bitų X, Y, Z ir rodykles su autoinkrementu ir poslinkiu, taip pat programų Flash skaitymą ir absoliučius šuolius (11). Abu branduoliai turi immediate, tiesioginį, netiesioginį ir santykinį adresavimą, tačiau AVR variantas yra žymiai turtingesnis ir patogesnis kompiliatoriams (11).

I/O posisteminié ir pertraukimai (12, 13)

COP8 mikrovaldikliuose įvesties/išvesties registrai yra išdėstyti duomenų atminties erdvėje (memory-mapped I/O): keli 8 bitų portai leidžia konfigūruoti išvadus kaip įvestį arba išvestį, laikmačiai ir skaitikliai realizuoja laiko ir dažnio matavimą, dalyje modelių įdiegtas ADC ir nuosekliosios sasajos, tokios kaip MICROWIRE ar UART (12). ATmega168 turi iki 23 bendros paskirties skaitmeninių I/O linijų, išdėstyty trijuose portuose, tris laikmačius (du 8 bitų ir vieną 16 bitų) su PWS, 10 bitų ADC su keliais kanalais, USART, SPI ir TWI (I²C suderinamą) magistralės, taip pat „watchdog“, išorinių ir „pin-change“ pertraukimų jėjimus, todėl periferijos rinkinys yra žymiai platesnis (12).

Pertraukimų sistema abiejose architektūrose yra vektorinė: skirtinti pertraukimų šaltinius turi atskirus vektorius programos pradžioje, o gržimui naudojama speciali instrukcija RETI, atstatant programos skaitiklį ir būsenos registrą (13). COP8 turi kelis pertraukimų šaltinius (išoriniai jėjimai, laikmačiai, „watchdog“, nuosekliosios sasajos), globalų leidimo bitą ir individualius leidimo bitus moduliams, o ATmega168 turi gerokai daugiau pertraukimų šaltinių (visi laikmačiai, ADC, USART, SPI, TWI, pin change ir kt.) ir globalų I bitą SREG registre, valdomą SEI ir CLI komandomis (13). Taigi bazinis principas abiejose architektūrose panašus, tačiau AVR sistemoje vektorių ir šaltinių skaičius yra didesnis, o integracija su branduolio būsenos registru – patogesnė (13).

Duomenų tipai ir aritmetika (14)

Aparatūros lygiu tiek COP8, tiek ATmega168 dirba su 8 bitų sveikaisiais skaičiais: ALU operacijos atliekamos 8 bitų masteliu, o platesnės reikšmės (16, 32 bitų) konstruojamos iš kelių baitų programiškai (14). Nei viena architektūra neturi įrengtos slankiojo kablelio ar fiksuoto kablelio aritmetikos vienetų – slankiojo kablelio operacijos, jei reikia, realizuojamos C ar asemblerio bibliotekomis, naudojančios sudėtį, atimtį, daugybę ir poslinkius (14). Sveikieji skaičiai su ženklu abu atvejais interpretuojami kaip dvejeto papildinio (two's complement) reikšmės, apie tai sprendžiama iš požymių bitų C, N, V ir S elgsenos, o specialių BCD, dešimtainių ar kompleksinių tipų komandos architektūros lygiu nepalaikomos.

Greitaveika ir našumas (15)

COP8 šeimos mikrovaldikliai dirbo maždaug iki 10–20 MHz taktiniu dažniu, nuo modelio ir maitinimo, o paprastos instrukcijos buvo vykdomos per 1–2 taktus, sudėtingesnės šakos ir posūkis – per kelis taktus, todėl reali greitaveika siekė kelis milijonus komandų per sekundę. Dėl akumulatorinio modelio ir riboto registrų skaičiaus C kompiliatoriaus sugeneruotas kodas dažnai reikalauja daugiau komandų tam pačiam algoritmui, todėl teorinis dažnis ne visada viršaus dideliu praktiniu našumu. ATmega168 gali veikti iki 20 MHz dažniu, o dauguma AVR instrukcijų yra vieno ciklo, todėl esant 16 MHz taktui nominali greitaveika artima 16 MIPS ir praktikoje viršija daugumą COP8 realizacijų, ypač dėl efektyvaus registrų failo ir avr-gcc generuojamо kodo. Pagal kainos ir našumo santykį ATmega168 ir labiausiai AVR šeima ilgą laiką buvo labai patrauklūs 8 bitų sprendimai, ypač dėl plačios įrankių grandinės ir bibliotekų pasiūlos.

Taikymo sritys ir programinės įrangos ekosistema (17, 18)

COP8 mikrovaldikliai buvo naudojami būtinėje ir paprastoje pramoninėje elektronikoje: skalbimo ir kitų būties prietaisų valdose, skaitiklių ir apsaugos sistemose, automobilinėje elektronikoje, nuotolinio valdymo pulteliuose ir asmeniniuose įrenginiuose, tokiuose kaip automobilių rakteliai ar paprasti komunikacijos moduliai (17). Tipinio panaudojimo pavyzdys – skalbimo mašinos valdiklis, kuriamo COP8 skaito vandens lygio ir temperatūros jutiklis, įjungiamas siurblį ir kaitinimo elementą, valdo variklio sukimosi kryptį ir greitį bei apdoroja vartotojo mygtukus ir LED indikaciją viename luste be atskiro procesoriaus (17). ATmega168 plačiai naudojo įterptinės valdymo ir mokomujų sistemų plokštėse, jutiklių mazguose ir paprastuose automatikos projektuose. Žymiausių pavyzdžių – Arduino Duemilanove lenta, kurios pagrindinis mikrovaldiklis skaito temperatūros, šviesos ar judesio jutiklius ir pagal tai valdo LED juostas, variklius ar reles valdomus įrenginius, pvz., mažą mobilujį robotą ar namų automatizavimo modulį (17).

Programinės įrangos ekosistemos šių architektūrų istorijoje išsivystė labai daug (18). COP8 laikotarpiu buvo sukurta uždarų „firmware“ būtinei technikai, skaitikliams, apsaugos ir automobilinėms sistemoms, tačiau didžioji šio kodo dalis viešai prieinama – tai gamintojų nuosavybė, kartais prie jos nėra galima prieiti tik per pavyzdžius application notes dokumentuose (18). COP8 ekosistemoje egzistavo keli komerciniai įrankių rinkiniai: oficialūs National assembleriai ir utilitos, Byte Craft C kompiliatorius, IAR Embedded Workbench su C kompiliatoriumi, linkeriu, bibliotekų įrankiu ir C-Spy derintuvu, taip pat programavimo per nuoseklią sąsają CP1 (ISP) OS artekančiose programose. Šiandien didelė dalis šios ekosistemos liko tik istoriniuose archyvuose, o naujuų projektų skaičius yra nedidelis (18).

ATmega168, priklausantis AVR šeimai, turi labai didelę ir iki gyvų ekosistemų: nemokamas avr-gcc + avr-libc toolchain, avrdude programavimui, Arduino IDE, Microchip/Atmel Studio, daug derinimo ir simuliavimo įrankių, tūkstančiai Arduino bibliotekų ir pavyzdinių projektų GitHub8). Tai reiškia, kad ATmega168 programinės įrangos bazė yra didžiulė, didelė jos dalis atviro kodo ir lengvai pasiekiamā, o įrankių ir bibliotekos leidžia greitai kurti tiek mokomoius, tiek praktinius įterptinius projektus (18).

Šaltiniai:

https://en.wikipedia.org/wiki/AVR_microcontrollers

https://bitsavers.org/components/national/_dataBooks/1994_National_COP8_Databook.pdf

https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2545-8-bit-avr-Microcontroller-ATmega48-88-168_Datasheet.pdf

<https://www.utmel.com/components/atmega168-microcontroller-features-pinout-and-datasheet?id=959&>

<https://www.ukwhitegoods.co.uk/appliance-industry-news/226-washing-machine-news/190-payg-pay-as-you-go-washing-machines>

<https://www.edn.com/edn-access-01-18-96-cryptographic-techniques-secure-your-wireless-design/>

<https://docs.arduino.cc/retired/boards/arduino-duemilanove/>

<https://hr.allelcoelec.com/blog/how-the-atmega168-microcontroller-works.html>

<https://en.wikipedia.org/wiki/COP8>