

INPUT-OUTPUT I-II.

Tartalom

INPUT-OUTPUT I-II.....	1
Perifériakezelési módszerek (Korrigált).....	2
A közvetlen memória hozzáférés (DMA)	4
Feladatok:	10
A megszakítás.....	12
Az Intel 8259 IT vezérlő működése, illesztése, programozása	12
8259A Interrupt Controller illesztése és felprogramozása	12
DMA.....	15
Az Intel 8237 DMA vezérlő működése, illesztése, programozása	15
8237 DMAC illesztése és felprogramozása	15
Mintafeladat:	15
Periféria illesztése a 8085' mikroprocesszorhoz	18
Periféria címzés dekóderrel	18

Perifériakezelési módszerek (Korrigált)

Feltétel nélküli

A perifériának (és persze a processzornak) mindig rendelkezésre kell állnia. Az átvitel in/out utasításokkal történik. Példák: kapcsolósor állapotának beolvasása, LED-ek kigyújtása latch-en keresztül. Előny: nagyon egyszerű megvalósítás. Hátrány: nem mindig alkalmazható.

Feltételes

A feltételes perifériakezelés megvalósítható:

- a) *állapotolvasással* • A periféria egy tároló beállításával (vonal szintjével, le/felfutó éllel, esetleg ezek együttesével) jelzi az átviteli igényét. • A processzornak folyamatosan olvasnia kell a periféria állapotát, egyébként az igény elveszhet! • Igény esetén a processzor végrehajtja az átvitelt. Előny: egyszerű hardver. Hátrány: Az állapotolvasás foglalja a processzort. Ha a processzornak más feladata is van, az állapotolvasás áttekinthetetlenné teszi a programot.
- b) *megszakítással* • A periféria a fenti módon jelzi az átviteli igényét. • A periféria jelzését közvetlenül vagy megszakításvezérlőn keresztül a processzor megfelelő bemenetére illesztettük. • A periféria igénye esetén a processzor által éppen végrehajtott program futása megszakad, az igény kiszolgálásra kerül, majd a megszakított program futása folytatódik. Előny: a processzornak nem kell a periféria állapotát figyelnie. Hátrány: valamivel bonyolultabb hardver, sok adat átvitele esetén a regiszterek értékének mentése és visszatöltése jelentősen ronthatja a teljesítményt.
- c) *DMA vezérlő segítségével* • A periféria a fenti módon jelzi az átviteli igényét. • A periféria jelzését a DMA vezérlő dolgozza fel. • A periféria igénye esetén a DMA vezérlő elveszi a rendszersínt a processzortól és az átvitelt maga hajtja végre. • Az átvitel megtörténtét a DMA vezérlőnek valamilyen módon jeleznie kell a processzor felé (például státusz beállításával vagy megszakítással)! • A DMA vezérlő visszaadja a rendszersínt a processzornak. Cikluslopásos átvitel alkalmazása esetén a rendszersínt a DMA vezérlő és a processzor időosztással használja.
Előny: Hatékonyabb adatátvitel – még a feltétel nélkülihez képest is, mivel nincs programvégrehajtás (például: in/out utasítások felhozása, mutatók és számlálók kezelése, stb.), az adatbuszon keresztül a memória és a periféria között csak az adatok áramlanak. Hátrány (az előnyök ára): bonyolultabb hardver (DMA vezérlő szükséges), a DMA vezérlőt fel kell programozni, és tipikusan megszakításkezelésre is szükség van.

Megjegyzés: A feltétel nélküli, az állapotolvasásos és a megszakításos esetekben az adatokat a CPU viszi át, míg DMA vezérlő alkalmazása esetén a DMA vezérlő hajtja végre az adatátvitelt.

Irodalom:

- Benesóczky Zoltán: Digitális tervezés funkcionális elemekkel és mikroprocesszorokkal, Műegyetemi Kiadó, Budapest, 1998. (143-144. o., 157-164. o.)
- Németh Gábor, Horváth L.: Számítógép architektúrák, 2. kiadás, Akadémiai Kiadó, 1993. (25. o.)

4.7. A közvetlen memória hozzáférés (DMA)

Forrás Németh G (25. oldal)

Az I/O-processzor mind adatblokkok átvitelére, mind byte-onkénti átvitelre felhasználható. Kizárólag adatblokkok átvitelére előnyösen felhasználhatók a *közvetlen tároló-hozzáférése*n alapuló eszközök (*DMA = Direct Memory Access*). A közvetlen tároló-hozzáférés lényege az, hogy a DMA modul magához ragadhatja a processzor tárolósínjének vezérlését, így az átvitelek a processzor közreműködése nélkül történnek. A processzor a DMA inicializálására átadja az operatív tárolóban kijelölt átviteli terület kezdőcímét, az átvendő blokk hosszát, az átvitel irányát, az átviteli üzemmódot és esetleges egyéb paramétereket. Ezeket a DMA modul vezérlőregisztereiben tárolja. Az adatblokk tényleges átvitelét az operatív tároló és a periféria (mely szintén lehet ugyanaz vagy egy másik tároló is) között a DMA modul irányítja, miután magához ragadta a processzor sínjének vezérlését. A processzorsín használatánál a konfliktusok elkerülése érdekében a DMA modul elkéri a sín használati jogát a processzortól, és a processzor azt megfelelő protokoll használatával átadja.

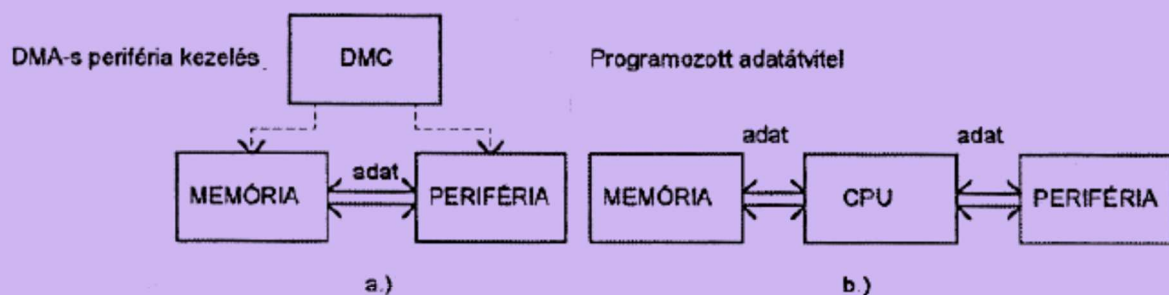
A DMA modul a processzorsínt az adatblokk átvitelének teljes időtartamára magához ragadhatja. Ebben az esetben a processzor működését fel kell függeszteni. A másik lehetőség - melyet *cikluslopásnak* neveznek - az, hogy a processzorsínt a DMA modul és a processzor időosztással használja (egy processzor használat, egy byte átviteles DMA használat ismétlődik). Ez utóbbi esetben a processzor működését nem kell felfüggeszteni a DMA működésekor. A *cikluslopás feltétele azonban*, hogy a tároló ciklusideje legfeljebb a fele legyen a processzor ciklusidejének, ez azonban *csak lassú, kis rendszerek esetén engedhető meg.*

A közvetlen memória hozzáférés (DMA)

[1] 157-162

Forrás Benesóczky

Az ún. DMA vezérlő segítségével a processzort kikerülve, közvetlen adatátvitel lehetséges a memória és a periféria között (4.50a ábra). Ezt nevezik közvetlen memória hozzáférésnek (Direct Memory Access). Ennek egyrészt az az előnye, hogy mivel az adatátvitelt egy speciálisan erre a célra kialakított hardver végzi, az többnyire gyorsabb, mintha a CPU végezné (nem kell közben a memóriából utasításokat olvasni és nem kell az adatot a CPU-n keresztül áramoltatni, (4.50b ábra). Másrészt, a DMA-s kezelés tehermentesíti a processzort (egyszerűbb lesz program). A DMA-s kezelést sokszor nagy sebességű, blokkos adatátvitelt igénylő perifériáknál alkalmazzák (floppy disk, winchester).



4.50. ábra. Az adat útja DMA-s és programozott kezelés esetén

Az átvitel lebonyolítását az ún. DMA vezérlő (DMC) végzi. Ez egy speciális periféria, amely felprogramozása után busz master funkciót képes ellátni. A DMA vezérlő főként a periféria és a memória közötti átvitel CPU-nál gyorsabb elvégzésére készített speciális hardver elem.

A DMA-s átvitel lebonyolítása a következő fázisokból áll: felprogramozás, átvitel kezdeményezés, busz vezérlési jogának átvétele, adatátvitel lebonyolítása, busz vezérlési jogának visszaadása, átvitel végének jelzése. Ezeket részletezzük a következőkben.

Felprogramozás

Az átvitel lebonyolításához a DMA vezérlővel közölni kell a következőket:

- az átvitel irányát (periféria → memória, memória → periféria, és lehetőség van memória-memória átvitelre is),
- az átvendő adatblokk memóriabeli kezdőcímét
- az átvendő blokk méretét (szószám)
- az átvitel módját (lásd később)

A DMC a felprogramozási fázis alatt szokványos perifériaként viselkedik, s a fenti információk a megfelelő regisztereibe kiírt adatként kerülnek bele.

Az átvitel kezdeményezése

- A periféria kezdeményezi, a DMC megfelelő kérés vonalán keresztül, vagy
- a program kezdeményezi a DMC-nek adott parancsszóval.

Az utóbbi főként memória→ memória átvitel esetén szokásos, ill. olyan perifériáknál, ahol egy kérés hatására megtörténhet a teljes blokk átvitele.

CPU-DMC arbitráció

A felprogramozás után, ha DMA átvitel kezdeményeződik, a DMA vezérlő az arbitrációs ciklus folyamán átveszi a busz vezérlési jogát a CPU-tól (busz mاستerre válik). A busz elkérésére a processzorok két jele szolgál.

DRQ Ezen a bemeneten jelzi a buszt elkérő egység, hogy használni kívánja a buszt.

DACK Ezen a kimeneten jelzi a CPU, hogy lekapcsolódott a buszról.

A busz elkérésére a processzorok az aktuális buszciklus befejezése után reagálnak. Ilyen módon akár egy utasítás végrehajtást is meg lehet szakítani a busz elkérésével. A processzorok egy része a buszhoz fordulást nem igénylő (belső) műveleteket ilyenkor is folytatni képes.

Adatátvitel

Az DMA-s átvitel során a DMA vezérlő megfelelő buszciklusok generálásával átviszi az adatforrásból az adatokat cél egységbe. A buszciklusok hossza általában bizonyos határok között programozható, ill. lehetőség van külső jellel annak meghosszabbítására (a wait kéréshez hasonlóan).

DMC-CPU arbitráció

Az átvitel befejezése után a DMC visszaadja a busz vezérlésének jogát a CPU-nak.

Az adatátviteli módtól (később részletezzük) függően vagy a teljes adatblokk átvitele alatt a DMC-nél van a busz vezérlésének joga, vagy időnként visszaadja. Ezért a DMC felprogramozásakor megadott számú adat átvitele során esetleg többször ismétlődik a következő három fázis: CPU-DMC arbitráció, adatátvitel, DMC-CPU arbitráció.

Az átvitel befejeződésének jelzése

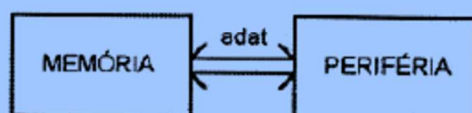
A teljes adatblokk átvitelének végét a DMA vezérlő vagy a periféria jelzi többnyire interrupttal.

Az átvitel befejezése után a program a bejövő adatokat feldolgozza ill. előkészíti a következő kimenő adat blokkot.

Attól függően, hogy egy szó átvitelét hány buszciklussal bonyolítja le a vezérlő, megkülönböztetünk egyciklusú és a kétciklusú átvitelt.

Egyciklusú DMA átvitel

Egyciklusú átvitelnél közvetlenül kapcsolódik össze a periféria és a memória (4.51. ábra) az adatbuszon keresztül, s egy byte átvitele egy ciklus alatt megtörténik. Ilyen átvitelre alkalmas az I8237 típusú DMC. (Az IBM PC DMA vezérlője is ilyen kialakítású.)



4.51. ábra. Adat áramlás egyciklusú DMA átvitel esetén

A 4.52. ábra egy egyciklusú memória-periféria átvitelre képes DMA vezérlő és a periféria kapcsolatát mutatja. A vastagon húzott jelek periféria → memória irányú átvitel esetén aktivizálódnak.

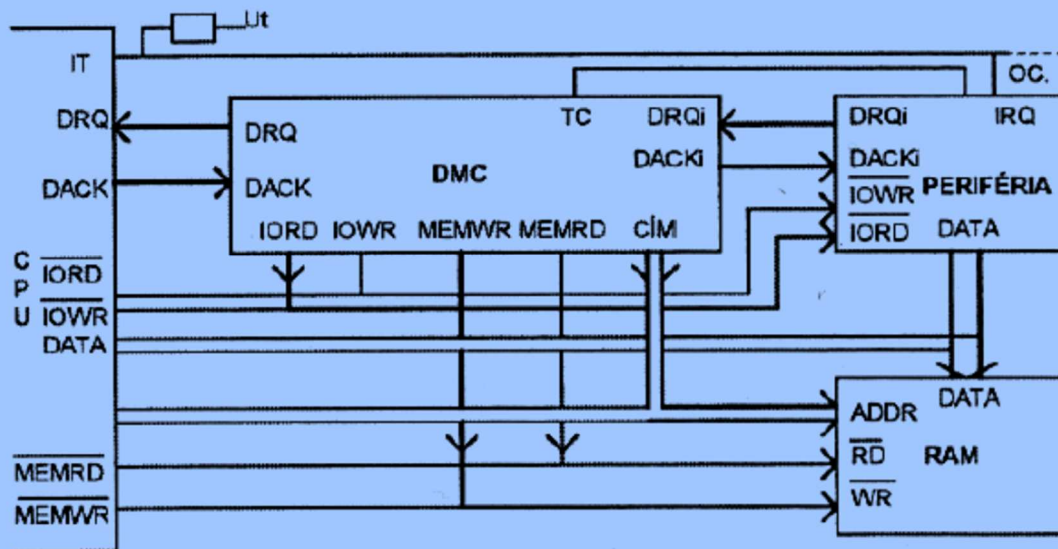
A DMA átvitel három részre bontható:

- a DMA vezérlő elkéri és megkapja a buszt (1. busz arbitrációs szakasz)
- elvégzi az átvitelt (adatátviteli szakasz, cím és vezérlőjelek kiadása)
- visszaadja a buszt (2. busz arbitrációs szakasz)

Az átvitel lefolyása részletesen:

- a periféria jelzi a DMC-nek, az átvitel igényét, a DRQ_i vonalon (egy vezérlő több DMA csatornával is rendelkezik, s mindegyik csatornához van egy DRQ_i-DACK_i jelpár),

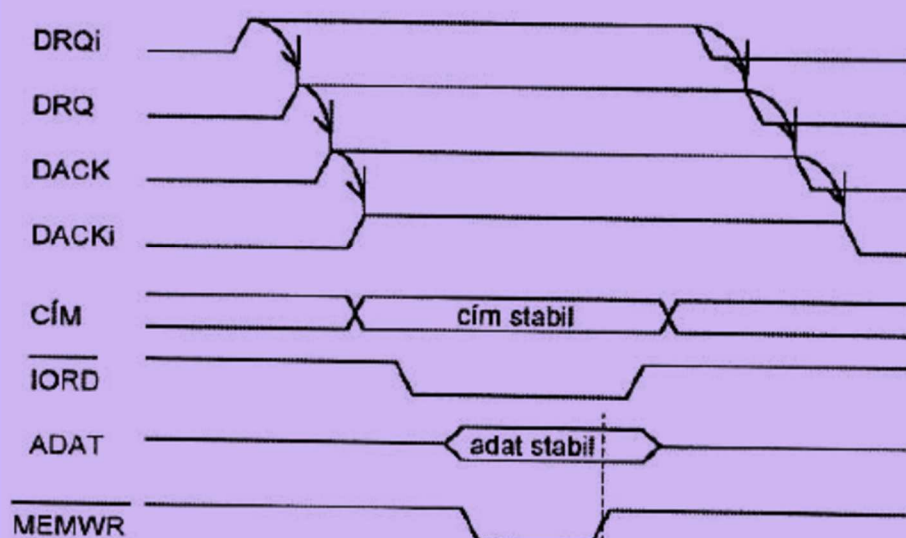
- a DMC a DRQ jel aktivizálásával elkéri a CPU-tól a buszt,



4.52. ábra. Egyciklusú DMA blokkvázlata

- a CPU befejezi az aktuális buszciklust, lekapcsolódik a buszról s ezt visszajelzi a DACK kimenetén,
- a DMC visszajelzi a DACKi vonalon a perifériának, hogy megkezdődhet az átvitel, s közben kiadja az aktuális memória címet, továbbá periféria → memória átvitel esetén aktivizálja az IORD, memória → periféria átvitel esetén pedig a MEMRD jelet,
- periféria → memória átvitel esetén a DACKi*IORD hatására a periféria az adatbuszra kapuzza az adatot, s annak stabilizálódása után a DMC a MEMWR jellel beírja azt a memóriába,
- memória → periféria átvitel esetén a megcímzett memóriarekesz tartalma a MEMRD jel hatására az adatbuszra kerül, melynek stabilizálódása után az DMC az IOWR jellel beírja azt a periféria regiszterébe,
- ha a teljes blokk átvitele megtörtént, azt többnyire egy interrupttal jelzi a DMC a processzornak (ezt a DMC blokkvéget jelző TC kimenete és a DACKi együttesen válthatja ki), és visszaveszi a DRQ jelet, (üzemmódtól függően a DRQ a teljes blokk átvitele előtt többször megszűnhet ill. aktivizálódhat),
- a processzor a DRQ megszűnését érzékelve visszaveszi a DACK jelet és folytatja működését a következő gépi ciklussal.
- a DMA folyamat a teljes blokk átvitele előtt megszakítható, TC vonal aktivizálásával, ugyanis ez kétirányú (belül nyitott kollektorosan meghajtott és figyelt jel)

A 4.53. ábra egy periféria → memória irányú egyciklusú DMA átvitel vázlatos idődiagramját mutatja.



4.53. ábra. Egyciklusú DMA idődiagramja (periféria → memória)

A DMA vezérlők többféle átviteli üzemmódra képesek. Az átviteli módok a következők lehetnek:

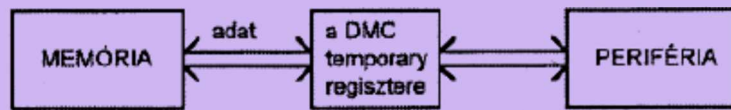
- *egyenkénti* (single), amikor minden egyes szó átviteléhez külön DMA kérés szükséges, folyamatos DMA kérés esetén minden DMA ciklust egy processzor ciklus követ,
- *blokkos*, amikor egyetlen DMA kérés hatására az egész blokk átvitele megtörténik,
- *demand*, amikor a DMA átvitel addig tart, amíg a kérés fennáll, egy teljes blokk átvitele esetleg csak több kérés hatására történik meg.

Az egyciklusú DMA esetén megfigyelhető, hogy a buszon egyszerre van memória cím, miközben a IORD vagy IOWR aktív. A perifériák a címbuszból csak kevesebb számú címbitot figyelnek, mint a memóriák (8-10-et). Ha egy közös (DMA-t nem használó) periféria címe megegyezik a címbuszon levő memória címmel a periféria által figyelt címbitekben, akkor ez a periféria azt hiszi, hogy megcímezték, és hibásan adat íródik bele, vagy meghajtja a buszt, a DMA átvitel irányától függően. Ennek elkerülésére a nem DMA-s perifériák címdekódját le kell tiltani az átvitel alatt (pl. a DACK jellel, ezt a célt szolgálja az IBM PC ISA busz AEN jele is).

Kétciklusú DMA átvitel

Ebben az esetben a DMA vezérlő az első ciklusban megcímszi a forrást és egy belső ideiglenes (temporary) regiszterében tárolja az onnan kiolvasott adatot (4.54. ábra). A második ciklusban megcímszi a célt és a temporary regiszter tartalmát beírja cél tárolójába. A forrás ill. a cél lehet memória és periféria. Egy ilyen temporary regisztert

tartalmazó DMC-vel memória-memória átvitel is lehetséges (pl. gyors blokkmozgatás a memóriában). A kialakítás előnye, hogy a DMA-s perifériák kiválasztó áramkörének felépítése azonos a normál perifériákéval. Hátránya viszont, hogy kb. fele olyan gyors, mint az egyciklusú átvitel.



4.54. ábra. Adat áramlás kétciklusú DMA átvitel esetén

Feladatok:

1. Példa feltétel nélküli periféria kezelésre

Illesszünk 8085' processzorhoz 8db kapcsolót (ellenállások felhasználásával) az 1Ch perifériacímre és 8 db LED-et (8 bites latch felhasználásával) az 1Dh perifériacímre. Írjunk programrészletet, amely beolvassa a kapcsolók állását és megjeleníti a LED-eken! (A program tárolásához szükséges ROM illesztésével most nem foglalkozunk.)

2. Példa állapotolvasásos periféria kezelésre (vezérléssel kiegészítve)

Illesszünk 8085' processzorhoz 8kB ROM-ot a 0000h címre, 8kB RAM-ot az E000h címre, és egy 2I/O címet elfoglaló perifériát a 06h/07h perifériacímre úgy, hogy a 07h címet íráskor vezérlésre, olvasáskor állapotolvasásra; a 06h címet pedig adatátvitelre lehessen használni! Vezérléshez a bitkiosztás: D₀: egy bájt kivitelenek igénye, D₁: folytonos kivitelen igénye, D₂: folytonos kivitelen vége, D₃: egy bájt bevitelenek igénye, D₄: folytonos bevitelen igénye, D₅: folytonos bevitelen vége. Státuszolvasásnál a bitkiosztás: D₀: egy bájt kivihető az eszközbe, D₁: egy bájt beolvasható az eszközből. Készítsünk programot, amely előbb folytonos átvittel 100 bájtot olvas be az eszközből és eltárolja valamilyen alkalmas memóriaterületen, majd a 100 bájtot folytonos átvittel fordított sorrendben kiírja a perifériába!

3. Részben ismétlődő jellegű feladat.

Illesszünk 8085' processzorhoz 3x 2kB ROM-ot a 0000h címtől, 1kB NVRAM-ot az 1800h címtől (nem teljes címdekódolás megengedett!) és 7x 8kB RAM-ot a 2000h címtől. Az NVRAM (0 aktív) NE bemenetét egy adott perifériacímre írt D₀ adatbittel lehessen vezérelni!

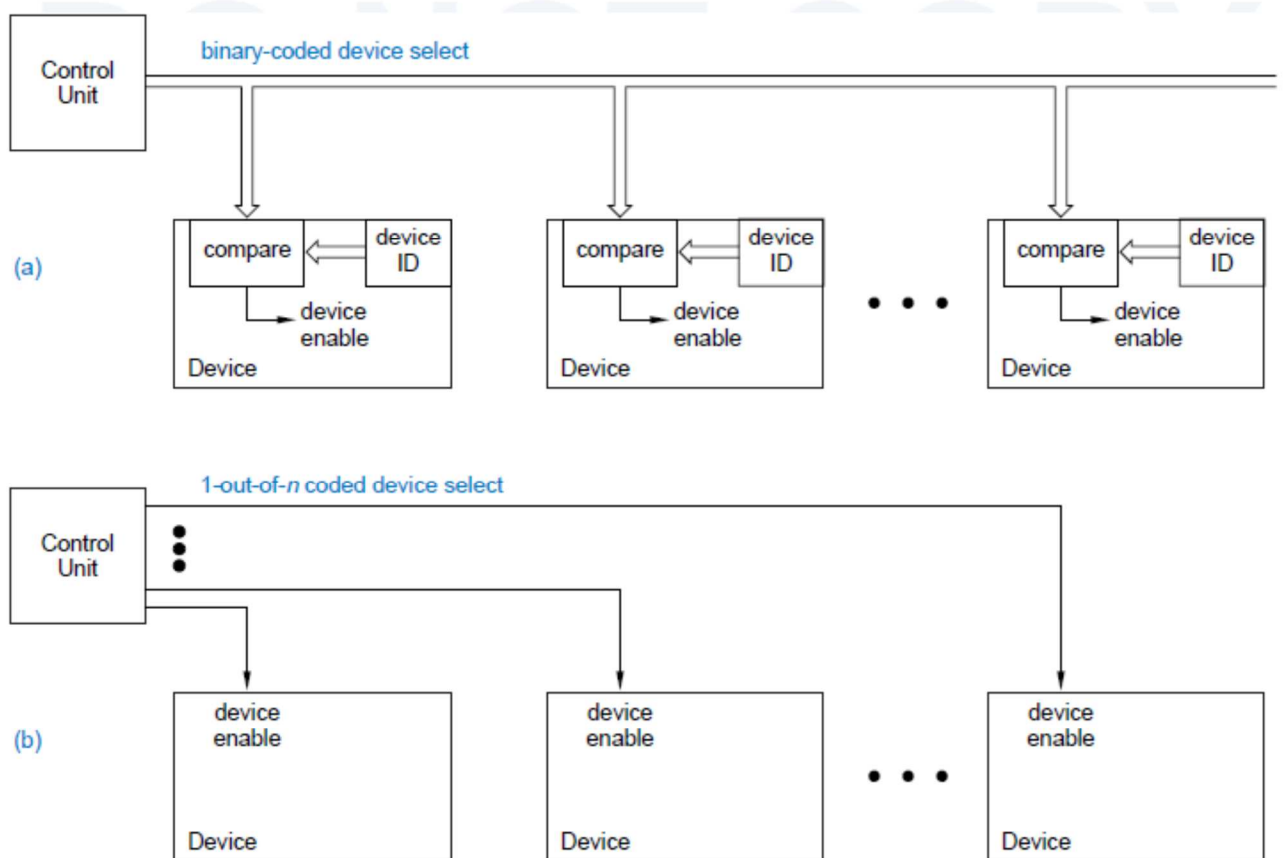


Figure 2-7 Control structure for a digital system with n devices: (a) using a binary code; (b) using a 1-out-of- n code.

A megszakítás

Az Intel 8259 IT vezérlő működése, illesztése, programozása

8259A Interrupt Controller általános megismerése

A 8259A megszakításvezérlőt csak példaképpen, korlátozottan ismertetjük. érdeklődőknek ajánljuk

A 8259A perifériaként programozható, két I/O címet foglal el. (Az A_0 címbitet kötjük bele.) Legfeljebb 8 megszakítás kiszolgálásra alkalmas (IR_{7-0}), de kaszkádósítható, így egy master és 8 slave egység használatával akár 64 megszakítást is ki lehet vele szolgálni. (Kaszkádosítással nem foglalkozunk.)

Felprogramozásához (Initial Command Word) szükséges: ICW1, ICW2, opcionálisan használható:

ICW3, ICW4 – ha hiányoznak, akkor minden bitjük 0-nak tekintett! (Figyeljünk oda A_0 értékére!)

Működési mód megváltoztatásához használható: OCW1, OCW2, OCW3.

Státuszlekérdezésnél $A_0=1$ címről a maszkregiszter értéke, $A_0=0$ címről az OCW3 utasítással kijelölt regiszter értéke olvasható ki. Ezekkel mi nem foglalkozunk.

8259A Interrupt Controller illesztése és felprogramozása

Adott 6db olyan input periféria, amelyek mindegyike egy-egy I/O portot foglal el, és úgy működik, hogy az out utasítással beleírt számú (0-255) bájtot lehet beolvasni róla egyenként. Ha egy bájt készen áll, ezt az IREQ lábán 0→1 átmenettel jelzi. Ezek mindegyikéről 250 bájtot kell beolvasni és egy közös pufferben eltárolni. A rendszerhez más perifériát nem kell illesztenünk. Tervezzük meg a hardvert és írjuk meg a programot!

Hardver tervezésének alapötletei:

A perifériák száma miatt azok megszakítás kimenetei nem köthetők közvetlenül a 8085 processzorra, hanem 8259A megszakításvezérlőt használunk.

Kihasználjuk, hogy a rendszerhez más perifériát nem kell illesztenünk, így nem teljes dekódolást használva az A_{2-0} címbitekre kötött 3/8 dekóderrel ki tudjuk választani a 6 perifériát (legyen 0-5) és a dekóder két felső (6, 7) kimenetére kötött ÉS kapuval a 8259A-t (amibe természetesen belekötjük az A_0 címbitet). A dekóder 1 aktív engedélyező bemenetére a 8085 IO/M jelét kötjük. A perifériák megszakítás kéréseit a 8259A IR_0-IR_5 bemenetére kötjük, IR_6 és IR_7 bemenetekre L jelszintet kötünk. A processzorhoz illesztünk még 8kB ROM-ot a 0000h címre és 8kB RAM-ot a 2000h címre.

Szoftver tervezésének alapötletei:

Ne írjunk 6 teljes IT rutint, hanem csak egyet, és a rutin meghívása előtt végezzük el azt, ami eltér (más portszámról kell olvasni). Ezt úgy tudjuk megtenni, ha a megszakítási szubrutinok kezdőcímei 8-cal térnek el egymástól! A puffer mutatóját tároljuk egy adott memóriaterületen (2 bájt). Ezt a mutatót kezdetben a puffer elejére kell állítanunk.

A következőkben közlünk egy **IT illesztés-minta programot**, de ezt csak az önálló megoldás után ajánlott elolvasni!

DMA

Az Intel 8237 DMA vezérlő működése, illesztése, programozása

8237 DMA Controller általános megismerése

A 8237 DMA vezérlőt csak példaképpen, korlátozottan ismertetjük. Referenciának elegendő:

- A 8237 DMA vezérlő perifériaként programozható, 16 I/O címet foglal el. (Az A_{3-0} címbiteket kötjük bele.)
- Legfeljebb 4 DMA-s periféria kiszolgálásra alkalmas ($DREQ_{3-0}$), de kaskádosítható (csak azzal mi nem foglalkozunk).

Illesztése és felprogramozása az alábbiakban bemutatott módon történik.

8237 DMAC illesztése és felprogramozása

Mintafeladat:

Adott egy olyan input periféria, amely két I/O portot foglal el (A_0 címbitet kell belekötni). $A_0=1$ esetén a parancs regiszterét érjük el, ide két egymást követő out utasítással (alsó bájtt, felső bájtt) kell beleírni, hogy mennyi adatot kérünk tőle. Ha az adatok készen állnak, ezt a REQ lábán $0 \rightarrow 1$ átmenettel jelzi. Ezután a megadott számú adat blokkosan (egymás után várakozás nélkül) kiolvasható az adat regiszteréből ($A_0=0$). A beolvasásra használjunk 8237 DMA vezérlőt! Illesszünk még az MCS-85 rendszerbe 8kB EPROM-ot a 0000h címre és 8kB RAM-ot az E000h címre! Tervezzük meg a hardvert és írjuk meg a programot! (Kérjük a perifériától 500 bájtt adatot és tároljuk az E000h címtől!)

A hardver tervezése:

Hardver tervezésének kritikus pontjai:

- Mivel a 8237 az A_{15-8} címbiteket a DB_{7-0} lábain adja ki, egy latch-et kell használnunk, amibe az adatbeírást (STB) a 8237 ADSTB jelével engedélyezzük, a kimenetét pedig a 8237 AEN jelével engedélyezzük!
- Nagyon fontos, hogy a perifériák címdekódereit a 8237 AEN jelével tiltani kell, nehogy DMA átvitel közben megjelenő cím esetlegesen egyező bitmintája miatt valamelyik perifériaengedélyezésre kerüljön!
- A 0-s csatorna használata esetén a periféria REQ kimenetét a 8237 $DREQ_0$ bemenetére kötjük, a 8237 0 aktív $DACK_0$ kimenetével pedig egyrészt egy (0 aktív logikában VAGY kapuként működő) ÉS kapun keresztül engedélyezzük a periféria kiválasztását, másrészt egy másik ÉS kapu segítségével gondoskodunk perifériába kötött legalsó címbit 0 értékéről is!
- A többi értelemszerűen megy

A szoftver tervezése:

Az ábra alatt található instrukciók alapján készítsük el periféria inicializálását és a DMA vezérlő felprogramozását végző programot! Az átvitel végét a főprogramból állapotolvasással figyeljük!

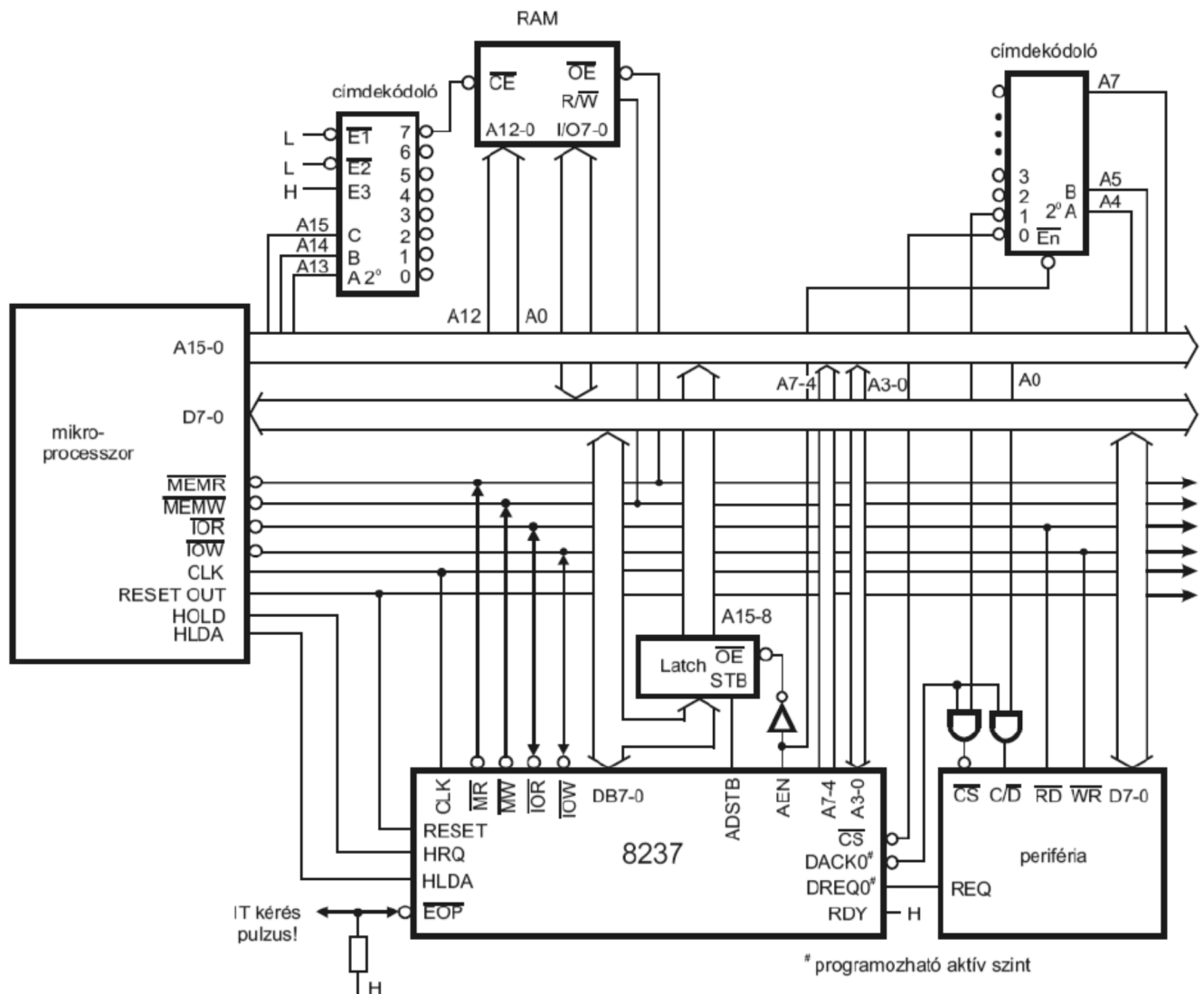
A 8237 felprogramozásának kritikus pontjai:

- A felprogramozást egy Master Clear (általános törlés) paranccsal kezdjük, ami a 0Dh címre való írással váltható ki. (A kivitt érték érdektelen.)
- A felprogramozás idejére az összes csatornát tiltjuk a 0Fh címre írt 1111b segítségével.
- A parancs regiszterbe (Command Register a 08h címen) 00000000b értéket írunk:
- DACK(L), DREQ(H), normál írás, fix prioritás, normál időzítés, DMA engedélyezése, x=közömbös, memória-memória átvitel tiltva.
- Ezután felprogramozzuk a 0-s csatornát (CH 0) a következők szerint:
- A mód regiszterbe a 0Bh címre 10000100b értéket írunk: blokkos átvitel, címnövelés, nincs AI, memória írás, 0-ás csatorna
- A 0-s csatorna címregisztere a 00h címen található, ide beírjuk az E000h memória kezdőcímet alsó bájt, felső bájt sorrendben (inicializáláskor a bájt mutató 0-ra állt be, ezért nem kellett nekünk külön törölni).
- A 0-s csatorna számláló regisztere a 01h címen található, ide beírunk a kért 500-nál 1-el kisebb (!) számot alsó bájt, felső bájt sorrendben. (A 8237 DMAC-t úgy tervezték, hogy nem 0-nál, hanem FFFFh-nál (-1) van TC (Terminal Count)!)
- Egyedileg engedélyezzük a 0-s csatornát a csatornánkénti maszkállítás módszerével: a 0ah címre 000b írásával: engedélyezés, 0-s csatorna.

A 8237 felprogramozása után utasítjuk a perifériát, hogy készítse el számunkra az adatokat és ha elkészült, indítsa el az átvitelt. Ezután folytonosan olvassuk a 8237 státusz regiszterét a 08h címen mindaddig, amíg a beolvasott érték 01h-val maszkolva (ANI 01h) 0-t ad eredményül. (JZ vissza az állapotolvasásra.) A bitenkénti logikai ÉS művelet 0-tól eltérő eredménye jelzi, hogy a kért számú adat átvitele megtörtént.

Gyakorló feladatok:

1. *A mintafeladat hardverét programozzuk fel úgy, hogy a DMA vezérlő az EPROM tartalmát másolja át a RAM-ba! (8237 felprogramozása, majd szoftver DMA kérés kiadása)*
2. *A mintafeladatot egészítsük ki azzal, hogy állapotolvasás helyett a 8237 0 aktív EOP kimenetén kiadott impulzus végét (felfutó él) egy 8259A ITC érzékelje, és jelezze az átvitel végét a 8085 processzornak. Programozzuk is fel a megszakításvezérlőt!*



mikroprocesszor

DMA vezérlő

periféria

- periféria inicializálása
- DMA vezérlő inicializálása (működési mód, móriacím, byte szám, stb.)
- periféria elindítása

- a DMA átviteltől független program
- a program végrehajtása a HOLD ciklusok idejére felfüggesztődik

← HRQ=1
HLDA=1 →

← REQ=1
DACK=0 →

memóriacím, $\overline{\text{MEMR}}$ és $\overline{\text{IOW}}$ vagy $\overline{\text{MEMW}}$ és $\overline{\text{IOR}}$

Egyszeres átvitel esetén egy kérésre egy byte átvitele történik meg, a következő byte átviteléhez újabb kérés (DREQ) kell. Blokk átvitel esetén egy kérésre a blokk hosszának megfelelő számú írási ill. olvasási ciklus történik.

Az utolsó ciklus alatt az állapotszóban beállítódik a TC bit és egy $\overline{\text{EOP}}$ pulzus keletkezik.

← HRQ=0
HLDA=0 →

DACK=1 →

az adatátvitel végének lekérdezése:

- állapotolvasással (állapotszó TC)

vagy

- IT kezeléssel ($\overline{\text{EOP}}$ pulzus!)

Periféria illesztése a 8085' mikroprocesszorhoz

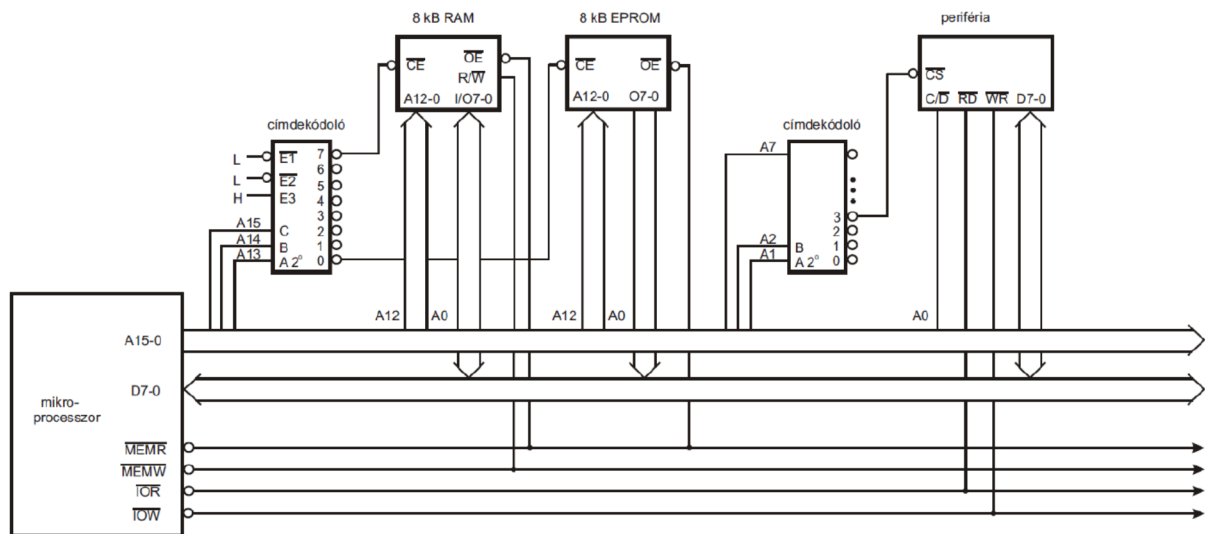
Periféria címzés dekóderrel

Példa periféria illesztésére

(Az 5. számú óravázlathoz)

Feladat:

Illesszünk 8085' processzorhoz 8kB ROM-ot a 0000h címre, 8kB RAM-ot az E000h címre, és egy 2 I/O cím elfoglaló perifériát a 06h/07h perifériacímre úgy, hogy a 07h címet íráskor vezérlésre, olvasáskor állapotolvasásra; a 06h címet pedig adatátvitelre lehessen használni!



Mésszünk 8085[®] processzorhoz 3x 2kB ROM-ot a 0000h címűl, 1kB NVRAM-ot az 1800h címűl (nem teljes címdekódolás megengedett!) és 7x 8kB RAM-ot a 2000h címűl. Az NVRAM (0 aktív) NE bemenetét egy adott perifériacímre írt D₀ adatbittel lehessen vezérelni!

