

Főszerkesztő: HORVÁTH IMRE

A szerkesztő bizottság elnöke: ANGYAL LÁSZLÓ

SZERKESZTŐ BIZOTTSÁG

BHG

Laczko Endre
Bernhardt Richárd
Dr. Eisler Péter
Dr. Gosztony Géza
Honti Ottó
Klug Miklós
Tölgyesi László

ORION

Jakubik Béla
Baracs Sándor
Csernoch János
Froemel Károly
Sass Károly
Szabó Károly

TERTA

Bánsági Pál
Baján Tibor
Benedek Elek
Egerszegi Béla
Hutter Mihály

BHG ORION TERTA

MŰSZAKI KÖZLEMÉNYEK

XXVII. évfolyam

1981

11. szám

A híradásipar mérőautomatáiról.

HOLÉCZY GYULA
KKVMF

III. rész

Vezérlési rendszerek, programozás

1. VEZÉRLÉSI RENDSZEREK

A mérőautomata vezérlő rendszerének feladata az aktuális vizsgálat program végrehajtása, a vizsgálatban közreműködő eszközök, készülékek működésének összehangolása. Ennek során a vezérlő rendszer értelmezi a vizsgálati program soron következő lépését, előállítja az ehhez szükséges beavatkozó jeleket (digitális vizsgálat esetén: a bemenetekhez tartozó „gerjesztő vektor”-t) és gondoskodik arról, hogy ezek az előírt ütemezésben eljussanak a vizsgált egységhez. Megfelelő késleltetés után beolvassa a vizsgált pontok állapotára jellemző elektromos jeleket (digitális vizsgálat esetén: a kimenetekhez tartozó „válasz vektor”-t), összehasonlítja ezeket a vizsgálati program előírásaival és az eredmény alapján dönt a vizsgálat további menetéről.

A legegyszerűbb, alternatív ellenőrzés során (GO-NO GO teszt) a vezérlő rendszer csak a válaszjel helyességét ellenőrzi és hiba esetén leállítja a vizsgálatot. A hiba jellegére némi tájékoztatást nyújt az, hogy hol állt meg a vizsgálati program végrehajtása, ezenkívül esetleg a vizsgált rendszer állapota a leállításkor. A bonyolultabb — többé vagy kevésbé diagnosztizáló jellegű — vizsgálatok során a vezérlő rendszer hibajel analízist is végez (pl. összehasonlítja a várt és a nyert értéket és vizsgálja az eltérés jellegét, esetleg számszerű értékét is) a hiba pontosabb behatárolása érdekében; esetleg a hiba jellegétől függően további vizsgálati lépésekre is sor kerülhet.

Az elmondottakból következik, hogy a vezérlő rendszerrel (és természetesen a programokkal) szemben támasztott követelmények igen eltérőek lehetnek. A továbbiakban először a vezérlő rendszer kiválasztásával és a mérőautomata rendszerbe való beillesztésével foglalkozunk, majd külön tárgyaljuk a programozás kérdéseit.

1.1. A vezérlő rendszer kiválasztása

Egy mérőautomata vezérlő rendszerének kialakításánál elsősorban a következő megfontolásokat kell szem előtt tartanunk:

(Az I. rész a BHG ORION TERTA Műszaki Közlemények 1979. 1. számában, a II. rész az 1980. 4. számában jelent meg.)

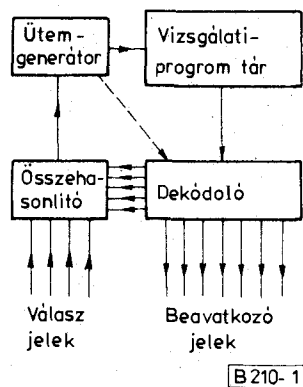
- a vizsgált áramkörök típusa (digitális, analóg vagy esetleg hibrid megoldások), választéka,
- a vizsgált áramkörök bonyolultsága, hozzáférhetősége,
- a vizsgálat jellege, a diagnosztizálás mélysége,
- egy átlagos kártyavizsgálat kívánatos időtartama,
- illeszkedés a rendszer más elemeihez, esetleg meglévő, egyéb rendszerekhez,
- a szakembergárda felkészültsége (a tervezés, programozás és vizsgálat szintjén),
- gazdasági megfontolások, beszerzési lehetőségek.

A felsorolt szempontok természetesen nem függetlenek egymástól, közöttük gyakran szoros összefüggés fedezhető fel. Így pl. a vizsgált áramkör bonyolultsága gyakorlatilag behatárolja a diagnosztizálás lehetőségeit, a kettő együtt erőteljesen befolyásolja a vizsgálat időtartamát. A hibák várható száma és eloszlása — amire a vizsgált áramkör bonyolultságából, alkatrészválasztékából a tapasztalatok alapján következtetni lehet — esetleg feleslegessé teheti a minden vizsgált kártyára kiterjedő részletes diagnosztizálást. Kisebb darabszámban gyártásra kerülő áramkörök esetén egyébként akkor is megfontolandó a részletes diagnosztizálás kidolgozása, ha bőséges és gyakorlott programozó gárda áll rendelkezésre.

A legegyszerűbb esetben a vizsgálat egyes lépéseit közvetlenül a fix adathordozón (lyukszalag, mágnesszalag stb.) rögzített program vezérli. Ez a vezérlési megoldás csak az egyszerű, digitális jellegű ellenőrzésre alkalmas, a kijelzési és a külső beavatkozási lehetőségek korlátozottak. A vezérlő program előállítására és rögzítésére általában számítógépet kell igénybe venni.

Egy ilyen vezérlő felépítését mutatjuk be az 1. ábrán.

A vizsgálati program egyes lépéseivel tartozó utasításkódokból a dekódoló segítségével állítjuk elő a vizsgált áramkör számára az aktuális beavatkozó jeleket, valamint a várt (etalon) jeleket. Utóbbiakat egy összehasonlító áramkör veti egybe a vizsgált áramkörtól érkező tényleges válaszjelekkel, majd az eredménytől függően vezérli az ütemgenerátort, mely vagy tovább lépteti a vizsgálati programot,



1. ábra. Vizsgálat rögzített programmal

vagy leáll és hibajelzést ad. Igényesebb megoldásnál a vizsgálati program tárolója, a dekódoló egyaránt PROM memória, az ütemgenerátor pedig ciklusszámlálókat is tartalmaz, melyek kapcsolatban állnak a dekódoló címzésével. Így egyes vizsgálati lépések szubrutin-jellegű végrehajtására is lehetőség van, esetleg a válaszjelek által módosított kiindulási állapotokkal.

A közvetlen vezérlésű mérő, ill. vizsgáló rendszerek beavatkozási és ellenőrzési lehetőségei korlátozottak, univerzális buszrendszerekkel való együttműködésre alkalmatlanok, így felhasználási területük a speciális célberendezésekre korlátozódik. Érdekes azonban megemlíteni, hogy a célberendezéssel esetleg jóval rövidebb vizsgálati idő érhető el, mint egy általánosabb célokra használható, processzoros vezérlésű mérőautomatával.

A korszerű univerzális mérőautomaták minden esetben processzoros, ill. számítógépes vezérlésűek. Csak így biztosítható ugyanis a vizsgálati lehetőségek széleskörű kiterjesztése és bővíthetősége, együttműködés univerzális buszrendszerekkel, különféle kijelzési és beavatkozási lehetőségek megvalósítása. A vizsgálati idő gyors működésű vezérlővel és optimális programozással csökkenthető. Az integrált áramkörti technika fejlődésével a processzoros vezérlő ára a mérőautomata rendszer egyéb eszközeihez képest egyre kisebb hányadot tesz ki. A vezérlés céljára általában egyetlen processzor elegendő, a követelményektől függően lehet mikroprocesszor, kalkulátor vagy kisszámítógép (ennél magasabb szintű konfiguráció csak összetett mérőautomata rendszerek együttműködtetése, komplex automatizálási feladatok esetén indokolt). A vizsgálati program előállításához felhasználható maga a vezérlő processzor is, de minden szempontból előnyösebb, ha ezt a feladatot külön számítógéppel végeztetjük el.

A mérőautomaták vezérlése általában nem igényel valamilyen sajátos, e célra fejlesztett processzort. Mint az ismertebb rendszerek példája bizonyítja, a vezérlés a legtöbb számítógépes, ill. processzoros rendszerrel megoldható. Kényelmes, de meglehetősen drága megoldás, ha a vezérlést olyan kisszámítógépes rendszerrel valósítjuk meg, amelyik már rendelkezik a lehetséges perifériák, háttértárak, buszillesztő áramkörök választékával és természetesen egy ennek megfelelő operációs rendszerrel is. Az ilyen rendszer szolgáltatásainak teljes kihasználása csak

egy igényes, sokoldalú és univerzális mérőautomata rendszerben remélhető. Mindenképpen olcsóbb megoldás, bár jóval több fejlesztési munkát igényel, ha a vezérlő rendszert pl. egy mikroprocesszoros rendszer áramkörválasztékából állítjuk össze, az alkalmazási terület igényeinek megfelelően. A processzoros mérőrendszer felépítését [1] részletesen tárgyalja, ezért erre itt nem szükséges kitérnünk.

A processzor kiválasztásánál a következő szempontokat tartjuk elsősorban szem előtt:

- szóhosszúság,
- utasításkészlet,
- ciklusidő,
- tárolókapacitás (operatív- és háttértár),
- perifériás rendszer kiépítési lehetőségei,
- megszakításkezelés (hardware interrupt),
- egyéb szempontok (beszerzési lehetőségek stb.).

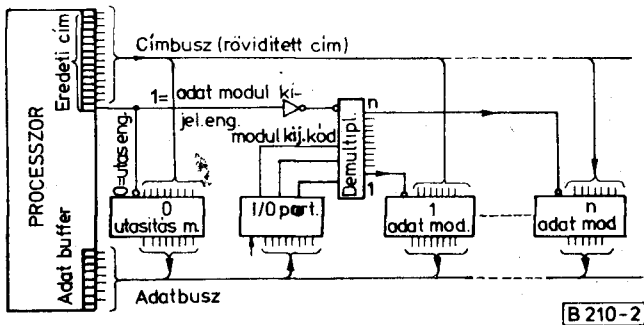
Megjegyzendő, hogy az azonos kategóriájú, különböző típusú processzorok sajátosságai között mutakozó eltérések gyakran kiegyenlítik egymást — különösen érvényesül ez a sokoldalú, univerzális rendeltetésű berendezéseknél. Így pl. egy 8 bites processzornál, mely elsősorban digitális vezérlési célokra készült, az utasításkészlet bytekezelésre orientált. Ezt a processzort mérőautomata vezérlésben alkalmazva, a perifériakezelő programok egyszerűek és rövidek lesznek. Ugyanakkor komoly programozási nehézségekre számíthatunk, ha pl. egy analóg mérőműszer lebegőpontos, 8–10 decimális számjeggyel beküldött mérési adatait kell kiértékelni. Fordított a helyzet, ha olyan processzort alkalmazunk, amelynek utasításkészletében a BCD kódolású aritmetikai műveletvégző utasítások dominálnak.

Természetesen vannak olyan szempontok is, amelyek esetenként szinte kizárólagosan érvényesíthetők. Így pl. egy IEC buszrendszerrel kiépített mérőautomata vezérléséhez elsősorban olyan processzorok jöhetnek szóba, melyek perifériás áramkörti készletében megtalálhatók az IEC busz-illesztők (feltéve, hogy ez a rendszer elfogadható időn belül be is szereshető).

A vezérlő processzor kiválasztásával kapcsolatos kérdések annyira szerteágazók, hogy részletes tárgyalásuk meghaladná ezen ismertetés kereteit. Ezért a következőkben inkább néhány jellegzetes összeállítás bemutatásával kíséreljük meg érzékeltetni az összefüggéseket és a konfiguráció nyújtotta lehetőségeket.

A processzoros vezérlő legegyszerűbb összeállítása: egy 8 bites mikroprocesszor maximális operatív tárolókapacitással, rendszervezérlő I/O egységekkel és egy manuális beavatkozó, ill. kijelző egységgel (kezelő pulttal). Alapperifériaként általában szükség van még egy lyukszalag-olvasóra és esetleg egy displayre vagy sornyomtatóra. A korlátozott utasításkészlet miatt várható, hogy a vizsgálati programot értelmező és végrehajtó programrészek terjedelmes szubrutinokból tevődnek majd össze. Az operatív tárolókapacitás általában 64 kbyte, adattárolás céljából azonban ez egyszerűen bővíthető további tármodulok beépítésével; az adatmodul-kijelölés perifériás I/O utasítással oldható meg.

A tárbővítés egy lehetséges megoldását be is mutatjuk (2. ábra). 16 bites címbuszot feltételezve, 32 k-s tármodulokat képezünk és a modulokon belül a tároló rekeszt a címbusz első 15 helyértékéből képezett, rövidített címmel jelöljük ki. A címbusz legfelső helyértéke választ az utasítás-, ill. adatmező között. Ha ennek a bitnek az értéke 0, akkor mindig az utasításmodul aktiválódik; ha az értéke 1, az I/O utasítással előzőleg meghatározott adatmodul kap kijelölést. Ez a megoldás főleg a vizsgálati programok tárolásánál előnyös, ahol nagy kapacitású ROM tároló tokokat használhatunk. Néhány száz kbyte kapacitásig olcsóbb egy diszk alkalmazásánál, programozása is egyszerűbb és jóval kisebb a hozzáférési idő.



2. ábra. Adattároló bővítés

Az értelmező és végrehajtó programrészek tárgénye elsősorban a mérőeszközök, készülékek számától és bonyolultságától függ. Jelentős többletigény lép fel, ha lebegőpontos decimális aritmetikai műveletek programjait is be kell építeni. Az így szükséges tárterület 8–24 kbyte-ra tehető, csak olvasható (ROM) memória. Ehhez járul még a pufferek, műveleti regiszterek, változó programrészek RAM tárgénye, ez azonban a teljes tárgényhez képest jelentéktelen, rendszerint 1 kbyte alatt marad (sok változót kezelő programnál esetleg néhány kbyte is lehet). Display alkalmazása esetén ide sorolható még a kiírás típuszövegeinek tárgénye is; attól függően változik, hogy milyen részletességű hibakiírásra, mennyire összefüggő üzenetekre tartunk igényt. Rövidítések alkalmazásával, a szövegfórmátum teljes leegyszerűsítésével a szükséges tárterület 1 kbyte alatt tartható; egyébként akár 8–10 kbyte is lehet.

Összesítve, az eddig tárgyalt programrészek rendszerint elhelyezhetők egy 32 kbyte-os tármodulban (melyben 0,5–1 kbyte RAM terület, a többi ROM, ill. PROM, esetleg EPROM).

A vizsgálati programok tartalmazzák tulajdonképpen egy-egy vizsgálandó objektum vizsgálatának egymást követő lépéseit, pl. rutincímek és konstansok formájában, melyeket értelmező és végrehajtó programok fejtenek meg és hajtanak végre. A vizsgálati programok tárgénye kizárólag a vizsgálati lépések számától függ – közvetve tehát attól, hogy milyen bonyolultságú a vizsgálandó objektum és milyen mélységű vizsgálatot, milyen szintű diagnosztizálást kívánunk megvalósítani. A digitális jellegű vizsgálati fázisok általában jóval kevesebb vizsgálati lépéssel oldhatók meg, mint a műszerbeállítás, mé-

rési eredmény értékeléssel járó analóg jellegű vizsgálati programrészek.

A vizsgálati programok terjedelme, tárgénye tehát széles határok között változhat. A következőkben néhány tájékoztató adatot közlünk, 8 bites vezérlő processzort, ill. adattárat feltételezve:

- egy 5000 dugaszpont körüli, kb. 70–80%-ban kitöltött, dugaszszávokból álló mező bekötéseinek ellenőrző programja kb. 6–10 kbyte terjedelmű (erősen függ az „egyedi” átkötések és a hosszú „hurkok” által elfoglalt pontok arányától);
- egy kis méretű, kb. 20 SSI áramkörrel felépített, 64 pontos dugasszal szerelt, digitális kártya egyszerű ellenőrző programja 2–4 kbyte, ugyanez részletesebb diagnosztizálással 8–10 kbyte;
- egy közepes méretű, kb. 40 SSI-MSI áramkörrel felépített, 128 csatlakozóponttal ellátott digitális kártya egyszerű ellenőrző programja, rövidített kiíró szövegekkel 12–16 kbyte, ugyanez részletesebb diagnosztizálással, értelmes mondatok kiírásával 20–30 kbyte;
- egy hasonló méretű, de részben analóg jellegű áramköröket és impulzustranzformátorokat tartalmazó kártya vizsgálati programja leg egyszerűbb esetben is 16–20 kbyte;
- egy nagy méretű, kb. 80 SSI-MSI áramkörrel felépített, 256 csatlakozóponttal ellátott, túlnyomórészt digitális jellegű kártya vizsgálatának programja várhatóan 30–60 kbyte (becsült érték);
- egy 40 lábú programozható LSI áramkör vagy egy 8×2048 bites PROM tároló tok funkcionális vizsgálata 3–5 kbyte;
- ugyanez etalon áramkörrel való összehasonlításra visszavezetve max. 1 kbyte;
- egy 8 bites LSI mikroprocesszor funkcionális ellenőrző programja, GO-NO GO teszt esetén 6–8 kbyte, célzott vizsgálat (részleges diagnosztizálás) esetén 10–16 kbyte tárolóterület igényel.

A vizsgálati programok objektumonként különböznek egymástól. A típusféleségek növekedésével – az operatív tár kapacitásának korlátozott voltából adódóan – elkerülhetetlenné válik valamilyen mágneses háttértár rendszerbe illesztése, ha lehetővé akarjuk tenni azt, hogy bármelyik vizsgálati program átdugaszolás (vagy a nehézkes, szalagról RAM tárterületre való beolvastatás) nélkül is bármikor hozzáférhető legyen. Az operatív tár kapacitását azonban háttértár alkalmazása esetén is célszerű akkorára tervezni, hogy a leghosszabb vizsgálati program is egyszerre lehívható legyen (a vizsgálati programtár ebben az esetben RAM modulokból épül fel). Ha az előbbi feltétel nem teljesül, a vizsgálati programban levő szubrutin jellegű vagy ugró hivatkozásoknál komoly nehézséget jelenthet a kérdéses programrészletek egyidejű lehívása. Itt jegyezzük meg, hogy ha a vizsgálati program hivatkozásokat (címkéket) tartalmaz, az ezek azonosítására szolgáló címtáblázat szintén a vizsgálati program tár-

igényét terheli és a program futása alatt teljes egészében az operatív tárban kell elhelyezkednie.

A mágneses háttértárral, kiíró perifériákkal ellátott, a programvezérelt eszközök, készülékek széles választékával rendelkező mikroprocesszoros mérő-automata rendszer lényegében minden vizsgálati technológiai feladat megoldására alkalmassá tehető. Ezeknél a rendszereknél a korlátok elsősorban abból adódnak, hogy a bonyolultabb feladatok elvégzése igen hosszú vizsgálati időt igényel s így a sorozatgyártásban való alkalmazás gazdaságtalan lenne. A vizsgálati időt egyrészt az értelmező és végrehajtó programok optimális kialakításával, másrészt a vizsgálati program lépéseinek csökkentésével rövidíthetjük. Utóbbi természetesen fordított arányban áll a vizsgálat mélységével, diagnosztizálási képességével. Érdemes megemlíteni, hogy elsősorban ez az a terület, ahol a mérőautomata felhasználójának módja nyílik az optimális megoldás kidolgozására.

A megoldások sokfélesége miatt itt is egyszerű lesz, ha néhány példán keresztül mutatjuk be az összefüggéseket és lehetőségeket. Továbbra is egy 8 bites mikroprocesszorral vezérelt rendszerből indulunk ki, a konfiguráció megfelel a tárgyigénnyel kapcsolatosan elmondottaknak. A tényleges vizsgálati idő két összetevőre bontható:

- a készülékek, eszközök beállításával kapcsolatos várakozási idő,
- a tényleges, programfutásra fordított idő.

A készülékek beállításából, a vizsgált objektum működésében előforduló késleltetésekből adódó várakozási idő — a számítástechnikai célokra használt számítógépekkel ellentétben — gyakorlatilag semmiféle programozási megoldással sem győzölköztethető ki; legfeljebb azt érhetjük el, hogy az egyidejűleg vezérelt készülékek működése jórészt átlapolja egymást. Az így adódó késleltetés esetén — a készülék jellegétől függően — néhány ms és néhányszor tíz ms között mozoghat. A lassú készülékeket működtető utasítások aránya a vizsgált objektumtól függően széles határok között változhat; lehetnek olyan áramkörök, melyeknél csak a vizsgálati program elején és végén van szükség beállításra s az ilyen programlépések aránya 1% alatt marad, más esetben viszont akár a 30%-ot is elérheti.

A tényleges programfutás ideje is erősen változhat, a vizsgálati programlépés jellegétől függően. Az előző feltételekből kiindulva, egy-egy gépi utasítás átlagos időigénye optimális utasításválaszték esetén sem csökkenthető 2–5 μ s alá. Egy-egy vizsgálati programlépés végrehajtásához igen egyszerű esetben (pl. várakoztatás) 50–100 gépi utasítás, egyszerűbb vezérlési, beavatkozási vagy értékelési programlépésekhez néhány száz, esetleg 1–2000 gépi utasítás szükséges. Összetettebb értékelő műveletek, többszörös ciklusszervezéssel járó kereső műveletek stb. gépi utasításigénye többször tízezer, sőt százezer is lehet. Természetesen ez nincs összefüggésben a tényleges tárterület igényel, hiszen a szubrutinokban, ciklusok belsejében levő utasításokat esetleg több százszor is felhasználjuk egymás után egy-egy művelet elvégzéséhez.

Ezek után már tehetünk néhány becslést. Egy egyszerű vizsgáló program, 2000 tényleges lépéssel,

lépésenként 1000 gépi utasítást és utasításonként 5 μ s ciklusidőt számítva 10 s alatt fut le. Ha feltételezzük, hogy a vizsgálati lépések 20%-ánál lesz átlag 5 ms-os várakozási idő, a futási idő 2 s-mal nő meg. A vizsgálati programot display kiírásokkal színesítve, a vizsgálati idő az előzőnek többszörösét is elérheti. Néhány — optimális esetre várható — érték:

- egyszerű kártya vizsgálata: néhány másodperc,
- bonyolultabb kártya teljes vizsgálata: néhány perc (várhatóan 10 perc alatt),
- 5000 pontos dugaszmező, 50 tényleges kötésel: 15 s,
- 5000 pontos dugaszmező, kb. 70%-ban kitöltve: 4–8 perc,
- 8 bites mikroprocesszor funkcionális ellenőrzése: 8–10 perc.

Érdemes megemlíteni, hogy a jó objektumok vizsgálati ideje elsősorban a vizsgálat alaposságától, mélységétől függ és elvileg független attól, hogy a program mennyire képes a hiba behatárolására, hiszen a diagnosztikai programrészeket csak hiba észlelése esetén kell végrehajtani. A gyakorlatban viszont egy diagnosztizálásra is alkalmas program felépítése sohasem lehet ideális, sok redundáns elemet tartalmaz és emiatt a vizsgálati idő a jó egységek esetén is meg fog növekedni.

Egy nagyobb teljesítményű, pl. 16 bites kisszámítógépes rendszerrel megvalósított vezérlés teljesítő-képessége sok tekintetben ugrásszerűen megnövekedhet. Egyrészt komoly előnyt jelent, hogy a kiválasztott konfigurációban a perifériás rendszer, a háttértárak, a könnyen kezelhető programrendszerek már rendelkezésre állnak; a rendszerbe illesztés feladatai leegyszerűsödnek. Ennek azonban veszélyei is vannak: a rendszerbe igen könnyen épülnek be felesleges, redundáns elemek, melyek egyrészt járulékos hibaforrást jelentenek, másrészt akadályozhatják a rendelkezésre álló eszközök optimális kihasználását. Egy szabványos buszrendszer például, univerzális felhasználásra alkalmas jelszekvenciával, egy korlátozottabb feladatkör megoldásánál nemcsak felesleges áramköri elemek beépítését jelenti, hanem a hasznos jelentés nélkül továbbított információelemek miatt a működési időben is indokolatlan növekedést okozhat. A programozás területén is jelentkezhet ilyen hatás, mert a magasabb szintű programrendszerekben megfogalmazva, egy-egy feladat végrehajtása jóval több gépi utasítást igényelhet, mint egy szigorúan az adott feladatra orientált programozástechnika esetén.

A nagyobb teljesítményű vezérlő alkalmazásával elsősorban a vizsgálati idő csökkentése várható. Ez a nagyobb szóhosszúság, a könnyebben kezelhető utasításkészlet és a kisebb ciklusidő együttes eredménye — feltéve, hogy a rendszernek nemcsak a kényelmesebb kezelhetőségét használjuk ki. A rendszer kiválasztásánál egyébként alapvető szempont lehet, hogy annak a többlétszolgáltatásai elsősorban az adott feladat megoldását segítsék elő. A tárolói igényre nézve lényegében állnak az előzőekben tett megfontolások, természetesen a nagyobb szóhosszú-

ság nyújtotta előnyök célszerű kihasználásával esetleg jelentős csökkenés érhető el.

Ebbe az irányba hat az is, hogy a bővebb utasítás-készlet segítségével egyszerűbb programokkal oldható meg ugyanaz a feladat.

1.2. A vezérlés beillesztése a mérőautomata rendszerbe

A vezérlő rendszer helyét a mérőautomata rendszerben [1] részletesen tárgyalta, az illesztő buszrendszerek kérdéseivel pedig [2] foglalkozott kimerítően, így itt az illeszkedés kérdését kizárólag a processzor oldaláról vizsgáljuk.

A mérőautomata rendszer egységei a vezérlő processzor szempontjából részben szokványos, részben egészen sajátos jellegű perifériás egységekként jelentkeznek. Előbbiekhez tartoznak a szokásos, kezelői együttműködést biztosító perifériás eszközök (Teletype írógép, display, sornyomtató, olvasó stb.), valamint a háttértárak vezérlő egységei. A második csoportot alkotják a mérőautomata rendszer programozható vagy vezérelhető műszerei, készülékei.

A két csoport eszközei többségükben vezérlési, illesztési szempontból egyaránt annyira különböznek, hogy célszerű az illesztésüket mind hardware, mind software oldalról különválasztani. Az alaperifériák perifériás buszrendszere és kiszolgáló programjai a legtöbb processzornál adottak, az adott rendszerbe illesztve legfeljebb kiegészítésre szorulnak. Egy, a rendszerben még nem szereplő perifériás eszköz illesztése is rutinfeladatnak tekinthető. A második csoportba tartozó sajátos készülékek illeszkedése már nem ennyire egyértelmű feladat. A buszrendszeren való együttműködés a byte-csoportok jól meghatározott szekvenciájú adását-vételét jelenti, kibővítve a hand-shake jelzéseivel. A buszrendszer illesztését az átviteli sebesség szempontjából is meg kell vizsgálni és ez felveti az átmeneti tárolás esetleges igényét is.

A nehézségek elsősorban akkor lépnek fel, ha a vizsgálati idő csökkentése érdekében több — viszonylag lassú működésű — eszközt kell egyidejűleg vagy egymást átlapolva működtetni. Ha a processzor működési idejéhez képest a buszrendszer is lassú, szóba jöhet több, független busz alkalmazása. Ezek között az egyes készülékeket úgy kell elosztani, hogy az egymást leginkább zavaró készülékek lehetőleg ne kapcsolódjanak azonos buszra. Ilyenkor általában nem nélkülözhető a jól kialakított, több szintű interrupt-rendszer. A megszakításkezelő rendszer egyébként minden esetben jelentős szerepet kaphat a teljes vizsgálati idő csökkentésében. Kivételt ez alól csak azok az egyszerű szervezésű vizsgáló rendszerek képeznek, amelyeknél a perifériás eszközök egyidejű működtetésének az igénye nem merül fel és így a processzor egyszerű várakozásos — lekérdezéses — üzemmódban működhet együtt a perifériás készülékekkel.

A processzor és a vezérelt rendszer kapcsolatáról szólva, még egy kérdést kell megemlítenünk, mely a speciális céleszközök vezérlésével kapcsolatos. Ezek

között ugyanis lehetnek olyanok, amelyek a processzor felé igen nagy számú perifériás pontként jelentkeznek; pl. egy dugaszmező vizsgálatánál esetleg több ezer ellenőrzendő pont, vagy egy nagyobb kapcsolóhálózat működtetési pontjai. Az ismert processzorok perifériás I/O rendszerének címzési lehetőségei viszont erősen korlátozottak, így a nagyobb számú perifériás egység kijelölése csak a buszon leküldött többlépcsős, szekvenciális címzéssel oldható meg. Az ilyen típusú perifériás egységeket ezért célszerűbb a processzor felől memória-modulként kezelni, így azonnal megszűnnek a címzési nehézségek. A memória-típusú perifériakezelésnek további előnyei is vannak. A használatos processzorok perifériás utasításainak köre ugyanis roppant szűk — általában csak az egyszerű adatmozgatásra korlátozódik — és a címzésnél is csak a direkt címkijelölés használatos. A memóriaként elérhető perifériás egységeknél ezzel szemben válogathatunk a különböző aritmetikai, logikai, adatkezelő utasítások között és az összetett címzési módok (indirekt, indexelt, relatív) széles köre áll rendelkezésünkre.

Az ismertetett megoldás természetesen csak olyan processzoroknál jöhet szóba, amelyekben a tároló-vonatkozású utasításoknál lehetőség van a processzor és a viszonylag lassú perifériás eszköz működésének összehangolására (wait-állapot). A memória-típusú perifériakezelés az esetleges interrupt-lehetőségeket nem korlátozza.

2. PROGRAMOZÁS

2.1. A programozási rendszer kialakítása

A mérőautomaták programozásának két sajátossága van:

- viszonylag nagy számú, sajátos rendeltetésű eszköz, készülék kezelése,
- ugyanazon a bázison sokféle, egymástól jelentősen eltérő felépítésű program futtatása.

Figyelembe kell venni ezenkívül azt is, hogy a tényleges vizsgálati programok készítéséhez várhatóan nem áll rendelkezésre nagyobb létszámú, képzett programozó gárda s így a program megírása esetleg az áramkör, ill. a vizsgált objektum tervezőjére hárul.

A vizsgálati programok megírását ezért csak egy magasabb szintű programnyelv (FORTRAN, BASIC típusú) segítségével képzelhetjük el. Alapvető szempont azonban, hogy a gépi kódban megjelenő bázisprogramok lehetőleg változatlanul felhasználhatók legyenek a különböző vizsgálati programok futtatása során, ezért ezeket PROM-ban tárolt szubrutinok formájában célszerű előállítani. Ha a vezérlő processzor rendelkezik FORTRAN vagy BASIC programozási lehetőséggel, megkísérelhető ezek kibővítése a berendezés vezérléséhez szükséges speciális programrészekkel, ettől a megoldástól azonban nem várhatunk optimális eredményt. Egyrészt azért nem, mert az alapprogramok — eltérő rendeltetésük miatt — sok felesleges elemet tartalmaznak, s ezek kiírása általában reménytelen feladat. Másrészt a FORTRAN vagy BASIC nyelvek gépi kódú reprezentánsai

rendszerint elég bőkezűen bálnak a gépi utasítás-készlettel s adattárolási rendszerük is elég nagyvonalú. A jelentősebb mérőautomata rendszerekhez ezért önálló vezérlő programot és vizsgálati programnyelveket dolgoznak ki (pl. EDITH, PEGAMAT-ATLAS, PEGAMAT-BASIC stb.).

A vezérlő programrendszer kialakításánál abból kell kiindulni, hogy milyen módon realizálódna majd a vizsgálati program egyes lépései. Már említettük, hogy a vizsgálati program készülékek, ill. gerjesztő-vektorok beállításából, majd mérési eredmények, ill. válaszvektorok értékeléséből tevődik össze. E „vektorok” előállítására többféle módszer ismeretes.

Az egyszerűbb — főleg digitális áramköröket ellenőrző — berendezések használják a szimulációs módszert. A vizsgálandó objektumot, áramkört részegységekre bontva írják le a szimulációs számítógép részére, mely analizálja az áramkört és ennek alapján előállítja a gerjesztő vektorokat és meghatározza a hozzájuk tartozó válasz-vektorokat. A gerjesztő vektorok előállítására egyszerű áramköröknél járható út lehet, minden lehetséges kombináció kipróbálása is, bonyolultabb rendszereknél azonban a számítógépek különböző szempontok alapján ki kell szűrnie a felesleges, megvalósíthatatlan vagy nem egyértelmű eredményt adó kombinációkat.

A szimuláció megvalósítható áramkörü alapelemként („tranzistor szinten”) is, a jelenleg használatos áramkörök leírása azonban ezzel a módszerrel így reménytelen vállalkozás lenne. Ezért legalább kapu-szintű szimulációt használnak, lehetővé téve összetettebb egységek beépítését is. Ez utóbbiaknál meg kell adni az egyes kivezetések elektromos paramétereit és jellemzőiket a szükséges vizsgálati szituációkban. Az így meghatározott egységekkel a szimulációs program mint kapu-szintű elemekkel tud számolni.

Bonyolultabb vizsgálati objektumoknál a szimulációs leírás már elég nehézkes, semmiképpen sem tekinthető gépies rutinmunkának; analóg működésű objektumok kezelése pedig erősen korlátozott s ebből már adódnak a szimulációs megoldás alkalmazásának korlátai is. A számítógép egyébként sem lesz alkalmas arra, hogy egy áramkör szellemét, rendeltetését megértse és ennek alapján döntsön a vizsgálati kombinációkról. A vizsgálatához szükséges gerjesztő-vektorok meghatározását legcélszerűbb az áramkör, ill. a vizsgálandó objektum tervezőjére, vagy alapos ismerőjére bízni. A válasz-vektorok előállítására ebben az esetben többféle lehetőség nyílik.

Alkalmazható a szimulációs eljárás egyszerűsített változata, ezt azonban nehézkessége miatt erre a célra nem szívesen használják. Abban az esetben, ha rendelkezésünkre áll egy teljesen jónak minősített vizsgálati objektum, célra vezető az etalon-módszer alkalmazása. Ennek két változata használatos. A komparáló (összehasonlító) módszernél a vizsgálat során a vizsgált és az etalon áramkör egyaránt a mérőautomatához kapcsolódik és az lépésenként maga végzi el a válaszvektorok összehasonlítását, az eredmény értékelését. A tanulós módszerrel viszont az etalon egységen végzett, első vizsgálatosorozat eredményeit

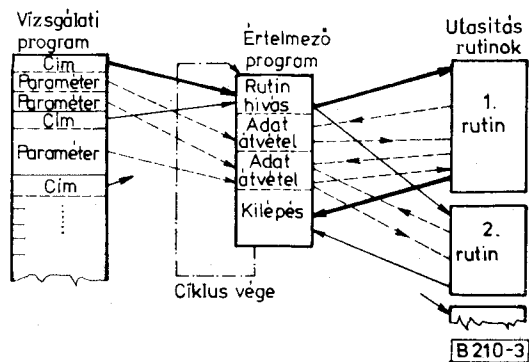
tárolja etalonként a processzor és ezzel hasonlítja össze a vizsgálat során a válasz-vektorokat. Emberi beavatkozásra mindkét esetben csak akkor van szükség, ha a válasz-vektorok egyes kombinációit ki kell rekeszteni a vizsgálatból. Az etalon-módszer nehézségei akkor jelentkeznek, ha a válasz-vektorok toleranciahatárai esetenként eltérőek.

A válasz-vektorok meghatározásának legáltalánosabban alkalmazható — és gyakran a leghatékonyabb — módja az emberi munka felhasználása. Természetesen ebben az esetben a legnagyobb a hibázás lehetősége is, emiatt a vizsgálati program hibafelismerő képességét célszerű ellenőrizni. Erre használhatók a szimulációs módszerek, ellenőrző vizsgálatok hibahelyzetek előidézésével stb.; ezek tárgyalása messze túlmutatna e cikk keretein.

A szimulációs eljárások kiinduló programnyelve valamilyen szimbolikus áramkör-leíró nyelv. A szimuláció eredményeként általában a kész vizsgálati programot kapjuk meg. Az emberi erővel készülő vizsgálati programok írására rendszerorientált szimbolikus nyelveket használnak. Ezek eszközbeállító, lekérdező, adatfeldolgozó, perifériakezelő, valamint különféle programszervező (ugró, szubrutinhívó stb.) utasításokból állnak, az egyes utasításokhoz egy vagy több operandus is tartozhat. Az utasításokat értelmes — esetleg rövidített — szavakkal, betűkombinációkkal, az operandusokat változókkal, konstans számértékekkel, esetleg összetett kifejezésekkel lehet megadni. Az így megírt vizsgálati programból egy- vagy többmenetes fordítóprogram segítségével állítható elő annak gépi reprezentációja, mely valamilyen közbenső adathordozó (lyukszalag, mágnesszalag) közvetítésével vihető be a vizsgálatot vezérlő processzor tárolójába.

A vizsgálati programrendszer felépítését a 3. ábrán mutatjuk be. A vizsgálati program célszerűen egy rutinhívó rendszerű program, melynek gépi megfelelője a rutincíméből és az esetleg ezt követő adatkonstansokból (paraméterekből) összeállított programegységekre tagolódik. Egy-egy programegység végrehajtásáról az értelmező program gondoskodik úgy, hogy először behozza a vizsgálati programmezőből az aktuális rutincímet, majd a program végrehajtását erre a rutinra tereli. A rutin végrehajtása közben a paraméterek átvétele az értelmező visszahívásával történik. Az aktuális programlépést az értelmezőben levő pseudo-utasításszámláló jelöli ki. A rutin végén a program visszakerül az értelmezőre és megkezdődik a következő programegység végrehajtása. Ez lehet a ténylegesen a tárban soronkövetkező programegység, alkalmazható azonban ugró utasítás is, amelynek rutinja felülírja az értelmezőben levő pseudo-utasításszámlálót. A vizsgálati programegységekhez ilyen esetben természetesen címkét kell rendelni, az ehhez tartozó effektív címet egy hivatkozási táblázatból keresi ki az ugró-utasítás rutin.

A vizsgálati idő csökkentésére a programozás oldaláról több lehetőségünk is van. Az egyik, alapvető szempont: az utasítás-rutinok optimális összeállítása. Különösen áll ez a gyakrabban használatos rutinokra; legkevésbé jelentős a lassú perifériás egységeket működtető készülék-rutinoknál. Néha egészen egyszerű megoldások hozhatnak jelentős ered-



3. ábra. Vizsgálati programrendszer felépítése

ményt. Így pl. egy többszörös szervezésű programciklusban érdemes lehet a szubrutin-hívások helyett a programrészeket megismételve beépíteni, ezzel futási időt nyerhetünk a tárolóterület rovására. Egy 8 bites mikroprocesszornál a szubrutinhívás és a visszatérés esetleg több időt vesz igénybe, mint egy rövidebb szubrutin hasznos utasításainak végrehajtása.

Az értelmező program szintjén a paraméterek átvételének optimális megszervezésével érhetünk el időnyereséget. Különösen áll ez a sok elemből álló paraméterekre (pl. összetett kifejezések átvétele). Ezzel egyébként már el is jutottunk a tényleges vizsgálati program írásához. Az időnyereség itt már a programozó kezében van.

A megoldás néha kézenfekvő: programrészek elhagyása, összevonása, paraméterek optimális megadása. Vannak azonban nehezebben felfedhető lehetőségek is. A készülék működtetése során a célszerű szervezéssel takaríthatunk meg időt, még akkor is, ha emiatt esetleg többletfeladat jelentkezik. A „többkevesebb” példajaként, inkább működjön esetleg négy készülék egyidejűleg, mint kettő egymás után. Az ilyen módszerek segítségével adott kártya vizsgálatánál még 20–30%-os csökkenés is elérhető a vizsgálati időben, ami nagy darabszámban gyártott kártyáknál igen jelentős lehet.

2.2. A fordítóprogramok felépítése

A fordítóprogram feladata, hogy egy — általában magasabb szintű programnyelven megírt — forrásprogramról egy, további fordításra alkalmas újabb forrásprogramot (többmenetes fordításnál), vagy egy végleges felhasználásra alkalmas tárgyprogramot állítson elő. A többmenetes fordításnak általában technikai oka van: kisebb processzoroknál így

csökkenthető a fordítóprogram tárigénye. Előfordul azonban, hogy a többmenetes fordítás segítségével különböző forrásnyelveken (esetleg különböző szintű) programrészeket fordítunk össze egyetlen közös tárgyprogramban.

A fordítóprogram több szegmensre tagolódik, ezek működését egy szervező program hangolja össze. Ez gondoskodik a háttértárban elhelyezett forrásprogram, valamint az aktuális programszegmensek behívásáról is. A fordítóprogram első szegmense végzi el a forrásprogram szintaktikai elemzését, összeállítja a változók, szimbolikus címkek értéktáblázatát, kiírja a fordítás közben észlelt hibákat.

A hibajavító programszegmens segítségével egyszerűen javítható a forrásprogram, majd az első szegmens meghívásával újból kezdődhet a fordítás.

A fordítóprogram következő szegmensei állítják elő a forrásprogramból — a kitöltött értéktáblázat felhasználásával — a tárgyprogram különféle változatait, melyek vagy a háttértárba kerülnek, vagy különböző adathordozókon (lyukszalag, mágnesszalag) jelennek meg. Ide sorolhatjuk a forrásprogramról készített programlistát is. A fordítóprogram ezenkívül tartalmazhat ellenőrző szegmenseket a tárgyprogram különböző változatai számára.

Nagyobb számítógéprendszerben a fordítás egyszerű és gyors, mert a teljes fordítóprogram és a forrásprogram is elhelyezhető a tárolóban, a részletes javító és szövegszerkesztő programok segítségével rövid idő alatt kijavíthatók a hibás programrészek. További előny, hogy egy ilyen rendszerben a fordítóprogramok előállítására is egyszerűbb feladat.

A mérőautomata vezérlő processzora ritkán alkalmas erre, így a vezérlő processzor kihasználásától függetlenül, a mérőautomata rendszerek többsége a vizsgálati tárgyprogram előállítását függetleníti a rendszer vezérlő processzorától és a fordítóprogramot valamelyik ismertebb számítógépre készítik el.

I R O D A L O M J E G Y Z É K

- [1] Dr. Kerpán István, Massziné Windisch Nóra: A híradásipar mérőautomatáiról. BHG ORION TERTA Műszaki Közlemények, 1979. 1. sz.
- [2] Temesvári Zsolt: A híradásipar mérőautomatáiról. II. rész. Interface rendszerek, programirányítású vizsgáló készülékek. BHG ORION TERTA Műszaki Közlemények, 1980. 4. szám.
- [3] Javaslat a kvázielektronikus távbeszélő központ kártyavizsgálatára. Tanulmány. KKVMF Vezetékes Híradástechnika Tanszék, 1978.
- [4] EDITH. Prüfmethode für IC-Schaltungen. Standard Telephon und Radio AG, 1975.
- [5] Siemens, Nachrichtennessgeräte 1979/80.