

TERVEZÉSI SEGÉDLET

Z80-as típusú mikroprocesszorhoz

*/Belső használatra/*

G A M F

1984.

## B e v e z e t é s

A Számítógépek perifériális berendezései című tárgy keretében a Számítógép technikai ágazatos hallgatóknak periféria vezérlőegységet kell tervezni, amelyhez a szükséges ismereteket az alábbi segédlet kívánja összefoglalni.

A periféria vezérlőegység, amit Control Unit-nak /CU/ is neveznek a számítógép buszrendszerére kapcsolódva lehetővé teszi az adatátvitelt a periféria és a számítógép között.

A feladatban a CU-t egy Z80-as alapú mikroszámítógép buszra kell illeszteni. A mikroprocesszoros hardware rendszer elemei a központi egység /CPU/, a memóriák /MEM/, valamint a CU-k egységes buszrendszeren keresztül kapcsolódnak egymáshoz. A buszra kapcsolódó CPU és CU-k között háromféle adatátvitel lehetséges:

- a./ programozott
- b./ megszakítást használó
- c./ közvetlen memória hozzáférés, DMA

a./ Programozott adatátvitelről beszélünk, ha minden egyes adat átvitelét, illetve minden periféria működését a CPU vezérli. Ez az átviteli mód lassítja a CPU működését, mivel a legtöbb processzor időt igényli, viszont a CU áramköri felépítése egyszerű lesz.

b./ Megszakítást használó adatátvitelnél a CU-k különböző események hatására /pl. egy adatbyte átvitel/ program-megszakítást kérhetnek. Ekkor a CPU a futó program végrehajtását felfüggeszti, kiszolgálja a megszakítást kérő perifériát /pl. megtörténik az adatátvitel/, majd ezután visszatér a félbeszakított programra. Mivel a CPU szinte kizárólag csak a megszakítási ok fennállásáig foglalkozik a perifériával, ez az átviteli mód jö-

val kevesebb processzor időt vesz el, mint az előbbi.

c./ A DMA átvitel a leggyorsabb, mivel a programvezérlés kihagyásával teszi lehetővé valamely periféria és általában a MEM közti közvetlen adatátvitelt, ami a CPU igénybevétele nélkül zajlik le. Ez a módszer főleg ott használatos, ahol nagymennyiségű információt visznek át általában blokk formában. Az említett előnyökért az ár, a bonyolultabb hardware felépítés.

### 1.1. Mikroszámítógép buszrendszere

A busz univerzális kommunikációs eszköz a rendszer elemei között, amelyen keresztül bonyolódik le az egységek közötti adat- és parancsátvitel. Rendszertechnikai szempontból a buszra kapcsolódó egységek funkciói:

- Rendszer vezérlő: az alaphelyzetbe állító /RESET/ valamint az órajelet /CLOCK/ adja a buszra. A rendszer csak egy ilyen egységet tartalmazhat.
- Busz-master: vezérlőjelet és címet ad a buszra, több ilyen is lehet a rendszerben, de egyidejűleg nem működhetnek.
- Busz-slave: vezérlőjeleket és a címet figyeli, saját címének felismerése után elvégzi a vezérlőjelek által előírt működést. A busz-slave lehet MEM és CU. A buszon általában több slave van, amelyeknek címükben különbözniük kell.
- Megszakítást kérő: a kiszolgálási kívánságát a megszakítást kérő /IRQ/ vonalra adja.
- Megszakítást figyelő: az IRQ vonalon érkező kívánságokat dolgozza fel. A buszon általában csak egy van belőle.

Az egyes funkciók nem feltétlen jelentenek külön kártyát is. Például az egyes, kettős, ötös funkciókat általában a CPU kártya valósítja meg a legtöbb kis rendszerben.

A busz a következő vonalakat tartalmazza, amelyek általában a 280  $\mu$ P jeleinek felelnek meg:

- cimbusz: a memóriák és perifériák címzésére, kiválasztására szolgál, 16 vonalat tartalmaz
- adatbusz: kétirányú adatátvitelt tesz lehetővé, 8 vonalat tartalmaz
- vezérlőbusz: a különféle vezérlőjelek továbbítására szolgál, 14 vonalat tartalmaz
- tápfeszültség vonalak: +5V és a GND számára.

A busz kártyacsatlakozók összekapcsolásával lett kialakítva. A buszra csatlakozó CU-k az egyes vonalakra maximum 1 TTL LS terhelést jelenthetnek, a CU adópontjának meghajtóképessége minimum 40 TTL LS legyen.

A csatlakozókiosztás az 1. táblázatban látható. A felülhúzott jeleknél az alacsony szint felel meg a logikai 1 szintnek.

Az egyes vonalak funkciói a következők:

AO - A15  
/address Bus/

16 bites cimbusz, háromállapotú kimeneti jelek. Memoria olvasás vagy írás esetén a memoria cím, perifériális művelet esetén az alsó 8 biten a periféria cím, memoria frissítésnél az alsó 7 biten a frissítési cím jelenik meg rajta. AO a legkisebb helyiértékű bit.

Csatlakozó	Jel	Csatlakozó	Jel
1-4	GND	25	A5
5	$\overline{M1}$	26	A6
6	$\overline{RFSH}$	27	A7
7	$\overline{WAIT}$	28	A8
8	$\overline{BUSRQ}$	29	A9
9	$\overline{BUSA1}$	30	A10
10	D0	31	A11
11	D1	32	A12
12	D2	33	A13
13	D3	34	A14
14	D4	35	A15
15	D5	36	$\overline{RD}$
16	D6	37	$\overline{WR}$
17	D7	38	$\overline{TORQ}$
18	-	39	$\overline{RESET}$
19	-	40	$\overline{MREQ}$
20	A0	41	$\overline{NMI}$
21	A1	42	$\overline{INT}$
22	A2	43	$\overline{HALT}$
23	A3	44	$\emptyset$
24	A4	45-47	VCC

1. táblázat

$D_0 - D_7$   
/Data Bus/

$\overline{M1}$   
/Machine Cycle One/

$\overline{MREQ}$   
/Memory Request/

$\overline{IORQ}$   
/Input/output Request/

$\overline{RD}$   
/Memory Read/

$\overline{INT}$   
/Interrupt Request/

8 bites háromállapotú kétirányú busz, amelyen a processzor és a külső eszközök közti adatforgalom zajlik.

Az utasítás lehívás /FETCH/ gépi ciklus jelzésére szolgál. A megszakítás elfogadás gépi ciklusban is aktív.

Háromállapotú kimenet azt jelzi, hogy a címbuszon érvényes memória cím van.

Háromállapotú kimenet, amely jelzi, hogy a címbusz alsó felén érvényes I/O cím van. A megszakítást elfogadó ciklusban  $\overline{M1}$ -gyel együtt aktivizálódik.

Háromállapotú kimenet, alacsony szintjével jelzi, ha a processzor kész adat fogadására a periféria vagy memória felől.

Maszkolható megszakítás kérés bemenővonala. Az  $\overline{INT}$  jelet a CU-k generálják. A processzor az utasítás ciklus végén megvizsgálja a vonal állapotát és ha a programmal állítható belső megszakítást engedélyező flip-flop /IFF/ engedélyezett, valamint a  $\overline{BUSRQ}$  jel nem aktív, elfogadja a megszakítás kérést. A kérés elfogadását

a processzor nyugtázó jellel jel-  
zi  $\overline{IORQ}$  és  $\overline{MI}$  egyszerre aktiv/  
a következő gépi ciklus elején.

INT  
/Non Maskable Interrupt/

Negatív élre hatásos bemenet. A  
nem maszkolható megszakítás kérés  
vonal magasabb prioritású mint az  
 $\overline{INT}$ . Hatására a processzor az IFF  
flip-flop állapotától függetlenül  
RESTART utasítást hajt végre a  
0066H címre.

A programszámláló tartalma automa-  
tikusan a stackba mentődik, így a  
megszakított programba vissza le-  
het térni.

RESET

A vonalon megjelenő alacsony szint  
hatására a processzor alaphelyzet-  
be kerül, azaz:

- 1./ a megszakítást engedélyező  
flip-flop letiltja a megszaki-  
tás kérés elfogadását.
- 2./ Az I regiszter COH
- 3./ Az R regiszter COH
- 4./ A 0-ás megszakítás üzemmódot  
állítja be.

A reset ideje alatt a cím- és ve-  
zérlőbusz lebegő, a vezérlő jelek  
pedig inaktív állapotba kerülnek.  
Memoria frissítő jelet nem ad ki  
a processzor.

BUSRQ  
/Bus Request/

Aktiv állapotának hatására a gépi  
ciklus végén az adat cím- és vezér-  
lőjelek lebegő állapotba kerülnek,

lehetővé téve, hogy a buszt más eszköz vezérelje.

/DMA controller/

BUSAK

/Bus Acknowledge/

Kimeneti jel, amelynek aktiv szintje jelzi, hogy a háromállapotú kimenő jelek lebegő állapotba kerültek, tehát más eszköz átveheti a busz vezérlését.

∅

Egyfázisú rendszer órajel.

WR

/Memory Write/

Háromállapotú kimeneti jel azt jelzi, hogy az adatbusz érvényes kimeneti adatot tartalmaz.

RFSH

/Refresh/

Aktiv állapota azt jelzi, hogy a cibusz alsó 7 bitje frissítési címet tartalmaz és az MREQ jel frissítő olvasásra használható.

HALT

/Halt state/

Kimeneti jel, aktiv állapota azt mutatja, hogy a processzor végrehajtott egy HALT utasítást és megszakításra vár. HALT alatt a processzor NOP utasításokat hajt végre és memoria frissítést végez.

WAIT

/Wait/

Bemeneti jel, amelyen a megcímzett MEM vagy CU jelzi, ha nem kész az adatátvitelre.

## 1.2. Busjelek funkcionális idődiagramjai

A Z80 CPU utasításciklusai a következő gépi ciklusokból épülnek fel:



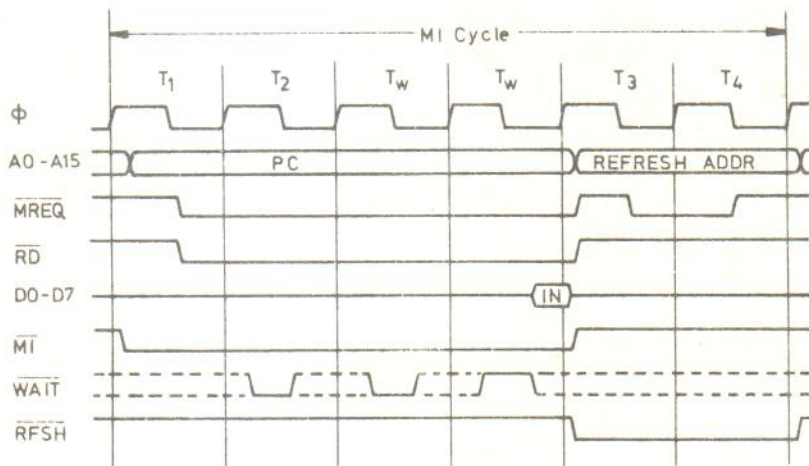
- Memoria írás/olvasás
- Input/output írás vagy olvasás
- Megszakítás elfogadás

A gépi ciklusok 3-6 órajel-periódussig tartanak, amik WAIT állapotok beiktatásával tetszőleges számú periódussal megnyújthatók. Így oldható meg a lassú perifériák ill. memoriák illesztése.

Az alap órajel-periódust ütemnek /T/ nevezzük. Minden utasításciklus első gépi ciklusa /A.1/ az utasítás lehvívási gépi ciklus /FETCH/, mely 4-6 ütem hosszú, és a következő végrehajtandó utasítás műveleti kódjának elérésére szolgál. A következő gépi ciklusokban általában adatmozgatás történik. Az adatmozgatásban résztvevő elemek a regiszterek, a memoria és a külső I/O eszközök.

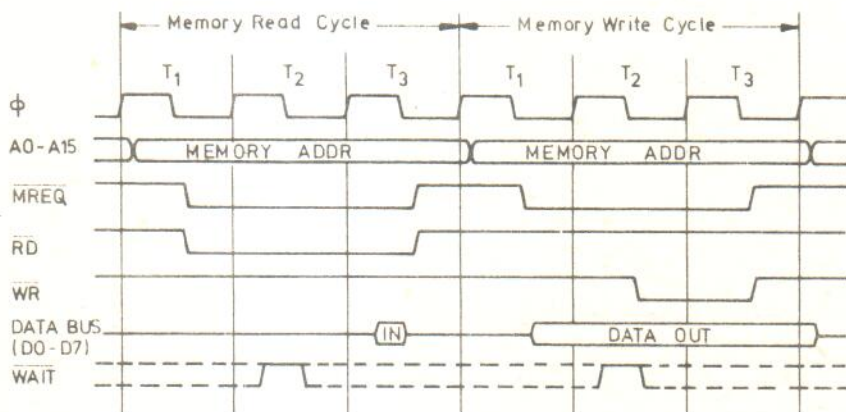
#### 1.2.1. utasítás elérési ciklus

A ciklus elején a PC tartalma a cibuszra kerül, fél órajellel később az  $\overline{MEMQ}$  jel aktív lesz, lefutó éle környezetében a cibuszon már stabil címek találhatóak. Az  $\overline{RD}$  jel is itt lesz aktív jelezve, hogy a memoriából kiolvasott byte a busz adatvonalaira adható. A CPU az adatvonalakat  $T_2$  ütem felfutó órajel élével mintavételezi, majd megszünteti az  $\overline{MEMQ}$  és  $\overline{RD}$  vezérlő jeleket. A  $T_3$  és  $T_4$  ütemek a dinamikus memoriák frissítésére használhatók fel. Ekkor a cibusz A0-A7 bitjei a memoria frissítő címet tartalmazzák, a  $\overline{RFSH}$  aktív szintje pedig jelzi, hogy a dinamikus memoriákon frissítő olvasás hajtható végre.  $T_3$  és  $T_4$  alatt  $\overline{RD}$  jel inaktív, ami kiküszöböli, hogy a frissítés során adatok kerüljenek az adatbuszra. Frissítés során a címek stabil fennállása csak  $\overline{MEMQ}$  jel aktív állapota alatt garantált. A CPU  $T_2$  és minden további  $T_n$  ütem az órajel lefutó élével mintavételezi a  $\overline{WAIT}$  vonalat, s ha a vonal aktív, a következő ütemben újabb WAIT állapot kerül beiktatásra. A WAIT állapotok beiktatásáról az itt leírtak a többi gépi ciklusokra is vonatkoznak.



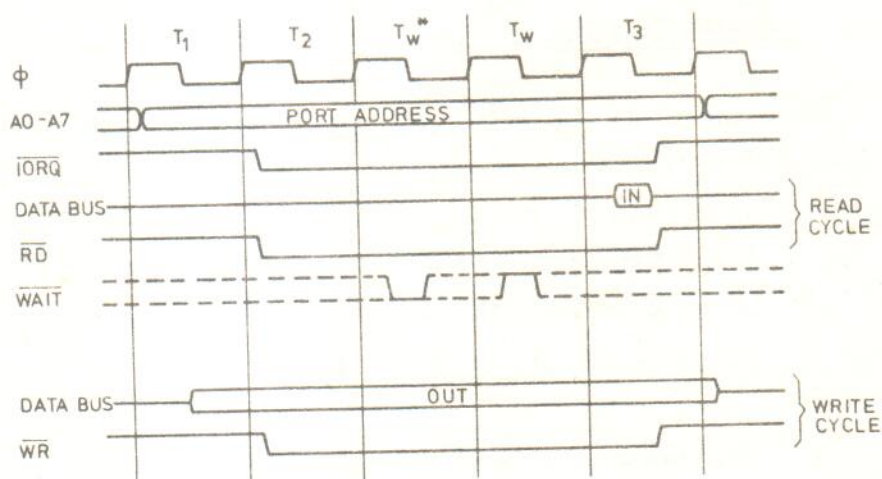
### 1.2.2. Memoria olvasás vagy írás

Olvasásnál az  $\overline{MREQ}$  és  $\overline{RD}$  jelek a FETCH ciklusban láttakhoz hasonlóan alakulnak, viszont a  $\overline{WR}$  jel akkor lesz aktiv, ha már stabil adat található az adatbuszon. A  $\overline{WR}$  jel megszűnését követően még fél órajelperiódus ideig stabil marad a cím vonalak és az adatvonalak állapota.



### 1.2.3. Input/output ciklusok

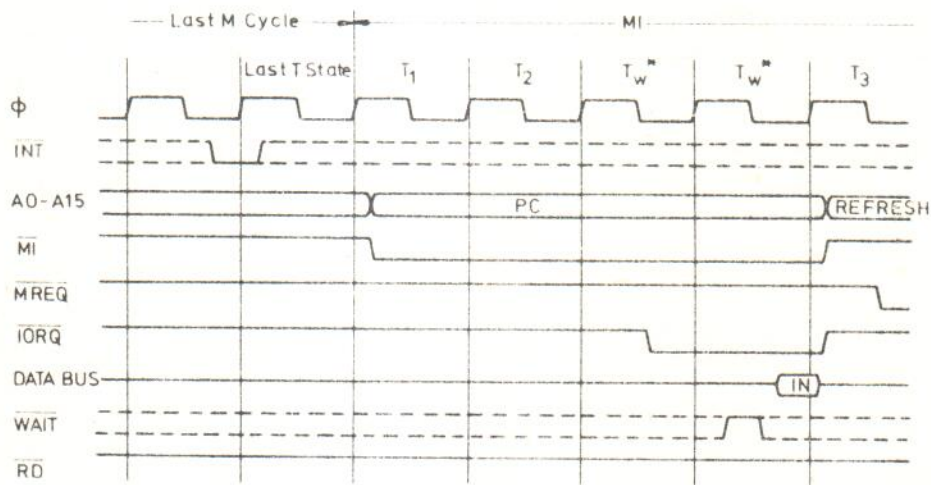
Az I/O műveletek során egy WAIT állapot automatikusan a gépi ciklusba iktatódik. Periféria olvasáskor a  $\overline{RD}$  jel a megcímzett CU adatának a processzor adatbuszára való kapuzását végzi. Periféria íráskor a  $\overline{WR}$  jel felfutó éle használható a CPU-tól érkező adat kapuzására a CU adatregiszterébe. Az ábrán a független input és output ciklusokat együtt tüntetjük fel, bár ezek egyszerre sohasem fordulhatnak elő.



### 1.2.4. Megszakítás kérés/elfogadás ciklus

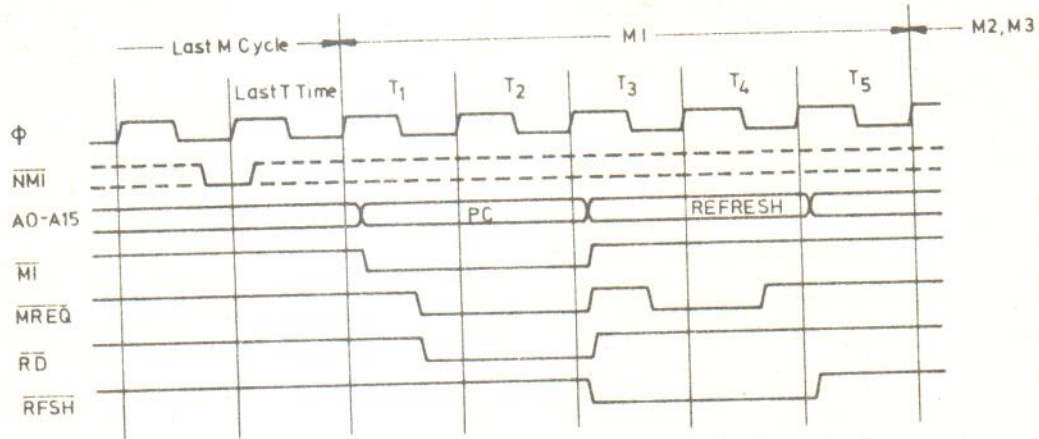
Az  $\overline{INT}$  jelet a CPU minden utasításciklus utolsó ütemének felfutó órajel élénél vizsgálja. A megszakítás kérést a CPU nem fogadja el, ha a software-vezérelt megszakítás engedélyező flip-flop nincs bebillentve, vagy ha a  $\overline{BUSRQ}$  jel aktív. Ha a CPU elfogadja a megszakítást, speciális M1 ciklust generál. Ezalatt az  $\overline{IORQ}$  jel is aktív lesz,

jelezve, hogy a megszakítást kérő eszköz egy 8 bites vektort adhat az adatbuszra. A megszakítási ciklus automatikusan meg van nyújtva hét WAIT ütemmel, szükség esetén további WAIT ütemek itt is beiktathatók.



#### 1.2.5. Nem maszkolható megszakítás elfogadás

Az  $\overline{NMI}$  jel bebillenti a CPU belső  $\overline{MI}$  latch-ét, melynek tartalmát a processzor minden utasításciklus végén megvizsgálja. A nem maszkolható interrupt nagyobb prioritású a normál megszakítás kérésnél és software vezérléssel nem tiltatható le.  $\overline{NMI}$  elfogadás után a processzor egy normál memória olvasás ciklust hajt végre azzal a különbséggel, hogy az adatbusz tartalmát nem veszi figyelembe. A processzor a PC tartalmát automatikusan elmenti a stack-be és a 0000H címre ugrik.



## 2. Programozott, programmegszakításos perifériák illesztése

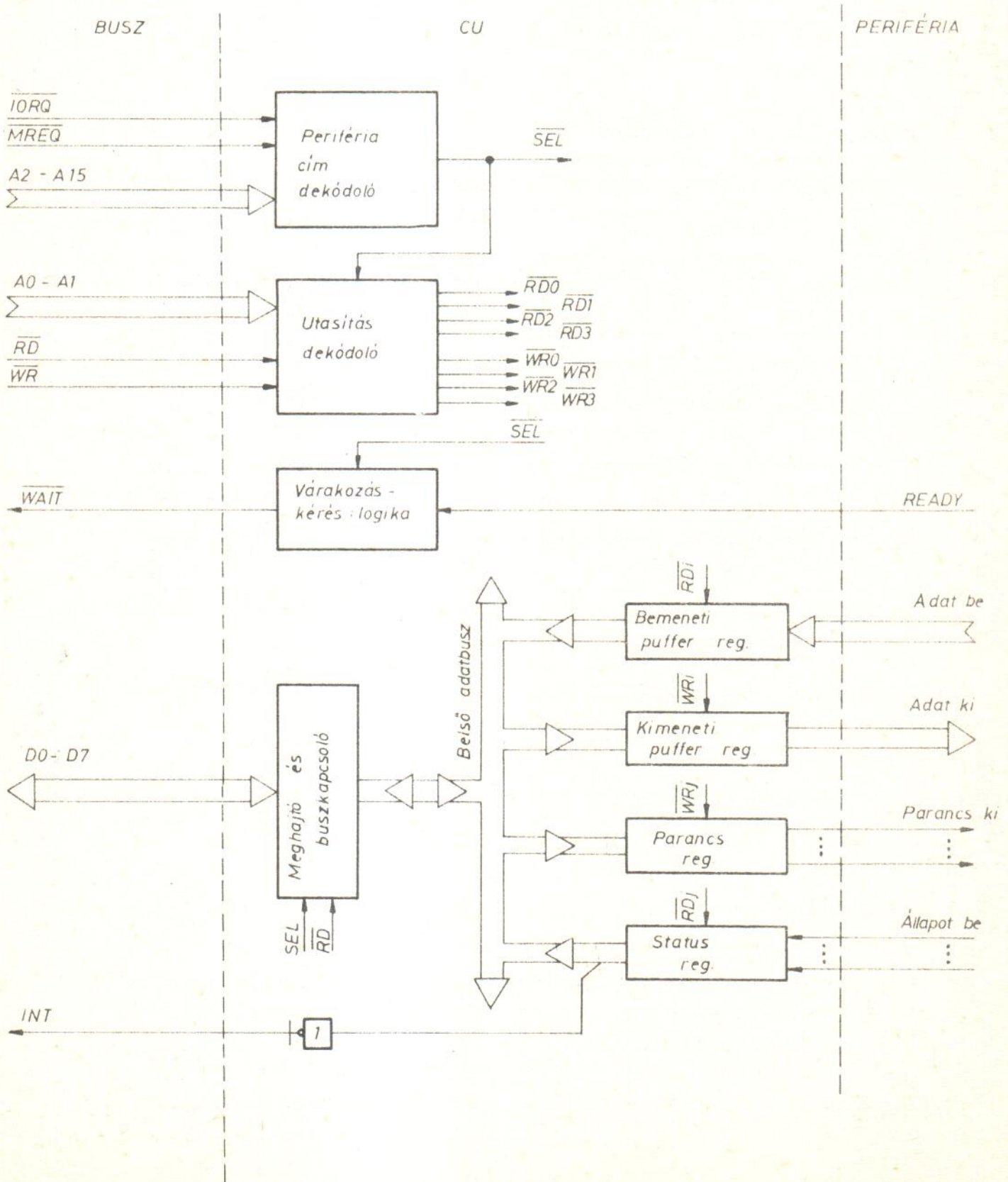
A mikroszámítógépes rendszerben a slave egységek döntő többsége programozott ill. programmegszakítást alkalmazó periféria. A buszra kapcsolódáshoz szükséges illesztő egységeknek tartalmazni kell bizonyos funkciókat megvalósító áramköri részeket. Ez alapján az illesztő egység tervezésének egyik célravezető módszere a funkcionális dekompozíció.

A CU funkcionális egységei a következők:

- periféria cím és utasításdekóder
- buszkapcsolók és meghajtók
- ki- és beviteli pufferek
- periféria flagrendszer
- megszakítás kérés logika
- várakozás kérés logika
- végjel azonosító hálózat
- szürendő karaktereket azonosító hálózat
- átkódoló hálózat
- perifériát vezérlő sorrendi hálózat

Az első hat között felsorolt részek minden csatoló egységben előfordulnak, a többi egység a CU feladatától függően van je-

len. A minimális illesztőegység blokkvázlata a 2.1. ábrán látható.

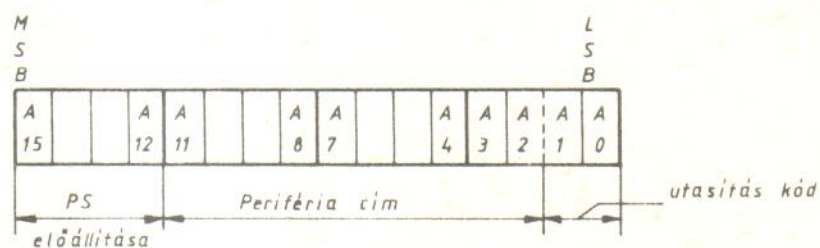


2.1. ábra

2.1. Periféria cím és utasításdekóder

A periféria csatolók azonosítása a CU-khoz rendelt címmel kétféleképpen történhet.

2.1.1. Memóriába leképezett periféria címzés esetén a CPU által címezhető memória-tartomány egy meghatározott része a csatolókhöz van rendelve. A csatolóknak fenntartott címtartományba tilos memóriát elhelyezni. A megoldás előnye, hogy nem szükségesek I/O utasítások, a teljes memóriareferenciás utasításkészlet használható a csatoló regisztereivel végzett műveleteknél. Hátrányként említhető a tényleges memóriatartomány csökkenése, ami kis rendszerekben egyáltalán nem jelent megkötést. A 8 bites szóhosszúságú mikroprocesszorok esetében általában 16 bites címmező áll rendelkezésre, amivel 64 kbyte memória érhető el. A tervezendő illesztőnél a memóriából 4 kbyte a perifériákhoz van rendelve. A periféria címzése az illesztőegységben szükséges címdekódoló egyszerűsítése érdekében a következőképpen történik.



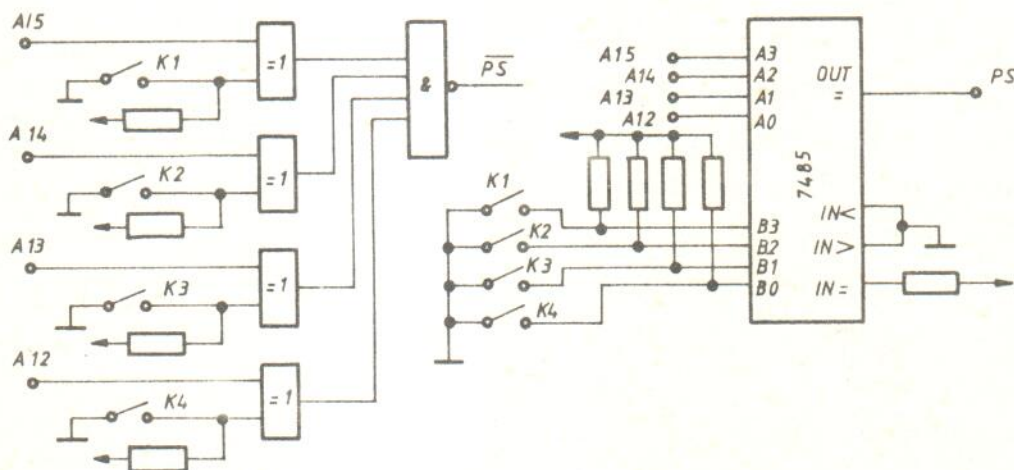
A 16 bites cím felső 4 bitje /A15-A12/ szolgál a periféria címtartomány azonosítására, a tartományon belül az A2-A11-bitek szolgálnak az egyes perifériák kiválasztására. A 10 bitből mindig csak egy vehet fel logikai egy értéket, miatt a többi kötelezően 0-ban kell tartani. Minden CU-hoz egy bit tartozik, ami azonosítja az illesztőt, s ennek figyelmével egyszerűen megoldható a dekódolás. Ezzel a mód-

szerrel tíz darab periféria különböztethető meg, maximálisan tehát ennyi lehet a rendszerben.

Ha több perifériát kívánunk elhelyezni, akkor pl. 2 bittel kijelölve egy perifériát /mindig 2 bit 1-es/

$$\binom{9}{2} = 36 \text{ egység lehet a konfigurációban.}$$

A két utolsó A0 -A1 bitet figyelembe véve egy perifériához négy tényleges cím rendelhető. Az  $\overline{RD}$ ,  $\overline{WR}$  jelek segítségével ezek olvasás ill. írás típusú címekre választhatók szét, amelyek így a CU számára 4 olvasás és 4 írás típusú parancsot jelentenek. Ezek az utasítások vagy tényleges írás, ill. olvasás irányú adatátvitel vezérlésére, különböző regiszterek feltöltésére és kiolvasására, vagy periféria funkciók indítására szolgálhatnak. Ez utóbbi esetben az utasítások adatátviteli funkcióját nem feltétlenül használjuk ki. Ha egy periféria működtetéséhez 4 írás és 4 olvasás utasítás nem elegendő, a perifériához két vagy több periféria cím is rendelhető. A fentiek alapján a címfelismerés és perifériális utasítás dekódolás egy lehetséges módja a 2.2 - 2.3. ábrán látható.



2.2. ábra

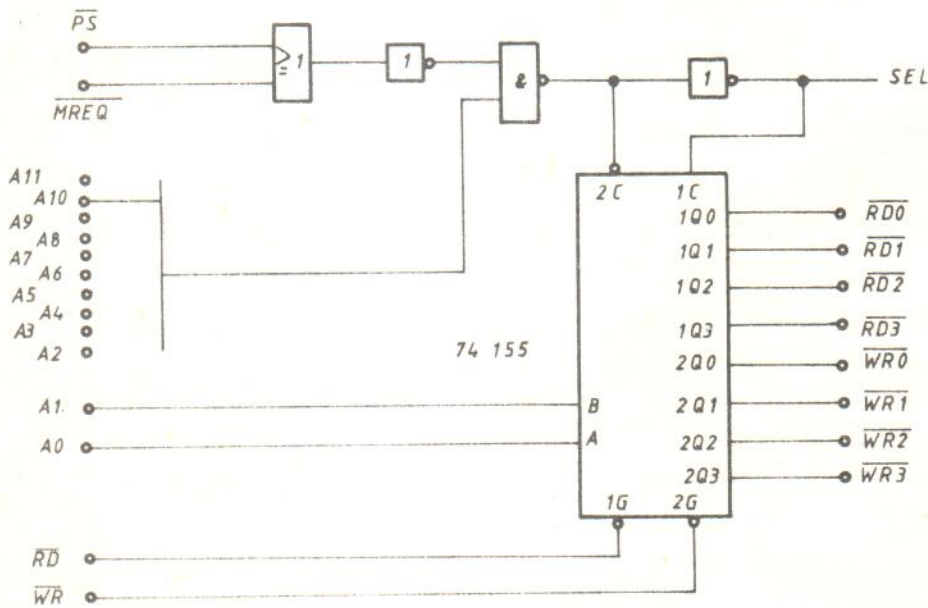
A K1-K4 kapcsolókkal az illesztő címe állítható be, zárt állásban a cimbit ponált értéke jut a NAND bemenetére.

Ha nem akarunk címváltoztatást engedélyezni, akkor a dekó-



dolás egyszerűsödhet. Kódkomparátor /7485/ használatával az az IC számot csökkenthetjük.

A PS jel hazárdos, így működtetésre nem használható, az  $\overline{MREQ}$  jellel a hazárdmentesítés megoldható.



2.3 ábra

A  $K_1$ ,  $K_4$  zárt,  $K_2$ ,  $K_3$  nyitott állásában és ha csak  $A_0=1$ , akkor a periféria címe 9400-9403-ig terjed. A dekódoló adatlapját figyelembe véve látható, hogy  $\overline{RD}=L$  esetén az  $1Q$  kimenetek közül az  $A_0$ ,  $A_1$  által megcímzett lesz aktív. Természetesen csak ha  $SEL=H$ , vagyis a CU kiválasztásakor. Ugyanez történik  $\overline{WR}=L$  esetén a  $2Q$  oldalon.

### 2.1.2. Input/output utasítást használó periféria kezelés

Ebben az esetben a címbusz  $A_0-A_7$ -ig terjedő vonalain jelenik meg a periféria cím, aminek érvényességét az  $\overline{IORQ}$  alacsony szintje jelzi. Az oktett felső 4 bitjének dekódolásával előállítható a PS jel. Ebben az esetben a rendszeren

belül különböző perifériákhoz általában különböző PS tartozik, amelynek hazárdmentesítése a  $\overline{TORQ}$  segítségével történhet. A 2.1.1. pontban leírtak alapján az ismertetett kapcsolások értelemszerű adaptálásával a dekódolás könnyen megoldható.

Mindkét esetben a dekódoló kimeneteihez  $\sqrt{RD\bar{i}}$  vagy  $\overline{WR\bar{i}}$  rendelhető parancsok a következők lehetnek:

Adatátvitel - Programozott periféria esetén a központi egység és a periféria között kivitel /CPU - PER/ vagy bevitel /PER - CPU/ irányú adatátvitel lehetséges. A kivitel általában pufferezt. A kivitt információ egy regiszterbe kerül, így az a periféria számára nemcsak az utasítás idejére, hanem folyamatosan is rendelkezésre áll. A bevitel lehet közvetlen vagy pufferezt. Közvetlen bevitelnél az utasítás pillanatában fennálló információt viszi be, míg pufferezt esetben a külső egység által megszabott időben a pufferezt regiszterben levő információ kerül beolvasásra. Az információ lehet adat vagy parancs output, illetve adat vagy állapot input esetén.

Kivitelnél az adatvonalakon /D0-D7/ megjelenő információból az írás utasítás  $\sqrt{WR\bar{i}}$  felfutó élével/ hatására vehetünk mintát. A  $\overline{WR\bar{i}}$  impulzus a megcímzett csatolóban keletkezik, gyakorlatilag a  $\overline{WR}$  jellel egyidejűleg.

Inputnál az illesztő egység adja rá az adatvonalakra a beviteli információt az olvasás utasítás  $\overline{RD\bar{i}}$  hatására. Az adatvonalakat csak az  $\overline{RD\bar{i}}$  impulzus alatt szabad meghajtani. Ha a  $\overline{WR\bar{i}}$  ill.  $\overline{RD\bar{i}}$  impulzusok idejénél többre van szüksége a perifériának, akkor WAIT állapot beiktatásával ezek az idők meghosszabbíthatók.

Vezérlő utasítások A perifériális utasítások az adatátvitelen kívül különféle vezérlési funkciót is ellátnak. A kiviteli utasítás pl. gyakran indítja a kiviteli egység működé-

sét, pl. adatkivitel a szalaglyukasztó regiszterébe egyben indítja a lyukasztást is. Az utasítások sok esetben adatátvitelt nem igénylő működést indítanak, pl. motor indítás, leállítás, floppy disk fej letétel, léptetés, stb.

#### A vezérlési funkciók kétféleképpen szervezhetők:

- Egy utasítás egyféle működést indít, s az adatvonalak tartalma közömbös.

Mind  $\overline{WR}$ , mind  $\overline{RD}$  utasítás felhasználható e célra, az előbbi használata azonban célszerűbb, mivel a másik esetben az adatvonalak tartalma bekerül a CPU akkumulátorába.

- Egy utasítás többféle működést indít. A többféle működés közül az adatvonalakon levő parancsszó választ. Ekkor vezérlésre csak a  $\overline{WR}$  utasítás használható.

#### 2.2. Buszkapcsolók és meghajtók

A CU-k kapcsolódásának rendszertechnikai és áramköri követelményeinek kielégítésére buszmeghajtókra, illetve vevőkre van szükség. Ezek szerepe biztosítani a megfelelő meghajtóképességet, illetve a megengedett terhelést.

A buszra való kapcsolódás történhet nyitott kollektoros kapuk huzalozott vagy kapcsolódásával, illetve háromállapotú buszmeghajtók segítségével. Az utóbbi alkalmazásánál biztosítani kell, hogy a buszt egyszerre csak egy adópontról lehessen meghajtani, ellenkező esetben a meghajtók károsodhatnak. Huzalozott vagy kapcsolásnál az esetleges több adópontból való meghajtás információvesztéshez vezethet, de áramköri hiba nem következik be.

Kétirányú háromállapotú meghajtók esetén biztosítani kell az illesztő belső vonalainak /belső busz/ és a buszvonalak megfelelő irányú és időzítésű összekapcsolását.

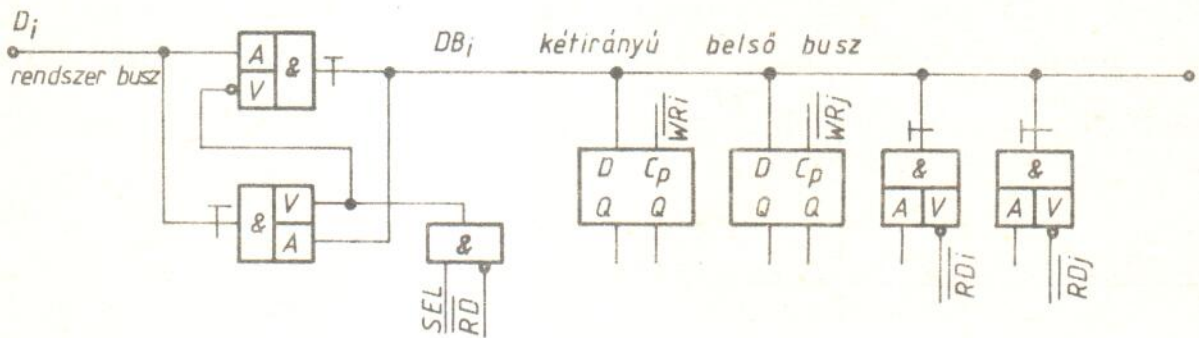
A cím- és vezérlőbuszt a master hajtja meg, az illesztőegységben vevők fogadják. Az adatbusz mint láttuk kétirányú, de a CU belső adatvonalai lehetnek két, de egyirányúak is.

- Kétirányú belső busz esetén a buszkapcsolókat az illesztő kiválasztásakor kell csak működtetni, mégpedig íráshoz a belső busz, olvasáshoz a rendszerbusz felé kell vezetni.

$ADATBE = SEL \cdot RD$  - A rendszerbusz felé vezetés feltétele.

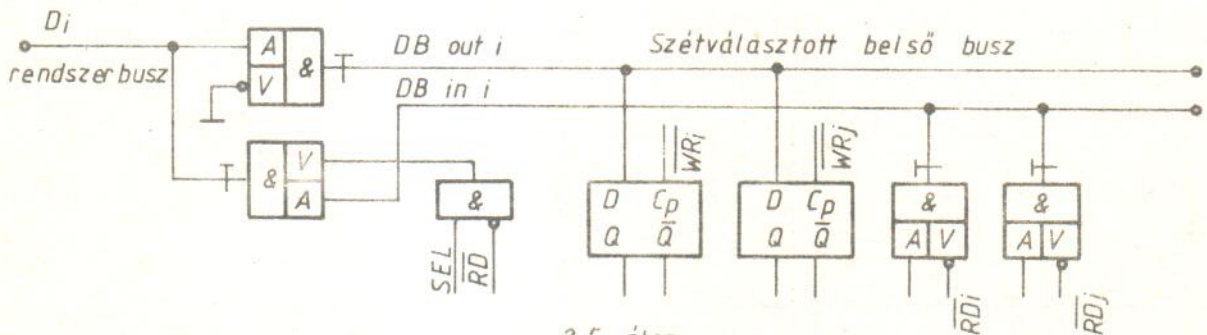
$ADATKI = SEL \cdot WR$  - A belső busz felé vezetés feltétele, vagy

$ADATKI = \overline{ADATBE}$



2.4. ábra

- Szétválasztott belső busz esetén a DB out állandóan meghajtható.



2.5. ábra

A belső buszra való csatlakozásra értelemszerűen az vonatkozik, ami a 2.2. pont elején lett ismertetve.

### 2.3. Periféria flag rendszere, periféria és CPU szinkronizálása

A periféria és CPU közötti kommunikáció biztosításához szükség van arra, hogy a CPU tudomást szerezzen a periféria állapotáról, illetve beállítsa a periféria üzemmódját. Az előbbire a periféria STATUS regisztere /SR/, az utóbbira a parancs regiszter CONTROL /CR/ szolgál.

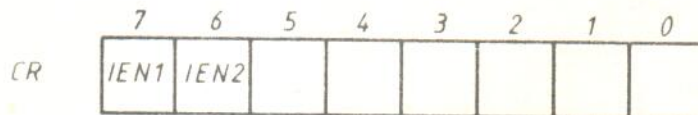
Az SR-ben a periféria aktuális állapotát tároljuk, az SR tartalma a CPU felől lekérdezhető. Az SR bitjeit jelzőbiteknek flag-eknek szokás nevezni. A flag-ek feladata jelezni a CPU számára, hogy egy megkezdett működés befejeződött-e, szabad-e a periféria. A CR a periféria vezérlésére szolgál.

Az SR általában csak olvasható, a CR általában csak írható, a program szempontjából a két regiszter ugyanazon a címen is szerepelhet, köztük az  $\overline{RD}$ ,  $\overline{WR}$  jel választ. A regiszterek hossza általában egy byte.

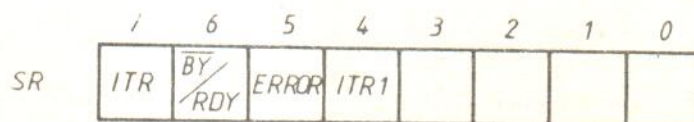
#### Regiszterek javasolt felépítése

Programmegszakításos működés esetén a CR-ben alkalmazni kell egy a megszakítást engedélyező bitet. /IEN - interrupt enable/

Ha a perifériánál több megszakítási ok fordul elő, és az általuk felmerült IT kérést külön-külön akarjuk engedélyezni, ill. letiltani, akkor további IEN biteket is felvehetünk.



#### Status regiszter felépítése



ITR=1 jelzi, ha a periféria megszakítást kér. Az interrupt kérés akkor következhet be, ha IEN=1 és a periféria megfelelő állapotban van. A megfelelő állapotot a  $\overline{BY}$  /RDY bit jelzi.

Kiviteli perifériák /passzivak/ esetében a "szabad" flag  $\overline{BY}$ =1 jelzi, hogy a periféria nem végez átvitelt. Programozott átvitelnél várakozó hurokban a program  $\overline{BY}$  állapotát vizsgálja. Ha  $\overline{BY}$ =1, akkor a következő működés indítható. Megszakításos működéskor, ha IEN=1 és  $\overline{BY}$ =1, az illesztő megszakítást kér

$$ITR = IEN \cdot \overline{BY}$$

A  $\overline{BY}$  flag bekapcsolásnál illetve, ha a periféria befejezte működését 1-be billen, és a működés indításakor törlődik.

Beviteli /aktív/ perifériák adatot állítanak elő, amit az RDY=1 jelez.

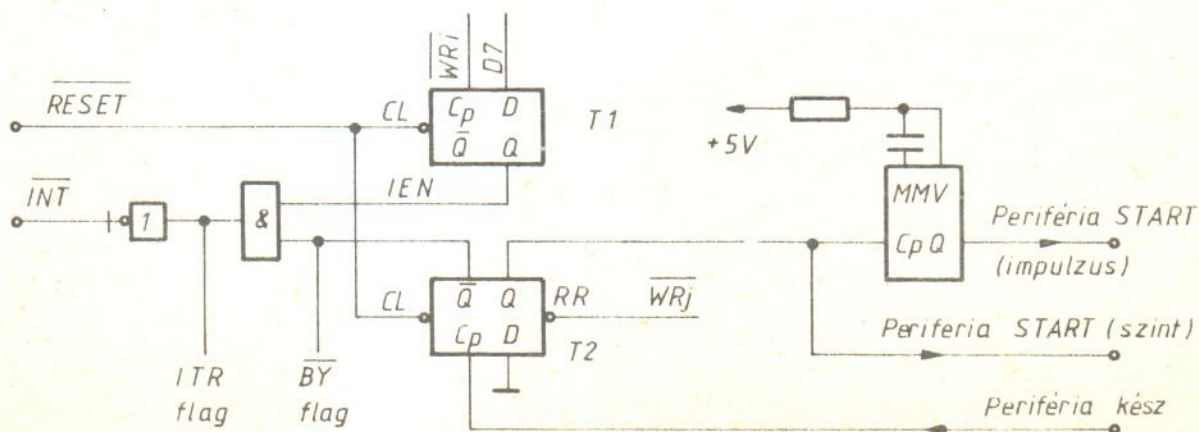
Programozott átvitelnél ezt kell vizsgálni, megszakításos működéskor az interrupt feltétele:

$$ITR = IEN \cdot RDY$$

Bekapcsoláskor és az előállított adat bevitele után RDY törlődik.

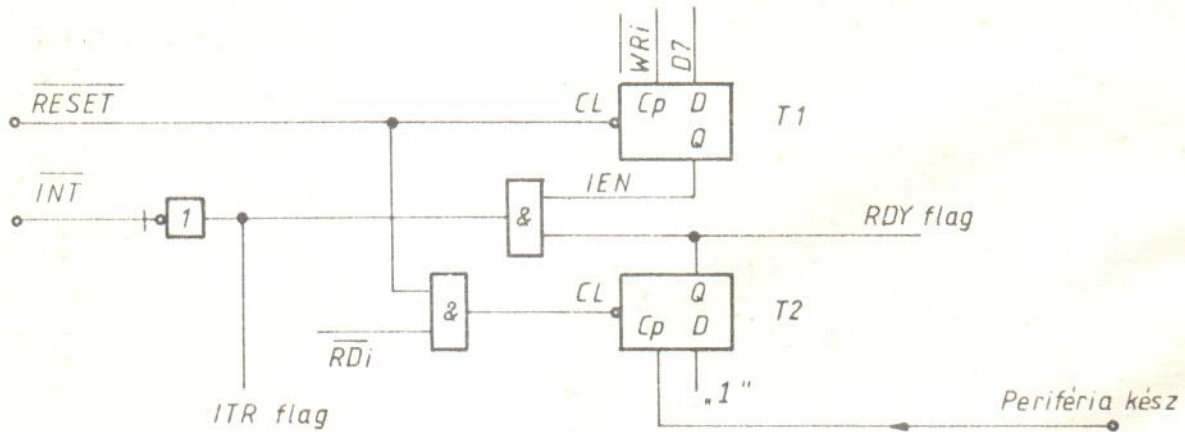
ERROR - a perifériában előforduló hibát, míg ITR1 egyéb okból származó interruptkérést jelöl. A fennmaradó bitek további jelzéseket szolgáltathatnak a periféria állapotáról.

BUSY -  $\overline{BY}$  flages változat



26 ábra

READY flages változat



2.7. ábra

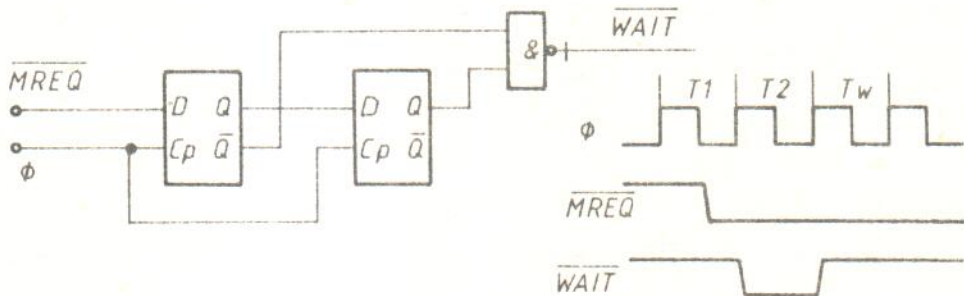
A  $\overline{BY}$  ill. READY flages szinkronizálási módszerre 2.6. és 2.7. ábrán látható példa. A CR-nek a T1, SR-nek a T2 D típusú flip-flop felel meg. Az ITR valamint a  $\overline{BY}$ /READY flagek az adatbuszra háromállapotú meghajtókon keresztül csatlakoztathatók.

2.4. Várakozás kérés

Ha a megcímzett memoria vagy CU nem képes az utasítást a processzor által meghatározott idő alatt végrehajtani, vagyis nem tudja a rendelkezésre álló idő alatt az adatokat fogadni, illetve az információt az adatvonalakra adni, akkor

a processzor működését átmenetileg meg kell állítani olyan időtartamra, hogy a megnövekedett idő az adatátvitel, illetve az utasítás végrehajtásához elegendő legyen. A processzor megállítása tetszőleges számú ütemre a  $\overline{\text{WAIT}}$  jel L szintre helyezésével lehetséges. Ha a rendszerben dinamikus memoriák vannak és frissítésüket a Z80-as processzor végzi, akkor a WAIT állapotok idejét korlátozni kell. Ugyanis ez idő alatt a processzor nem végez memoriafrissítést, így a memória tartalma elveszhet.

A várakozást kérő áramkörök kialakítása attól függ, hány ütem WAIT-et kell kérni. Egy ütem várakozás beiktatása memoriatranszfer esetén a 2.8. ábrán látható:



2.8. ábra

Ha memoria tartományban elhelyezkedő CU-t illesztünk, akkor a 2.8. ábra kapcsolásánál az  $\overline{\text{MREQ}}$  jel helyére  $\overline{\text{SEL}}$ -et kell kötni, így várakozás kérés csak a periféria kiválasztásakor történik.

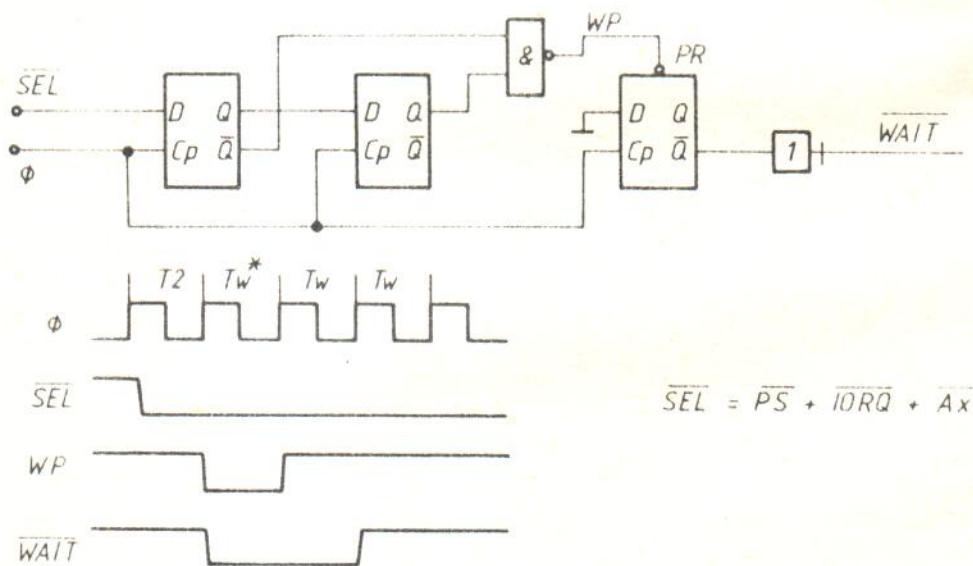
Input/output utasítást használó perifériakezelés ese-



tón a 2.8. ábrán látható kapcsolás a következő módosítással használható:

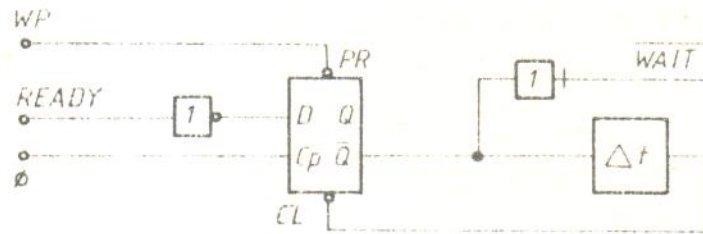
- Az  $\overline{MREQ}$  jel helyére  $\overline{IORQ}$ -t kötve minden I/O utasítás esetén plusz egy WAIT állapot kerül beiktatásra
- Az  $\overline{MREQ}$  jel helyére  $\overline{SEL} = \overline{PS} + \overline{IORQ} + \overline{A_x}$  kötve csak a megcímzett periféria kér várakozást.

Két ítem várakozás beiktatására a 2.9. ábrán látható példa :



2.9 ábra

Változó számú WAIT állapotot akkor kell alkalmazni, ha a buszciklus hosszát a periféria működésétől függően változtatni kell.



2. 10. ábra

A várakozás kérést a WP jel indítja el /előállítás a 2.9. ábrán/. Ha a periféria kész, a READY jel H szintű lesz és az órajel visszabillenti a WAIT kérő D flip-flopot. Ha valamilyen hiba folytán a READY jel nem alakulna ki, akkor a processzor állandóan WAIT állapotban maradna, a ciklus "befagyna". Ezért egy ún. watch-dog áramkört kell kialakítani, ami egy beállítható idő után törli a WAIT kérést. Ez a szerepe a  $\Delta t$  késleltetésnek. A  $\overline{\text{WAIT}}$  vonalat nyitott kollektoros kimeneten keresztül kell meghajtani.

A minimális illesztő részegységeinek megismerése után, ezek együttes alkalmazására az alábbiakban tekintsünk egy példát.

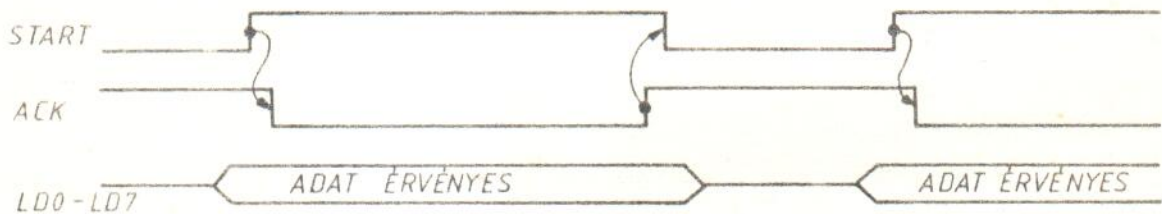
### 3. Feladat

Tervezzen Z80-as mikroprocesszorhoz lyukszalag lyukasztó kezeléséhez illesztő egységet, amely programozott és megszakításos átvitelt tesz lehetővé.

- illesztő egység specifikációja: Az illesztő egység ismerje fel a CPU felől érkező "RETURN" /hexa kódja 0D/ karaktereket és ennek kilyukasztása után lyukasszon ki egy darab

"LINE FEED" /hexa kódja 0A/ karaktert is. Kész jelzést a CPU felé csak a 0A lyukasztása után adjon. Az illesztő egység a Z80-as processzor I/O utasításai révén legyen elérhető. Az illesztő címe hexa 38 - 3B tartományban legyen. Minden, az illesztőhöz való adat illetve információ átvitel esetén, plusz két WAIT ütem kerüljön beiktatásra az I/O ciklusokba.

- Lyukasztó interface specifikációja: A lyukasztó a START jel hatására kilyukaszt egy karaktert. A karaktert legkésőbb a START jel megjelenéséig rá kell kapcsolni a lyukasztó LD0 - LD7 adatvonalaira és az ACK válasz jel megjelenéséig rajta kell tartani. A lyukasztó 8 csatornás ASCII kódolású szalagot lyukaszt, a jelek TTL szintűek.



3.1. ábra

Működési leírás /Kapcsolás a 3.2. ábrán látható/

A csatoló regiszterei a Z80 I/O utasításaival a 38-3BH címeken érhetők el. A 30H tartományt a 7485 komparátor azonosítja. A  $\overline{SEL}$  jelet, ami a CU-t kiválasztja a korábban ismertettekhez hasonló áramkör állítja elő. A periféria parancsok a 74155 dekódoló kimenetein jelennek meg, és a következők lehetnek:

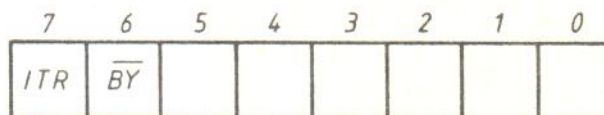
Statusz olvasás: - 39H címről olvasva  $\overline{RD}$ =L lesz, ami a 74125 háromállapotú meghajtók kimeneteit engedélyezi, vagyis a statusz regiszter tartalma az accumulatorba kerül.

Vezérlő regiszter beírása: - Kivitel /írás/ a 39H címre a 74155  $\overline{WR}$  kimenetét a output idejére L szintre helyezi, felfutó éle használható az adatvonalak mintavételezésére. Kivitelnél az accumulator tartalma kerül az adatbuszra, tehát ebben az esetben output előtt a vezérlőszót kell az accumulatorba tölteni.

Adatátvitel: Lyukszalag lyukasztó esetében csak adatkivitel szükséges, amit a 38H címre való irással lehet kezdeményezni. Ekkor  $\overline{WR}$ =L lesz, aminek következtében az adatbusz tartalma - jelen esetben a lyukasztandó adat - beke-  
rül a CU adatregiszterének szerepét betöltő 8212-es típusú latch-be. Az adatkiviteli parancs egyuttal a lyukasztást is indítja,  $\overline{WR}$  felfutó élével.

Control regiszter: Jelen esetben egy bites és a  $D_2$  jelű D flip-flop valósítja meg, ami az adatbusz  $D_7$ -es bitjét tárolja. Ez a bit használatos a megszakítások engedélyezésére.

Statusz regiszter: Két bites, amit a  $D_1$ - $D_2$  flip-flopok tárolnak.



Megszakításos periféria kezelésnél az ITR vizsgálata alapján dönthető el, hogy melyik csatoló kért IT-t.

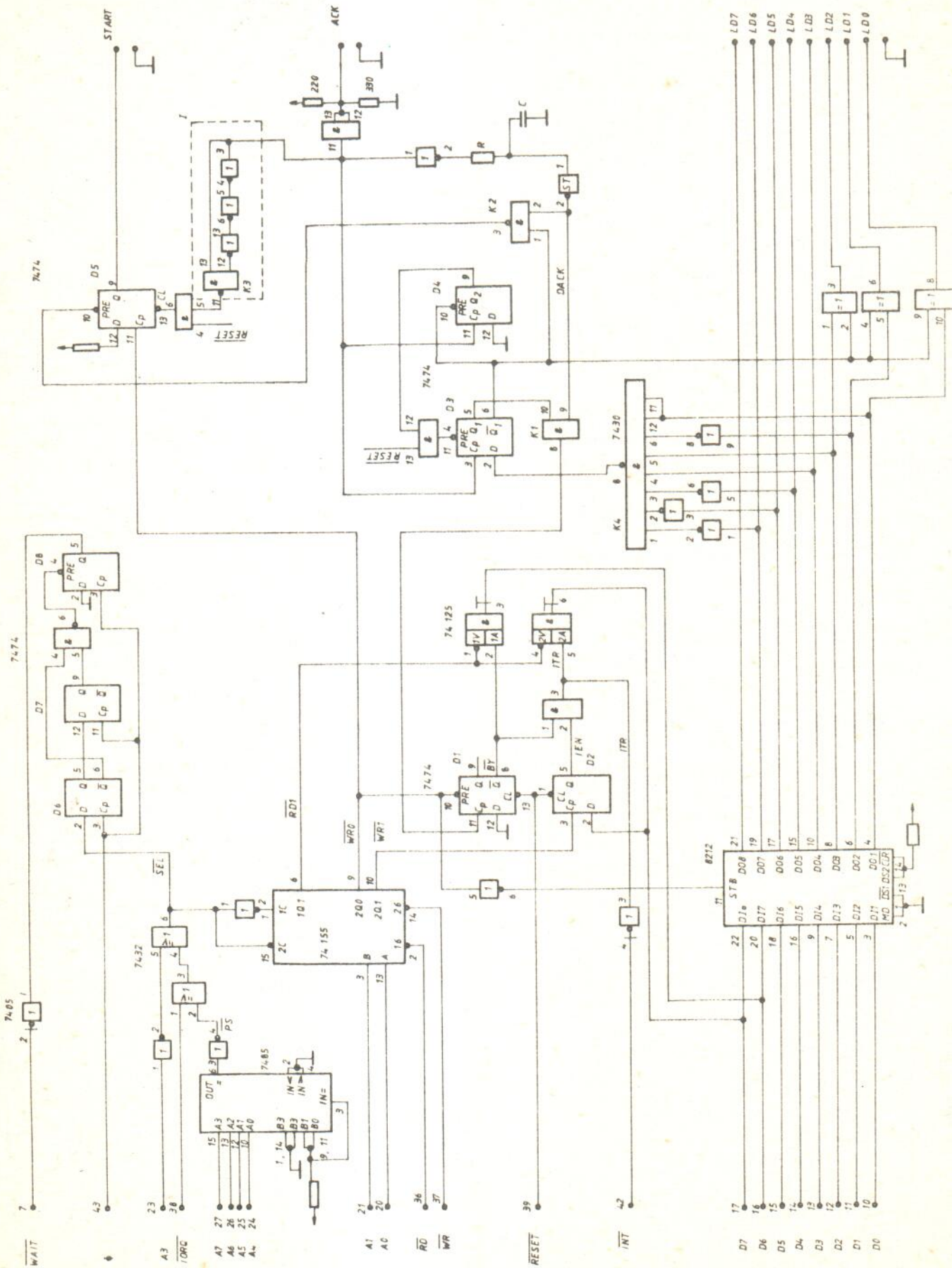
$\overline{BY}=L$  esetén a lyukasztó foglalt,  $\overline{BY}=H$  esetén lehetséges az újabb adatkivitel, s ezzel egyuttal a lyukasztás indítása. A kapcsolás eddig a WAIT logikával együtt /D6-D8/ megegyezik a korábban ismertetettel, plusz elemeket a CR, LF lyukasztása jelent.

A mikroszámítógép bekapcsolásakor megjelenő  $\overline{RESET}$  jel alaphelyzetbe állítja a CU regisztereit és tárolóit. Ekkor  $D_3/Q_1=H$   $D_4/Q_2=H$  szintre kerül.  $\overline{WRC}$  felfutó éle  $D_5/Q$ -t l-be írja,  $START=H$  lesz és elindul a lyukasztás  $\overline{BY}=L$ . Az ACK felfutó élére az l jelű impulzus formáló áramkör kb. 20 nsec szélességű L szintű impulzust ad a  $K_3$  kimenetén megszüntetve a START jelet. Ha a lyukasztott karakter nem CR volt, akkor az RC tag által késleltetett ACK jel a DACK felfutó éle  $K_1$ -en keresztül  $D_1$  tároló  $\overline{Q}$  kimenetét l-be írja  $\overline{BY}=H$  lesz, jelezve a lyukasztás megtörténtét. Ha a lyukasztott karakter CR volt, akkor  $K_4$  /ami a OD-t de-kódolja/ kimenete L szintre kerül. A CR lyukasztása utáni ACK felfutó éle mintavételezve  $K_4$  kimenetét  $D_3/Q_1$ -t 0-ba írja, s így a DACK nem tudja a  $D_1$  flip-flop foglaltság jelzését megszüntetni. A  $D_3/\overline{Q_1}$ -en megjelenő H szint a 7486-os kizáró vagy kapuk kimenetén az adatvonalak alsó három bitjének az inverzét eredményezi.

/ OD = 0000 1101      OA = 0000 1010 /

Tehát a lyukasztó adatvonalain az LF kódja jelenik meg, ezután a DACK  $K_2$ -n keresztül /mivel  $D_3/\overline{Q_1}=H$ / a  $D_5$  preset bemenetére 0-at adva a START jelet l-be írja, indítva az LF lyukasztását.

Míg  $D_3/\overline{Q_1}=H$   $D_4$  nincs presetelve, ezért a lyukasztás utáni ACK felfutó éle  $D_4/Q_2$ -t 0-ba billenti. Ez  $D_3/Q_1$ -t l-be írja, ezzel  $K_2$ -n keresztül letiltja a további START jel kiadását, viszont  $K_1$ -n keresztül engedélyezi, hogy a DACK felfutó élére megszűnjön a foglaltsági jelzés  $\overline{BY} = l$ .



84286. GAMF rota