Motorola 68000

Részletesebben: https://hu.wikipedia.org/wiki/Motorola_68000
A család modelljei: https://hu.wikipedia.org/wiki/Motorola_68000_processzorcsalád

AMotorola 68000 16/32 bites CISC (Complex Instruction Set Computing) mikroprocesszor mag (és konkrét processzor), amelyet a Freescale Semiconductor, azaz a Motorola korábbi félvezetőgyártó üzemága *(Semiconductor Products Sector)* tervezett és gyártott. Az 1979-ben bemutatott HMOS technológiájú processzor az első tagja volt a sikeres 32 bites processzorcsaládnak, amely előre-kompatibilis volt a sorozat későbbi tagjaival, a 16 bites szélességű külső sín korlátai ellenére. 30 évvel később a 68000-es architektúra még mindig használatban van.

A 68EC000 a 68000 olcsó változata, beágyazott vezérlő alkalmazásokhoz tervezték. A 68EC000 8- vagy 16 bites adatsínt képes használni, és ezek között váltani is tud, egy reset segítségével.[11]

Architektúra

Az M680xxarchitektúra igen fejlett jellemzőkkel rendelkezett, amelyek más felépítésekben csak sokkal később jelentek meg. Ilyenek például a privilégiumszintek vagy a fejlett kivételkezelés. Az m68k egy belsőleg 32 bites architektúra, amely azonban mindig tartalmazta a 16 bites rendszerekhez való illesztés lehetőségét.

Az MC68000 főbb jellemzői:

- 32 bites adat- és címregiszterek
- 16 MiB lineárisan címezhető memória (24 bites címsín)
- 56 utasítástípus
- öt alapvető adattípus
- memóriába leképezett I/O (nincsenek külön I/O portok)
- 14 címzési mód

A 68000 család bájtsorrendje big-endian: az egy word/dwordben tárolt bájtok logikai és fizikai sorrendje megegyezik, hasonlóan a hálózati bájtsorrendhez. Alapvető adattípusok: bitek, BCD számjegyek, bájt (8 bit), szó (16 bit), duplaszó (v. *long*, 32 bit).

Címsín

A 68000 processzor elméletileg 24 bites címzést használt, de a 24 bites címet 23 bemeneti címvonal (A1-A23) és két vezérlőjel (UDS, LDS) segítségével határozta meg, amelyek az A0 értékét állították, így állt elő a 24 bites cím (A0-A23). Ezzel 16 MiB fizikai memóriát tudott megcímezni byte-os felbontásban. A címszámítás és tárolás minden esetben 32-bitet használt, csak a legfelső helyiértékű byte-ot nem vette figyelembe, a címvonalak fizikai hiánya miatt. Ez lehetővé tette a folytonos 32 bites címteret használó programok futtatását. A Motorola ezzel a belső 32 bites címtér előre-kompatibilitását igyekezett biztosítani, lehetővé téve a teljes 32 bites címzés használatát, amikor azt később megvalósítják.[13]

Ez azonban nem tartotta vissza a programozókat az előre-inkompatibilis kód írásától. A 24 bites szoftver, amely elhanyagolta a legfelső címbájtot vagy más célokra használta azt, hibát okozhatott a 32 bites 68k hardveren. Például a korai, 7.0 előtti Apple Mac OS verziók a memóriablokk fő pointereinek felső bájtját jelzőbitek tárolására használták (pl. 'zárt' v. 'törölhető'). A későbbi verziókban a jelzőbitek máshova kerültek és az Apple a Mac IIci 1989-es megjelenésével elkezdett "tisztán 32 bites" ROM-okkal ellátott gépeket szállítani.

A 68000 regiszterei

A 68000 processzorban a regiszterek 32 bitesek (néhány kivételtől eltekintve), az adat- és a címregiszterek külön vannak választva. Az **adatregiszterek** abban különböznek a címregiszterektől, hogy lehetővé teszik a 8- és 16 bites adatok írását úgy, hogy a magasabb

helyiértéken lévő bitek értéke nem változik, míg a **címregiszterek** írásakor mind a 32 bit megváltozik, és azokba nem is megengedett a 8 bites írás. A regiszterkészlet:

- 8 db 32 bites adatregiszter: D0, D1, D2, D3, D4, D5, D6, D7
- 7 db 32 bites címregiszter: A0, A1, A2, A3, A4, A5, A6
- Az A7 regiszter egyben a verremmutató (*stack pointer*), jele SP: az assemblerek ilyen néven is hivatkozhatnak rá. Az A7 felhasználói módban az USP, szupervizor módban a SSP regisztert címzi.
- 4 speciális regiszter:
  - PC – 32 bites programszámláló (a 68000 processzorban 24 bites)
  - SR – 16 bites állapotregiszter, az alsó bájtjára CCR néven lehet hivatkozni (*condition code register*, feltételkód regiszter)
  - USP – 32 bites felhasználói stack pointer
  - SSP – 32 bites szupervizor stack pointer

A **programszámláló** belsőleg 32 bites, de a 68000 processzor 24 bites címsína miatt a legfelső bájtot nem használja, így a PC értéke a 24 bites címtéren belül marad.

A regiszterek száma lehetővé teszi, hogy a processzor gyorsan reagáljon a megszakításokra, ilyenkor ugyanis a 8 adatregisztert (D0–D7) és 7 címregisztert (A0–A6) kell elmenteni, összesen 15-öt, viszont a normál működéshez is megfelelő a regiszterek száma. (Az A7 regiszter is menthető szupervizor módban).

A processzor **állapotregisztere** 16 bites és két bájtra osztható. Az alsó bájt, azaz a 0-7 bitek alkotják a "felhasználói bájt"-ot, amelyben a 68000 összehasonlító, számtani és logikai műveletei különböző biteket állítanak, amiket később feltételes utasításokban (pl. ugrásoknál) lehet felhasználni. Ezek: "zéró" (Z), "carry" (C), "túlcsordulás"/"overflow" (V), "extend" (X), és "negatív" (N) jelzőbitek. Az "extend" (X) jelző különbözik a "carry" átvitelbittől, a nem túlcsordulás okozta kilépő bitet tárolja. A 8-15 bitek alkotják a "rendszer bájt"-ot, ebben egyéb vezérlőbitek vannak, pl. a megszakítási maszk, szupervizor állapotjelző és a trace bit.

Utasításkészlet

A 68000 processzorcsalád egyik jellegzetes vonása a nagymértékben ortogonális utasításkészlet: majdnem mindegyik gépi utasítás használhat minden típusú címzést. Az utasítások 0, 1 vagy kéttényezősek lehetnek; az utóbbiak alakja:

```
CMD.S src, dst
```

ahol az **CMD** az utasítás mnemonikja, **.S** az adattípus utótagja, az **src** és **dst** a tényezők. Az **.S** utótag azt jelzi, hogy az utasítás milyen méretű adaton dolgozik: **S** helyett { B, W, L } állhat: a .B 8 bites (bájt), .W 16 bites (szó), .L 32 bites (long) adatot jelent. A kétoperandusú utasításoknál az **src** a forrásoperandust, az **dst** a céloperandust határozza meg.

Az utasításokat az utasításkód és címzési mód határozza meg, a minimális utasításméret 16 bit (2 bájt). Sok utasítás és címzéskombináció hossza ezt meghaladja, ekkor a teljes utasítás kiegészítő szavakat tartalmaz: az utasítások hossza egytől öt szóig terjedhet.

Címzési módok

- Regiszter indirekt címzés
- PC-relatív címzés
- Abszolút adatcímzés
- Regiszter direkt címzés
- Közvetlen címzés *(immediate)*
- Beleértett címzés *(implied)*

Abszolút vagy közvetlen címzésnél szimbolikus nevek is használhatók (ezt az assemblerek lehetővé teszik). A PC-relatív címzés pozíciófüggetlen kód esetén haszos. A gyors közvetlen módú

címzésnél az érték magában az utasításkódban tárolt 8 biten van megadva, és előjel-kiterjesztéssel lesz 32 bites értékre alakítva.

Utasítások

A 68000 processzornak 56 utasítása van. Az utasítások az alábbi csoportokba oszthatók:

- *Adatmozgató utasítások*: adatok mozgatására az univerzális MOVE utasítás szolgál, amely lehetővé teszi tetszőleges típusú adat mozgatását (másolását) a tárban, tárból regiszterbe és fordítva, valamint regiszterek között. A címmozgató utasítások szavas és duplaszavas adatok átvitelét teszik lehetővé. Az általános adatmozgató utasításokon kívül speciális változatok is léteznek.
- *Aritmetikai műveletek*: ADD, ADDX, SUB, SUBX, MULU (előjel nélküli szorzás), MULS (előjeles szorzás), DIVU, DIVS, CLR (törlés), NEG, NEGX (additív negáció), és CMP (összehasonlítás, amely a kivonáshoz hasonlóan állítja az állapotbiteket, de az eredményt nem tárolja), EXT (előjel-kiterjesztés), TAS, TST (tesztek). Ide tartoznak a gyors ADDQ, SUBQ utasítások is.
- *Decimális aritmetika*: ABCD, SBCD, NBCD (BCD negálás)
- *Logikai műveletek:* AND, OR, NOT (logikai tagadás, invertálás), EOR (kizáró vagy)
- *Léptető/eltoló és rotáló utasítások*: Bitkezelő utasítások a memóriában: BSET (1-re állít), BCLR (0-ra állít), BCHG (bit ellentettje) és BTST (ellenőrzés: az állapotregiszter Z bitjét 1-re állítja, ha a vizsgált bit 0)
- *Multiprocesszoros üzemmódot támogató utasítás*
- *Folyamatvezérlő utasítások*
- *Elágazások, Csökkentés-és-elágazás* DBcc (ahol a "cc" ugyanaz mint a feltételes elágazásoknál): ez az utasítás csökkenti egy adatregiszter értékét, ezután elugrik, ha a feltétel értéke még igaz **és** a regiszter tartalma még nem érte el a -1-et.
- *Rendszervezérlő műveletek*

A felsorolás nem teljes. Az utasításkészletet részletesen leírják a 68000 programozói kézikönyvek.

Működés

Privilégium-szintek

A CPU és később az egész processzorcsalád két privilégium-szintet valósított meg. A felhasználói mód *(user mode)* mindenhez hozzáférést adott a megszakítási szintek vezérlését kivéve. A szupervizor privilégium szint mindent megengedett. A megszakítások végrehajtása mindig szupervizor módban történt. A szupevizor módot a szupervizor bit jelezte a státuszregiszterben, ami látható volt a felhasználói módú programok számára is.

Az előnye ennek a rendszernek, hogy a szupervizor szint egy külön veremmutatóval *(stack pointer)* rendelkezett. Ez lehetővé tette a többfeladatos*(multitasking)* rendszereknek, hogy igen kicsi vermeket foglaljanak a feladatok (taszkok) számára, tehát a tervezőknek nem kellett külön nagyméretű memóriaterületet lefoglalni, amely képes befogadni az összes veremkeretet maximális számú megszakítás kiszolgálása esetén.

Megszakítások

A CPU 7 megszakítási szintet ismer. A szintek 1-től 7-ig szigorúan priorizáltak, tehát a magasabb szintű/számú megszakítás mindig meg tudja szakítani az alacsonyabb számú megszakítás kiszolgálását. Egy privilegizált utasítás szolgál az aktuális minimum megszakítási szint beállítására az állapotregiszterben, így blokkolhatók az alacsonyabb prioritású megszakítások. A 7. szint a **nem maszkolható megszakítás***(non-maskable interrupt, NMI)*. Az 1. szintet bármely magasabb szint képes megszakítani. A 0. szint azt jelenti, hogy nincs megszakítás. A megszakítási szint értéke az állapotregiszterben van tárolva és a felhasználói szintű programok számára látható.

A hardvermegszakítások kérésére a CPU felé három bemeneti vonal (láb) szolgál, amelyek kódolják a legmagasabb függőben lévő megszakítási prioritást. A megszakítások kódolására

általában egy külön megszakításkezelő szükséges, bár azok a rendszerek, amelyeknek nincs szükségük háromnál több hardvermegszakításra, közvetlenül csatlakoztathatók a megszakításvezérlő bemenetekre, kissé komplexebb megszakításkezelő szoftver árán. A megszakításvezérlő lehet akár egy egyszerű 74LS148 prioritáskódoló, vagy a VLSI perifériacsip része lehet, mint pl. a MC68901 multifunkciós perifériában, amit az Atari TT030-ban használtak, és amelyben volt UART, timer, és párhuzamos I/O is.

A "kivétel-táblázat", azaz a megszakítási vektorok (megszakításkezelők által használt címek) helye rögzített a 0-tól 1023-ig terjedő memóriaterületen, ez 256 db. 32 bites címet biztosít a megszakítások számára. Az első vektor a kezdő veremcímet, a második az indítókód kezdőcímét tartalmazza. A 3-tól 15-ig terjedő vektorok különböző hibák kezelőire mutatnak: sínhiba, címhiba, illegális utasítás, nullával való osztás, CHK és CHK2 vektor, privilégiumsértés; néhány fenntartott vektor a "1010 kód emulátor" és "1111 kód emulátor" számára (ezek a hexadecimális A és F kódokkal kezdődő, később bevezetendő utasítások, pl. koprocesszor-utasítások emulálására szolgálnak), és hardveres töréspont vektora. A 24-es vektor kezdte a **valódi** megszakítások sorát: hamis megszakítás (amikor nincs hardveres nyugtázás), az 1-től a 7. szintig terjedő autovektorok, ezek után 16 db. TRAP (csapda) vektor, ezek után néhány fenntartott, végül a felhasználói vektorok találhatók.

Mivel induláskor vagy resetkor legalább a kezdőkód címvektornak érvényesnek kell lennie, a rendszerek általában valamilyen állandó memóriát (pl. ROM) alkalmaztak a 0. címen, amelyek tartalmazták a vektorokat és az kezdeti indítókódot *(bootstrap code)*. Azonban az általános célú rendszerekben kívánatos, hogy az operációs rendszer képes legyen változtatni a vektorokat futási időben. Ezt vagy úgy érik el, hogy a ROM-ban lévő vektorok egy már a RAM-ban lévő ugrási táblázatra mutatnak, vagy valamilyen memóriaváltási mechanizmust alkalmaznak, amely a ROM-bankot egy RAM-memóriabankra váltja futási időben.

A 68000 nem felelt meg a Popek és Goldberg-féle virtualizációs követelményeknek a teljes processzor virtualizáció tekintetében, mert tartalmazott egy *MOVE from SR* utasítást, amely lehetővé tette a felhasználói módú programok számára a csak-olvasható hozzáférést a privilegizált állapot egy kis részletéhez.

A 68000 nem volt képes a virtuális memória egyszerű támogatására, amelynél szükség van a hibás memória-hozzáférések elfogására, majd helyreállításukra. A 68000-nél van ugyan sínhiba-kivétel, ami használható a hiba elfogásához, de ez nem ment el elég információt a processzor állapotáról, ami lehetővé tenné a program folytatást a hibát okozó utasítás után, miután az operációs rendszer lekezelte a kivételt. Több cég is készített 68000-alapú virtuális memóriát használó Unix munkaállomásokat, amelyekben két párhuzamos, de eltérő fázisú órajellel futó 68000 processzorral oldották meg ezt a problémát. Mikor a "vezető" 68000 hibás memória-hozzáféréssel találkozott, egy külön hardver megszakította a "fő" 68000 futását, megakadályozva ezzel, hogy az a memóriahibával találkozzon. Ez a megszakítási rutin kezelte a virtuális memóriafunkciókat és indította újra megfelelő állapotban a "vezető" processzort, hogy folytassa a megfelelően szinkronizált végrehajtást mikor a "fő" processzor visszatér a megszakításból.

Ezeket a problémákat kijavították a 680xx architektúra következő nagyobb revíziójában, amely az **MC68010** megjelenésével indult. A sínhiba- és címhiba-kivételek elég állapot-adatot helyeztek a szupervizor-verembe, lehetővé téve a helyreállítást, és a *MOVE from SR* utasítás privilegizált lett. Helyette a felhasználói módú programok új nem-privilegizált *MOVE from CCR* utasítást használhattak, az operációs rendszer pedig elfoghatta és emulálhatta a felhasználói módú *MOVE from SR* utasítást, ha erre szükség volt.

Koprocesszor használata

Az MC68000, MC68008, MC68010 nem tartalmaz koprocesszor-interfészt, mégis lehetőség van a Motorola 68881matematikai koprocesszor használatára. Ezeknél a processzoroknál a koprocesszor egy perifériavezérlő processzorként illeszthető a rendszerbe, amelynél a (lebegőpontos) regiszterek a memóriába vannak leképezve, és a koprocesszor interfész szoftveresen emulálható. Erre a célra rendelkezésre áll az "1111 kód emulátor" csapda, amely minden hexadecimális F kóddal kezdődő

opkód észlelésekor végrehajtja a megszakítást. A megszakítás-kezelőben emulálható a teljes lebegőpontos utasításkészlet, vagy meghajtható a fizikailag jelenlevő matematikai koprocesszor.