

MŪSDIENU DATORU PROCESORU STRUKTŪRA MODERN COMPUTER PROCESSOR STRUCTURE

Autors: **Arturs ŠKAPARS**, e-pasts: varabei@inbox.lv, tālrunis.:+37122485481
Autors: **Aigars LADUŠEVS**, e-pasts: aigaigarss@inbox.lv, tālrunis.:+37129198285
Zinātniska darba vadītājs: **Pēteris Grabusts Dr.sc.ing. profesors**, e-pasts:
peteris.grabusts@ru.lv
Rēzeknes Tehnoloģiju akadēmija
Rēzekne, Atbrīvošanas aleja 115

Abstract: *This article is about computer processor structure, that has improved over time and is still evolving. The goal of this work is to delve into specifics of computer processors and the logic they obey. This paper will overlook parts of modern computer processor structure to understand how they work.*

Keywords: *arithmetic logic unit, central processing unit, registers.*

Ievads

Vecākajām skaitļošanas iekārtām, piemēram, ENIAC (*Electronic Numerical integrator And Computer*) nācās veikt manuālu atsevišķu komponentu pieslēgšanu un atslēgšanu, lai veikt dažādus uzdevumus. Kopš termins CPU (*Central Processing Unit*) tika definēts kā ierīce priekš programm izpildes, agrīnās ierīces, kuras varēja tikt sauktas par CPU nāca kopā ar “uzglabāto-programmu datoru” atnākšanu. “Uzglabāto programmu dators” saglabāja programmas instrukcijas elektroniskajā atmiņā.

Agrākajiem CPU tika piešķirts individuāls dizains, kurš tika izmantots kā daļa no lielāka vai īpatnēja datora. Savukārt šī īpatnējā izstrādes metode deva ceļu procesoru izstrādei ar vairākiem pielietojumiem (*multi-purpose processor*). Šī standartizācija atvēra ceļu uz jaunu tranzistoru lieldatoru un minidatoru ēru un strauju integrēto shēmu (IC) popularizāciju. Integrētās shēmas pieļāva sarežģītāku procesoru izstrādi un ražošanu. CPU minituarizācija un standartizācija palielināja digitālo ierīču klātbūtni modernajā dzīvē, salīdzinot ar laikiem kad ražoja ENIAC datorus. Modernie mikroprocesori ir sastopami vairākās ikdienas elektronikas iekārtās, sākot no automobiļiem, telefoniem un pat rotaļlietām.

Releji un vakuumlampas tika izmantoti kā slēdžu elementi, pilnvērtīgam datoram bija nepieciešami vairāki tūkstoši šo slēdžu ierīču. Sistēmas ātrums tika limitēts ar slēdžu pārslēgšanās ātrumu. Beigu beigās vakuumlampu datori kļuva dominējoši, pateicoties to ātrumam, pārsvērot uzticamību problēmas. Vairākums šo agro sinhrono procesoru darbojās zemos takts signālos, salīdzinot ar modernajiem mikroelektronikas dizainiem. Takts signālu frekvences svārstījās no 100 KHz līdz 4 MHz un tas bija diezgan parasts svārstību intervāls priekš tā laika, kas tika ierobežots sakarā ar slēdžu ierīču uzbūvi.

Centrālais procesors

Centrālais procesors (CPU) ir datora mezgls, kas paredzēts instrukciju nodošanai pēc programmas pieprasījuma izmantojot aritmētiskās, loģiskās, kontroles un ievad/izvad operācija. CPU termins tika izmantots kopš 1960. gadu sākuma un vairāk attiecās uz kontroles un apstrādes iekārtu, nošķirot atslēgas elementus no ārējām datora komponentēm, no izvad/ievad mezgliem un atmiņas nesējiem.

Dizaina un īstenošanas ziņā CPU ierīces mainījās laikam ejot, bet fundamentālās operācijas palika nemainīgas. Galvenās procesora sastāvdaļas palika:

- Aritmētiski loģiskā iekārta (ALU, *arithmetic logic unit*), kura atbild par aritmētiski loģisko operāciju izpildi;
- Procesora reģistri, kuri nodrošina operandus aritmētiski loģiskajai iekārtai;

- Kontroles iekārta, kura vada datu izvēles un atlasēšanas procesus, instrukciju izpildi vadot jau koordinētās aritmētiski loģiskās iekārtas operācijas, reģistrus un citas komponentes. [1]

Lielākā daļa centrālo procesoru ietilpst mikroprocesoru kategorijā, tas nozīmē, ka tie ietilpst vienā integrētās mikroshēmas (IC) čipā. Integrētā shēma saturošā centrālo procesoru tāpat satur arī atmiņu, perifēros interfeisus un citas datora komponentes, šādas integrētās ierīces tiek sauktas par mikrokontrolieriem vai vienkristāla shēmām (no angļu *system on a chip* – SoC). Daudzi datori izmanto vairāk-kodolu procesorus, kur viena shēma satur divus vai vairākus centrālos procesorus, kurus sauc par kodoliem.

Centrālā procesora mērķis ir izpildīt uzkrāto instrukciju secību, kuru sauc par programmu. Programma ir skaitļu sērija, kura tiek turēta datora atmiņas ierīcē. Programmas izpildes procesam ir četri soļi, kurus lieto gandrīz vai visi procesori, tie ir: saņemt, atšifrēt, izpildīt un atbildēt. Galvenās procesora iezīmes ir takts frekvence, ātrdarbība, enerģijas patēriņš, arhitektūra un litogrāfijas procesu normas izgatavojot procesorus. Algoritmu izpildes laikā, procesors veic milzīgu skaitu vienkāršu soļu, kurus sauc par taktīm un to skaitu, ko procesors izpilda vienā sekundē, sauc par procesora takts frekvenci. Takts frekvence tiek mērīta megahercos (MHz), agrāk tā tika mērīta kilohercos (1KHz = 1000Hz), bet mūsdienās, datoriem attīstoties takts frekvence tiek mērīta megahercos (1MHz = 1000KHz), un bieži tā tiek izteikta gigahercos (1GHz = 1000MHz).

Mūsdienu procesora galvenās sastāvdaļas:

- Vadības ierīce – atkodē no atmiņas saņemtās komandas un formē to izpildei nepieciešamos vadības slāņus.
- Aritmētiski loģiskā ierīce – atbild par aritmētisko funkciju izpildi (saskaitīšana, atņemšana, reizināšana un dalīšana), loģiskās (skaitļu salīdzināšanu) un citas datu apstrādes operācijas.
- Reģistru bloks – satur reģistrus, t.i., maza apjoma, ātrdarbīgu atmiņu, kura paredzēta datu īslaicīgai uzglabāšanai. Universālos reģistrus izmanto starprezultātu glabāšanai, bet atmiņas reģistros esošos datus – datu vai komandas adreses vairošanai. Procesors tāpat satur arī specializētus reģistrus, kuri paredzēti konkrētai informācijai, piemēram komandas kodi vai turpmāk izpildāmās komandas adreses.
- Kešatmiņa – izmantojas komandu, datu un starprezultātu glabāšanai. [1]

Aritmētiski loģiskā ierīce (ALU)

Ievads ALU shēmā ir dati, kurus sauc par operandiem un kods norāda, ka ir jāizpilda kāda operācija. ALU izvade ir izpildīto operāciju rezultāts. Dažādos ALU dizainos papildus informācija tiek apmainīta ar statusa reģistru, kurš attiecas uz rezultātu no tagadējām vai iepriekšējām operācijām (skatīt 1. attēlu).

Aritmētiski loģiskās ierīces simboliskā representācijā ar bultiņām attēloti ievada un izvada signāli, katra no bultām attēlo vienu vai vairākus signālus. Kontroles signāli nāk no kreisās puses un statusa signāli iziet no labās puses, dati plūst no augšas uz leju. Aritmētiski loģiskajai iekārtai pastāv vairāki izvada un ievada tīkli, kuri fiziski ģenerē digitālos jeb ciparveida signālus starp ALU un ārējo shēmu. ALU darbojoties ārējā shēma padod signālus uz ALU ievadiem, savukārt ALU ģenerē un vada signālus uz ārējo shēmu, izmantojot izvadus. Parastam ALU ir trīs paralēli datu tilti jeb datu maģistrāles, kuras sastāv no diviem ievada operandiem (A un B operandi) un rezultāta izvads (Y). Katra datu maģistrāle ir signālu grupa, kura nogādā vienu bināro ciparu. Parasti A, B un Y maģistrāļu platumi ir identiski un sakrīt ar vajadzīgajiem ārējās shēmas izmēriem.

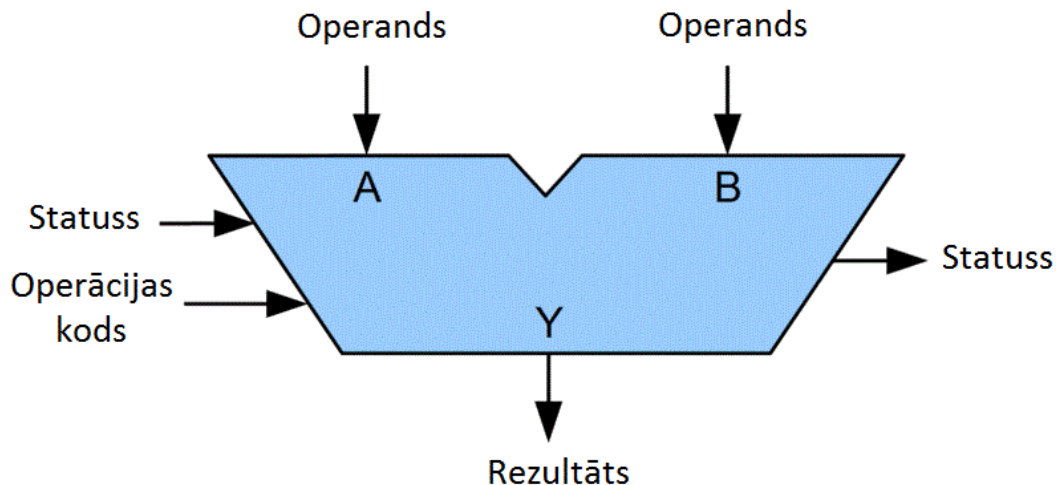
Opkoda (operāciju kods) ievads ir paralēla maģistrāle kura padod operācijas sekciju kodu uz aritmētiski loģisko ierīci. Koda sekcijas tiek uzskaitītas dodot ALU instrukcijas par vajadzīgo aritmētisko vai loģisko operāciju. Opkoda maģistrāles platumš saistīts ar operāciju skaitu ko ALU var veikt; piemēram, četru bitu opkods var specificēt līdz 16 dažādām ALU

operācijām. Parasti ALU opkods nav tādā pašā valodā kā datora, vai citas mašīnas opkods, dažos gadījumos tas var būt šifrēts kā bitu lauks pašas mašīnas opkodā.

Statusa izejas ir individuāli signāli kuri nodod papildus informāciju par ALU operāciju rezultātu. Šie izejas signāli parasti tiek uzglabāti reģistros turpmākai izmantošanai ALU operācijās, lai vadīt nosacījuma zarojumus. Bitu reģistru kopa kura uzglabā statusa izvadus bieži tiek apstrādāta, kā vien, daudzbītu (*single-bit, multi-bit*) reģistrs, citiem vārdiem, kā statusa reģistrs. Parasti ALU satur šādus statusa signālus:

- *Carry-out* - ciparu transportēšana papildus operācijas rezultātā, aizņemšana atņemšanas rezultātā - vai bitu pārpilde binārās maiņas rezultātā.
- *Zero* – norāda, ka visi izvada biti ir loģiskā nulle.
- *Negative* – norāda, ka aritmētiskās operācijas rezultāts ir negatīvs.
- *Overflow* – norāda, ka aritmētiskās operācijas rezultāts ir pārsniedzis cipara diapazona izvadu.
- *Parity* – norāda, ka pāra vai nepāra bitu skaits ir loģiskais vieninieks.

Statusa ievads ļauj ALU piekļūvi pie papildus informācijas operāciju izpildes laikā.[3]



1. attēls. Aritmētiski loģiskā ierīce (ALU)

Vadības iekārta

Kontroles iekārta, jeb CU (*control unit*) ir CPU komponente kura vada procesora operācijas. Tā nosaka kādā veidā datora atmiņa, aritmētiski loģiskā iekārta un izvad/ievad ierīces reaģē uz programmas instrukcijām.[4]

Kontroles iekārta vada citu vienību operācijas nodrošinot kontroles signālus un laika noteikšanu. Liela daļa datora resursu tiek vadīti ar kontroles iekārtas palīdzību, CU vada datu plūsmu starp CPU un citām iekārtām. Modernās konstrukcijās CU parasti atrodas iekšējā CPU daļā un laika gaitā CU operācijas īpaši nemainījās.

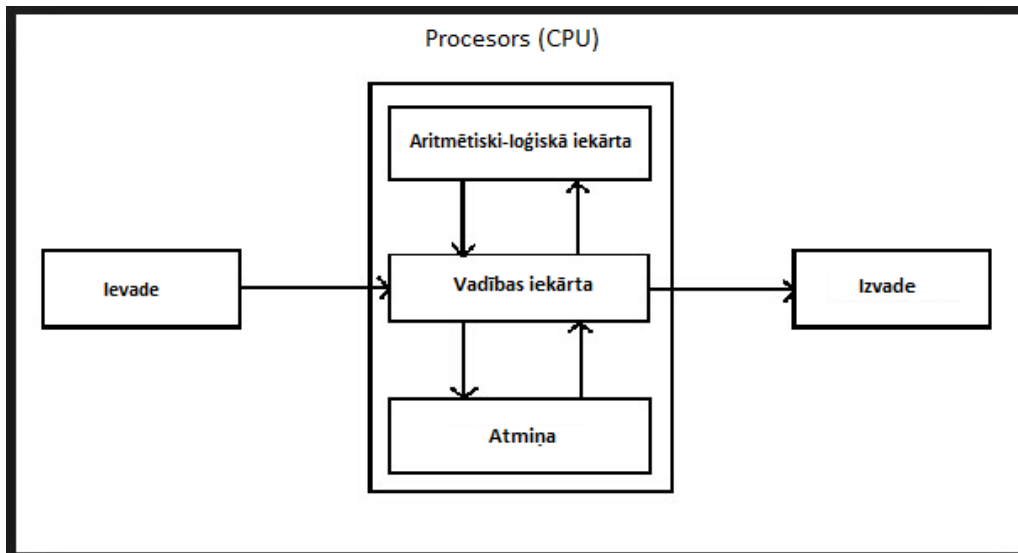
Kontroles iekārta ir digitāla shēma procesorā, kura koordinē datu plūsmas secību uz, no un starp daudzām CPU apakšvienībām. Rezultātā dati tiek izveidoti un nodoti programmai caur vairākām apakšvienībām. Kontroles iekārta vada datu plūsmu iekš procesora un papildus nodrošina kontroles signālus datoram, lai turpmāk vadīt datus un instrukcijas uz/no procesora ārējiem galapunktiem, CU var nosaukt par “smadzenēm smadzenēs”.

Kontroles iekārta nepieciešama gan procesoriem, gan grafiskajiem procesoriem. Kontroles iekārta saņem ārējās instrukcijas vai konvertē komandas kontroles signālu secībās, kuras CU pielieto datu plūsmas ceļā, lai izpildīt reģistru pārsūtes operāciju secību. Konkrētāk, kontroles iekārta ir sarežģītu digitālo shēmu savienojums un kontrolē daudz

vienību izpildi centrālajā procesorā. Kontroles iekārta ir pirmā CPU iekārta, kura saņem instrukcijas no ārējās programmas. CU atšifrē instrukciju sekojošajos soļos, kuri kontrolē un koordinē procesora iekšējās darbības pareizai datu manipulācijai. Sekojošo soļu konstrukcija ir atkarīga no katras instrukcijas vajadzībām un var atšķirties soļu skaitā, izpildes kārtā un iesaistītajās izpildes vienībās. Izmantojot instrukciju programmu atmiņā, CU konfigurē CPU datu plūsmu saskaņoti vajadzībai veikt darbības ar datiem pareizā veidā.

Progresīvākas kontroles iekārtas var tulkot instrukcijas, bet ne datu sekcijas, vairākās mikro-instrukcijās un CU vada mikro-instrukciju kārtību starp izvēlētajām izpildes iekārtām, kurām dati tiek nodoti un izmainīni izejot no izpildes vienības funkcijām (skatīt 2. attēlu).[4]

2. attēls. Vadības iekārtas funkcionālā shēma



2. attēls. Vadības iekārtas funkcionālā shēma

Procesora reģistri

Procesora reģistrs ir daļa no mazāka datu uzglabāšanas komplekta, kurš ir procesora daļa. Reģistri parasti sastāv no neliela atmiņas daudzuma, dažiem reģistriem piemīt specifiskas funkcijas un daži reģistri var būt tikai nolasāmi (*read-only*) vai tikai rediģēti (*write-only*). Reģistri parasti tiek adresēti ar pārējās atmiņas mehānismiem.

Reģistri var saturēt instrukcijas, datu uzglabāšanas adreses vai dažādus datus, piemēram bitu kārtību, vai individuālus simbolus. Programma var saturēt instrukciju, ka divu definētu reģistru saturu, pieņemsim veselos skaitļus, nepieciešams saskaitīt un uzglabāt noteiktā reģistrā. 64-bitu datorā reģistra garumam jābūt ar izmēru 64 biti. Dažādās datoru arhitektūrās tiek izmantoti mazāka izmēra reģistri, piemēram, pusreģistri, paredzēti mazāka jeb īsāka izmēra instrukcijām. Atkarībā no procesora arhitektūras un valodas noteikumiem, reģistri var tikt numurēti vai tiem var tikt piešķirti patvaļīgi nosaukumi.

Procesors parasti satur vairākus indeksa reģistrus. Indeksa reģistri ir shēmas, kuras saņem, uzglabā un izvada instrukcijas paredzētas koda maiņai datorā. Šīs shēmas tāpat arī sauc par adrešu reģistriem vai arī par modifikāciju reģistriem. Efektīvā adrese satur bāzi, rādītāju un relatīvo adresi, tas viss tiek saglabāts indeksa reģistrā, jeb rādītāja reģistrā. Reģistri parasti tiek mērīti bitu skaitā, kurus tie spēj saglabāt, piemēram 8-bitu reģistrs, vai 32-bitu reģistrs. Procesorā parasti ietilpst vairāku tipu reģistri, kuri var tikt klasificēti saskaņā ar to instrukcijām:

- Lietotājam pieejami reģistri var tikt nolasīti un rediģēti izmantojot mašīnas instrukcijas. Plašāk pieejamie lietotāju reģistri atrodas datu un adrešu reģistros.
 - Datu reģistri var saturēt veselos skaitļus, dažas arhitektūras pieļauj decimālo skaitļu, simbolu, nelielu bitu masīvu un citu datu saglabāšanu.

- Adrešu reģistri satur adreses, kuras tiek izmantotas instrukciju izpildes laikā. Instrukcijas piekļūst pie galvenās atmiņas netiešā veidā.
 - Daži procesori satur reģistrus, kuri var tikt izmantoti, lai uzglabātu adresi vai tikai skaitļus.
 - GPRs (*general-purpose registers*) – reģistru tips, kas var saturēt gan datus, gan adreses, reģistru faili retos gadījumos var saturēt arī decimālos skaitļus.
 - Statusa reģistri satur *true/false* mainīgos, izmanto, lai noteiktu, kādas instrukcijas jāizpilda un kādas nē.
 - Decimālo ciparu reģistri, jeb FPRs (*floating-point registers*) satur decimālos ciparus.
 - Konstantes reģistri, satur tādus mainīgos, kā nulle, viens vai arī skaitli *PI*.
 - Vektora reģistri satur datus par vektor apstrādi, kura nepieciešama SIMD instrukcijām (*single instruction, multiple data*).
 - Speciālu mērķu reģistri (SPR) satur programmas stāvokli, tajos ietilpst programmas kārtotāju, vai citiem vārdiem, instrukciju vadības un statusa reģistru. Statusa reģistrs un programmas kārtotājs var tikt apvienoti PSW (*program status word*) reģistrā.
 - Dažādās arhitektūrās modeļa-specifiskie reģistri satur datus un iestatījumus, saistītus ar pašu procesoru. Šie reģistri ir atkarīgi no procesora arhitektūras modeļa un arhitektūras modeļi mainās laikam ejot.[2]
- Iekšējie reģistri – reģistri, kuriem nevar piekļūt instrukcijas, izmanto procesora operācijām.
- Instrukciju reģistri satur pašlaik izpildāmās instrukcijas.
 - Reģistri saistīti ar informāciju izvadi no RAM, ir uzglabāšanas reģistru krājums, kurš atrodas uz atsevišķas no procesora shēmas. [2]

Detāļu reģistri, jeb komponentu reģistri (*hardware register*) ir līdzīgi augstāk minētajiem, bet darbojas ārpus procesora. [2]

Secinājumi

1. Arhitektūras un īstenošanas ziņā CPU ierīces mainījās laikam ejot, bet fundamentālās operācijas palika nemainīgas.
2. Laika gaitā uzlabojās procesora takts frekvence, ātrdarbība, samazinājās jaudas patēriņš un pilnveidojās arhitektūra.
3. Mūsdienās lielākā daļa datoru izmanto vairākkodolu procesorus, kuru viena shēma satur divus vai vairākus centrālos procesorus; tādā veidā ļaujot izpildīt uzkrāto instrukciju secību efektīvāk.

Summary

Nowadays, most computers use multi-core processors, with one circuit containing two or more central processors, thus allowing the sequence of stored instruction to be performed more efficiently. As time goes by, the processor power consumption has decreased but clock speed, performance, and architecture have improved. In terms of design and implementation, CPU devices changed as time went by, but basic operations remained unchanged.

Literatūras saraksts

1. https://en.wikipedia.org/wiki/Central_processing_unit
2. https://en.wikipedia.org/wiki/Processor_register
3. https://en.wikipedia.org/wiki/Arithmetic_logic_unit
4. https://en.wikipedia.org/wiki/Control_unit