

Kapcsolómező vezérlése mikroprocesszorral

HOLÉCZY GYULA

KKVMF Híradásipari Intézet



ÖSSZEFOGLALÁS

A cikk a 8 bites mikroprocesszorok alkalmazási lehetőségeit vizsgálja kapcsolórendszerek vezérlő áramköreiben. Ismertet egy rendszert, mely különböző irányú fejlesztések alapjául szolgálhat és kitér egy konkrét alkalmazásnál felmerült kérdésekre is.

1. Bevezetés

A számítástechnika gyors fejlődésének hatására a kapcsolóközpont-technikában is teret nyert a tároltprogramú vezérlés. Jellemző, hogy az első időszakban szinte valamennyi cég kidolgozta — részben műszaki, részben gazdasági szempontokkal indokolva — a saját speciális processzorát, ill. processzorait és természetesen hasonló volt a helyzet a software-fejlesztés területén is. Az integrált áramköri technológiák gyors fejlődése azonban hamarosan valószínűsítette, hogy a kapcsolórendszerek vezérlésében is jelentős szerephez juthatnak a mikroprocesszoros rendszerek.

A KKVMF Vezetékes Híradástechnika Tanszékén kezdtünk 1976-ban foglalkozni annak vizsgálatával, hogy a mikroprocesszorok milyen feltételekkel alkalmazhatók kapcsolómező-egységek vezérlésére. Ennek során elsősorban a következő kérdéseket tanulmányoztuk:

- a mikroprocesszorok teljesítőképessége és annak optimális kihasználása;
- a mikroprocesszor programozási lehetőségei;
- mikroprocesszorok együttműködése és a feladatok célszerű megosztása;
- a kapcsolástechnikai hardware célszerű kialakítása és illesztési lehetőségei;
- az üzembiztonság egyes kérdései.

Kutatásainkhoz egy megbízásos K+F tevékenység is kapcsolódott, részben ennek célkitűzései határozták meg vizsgálataink körét és lehetőségeit is.

2. Kiindulási feltételek

Vizsgálatainkat kisméretű (forgalomtól függően max. néhányszor 100 vonal kapacitású), 8 bites mikroprocesszorral vezérelt kapcsolóegységeken végeztük. Ennek során igyekeztünk kialakítani egy olyan, lehetőleg univerzális kapcsolórendszert, mely könnyen adaptálható különböző feltételekhez, funkcionális követelményekhez s így lehetővé teszi, hogy egymástól eltérő,

HOLÉCZY GYULA

oki. villamosmérnök egyetemi tanulmányait 1959-ben a BME Villamosmérnöki Karán fejezte be. Ezt követően, mint a BHG fejlesztő mérnöke, a jelzőgós és Crossbar központok honosításában vett részt. 1966-tól a KKVMF Vezetékes Híradástechnika

Tanszéke, majd annak utódja, a Híradásipari Intézet tanára, megalakulása óta a Kapcsolástechnika szakcsoport vezetője. Érdeklődési területe: a központok tároltprogramú vezérlésének, valamint a gyártás és ellenőrzés számítógépes automatizálásának hardware és software kérdései.

esetleg speciális követelményeket támasztó alkalmazási területekre (alközpont, diszpečser rendszer stb.) viszonylag rövid idő alatt is gazdaságosan lehessen rendszert kialakítani. A processzoros vezérlésnek ebben az esetben jelentős előnyei vannak:

- a) Az ilyen rendszerek funkcionális paraméterei a fejlesztési munka során esetleg menet közben is módosulhatnak. Így előnyös, ha egy univerzális, egyszerű és áttekinthető funkciót megvalósító hardware rendszer mellett a könnyen, olcsón változtatható software látja el a vezérlés minden bonyolultabb feladatát.
- b) A speciális célú fejlesztések esetén — eltekintve néhány előzetes labormodell-kísérlettől — a rendszer prototípusa általában egyúttal a végtermék is. Ezért érdemes olyan megoldásra törekedni, melynél a nagyobb darabszámban készülő áramkörökben minimálisra csökkenthető a tervezési és a gyártási hibák esélye s a várható hibák, javítások részben a software, részben egyes, néhány példányban készülő, koncentrált hardware funkciókat megvalósító áramkörök felé terelődnek.
- c) A processzoros vezérlés esetén könnyen megvalósítható az intelligens együttműködés más csatlakozó egységek, rendszerek processzoraival.
- d) A hatékony hibaellenőrzés, behatárolás a processzoros vezérlésnél viszonylag olcsó eszközökkel oldható meg és túlnyomórészt software igényeket támaszt (ugyanakkor a processzoros vezérlés esetén jóval több lesz a hibalehetőségek forrása is).

Kiindulási feltételként rögzítettük, hogy a vizsgálatokat az 18080 mikroprocesszoros rendszerre alapozzuk. Ezt elsősorban az akkori beszerzési lehetőségek, a jó software támogatás, perifériás áramkörökkel való ellátottság indokolta [1]. Az 18080 utasításkészlete, működési sebessége, a buszrendszer jellemzői alapján nyilvánvalóan alkalmas volt a tervezett rendszer vezérlésére, az optimális konfigurációt azonban további vizsgálatokkal kellett meghatározni (erre, valamint

Beérkezett: 1985. XII. 10. (#)

a más processzorokkal való összehasonlítás néhány kérdésére a továbbiakban még visszatérünk).

A várható üzemi követelményeket, valamint a beszerzési lehetőségeket és a gyárthatóság szempontjait figyelembe véve, a rendszerben a kapcsolómezőt áramtartásos reed-jelfogókkal alakítottuk ki. A vonalak illesztő áramköreiben a kapcsolási feladatokra és az egyenáramú hurok-figyelésre szintén reed-jelfogókat alkalmaztunk, részben az előbbi szempontok miatt, részben mert így a szokásosnál magasabb vonali zavar szintek esetén is megbízható működésre számíthatunk. A vonallezárást, mikrofontáplálást transzformátorral oldottuk meg. Az átviteltechnikai paraméterek így lényegében megfeleltek a hasonló rendszereknél szokásos követelményeknek.

3. A rendszer kialakítása

Rendszerünket három nagy egységre osztottuk:

- a kapcsolóhálózat,
- a vonalillesztő áramkörök,
- a processzoros vezérlő.

A processzoros vezérlő konstrukciós elkülönítését a koncentrált vezérlési funkciók mellett a tápáramellátás és a zavarvédelem szempontjai is indokolják. A processzoros vezérlés a rendszer többi egységével és az esetleges külső kapcsolódó egységekkel is zajvédett, párhuzamos buszvezetékeken tart kapcsolatot. A kapcsolómező és a vonaláramkörök elkülönítése már nem ilyen egyértelmű. Az összekötő kábelek mennyisége pl. jóval kisebb abban az esetben, ha az illesztő áramköröket közvetlenül a megfelelő kapcsolómező-egységek mellé telepítjük. Esetünkben azonban a variálhatóság követelményei: a kapcsolómező rugalmas kialakítása, esetleges bővítése, speciális vonalillesztők alkalmazási lehetősége csak úgy biztosíthatók, ha ezek az egységek egymástól el vannak különítve. Mivel vezérlésük is funkcionálisan elkülönített, ez is a választott megoldást indokolja.

3.1. Kapcsolómező egység

Miután a kapcsolómező a rendszerben önálló egységet alkot, mely a többi egységgel csak a bemenő és a kimenő vonalak, valamint a vezérlő busz vezetékeim tart kapcsolatot, megoldása mindig optimálisan illeszkedhet a kapcsolási feladatokhoz. Így kialakítható kisebb vagy nagyobb mátrix egységekből, esetleg többfokozatú linkkapcsolással, 2 vagy 4 huzalos továbbkapcsolásra. A rugalmas kialakítási lehetőség érdekében a vezérlő busz dekódoló áramköreit előnyösebb a kapcsolómező-egységben elhelyezni, viszont a zavarvédelem érdekében a kapcsolómező-blokkban nem használtunk semmiféle tároló-funkciójú elektronikus elemet (természetesen, ha a kapcsolómezőt elektronikus keresztpontokra dolgoznánk át, az elektronikus kapcsolómemóriát is ott kellene elhelyezni).

Konkrét vizsgálatokat egy diszpečersközpont céljára kialakított kapcsolómező egységen végeztünk, mely max. 150 bemenő vonalat kapcsolt 8 kezelői áramkör felé. A kapcsolási követelmények miatt ideális mátrixot alkalmaztunk. A részleges kiépítés lehető-

sége, valamint a gyors hibaelhárítás érdekében minden mátrix-oszlopot önálló konstrukciós kártyaegységként alakítottunk ki, beépítve a reed-kijelölő dekódolókat is. A mátrix-oszlopokat kiválasztó többlépcsős dekódolók külön kártyákon helyezkedtek el. Ez a megoldás természetesen helykihasználás szempontjából semmiképpen nem lehet optimális, erre azonban az adott esetben nem is törekedtünk.

A kapcsolómező-egységben elhelyeztünk ezenkívül egy párhuzamos lekérdező-sínt is, amelyen keresztül a vezérlés bármelyik mátrix-elem pillanatnyi állapotát azonosíthatja.

3.2. Vonalillesztő áramkörök

A speciális követelmények miatt — az előzőekben már körvonalazott szempontok szerint — többféle illesztő áramkört kidolgoztunk és kipróbáltunk:

- a) LB távbeszélő készülék illesztése, a jelzés mindkét irányban váltakozó áramú csengetőjel. Lehetőség van több felfűzött készülék csatlakoztatására is, ebben az esetben ezek szelektíven is hívhatók rövidhosszú csengetéskombinációkkal.
- b) CB készülék számtárcsa nélkül. A készülék hurokzárással jelzi a hívást, a készülék felé a jelzések hangfrekvenciásak.
- c) CB készülék számtárcsával. Az előzőeken túl a vezérlés tárcsázási hangot is küldhet és regisztrálja az impulzussorozatokat.
- d) Kétirányú fővonalai áramkör. Egy távbeszélő készüléket helyettesít, tehát alközpont, főközpont előfizetői vonalára csatlakoztatható. Figyeli a bejövő csengetést és a 400 Hz-es jelzéseket; a kiadott jelzések egyenáramúak.
- e) Kapcsológép vezérlésére alkalmas illesztő áramkör, revertív impulzállással.
- f) MFC jelzésű vonalakat illesztő áramkör.

Az a)–d) alatti áramkörök működését üzemszerű viszonyok között is vizsgáltuk. A másik két áramkörrel csak modell-áramkörön tanulmányoztuk a vezérlővel való együttműködés lehetőségeit.

Az egyes illesztő áramkörök helyigénye funkcióiktól függ. Így LB illesztőből 4, CB illesztőből 2, fővonalai áramkörből 1 áramkört lehetett elhelyezni egy kártyaegységen. Az univerzális felépítés érdekében először azt terveztük, hogy bármelyik csatlakozó pozícióba bármilyen típusú illesztő kártya bedugasztható legyen. Ez azonban végül túl sok felesleges csatlakozási pontot jelentett volna, előnyeit viszont nem sikerült hasznosítani. Így inkább azt a megoldást választottuk, hogy az egyes áramkör-típusok számát maximáltuk s ezek számára biztosítottunk elhelyezést, fenntartva a részleges kiépítés, ill. az egymás rovására történő esetleges bővítés lehetőségét.

Az illesztő áramköri kártyákon kaptak helyet a jelzésadó funkciókat kiválasztó dekódolók (pl. hangjelzés kiadás, hurok zárás-bontás stb.). Olyan kártyáknál, melyeken több áramkör helyezkedik el, ezek a dekódolók végzik a megfelelő áramkör kiválasztását is. Ez a megoldás hardware szempontból volt előnyös, a vezérlés szempontjából kényelmesebb lett volna az egyes áramkörök kijelölő pontjait külön-külön hozzá-

férhetővé tenni. Az egyes illesztő-kártyákat kiválasztó, többlépcsős dekódolókat itt is külön kártyaegységben helyeztük el. Külön szerelve helyeztük el a vonallezáró transzformátorokat s az esetleges vonalimpedancia korrektorokat is.

A rendszer működéséhez elegendő lenne, ha a vezérlő csak a bejövő jelzéseket figyelné (hívásérzékelő jelfogók, hangfrekvenciás szelektív vevő állapota). Az üzembiztonság, könnyebb ellenőrzés érdekében azonban lehetővé tettük a jelzések kiadását kapcsoló eszközök pillanatnyi állapotának lekérdezését is. Ezt a feladatot ugyanolyan lekérdező busz valósítja meg, mint a kapcsolómező egységnél (természetesen a vezérlőben nem szükséges e busz minden jelét felhasználni).

3.3. A vezérlő rendszer

A processzoros vezérlő kialakításánál elsősorban azt kellett eldönteni, osztott vezérlést alkalmazunk-e, s ha igen, milyen legyen ennek a struktúrája. Tanulmányozva a processzor utasításkészletét, az egyes utasítások végrehajtásához szükséges időket, meghatároztuk a legfontosabb vezérlési alpműveletek (pl. cím-meghatározások, lekérdezési ciklus, üzenetváltás stb.) időigényét. Ezt összevetve a várható forgalmi követelményekkel, arra a megállapításra jutottunk, hogy koncentrált vezérlés legfeljebb 100 vonal alatti, kisebb kapcsolóegységeknél alkalmazható megbízhatóan. Itt azt is figyelembe kell venni, hogy — amennyiben a lekérdező busz szolgáltatásait teljes mértékben kihasználjuk — egy 8 bites mintával nem lehet pl. 8 vonaláramkör állapotát beolvasni, hanem csak az egy kártyaegységen levőket. Így pl. egyetlen kétirányú fővonalis áramkör lekérdezése egy teljes buszműveletet igényel.

Az osztott vezérlés lehetőségeit vizsgálva, viszonylag kis kapacitású rendszerünkben a funkcionálisan osztott vezérlés alkalmazása bizonyult gazdaságosabbnak. A következő vezérlési feladatköröket különítettük el:

- lekérdező, „scanner” funkciók (vonallámpák, kapcsolómező egységek ellenőrzése);
- számjegykezelő, „regiszter” funkciók (impulzusüzemű jelzések, így számtárca-impulzusok, LB szelektív hívójelzések kezelése);
- beavatkozó műveletek (a Crossbar vezérlők analógiájára ezeket „marker” funkcióknak neveztük);
- I/O funkciók, esetleges külső eszközök kezelésére.

A terhelés mértékétől függően a funkciók összehozhatók. Így az általunk vizsgált kapcsolóblokkok esetében, ha a regiszter-műveletet igénylő áramkörök aránya nem volt túl nagy, a regiszter-funkciókat részben a scanner, részben a marker funkciókba olvaszthattuk be; kisszámú külső egységnél az I/O funkciókat teljes egészében a marker-blokk láthatta el. Így alapvetően egy kétblokkos vezérlő rendszer alakult ki.

A vezérlő blokkok tényleges felépítése megfelel a szokásos I8080 konfigurációnak, így ezzel részletesebben nem érdemes foglalkozni. A sajátos vezérlési feladatokról a következő fejezetben lesz szó.

4. Együttműködés a rendszer egységei között

A kapcsolórendszer megbízható működésének előfeltétele, hogy az egységek közötti kapcsolatot megfelelően alakítsuk ki. A részletek mellőzésével itt csak az egyes megoldások fontosabb sajátosságaira térünk ki.

4.1. Kapcsolat a vonalillesztő és a kapcsolómező egységek között

Itt a kapcsolat gyakorlatilag csak a vonalágak (a—b vezetékek) csatlakoztatására korlátozódik. Az összekötéseket mindkét végén dugasszal ellátott kábelekkel valósítottuk meg.

4.2. Kapcsolat a lekérdező egységgel

A scanner-blokk kezelheti egyedül a vonalillesztő egység, valamint a kapcsolómező egység lekérdező buszána jeleit. A kapcsolat itt szimmetrikus vonaladó és vonalvevő áramkörökkel valósul meg. Egyik irányban a scanner kiadja a kérdéses egység felé a kijelölt áramkör dekódolásához szükséges címet, majd a másik irányban a 8 bites lekérdező buszon beolvassa a megcímezett áramkör, ill. mátrix-egység állapot-kódját.

A lekérdezést nem az I8080 szokásos I/O utasításával, hanem „tárba ágyazott” címzéssel oldottuk meg. Ezt elsősorban a szóba jöhető egységek nagy száma indokolta; kijelölésüket perifériás címzés esetén egyetlen címzési lépcsőben nem lehetett volna megoldani. Ezenkívül így bővült a lekérdezési műveleteknél alkalmazható utasítások készlete is, ami egyszerűbb és gyorsabbá tette a lekérdező programot. Mivel a lekérdezt egységek válaszüzeje jóval nagyobb volt, mint a memória ciklusidő, azt a megoldást választottuk, hogy a lekérdezési címtartományban a processzor hardware úton „wait” üzemmódba kerül és kivárja a megcímezett egységtől érkező válaszjelet. A lekérdező buszon „bejelentkező” jelet nem alkalmaztunk.

4.3. Beavatkozó jelek leküldése

Beavatkozó jeleket csak a marker-blokk küldhet akár a vonalillesztő egység, akár a kapcsolómező egység áramkörei felé. Mivel itt kizárólag reed-működtető (mehúzó vagy elengedett) műveletek vezérléséről lehet szó, a reed jelfogók működési idejének áthidalásához átmeneti puffereles szükséges. Ezt úgy oldottuk meg, hogy a kapcsolómező egy-egy „logikai” mátrix-blokkjához egy I8255 áramkörrel kialakított átmeneti tárolót rendelünk s ennek kimeneti jelét kb. 5 ms késleltetésű monostabil multivibrátorral kapuztuk meg. Hasonló megoldást alkalmaztunk a vonalillesztő kártyák vezérlésénél is.

A markerben egy működtető üzenet kódja sosem kerül közvetlenül az I8255 áramkörökhöz; a feldolgozó program ezeket mindig a megfelelő áramkörhöz tartozó RAM puffereles helyezi el. A puffereles ürtését és az I8255 áramkörök feltöltését, aktiválását egy 5 ms-os belső órától indított megszakítási program végzi. Így könnyen kézben tartható az egyes műveletek sorrendi prioritása (pl. számjegy-impulzusok adásakor így biztosítható a jel-szünet arány pontos betartása).

A logikai mátrix mérete, tehát az egy puffer egységről vezérelhető kártyaegységek száma a kártyák típusától és a forgalmi viszonyoktól függ. Itt figyelembe kell venni, hogy adott puffernél a műveletek 5 ms-os ciklusokban követhetik egymást. Vizsgált egységeinknél — feltéve, hogy a kártyákat a forgalomtól függően csoportosíthatjuk — a maximális logikai mátrix-méret 8×32 -re adódott.

4.4. A vezérlő blokkok együttműködése

A vezérlő egységek együttműködésére alapvetően két módszer kívánkozik. A közös memóriaterület használata igen hatékony, azonban buszkezelési és program-szinkronizálási problémákat vet fel. Esetünkben ennek a módszernek a marker—scanner relációban lehet jelentősége, mivel a kártyaegységek memória-címzése folytán azok állapotához a marker-blokk így közvetlenül is hozzáférhet. Célszerű viszont a scanner-blokk tényleges memóriaterületét elzárni a marker beavatkozásai előtt. A két egység közötti üzenetváltáshoz ezért egy puffereit, párhuzamos üzemmódú perifériás típusú adatcserét is megvalósítottunk mindkét irányban. Hasonló megoldás kínálkozik akkor is, ha pl. külön regiszter funkciót megvalósítható vezérlő blokkot alkalmaznánk.

A processzor-busz egy I/O művelet során 8 bites adatcserét tesz lehetővé; a vezérlők közötti üzenetváltás jelblokkja így 3—4 byte átvitelét igényli. A processzorok műveleti idejének jó kihasználása érdekében ezért bizonyult előnyösebbnek a párhuzamos átvitelű puffertárakkal megvalósított adatcsere, bár ennek hardware eszközigénye jóval magasabb, mintha egyszerű soros adatbuszt alkalmaztunk volna.

4.5. Együttműködés külső eszközökkel

Ezt az együttműködést rendszerünkben kizárólag a marker vezérlőblokk valósíthatja meg (kivéve néhány, a processzorok működését közvetlenül befolyásoló perifériás eszközt). A külső eszközöket, melyekkel a kapcsolattartás szükséges lehet, két csoportba sorolhatjuk:

- egy kapcsolódó másik rendszer processzora;
- szokványos perifériás eszközök.

a) A legtöbb esetben egy külső processzorral való együttműködés biztosítására elegendő egy, megfelelő sebességű soros adatbusz (pl. RS—232 vagy V. 24.). Erre a célra kialakítottunk egy I8251 áramkörre alapozott, TTL kimenetű interface kártyát, ezt azonban később a tényleges adatcserére nem használtuk fel, részletes kezelő software sem készült hozzá (a beméréshez a kártyát ideiglenesen az egyik mátrix-blokk működtető kártyájának pozíciójába helyeztük).

Néhány alkalmazási területen viszont előfordulhat, hogy egy másik rendszer együttműködő processzorával szoros és igen aktív jelzést váltás szükséges (ilyen lehet pl. egy diszpécser központ esetén egy, a kezelők tevékenységét koordináló processzor). A hatékony, a processzorok idejét minimálisan lekötő jelzést cseré érdekében erre a célra egy handshake üzemi kétirányú, puffereit párhuzamos adatcsatornát is kialakítottunk, szimmetrikus vonaladó-vonalvevő párokkal

működtetett alacsony zavarérzékenységu adatvonalakkal.

b) A perifériás eszközök közül háttértár-illesztő alkalmazását nem vettük tervbe, ilyenhez software sem készült. A bemérések, kísérletek időszakában adathordozóként lyukszalagot alkalmaztunk, erre a célra illesztőkártya készült egy FS I500 olvasóhoz és egy PERFOMOM lyukasztóhoz. Ezek kezelő rutinjait valamennyi vezérlő blokk monitor programjában elhelyeztük. Párhuzamos illesztőkártya készült ezenkívül egy ORION ADP—2000 displayhez, azonban az ehhez szükséges megbízható működtető software méretei túl nagyok bizonyultak a rendszer arányaihoz képest, s így a monitor programokba nem került beépítésre.

A rendszer működéséhez szükséges elemi beavatkozási műveleteket a vezérlő blokkokhoz rendelt, a processzor-buszon át közvetlenül lekérdezhető kapcsolósorral valósítottuk meg, melyet pl. megszakításkérő billentyűvel aktiválhatunk. A vezérlőrendszer állapotát egy latch-es kijelzőtáblán ellenőrizhetjük, ezenkívül a legfontosabb buszvonalak állapotát közvetlen LED kijelzőkön állandóan szemmel tarthatjuk. Beméréskor a processzorok léptetett üzemmódban is működtethetők, s egy opcionális kártya segítségével hardware break-point is beiktatható.

5. A software kialakítása

A vezérlő blokkok önálló software rendszerrel rendelkeznek, ezek működését a kölcsönös együttműködési jelzések hangolják össze. A vezérlő alap-software EPROM modulokban helyezkedik el (ellenőrzés céljára azonban egyes részei áttehetők RAM modulba). Az alap-software főbb részei: egy kisebb monitor-program, mely az interrupt-kezelőt is tartalmazza; a jóval terjedelmesebb üzemi program és végül az ellenőrző programok. A rendszer kiszolgálásához szükséges táblázatok, változó adatblokkok jelentős RAM területet is igényelnek. Utóbbi statikus RAM áramkörökből épült fel.

5.1. A lekérdező rendszer

A scanner-blokk software feladata elsődlegesen a vonalillesztő egységek állapotának folyamatos felügyelete, az állapotváltozások értékelése és az eredmény továbbítása a marker-blokkhoz. A lekérdezési ciklusok szabályosságát hardware óráramkör biztosítja megszakításkéréses bejelentkezéssel. Regiszter funkciók esetén az erre kijelölt egységeknél sűrített lekérdezési ciklust alkalmazunk. A lekérdezési ciklus periódusát a processzor műveleti sebessége, a lekérdező rutinok hossza és a figyelt áramkörök száma meghatározza s ez egyúttal a rendszer teljesítőképességének felső határát is megszabja (ezért szükséges pl. az időigényes regiszter funkciók elkülönített kezelése).

A kiértékelő program állapot-táblázat segítségével dolgozik. A táblázatban minden lekérdezhető egységhez három memóriarekesz tartozik; ezek tartalma:

- a megelőző állapot kódja;
- egy (még nem regisztrált) állapotváltozás alatt eddig végrehajtott lekérdezési ciklusok száma;

— maszkregiszter, mely kijelöli a vizsgálandó bitet, ill. biteket.

Az értékelő program tehát állapotjelző bitenként szelektált és a ciklusok leszámolása alapján időtartamkódolással jelzések vizsgálatára is alkalmas. Ennek megfelelően, nem egyszerű állapotváltozásokat, hanem egy definiált jelkészlet valamelyik üzenetét adja tovább a marker-bloknak (pl. hurokzárás, hurokbontás, TH jött, FH jött, csengetés jött stb.).

A kapcsolómező egységeit a ciklikus lekérdezésbe nem vonjuk be; ezek állapotát csak az ellenőrző, felügyelő programok vizsgálják.

5.2. A beavatkozó rendszer

A marker-blokk gyűjti össze a beavatkozásokhoz szükséges összes információt, majd ennek alapján dönt a szükséges kapcsolási műveletekről és kiadja a megfelelő utasításokat; a megváltozott állapotokról szükség esetén jelzést küld a kapcsolódó vezérlő blokkokhoz is.

A marker-blokk alapprogramjai is EPROM-ban vannak elhelyezve s a táblázatok itt is jelentős RAM területet foglalnak el. A kapcsolómező és a vonalillesztő áramkörök állapotát ui. a markerben is nyilvántartjuk, ami a gyorsabb programfutást és a hatékonyabb hibaelenőrzést segíti elő. A marker-blokk RAM területén vannak ezenkívül a regiszter-jelzések kiadásához szükséges átmeneti számjegytárak is.

A marker-blokkban is van egy hardware óra, mely a reed-működtető áramkörök 5 ms-os ciklusait szinkronizálja; ez azonban csak akkor van aktiválva, ha van valamilyen kapcsolási műveleti igény (azaz nem ürült ki valamennyi átmeneti tár). A működtető áramköri egységek korlátozott száma miatt ui. szükség lehet a műveletek rövid idejű várakoztatására, regiszter műveleteknél pedig az impulzussorozatok végéig a számjegyek tárolására; a várakozó sorok kezelése a marker programok feladata. A hardware óra másik funkciója, hogy meghatározott ciklusokban ellenőrző, felügyelő programokat indít.

5.3. Megszakításkérés-kezelés

A processzoros vezérlésű kapcsolórendszerek egységei bonyolult real-time rendszert alkotnak s ebben gondosan össze kell hangolni a különböző műveletsorokat, rögzíteni kell az egyes műveletek, funkciók hierarchikus rendjét. Ezt csak egy jól kialakított megszakításkezelő rendszerrel tudjuk megoldani. Bár az I8080 processzor csak egyetlen, tiltható megszakításkérő be menettel rendelkezik, áramkörkészletében több különböző, többszintű hardware megszakításkezelő áramkört is találhatunk. Ezek segítségével alakítottuk ki a processzoraink megszakításkezelési rendszerét, melyben az egyes megszakítási szintekhez a következő műveletek tartoznak:

1. Magasabb szintű processzor üzenete.
2. Regiszter-funkciókat vezérlő óra.
3. Alacsonyabb szintű processzor üzenete.
4. Alapciklust vezérlő óra.
5. (Esetleges későbbi igények céljára fenntartva.)

6. (Az I/O eszközök számára fenntartva.)

7. Kezelő-kapcsoló sor bejelentkezése.

8. (Nincs felhasználva.)

Legmagasabb prioritású az I. szintű bejelentkezés. Az alpprogram szinten a scanner-blokkban a lekérdező alapciklus fut, a marker-blokk ellenőrző programokat hajt végre.

5.4. Ellenőrzés, hibafelderítés

A processzoros vezérlő rendszerekben nagy súlyt helyezünk a belső ellenőrzésre, hibafeltárára. Megfelelő programok segítségével a nem-katasztrofális hibák szűk körben behatárolhatók, a hiba következményeinek továbbterjedése megakadályozható. Az ehhez szükséges software terjedelme alkalomadtán túllépheti az üzemelési software méreteit is. Rendszerünkben az ellenőrzési folyamatokat túlnyomórészt a marker-blokk koordinálja, bár minden processzornak van saját ellenőrzési rendszere és önállóan futtatható teszt-program-csomagja is. Az ellenőrzésre az alábbi lehetőségek adódnak:

a) A RAM adatblokkok és a buszok ellenőrzésére minden modulban bizonyos, más célra nem használt tárrekeszekbe az inicializáláskor meghatározott kód-kombinációt írunk és azt üzem közben a processzor rendszeresen visszaolvassa. Ha eltérést észlel, megkísérli a rekesz tartalmát visszaállítani és az eredménytől függően ad hibajelzést.

b) A vezérlő program esetleges szétfutásának megakadályozására a program kritikusnak vélt pontjain csapdákat építünk be.

c) A hardware rendszert, valamint a kapcsolóhálózatot az eszközök állapotának rendszeres lekérdezésével ellenőrizhetjük. A kapott eredmények, valamint az állapotár által mutatott (vélt) állapotok összevetésével a hibák felderíthetők és be is határolhatók.

d) További lehetőség nyílik a teljes rendszer működésének ellenőrzésére az esetleg ki nem használt áramkörök összekapcsolásával felépített hívásokkal (pl. egy CB készüléket illesztő áramkör és egy fővonalis áramkör, más esetben két LB illesztő áramkör a—b ágainak összehurkolása útján).

e) A rendszerhez természetesen a tápellátás ellenőrzése is hozzátartozik, ez azonban túlnyomórészt hardware úton valósul meg. Így valamennyi biztosított tápfeszültséghez figyelő áramkör tartozik, mely a feszültség kimaradása esetén riasztást ad (a riasztórendszer önálló, független tápellátással rendelkezik). Ha a hiba a kapcsolástechnikai részben következik be, fennáll az a lehetőség is, hogy a hibajelzéseket a processzoros vezérlés lekérdezze és a hiba mértékétől függően tegyen lépéseket (pl. adott vonalcsoport áramköreinek kizárása, esetleg a tápfeszültség megjelenésekor inicializálás és visszakapcsolás normál üzemmódba stb.). Katasztrofális jellegű tápfeszültségkimaradás esetén a rendszer leáll és csak külső beavatkozással indítható újra.

Talán érdemes megemlíteni, hogy vizsgálataink során a hibáknak nagyobbik része adódott a tápáramellátó-rendszer valamilyen problémájából.

5.5. Programozástechnika

Vizsgálataink egyik célja a mikroprocesszorok optimális kihasználása volt, ezért rendszerünk programjai az I8080 assemblerben készültek. A programoknál nem törekedtünk optimális tárkihasználásra, ezért a futási idő szempontjából kritikus programrészeknél gyakran használtunk tárgyigenyes megoldásokat — pl. szubrutinok helyettesítése macroval —, valamint a hozzáférés szempontjából kritikus adatblokkok esetében is. Így pl. az egyszerű, ciklikus adatkezelés érdekében célszerű volt egy puffertár számára egy teljes 256 byte-os RAM modult lefoglalni, bár így kapacitása általában még 20%-ig sem volt kihasználva. Más területeken, ahol a futási idő nem volt kritikus, a program áttekinthetősége, a programírás egyszerűsítése érdekében engedtünk meg felesleges átfedéseket (pl. egy, felhasználásra nem is kerülő cím gépies meghatározása stb.).

A programíráshoz, ellenőrzéshez az első időszakban semmiféle önálló fejlesztő rendszer nem állt még rendelkezésre, ezért TPAi kiszámítógépünkön alakítottunk ki egy egyszerűsített fejlesztési elrendezést, melyet később egyes szimulációs vizsgálatoknál is hasznosítani lehetett.

6. Értékelés

Kutatásaink célkitűzése az volt, hogy meghatározzuk a mikroprocesszorok optimális alkalmazási lehetőségeit, hardware struktúráját és programozási módszereit kapcsolómező vezérlési feladatok esetén. Sikertült kialakítani egy olyan rendszert, melyen a különböző funkciók működését üzemi körülmények között is ellenőrizhettük.

A vezérlő rendszer működését statisztikusan jelentkező igények esetén részleges szimulációval vizsgáltuk. Ezt megkönnyítette az a tény, hogy a vezérlő a rendszer központtechnikai egységeivel egyszerű felépítésű lekérdező- és vezérlő-buszok segítségével tart kapcsolatot. E buszvonalakra pufferek segítségével illesztettük a TPAi számítógép perifériás buszvonalaait s így sikerült véletlenszerű állapotváltozásokat szimulálni s a vezérlő processzorok reakcióit ellenőrizni. Ennek során elsősorban a software-ben bukkantak elő igen kritikus pontokon levő, esetenként nagyon tanulságos hibák.

Az I8080 mikroprocesszorral egyébként kedvező tapasztalatokat szereztünk; néhány „szűk keresztmetszettől” eltekintve a programok egyszerűen írhatók, logikusan strukturáltak. A rendszernek van néhány különösen előnyös tulajdonsága, így a memória-címmezőbe illeszthető perifériák, önálló stack-memória kialakítási lehetősége, kétdimenziósan kezelhető adatblokkok. Utóbbit az teszi lehetővé, hogy a regiszterpáron keresztül, indirekten címzett adatblokknál a regiszterpár két regisztere önállóan is inkrementálható vagy dekrementálható s így az adatblokkban kétdimenziós mozgást biztosít.

Vizsgálataink gyakorlati részét sajnos a külső körülmények viszonylag szűk körre korlátozták (áramtartásos reed mező, I8080 vezérlő rendszer). Ilyenkor óhatatlanul felmerül az az igény, hogy felderítsük, milyen irányban és hogyan bővíthetnénk lehetőségeinket. Esetünkben elsősorban azt volt érdemes vizsgálni, hogyan alkalmazhatók céljainkra másféle típusú mikroprocesszorok.

A processzorok összehasonlítására sokféle lehetőség nyílik (pl. [2], [3]), meg kell azonban állapítani, hogy minden ilyen próbálkozás szubjektív jellegű és felhasználó-orientált. Így számunkra is elsősorban az volt érdekes, hogy a programjaink kritikus részei hogyan valósíthatók meg más típusú mikroprocesszorokkal. Ilyen irányú vizsgálódásaink a processzor-típusok gyors avulásával, a 16 bites mikroprocesszorok elterjedésével jórészt aktualitásukat veszítették, ill. csak elméleti jelentőséggel bírnak. Hazai lehetőségeinket tekintve talán a Z80 rendszerrel kapcsolatos néhány megállapítás lehet érdekes.

A Z80 magasabb órasebességéből adódó teljesítménynövekedés viszonylag egyszerűen értékelhető, így ezzel külön nem foglalkoztunk. Érdekesebbek azok a lehetőségek, melyek a központi egység struktúrájából, bővebb utasításkészletéből [4] adódnak. Ezek közül a fontosabbak:

- a belső regiszterek nagyobb száma,
- adatblokk-kezelő műveletek,
- speciális léptetési műveletek,
- bitkezelő műveletek.

A felsorolt műveletek egy része azonban a gyakorlatban sem futási időben, sem tároló területben nem jelent érdemleges megtakarítást, inkább csak programozástechnikai előnyöket nyújt. Más részük éppen ritka előfordulása miatt nem bír különösebb jelentőséggel (bár egy-egy kritikus programrészben roppant nagy segítséget nyújthatnak). Nagy előnyt jelent természetesen a kettős regiszter-készlet, mellyel elsősorban a többszörös stack-mentési műveleteket lehetett kiküszöbölni, programozástechnikai szempontból azonban kezelésük nehézkes. Előnyt jelent a bitkezelő utasítások alkalmazása is, bár nagyon sokszor nem helyettesíthették az I8080 maszkolásos technikáját, mivel a bitpozíció-kijelölés nem változtatható dinamikusan.

A Z80 processzorral végzett összehasonlítások azt mutatták, hogy — bár alkalmazása előnyösebb — az I8080-al nyert tapasztalataink túlnyomórészt érvényben maradtak.

IRODALOM

- [1] MCS—80 User's Manual (Intel Corp. 1977 oct.)
- [2] Henkel, T.: Hardware round up (Computer World, 1981 jul.)
- [3] Withworth, Ian R.: 16-bit microprocessors (Granada, New York)
- [4] Z80-as sorozat: CPU (Ipari Informatikai Központ, Budapest, 1984.)