

# A Primotest2 program ismertetése

## A program fő célja

A program az A sorozatú Primo alapvető hardveres tesztelését segíti például egy utánépítés során az első indításkor. A kód kifejezetten úgy lett kialakítva, hogy a hardver minimális működése és az első ROM elérése esetén már képes legyen értékelhető segítséget nyújtani, azaz sem a többi ROM, sem a RAM, illetve a képszerkesztés vagy a billentyűzet működése nem befolyásolja az elindulását.

A speciális igények miatt a kód is speciális eszközök használatával készült. Mivel RAM meglétére és működésére nem támaszkodhattam, így szubrutinhívás nélkül, kizárólag a CPU-ban közvetlenül elérhető regisztereket használhattam. A szubrutinok kiváltására makrókat alkalmaztam, amik természetesen kirívó mértékben meghosszabbítják a lefordított kód méretét (mivel minden egyes felhasználásnál újra és újra behelyettesítődik a makró kifejtése a kódba), de ez ennél az alkalmazásnál nem jelent hátrányt. Az assembly forrás úgy lett kialakítva, hogy a tesztprogram mögötti ROM részek - az utolsó 8 byte kivételével - véletlenszerű adatokkal legyenek feltöltve, ez a ROM tartalom teszteléséhez segítség. A Primo ROM területének legvégén, az utolsó 8 byteban vannak letárolva a legenerált ROM tartalom egyes, külön tesztelendő részeinek kiszámított ellenőrző összegei.

## A program futásának menete

### Kimeneti kapu inicializálása

A legelső művelet a rendszer kimeneti kapura egy alapértelmezett érték kiírása. Az érték a következő beállításokat tartalmazza:

- 7 - NMI tiltása (0)
- 6 - joystick CLK értéke (0)
- 5 - külső magnetofon kikapcsolása (1)
- 4 - csipogó kimenet értéke (0)
- 3 - a megjelenített képernyő lap a normál működés szerinti felső lap (1)
- 2 - RSR232 kimenet értéke (0)
- 1,0 - magnetofon kimenet értéke 0mV (01 bitkombináció)

### 1 csippanás megszólaltatása

Amennyiben ezt a csippanást halljuk a Primo csipogóján, akkor a program végrehajtása sikeresen elindult.

### ROM #0 végigolvasása és ellenőrzése

Ez a programrész a Primo legelső ROM foglalatában lévő ROM tartalmát olvassa végig, miközben a beolvasott byteokból két különböző, egyszerű módszerrel (összeadással illetve bitenkénti kizáró vagy kapcsolatokkal) képez ellenőrző összegeket. A teljes ROM terület végigolvasása után a számolt értékeket összehasonlítja a fordítás során kiszámolt és a programban eltárolt elméleti értékekkel, hiba esetén 2 csippanással jelez, majd egy kis várakozás után a tesztprogram elejére ugrik vissza.

## ROM #1 végigolvasása és ellenőrzése

Ez a programrész a Primo második ROM foglatában lévő ROM tartalmával végzi el a ROM #0 ellenőrzésénél leírtakat. Hiba esetén 3 csippanással jelez, majd egy kis várakozás után a tesztprogram elejére ugrik vissza.

## ROM #2 végigolvasása és ellenőrzése

Következik a Primo harmadik ROM foglatában lévő ROM tartalmának ellenőrzése a ROM #0-nál leírtak szerint. Hiba esetén 4 csippanással jelez, majd egy kis várakozás után a tesztprogram elejére ugrik vissza.

## ROM #3 végigolvasása és ellenőrzése

Végül az utolsó, negyedik ROM foglatban lévő ROM tartalmával végzi el a ROM #0-nál leírt ellenőrzést. Hiba esetén 5 csippanással jelez, majd egy kis várakozás után a tesztprogram elejére ugrik vissza.

## A RAM első 256 bytejának ellenőrzése

Mivel 4000H címtől kezdődően minden Primoban működő RAM-nak kell lenni, így innen kezdődően 256 byte-ot ellenőriz minden egyes memóriarekeszbe egy tesztadat beírásával, majd a visszaolvasott adat ellenőrzésével. Ha valahol problémát érzékel (a visszaolvasott nem egyezik a beírt tesztadattal), akkor 6 csippanással jelez, majd egy kis várakozás után a tesztprogram elejére ugrik vissza.

## A RAM méretének megállapítása

Miután a program az előző lépésben megbizonyosodott arról, hogy legalább a RAM első 256 byteja használható, így beállítja ennek a területnek a végére a stack pointert. Innentől kezdve használható a CPU stackje, lehetne szubrutinhívásokat is használni a programban, bár jelenleg ez nincs kihasználva. A következőkben 4100H címtől kezdődően elkezd a memóriarekeszeket 4 különböző érték beírásával valamint visszaolvasásával ellenőrizni. A memóriarekeszek ellenőrzésekor a címet folyamatosan növelve addig teszteli azokat, amíg egy hibásat nem talál. Ez lehet a gépbe épített fizikai memória utáni első üres terület (A32 vagy A48 gépek esetén), illetve a címzés körbefordulásakor a 0000H cím (A64 esetén), ahol természetesen nem fog sikerülni az írás-olvasás, lévén ott a ROM #0 kezdődik. A megállapított összes RAM méretét a 4000H címre kétbyteos adatként letárolja későbbi felhasználásra. A RAM méretéből kiszámolja a Primo normál működése esetén használt képernyőmemória (felső lap a RAM tetején) kezdőcímét, majd ezt a 4002H címen kétbyteos adatként eltárolja.

## “Számlálás” a képernyőn

A képernyő teljes területét feltölti egy bináris értékkel. Rövid várakozás után a feltöltési értéket csökkenti és megismétli a feltöltést. Az egész feltöltés-várakozás ciklust addig folytatja, amíg a feltöltési érték eléri a nullát. A képernyőn a feltöltési ciklusok során különböző módokon függőlegesen csíkozott ábrákat kell látnunk. Ha a képernyőmemória helyesen működik, akkor a függőleges csíkokban nincsenek “rojtózódások”, kimaradt vagy extra pixelek, amik kilógnának a sorból.

## Képernyőkép invertálgatása

A képernyőt feltölti FOH értékekkel, aminek hatására 32 szabályos, függőleges csíkot kell lássunk. Ezután elkezd a képernyőmemória tartalmát olyan módszerrel invertálgatni, hogy a memóriából kiolvasott érték bináris ellentettjét írja vissza a képernyőmemória minden egyes rekeszébe. A teljes képernyőmemória átforgatása után kis várakozást követően megint végrehajtja az invertálást, és ezt jónéhányszor megismétli. Ha a képernyőmemória működése rendben van, akkor a műveletek során nem szabad megjelennie a képernyőn “szemétnek”, extra, a szabályos csíkokból kilógó, eltérő színű pixeleknek.

# Ugrás a teszt elejére

A teszt a legelejére ugrik, az egész teszt előlről kezdődik.

## A teszt használata

A primotest2 program lefordítása egy 16KB hosszú bináris illetve ugyanezzel a tartalommal ekvivalens Intel hex formátumú file generál. A tesztprogram tehát a Primo mind a négy eredeti, egyenként 4KB-os ROM területét lefedi. Amennyiben az eredeti ROM-ok helyére tesszük a tesztprogramot tartalmazó IC-eket, akkor 4db, egyenként 4KB-os ROM-ot kell megírunk. Ekkor a 16KB-os terület 4KB-os darabjait kell használni, vagy az íróprogramban a 16KB-os file-on belüli kezdőcím és hossz megadásával, vagy az írás előtt pl. a bináris file más eszközökkel történő feldarabolásával létrehozott 4KB-os file-okkal. Ha ROM-feltét panelt használunk, akkor egy teljes primonyi, 16KB-os területet kell biztosítanunk a tesztprogram számára.

## A teszt kiértékelése

Az alábbi áramköri elemek normál működőképessége esetén a program már el tud indulni:

IC1 - Z80; központi egység  
IC5 - 2732; ROM #0  
IC61 - 74LS139; memória címdekóder  
IC59 - 74LS155; I/O címdekóder  
IC55 - 74LS273; rendszer kimeneti kapu, csipogó  
IC60 - LM324; RESET áramkör  
IC42 - 74LS04; CPU órajel meghajtás  
IC39 - 74LS14; órajel generátor  
IC40 - 74LS74; órajel frekvenciaosztó  
IC41 - 74LS92; órajel frekvenciaosztó

Az elkészített, primotest2-t tartalmazó ROM(ok) behelyezése után a gép bekapcsolását követően azonnal kell hallanunk egy csippanást. Amennyiben ez nem történik meg, úgy elsősorban az előbbi listán lévő alkatrészek körül kell vizsgáldni, például órajelet mérni a CPU-n, a bekapcsoláskori RESET-et ellenőrizni, huzalozási hibákat keresni (szakadás, rövidzár).

A kezdő csippanás után a ROM tesztek alatt a képernyőn többé-kevésbé véletlenszerű, de statikus képnek kell lennie, "izgó-mozgó", vibráló, villódzó pontok vagy akár az egész képernyő területén ilyen jelenségek jelenléte a képszerkesztés és a memória kapcsolatának, vagy magának a képmemóriának a problémáira utalnak. Amennyiben a ROM területek ellenőrzésekor hibát tapasztal a program (az elméletitől eltérő ellenőrző összegeket számol), akkor a megfelelő számú csippanással (2, 3, 4 vagy 5 rendre az első, második, harmadik vagy negyedik ROM problémája esetén) jelzi ezt, majd egy kis várakozás után az egész tesztet előlről kezdi. Ilyenkor tehát a további tesztlépések nem is lesznek végrehajtva, a további tesztek előtt a ROM elérés hibáit kell kijavítani a gépben. ROM hiba észlelésekor főként a címdekóderek környékén kell vizsgáldni a Primo kapcsolási rajza alapján. A ROM tesztek alatti hiba esetén a további tesztekben semmi nem fog már lefutni.

A ROM teszteken sikeresen túljutva a RAM első 256 bytejának ellenőrzése következik, amennyiben ez sikertelen, akkor 6 csippanással jelzi a hibát, majd kis várakozás után a tesztet előlről kezdi. Ilyenkor a memóriaméret megállapítása és a képernyőmemória tesztek nem kerülnek sorra. A RAM észlelt hibája esetén elsősorban a memóriakonfigurációt kezelő átkötések (jumperek) állását, a címdekóderek és a címmultiplexerek működését kell ellenőrizni.

Ha a RAM első 256 byteja jónak bizonyul, akkor a RAM méretének a megállapítása és a képernyőmemória kezdőcímének a kiszámítása következik. A RAM méretének megállapítása során a képernyőterület is végig kell íródjon azonos adatokkal, amik azt eredményezik, hogy a képernyőn szabályos függőleges csíkozást fogunk látni. Ezután következik a képernyőterület csökkenő értékekkel történő feltöltése majd a képernyőtartalom invertálása tesztlépések (lásd fent

részletezve). Ezek a tesztek mind vizuális tesztek, azaz egy jól működő gép esetén mindig valamilyen szabályos függőleges csíkozást kell a képernyőn látnunk, a szabályosból kilógó, eltérő színű pixelek nélkül. Amennyiben “zizegés”, eltérő pixelek, “szemetelés” tapasztalható a tesztek során, akkor a képernyőmemória elérés körüli bizonytalansággal állunk szemben. Célszerű a RAS/CAS/MPX időzítésekénél jelen lévő RC tagok ellenőrzése, esetleg a kondenzátorok megmérése.

Mivel a tesztprogram csak az alapvető funkciókat tudja letesztelni a gép “belsejéből nézve”, az általa felderített anomáliákat nekünk kell kijavítanunk, a pontos okot nekünk kell mérésekkel és a kapcsolási rajz tanulmányozásával behatárolni. Sajnos a legtöbb ilyen jellegű hiba keresése és megtalálása elég komoly kihívást jelenthet és komolyabb műszerek (legalább egy 10-15MHz határfrekvenciájú oszcilloszkóp) is szükségesek hozzá. Ennek ellenére egy sima feszültségmérővel is nagyon sok olyan információhoz juthatunk, ami már szűkítheti a hibalehetőségeket, például a tesztprogram működése során a memóriákhoz vezető cím- vagy adatvezetékekre mérve gyanús lehet, ha bármelyik végig folyamatosan alacsony vagy magas állapotban van.