

Mikroprocesszoros újdonságok a telefonközpontban

TÖLGYESI LÁSZLÓ
BHG

A távközlési ipar jó úton halad előre afelé, hogy a mikroprocesszorok alkalmazásával a korábbi elektro-mechanikus vagy szabványos „diszkrét” integrált áramkörökkel felépített kapcsolástechnikai berendezéseknél sokkal jobb központokkal szolgáljon felhasználói számára, amelyek új szolgáltatásokkal rendelkeznek, alacsonyabb áron, kisebb teljesítményfelvétellel kevesebb karbantartást igényelnek, és sokoldalúbban tudják követni a későbbi változásokat.

Szemlélnk a Communications International 1981. októberi száma alapján a National Semiconductor új generációjú 16 bites mikroprocesszorának néhány olyan jellemzőjét ismerteti, amelyek a telefonközpontok tervezése, üzemeltetése és karbantartása során műszaki és gazdasági előnyöket jelentenek az új programozhatósági jellemzők, memória szervezés, bit-manipulációk és slaveprocesszoros elrendezés révén.

Nagy teljesítőképességük, alacsony árak, rugalmas programozhatóságuk, kis teljesítményszükségletük miatt a mikroprocesszorok ma már gyakorlatilag nem hiányozhatnak az új telefonközpontokból, akár a kiskapacitású alközpontok, akár a nagy főközpontokat tekintjük. A népszerű 8-bites mikroprocesszorok a kisközpontok központi vezérlő egységeként a hívásfeldolgozás és koordinálás kifinomultabb feladatait is ellátják; a nagy központok elosztott vezérlési rendszerében az egyszerűbb, ismétlődő feladatokat, mint például a vonalfigyelés, jelzés előállítás végzik, hogy a központi processzor terhelését csökkentsék és a telefonközpont működését gyorsítsák vele.

Egyes cégek becslései szerint a telefonközpontok fejlesztési és karbantartási költségeinek több, mint fele software tervezési költség, és ez az arány csak növekszik, ahogy a vásárlói igények és a piaci verseny nyomására egyre bonyolultabb és kifinomultabb szolgáltatásokkal látják el a központokat. A programfejlesztés során jelentős költségmegtakarítást lehetne elérni, ha az assembly nyelvek helyett magas szintű nyelveken történne a programozás. További megtakarítás lehetséges a programok bővítése és karbantartása során. Különösen fontos ez utóbbi a telefonközpontoknál, mivel a szolgáltatások és az előfizetők száma gyorsan növekszik. A Brit Posta szerint egy új alközpont felhasználási ideje alatt a korszerűsítések során szolgáltatásainak száma 150%-kal is több lesz mint eredetileg.

Az 1970-es évek közepéig csaknem valamennyi telefonközpontot assembly nyelven programozták, azért, hogy minimalizálják a drága memória hardware költségeit és biztosítsák, hogy a programok a

lehetséges legrövidebb idő alatt fussanak le a real-time feldolgozási követelmények kielégítéséhez. Az elmúlt öt évben néhány tényező jelentősen megváltozott, emiatt a magasszintű programnyelvek kedveltebb eszközzé váltak a legtöbb telefonközponti program számára.

A korábban drága félvezető memóriák ára drasztikusan lecsökkent. A jelenlegi ár körülbelül 5 százaléka annak, ami a 70-es évek közepén volt. A processzorok körülbelül tízszer gyorsabbak most mint korábban, a magasszintű nyelvek hatékonyabb utasításokkal rendelkeznek, a fordítóprogramok hatékonyabb kódokat generálnak, a programozók munkaköltségei növekedtek, miközben a piac elvárásai egyre összetettebb programokat igényelnek a termék élete során.

A programozás tiszta termelékenysége magasszintű nyelven rendszerint legalább kétszer akkora, mint assembly nyelven. Egy nagy telefonközpontnál ez az abszolút idő száz emberévekben mérhető.

Így elavultak a memóriaterületre és a gyors real-time végrehajtásra vonatkozó régi követelmények a sokkal olcsóbb memóriák, a hatékonyabb fordítóprogramok, gyorsabb integrált áramkörök alkalmazása következtében, valamint azért, hogy a nagy sebességeket igénylő, gyakran végrehajtásra kerülő kis programokat assembly nyelven programozzák. Például egy japán központban a működési időben végrehajtott utasítások 90 százaléka a kódok csupán 4 százalékából származik (6K a 160K-ból).

A magasszintű programnyelvek érzéketlenek a hardware változásokra, így a gyártó előnyösen alkalmazhat újabb technológiákat a szolgáltatások bővítésére vagy akár a költségek csökkentésére anélkül, hogy a korábbi software befektetése kárba vesznének akár a kipróbált software megbízhatóságának csökkenése miatt, akár további nagy software fejlesztési költség bekerülésével.

Jelenleg általánosan elfogadott az a vélemény, hogy a magasszintű programnyelvek használata nagyobb előnyöket nyújt, mint a memória többlet okozta hátrányok, így a legtöbb új közepes és nagy telefonközpontot úgy tervezik, hogy azok magasszintű programnyelveket alkalmazzanak. Példaként: a Brit Posta CDSS1 alközpontja Coral 66-ot alkalmaz assembly nyelvű betétekkel; az ITT 4080 Unimatja CDL (hívás leíró nyelv) nyelvet használ a hívásfeldolgozó funkcióira; a Northern Telecom SL-1-e egy olyan magasszintű nyelvet használ, amit szintén SL-1-nek hívnak, és az ALGOL-W-hez hasonló; az LM Ericsson AXE-je a PLEX nyelvet alkalmazza.

A CCITT tagjai a CHILL nevű, magasszintű nyelvet dolgozták ki, amelyet a múlt évben hoztak nyilvánosságra. Korábban is programoztak már mikroprocesszorokat is magasszintű nyelveken, de így azok fontos előnyeiket veszítették el az alacsony végrehajtási sebesség és a nem túl hatékony fordítás miatt szükséges jóval nagyobb memória miatt.

A National Semiconductor új mikroprocesszorát, az NS 16 000-t magasszintű nyelven való programozásra optimalizálták. Ezzel egy PASCAL nyelven írt tesztprogram memóriaszükséglete csupán 20%-kal nagyobb, mintha a programot assembly nyelven írták volna. A program PASCAL-ba történő kódolása viszont csak harmada időbe került, mint assembly nyelven.

Ez a nagyobb hatékonyság a célszerűbb utasítások, az utasítás szimmetria, az egységes címzés és a virtuális memória eredménye. Természetesen azok a program modulok, amelyek nagyon gyakran és nagy sebességgel kerülnek végrehajtásra, továbbra is programozhatók assembly nyelven.

Az utasítás szimmetria itt azt jelenti, hogy a központi processzor szolgáltatásai korlátozás nélkül kombinálhatók. Ez egyszerűbbé és gyorsabbá teszi a programozást, mivel a programozónak nem kell attól tartania, hogy programja valami olyan korlátozás miatt nem fog lefutni, amiről esetleg megfélekedett. Az NS 16 000-ban bármelyik utasítás, bármelyik címzési módon elérhet operandust, regisztert vagy memóriát.

MODULÁRIS SOFTWARE

A programozás idejének és költségeinek csökkentésére elterjedt módszer, hogy az egyes program modulokat egymástól függetlenül írják meg, majd a végén együtt fordítják őket. Így annyi munkát végezhetnek párhuzamosan, amennyi csak lehetséges. Ha azonban egy modult változtatni kell, akkor az összes modult újra együtt kell lefordítani, hogy biztos legyen az, hogy a megváltoztatott modulból és fordítva, mindegyik modul tud hívni szubrutinokat vagy adatokat. Ez a módszer jó akkor, ha a software-t tényleg „lágú” közeg tartalmazza, mint például az eredetileg alkalmazott forrásnyelv. Néha viszont az a hasznosabb, ha a felhasználó néhány software modul hardware-be tesz át, vagyis ROM-okba. Ez egyszerűbbé teheti az elosztást, növelheti a megbízhatóságot, és megnehezítheti az illegális másolást. Ehhez a ROM-nak a lefordított tárgykódot kell tartalmaznia. Ha később változtatás szükséges — és ez nagyon jellemző a telefonközpontokra —, akkor az összes ROM-ot ki kell cserélni, hogy az újra fordított kódot tartalmazzzák.

Az NS 16 000 az első olyan mikroprocesszor, amelynél a software modulok ROM-ban tárolhatók, de nem kell az összes ROM-ban tárolt programot kicserélni, ha egy modult megváltoztatnak. Ezt úgy érik el, hogy a ROM-ban nincs abszolút címzés hivatkozás. Helyette van egy külön „linker” táblázat a mikroprocesszor központi memóriájában és ez tartalmazza a különböző ROM-software modulok összes címzési információit. Ha ROM-software modult kell változ-

tatni, csak a „linkert” kell módosítani, az összes többi modul tovább használható. A modul író programozónak nem kell azt figyelnie, hogy modulja hol fog elhelyezkedni a memóriában. A rendszerprogramozó ezt a „linker” segítségével végzi el, ugyanis a „linkerre” bízhatja annak eldöntését, hogy egy adott modul hova kerüljön a memóriában.

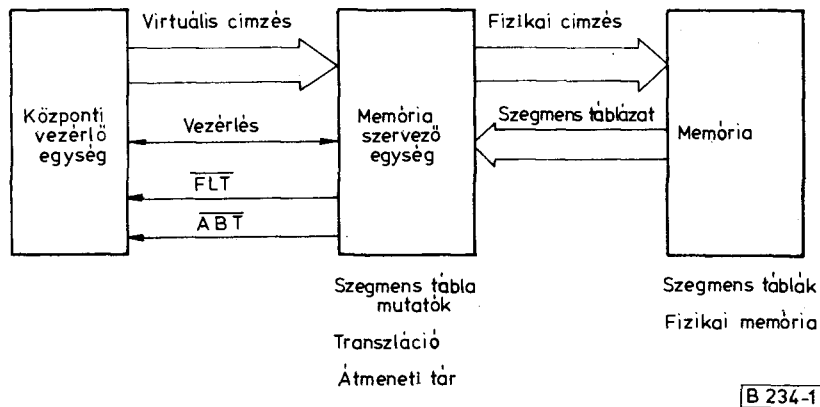
A software befektetések megőrzése érdekében lényeges, hogy bármelyik új generációjú eszköz a lehető legnagyobb mértékben legyen visszafele is kompatibilis. Az NS 16 000 család esetében az egyik elem, az NS 16 016 nevű például a „szülő” NS 16 000 kódján kívül a 8080 mikroprocesszor kódot is végrehajtja. Így a meglévő központok jól bevált 8080-as software-ja az NS 16 016-on is futtatható, méghozzá négyszeres sebességgel mint az eredeti 8080-on lehetséges.

VIRTUÁLIS MEMÓRIA

A software fejlesztés könnyebb, ha a programozónak nem kell a központi (RAM) memória pontos méretére ügyelnie. Mivel a 16-bites címzés csak 64K memóriaelem közvetlen hozzáférést teszi lehetővé, a programozónak memóriaszervezési feladatokkal is kell foglalkoznia, mivel a program a 64 Kbyte határon túl is terjedhet. Ilyenkor tudni kell, hogy mi van a központi memóriában, mi van mágneslemezen, és hogy melyik tár mekkora méretű. A kisebb központok igényeit a 64 K címzés kielégítheti, a közepes és nagyobb központok programjai 64 K-tól több mint egymillió szóig is terjedhetnek.

A legtöbb 16-bites mikroprocesszor úgy tud 64 K-nál nagyobb memóriát elérni, hogy a 16 címvezeték két 16-bites szót multiplexeinek. Gyorsabb memóriahozzáférést további címvezetékek alkalmazásával lehet elérni. Az NS 16 000 24 címzővezetékkel rendelkezik, így 16 Mbyte-os memóriát tud közvetlenül címezni. Alig hihető azonban, hogy az átlag felhasználó 16 Mbyte-os félvezetés RAM tárra vágyik, lévén a mágneslemez jóval olcsóbb; a teljes memória a gyors RAM és a lassúbb mágneslemez között oszlik meg. Erre a megosztásra szolgál például az NS 16 082 *szolga* processzor, amely a program virtuális címe és a központi memóriában levő fizikai cím közötti translációt végzi el. Ezzel a virtuális memória rendszerrel a programozó feladata könnyebbé válik, mivel úgy tekintheti, mintha egy egyformán gyors memória állna rendelkezésére, amely a központi és a háttér memória összege.

Az NS 16 000 normál memória ciklusideje 400 ns. Telefonközpontokban a hívásfeldolgozás, forgalomirányítás, hálózatfelügyelet és karbantartás közben az egyik leggyakoribb művelet az egybites, vagy kisértékű bitekkel végzett művelet. Ilyen az előfizetői hurkok állapotának letapogatása, ahol a nyert adatok 16-bites szavakba rakhatók össze, amint az a következő táblázatban látható. A különböző idejű minták összehasonlításával, logikai és bitkereső utasítások végrehajtásával a processzor megállapíthatja, hogy melyik előfizető milyen választ igényel; például: tárcsahang kapcsolása, ha az előfizető felemeli a kézibeszélőt, vagy a hívott lekapcsolása, ha a hívó leteszi a hallgatót.



1. ábra. Slave processzor – memória szervező egység

Előfizetői vonal	Bitpozíciók a szóban															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B: előző áll.	1	0	1	0	1	1	1	0	0	0	1	0	1	0	1	0
C: áll. váltás	0	1	1	1	0	1	0	1	0	0	1	1	0	1	1	0
D: híváskezd.	1	1	0	1	1	0	1	1	0	0	0	1	1	1	0	0
E: hlvásbef.	1	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0
A: jelen áll.	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	0

Megjegyzés: „0”: előfizetői hurok szakadt,
„1”: előfizetői hurok zárt.

Látható, hogy a „C” sor az „A” és „B” sorok antivalencia kapcsolata, az „E” sor a „B” és „C” ÉS kapcsolata, míg a „D” sor az „A” és „C” sor logikai összege. Az előfizetők azonosításához fontos, hogy a telefonközpont vezérlője gyorsan tudja meghatározni, hogy egy szóban mely bitpozíciókon található „1”. Erre a célra az NS 16 000 sorozat gyors bitvizsgáló és beállító lehetőségekkel rendelkezik, és könnyen tud változó, illetve nem szabványos hosszúságú elemekből álló adatstruktúrákat feldolgozni. A vizsgáló és beállító utasítások végrehajtási ideje kb. 2 μ s. Az adatok hossza 1-től 32 bitig terjedhet, és ezek egy nagyon hatékony címzési rendszer, valamint egy különleges bitműveleti utasításkészlet segítségével a memóriából akárhonnán közvetlenül elérhetők. Az NS 16 000 tehát nem-byte hosszúságú adatelemeket is kezelhet, és az adattárat a tipikus sebességhátrány bekövetkezése nélkül sűritheti. Egy másik olyan NS 16 000 utasítás, amely az állapotműveleteket gyorsítja meg, a „Keresd az első 1-t” utasítás, amely

egy szó letapogatásával annak a bitpozíciónak a számát adja meg, ahol az első 1-t találta. Ennek az utasításnak a végrehajtási ideje egy 32 bites kétszeres szóra 7 μ s körül van.

SLAVE PROCESSZOR

Ha egy valamilyen méretűre tervezett telefonközpontot később jelentős mértékben bővítenek, a mikroprocesszoros vezérlő teljesítőképességét nagymértékben növelhetik, ha *slave* processzorokat adnak a rendszerhez. Ennek megvalósítása az NS 16 000 sorozat esetében nagyon egyszerű, mivel semmiféle software változtatásra nincs szükség. A hozzáadott processzor a programozó számára átlátszó, vagyis úgy írhatja a programokat, mintha minden feldolgozást a központi mikroprocesszor végezne el. Az ábrán látható a *slave* processzor megvalósításának egy példája. Ez a memória szervező egység, amely a virtuális címeket fizikai címekké alakítja át. A központi processzor a rendszeren illegális memória címeket elkapja és az adatokat a memória szervező egységbe továbbítja, amely a fizikai címmé való transzlációt elvégzi.

KÖVETKEZTETÉS

A mikroprocesszorok nyújtják a felhasználóknak az árukhoz viszonyított legmagasabb teljesítményt. Alkalmazásuk már számos iparágat forradalmasított. Az új generációs 16 bites mikroprocesszorok feldolgozási kapacitása meghaladja a korai miniszámítógépeket és hasonló az új miniszámítógépekéhez.