

Számítógép architektúrák alapjai

Hallgatói jegyzet

I/O rendszer

2023

Tartalom

1.	I/O rendszer.....	1
1.1.	Fejlődése	1
1.2.	Fajtái	1
1.3.	Programozott I/O	2
1.3.1.	I/O port részei	2
1.3.2.	Különálló I/O címtér	2
1.3.3.	Memóriában leképzett I/O.....	3
1.3.4.	Működés.....	4
1.4.	DMA	5
1.4.1.	Működése	5
1.4.2.	Blokkos átvitel	6
1.4.3.	Cikluslopásos átvitel	7
1.5.	I/O csatorna	7
1.5.1.	Szelektor csatorna.....	8
1.5.2.	Multiplexer csatorna	8

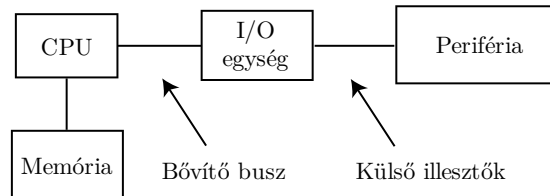
Készítette: Kováts Máté

A jegyzet Durczy Levente 2019 őszi Számítógép architektúrák alapjai előadás videói alapján készült.

Segítséget jelentett Uhrin Ádám és Nagy Enikő korábbi jegyzetei.

1. I/O rendszer

Alapvető képessége a számítógépeknek, hogy más egységekkel adatot cseréljenek, és ehhez a kommunikációs képességhez szükség van: CPU-ra, memóriára és perifériákra. Ezeket köti össze a buszrendszer, de a kettő között van még egy eszköz, az I/O egység. Ez választja el a processzort a perifériától. Az I/O egység adatregiszterei még a memória címtérhez tartoznak.

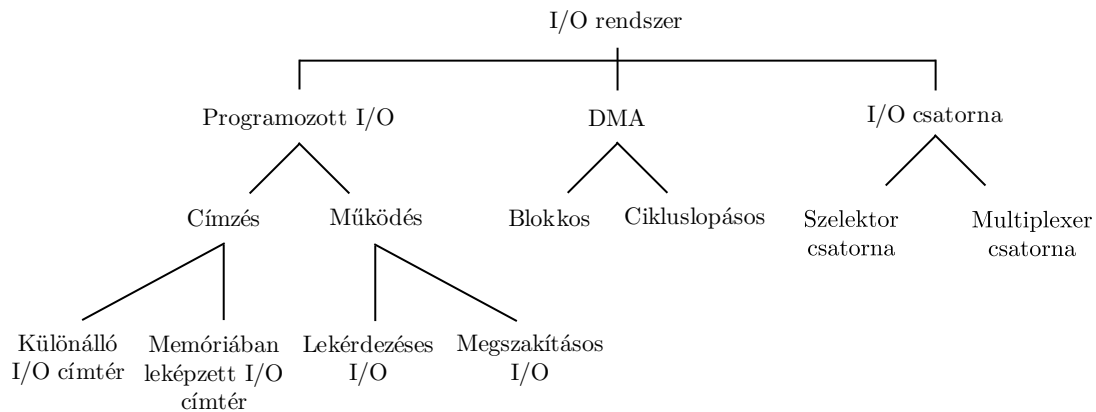


1.1. Fejlődése

1. A CPU közvetlenül vezérli a perifériát. Ezt úgy nevezik, hogy megszakítás nélküli programozott I/O (wait for flag). A processzor kiadta az utasítást a perifériának és nem csinált mást csak várt a válaszra.
2. Egy I/O modult (vezérlőt) alakítanak ki. Ez vezérli a kommunikációt a perifériákkal
Kétféle változata volt:
 - Megszakítás nélküli programozott I/O-t használ
 - Megszakításos vezérlés: a processzornak nem kell várni a I/O művelet befejezésére, mert az I/O egység küldi a kérést a perifériának és várja a választ. Amikor azt megkapja, küld egy megszakításkérést a CPU-nak → nő a teljesítmény
3. DMA: I/O modul számára DMA segítségével közvetlen hozzáférés a perifériához. Ez lényegesen nagyobb sebességű.
4. I/O csatorna: lassabb perifériák számára. I/O-ra specializált utasításkészlettel rendelkezik.
5. I/O processzor: I/O célú utasításokat dolgoz fel, saját operatív tárral rendelkezik.

1.2. Fajtái

1. Programozott I/O egység: A CPU által irányítottan történik az adatátvitel. A processzor vezérli, irányítja és lezárja az I/O műveletet. Megvalósítás egyszerű, de jelentős processzor időt vesz el.
Kétféleképpen működhet:
 - a. CPU aktív részvételével (lekérdezéses átvitel): minden műveletet a CPU irányít és vezérel
 - b. Megszakításos vezérléssel
2. CPU közreműködése nélkül történik az adatátvitel → plusz vezérlő egység kerül a rendszerbe.
Példa: DMA
3. I/O csatorna



1.3. Programozott I/O

- Különálló I/O címtér annyit tesz, hogy a CPU két címteret lát, egyet a perifériáknak és egyet a memóriának, ezáltal ugyanaz a cím szerepelhet memóriacímként és I/O címként is.
- Közös buszhasználat van.
- Létezik az M/IO vezérlőjel, mely megmondja, hogy az adott időpillanatban memória- vagy I/O-cím van a címsínen.
- Azon regisztereket, amelyeken keresztül a processzor a perifériákkal kommunikálhat, I/O portnak nevezzük.
- Az I/O port fizikailag a vezérlőkártyában helyezkedik el.

1.3.1. I/O port részei

- parancs regiszter (command): a CPU ide írja a „kívánságait”
- adatregiszterek: ez tekinthető a bővítőbusz végállomásának
 - o data input regiszter: : ebből olvassa a CPU a perifériáról kapott adatokat
 - o data output regiszter: ebbe írja a processzor a periféria számára küldött adatokat
- állapotregiszter: állapotinformációkat közöl a perifériáról. Minden bitje mást jelent. Legjellemzőbb állapotok például a megszakításkérés, Ready, Busy.

Lehet közös:

- az állapot és parancs regiszter, mert a CPU felől az egyik írva, másik olvasva van
- adatregisztereknél pedig a data input és output

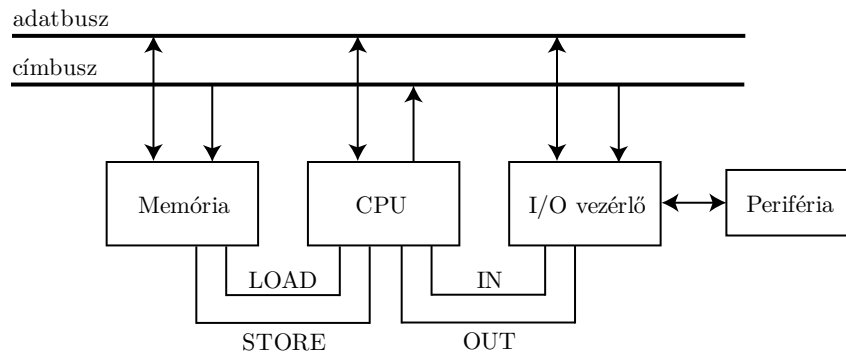
Egyéb regiszterek:

- jelenlét ellenőrző regiszter: megmondja van-e eszköz kapcsolva a I/O portra
- eszköz tulajdonságait tartalmazó regiszter (plug and play funkciót teszi lehetővé)
- egyes funkciókhoz akár több regiszter is tartozhat

1.3.2. Különálló I/O címtér

A CPU a memóriával a kommunikációt LOAD/STORE utasításokkal végzi. Megcímzi a címbuszt, és utána a memóriával adatátvitellel kommunikál. Ugyanígy címzi a I/O vezérlőt, viszont külön

utasításokkal (IN és OUT) biztosítja az adatátvitelt. Az I/O vezérlőn megtalálhatóak a portok, melyek biztosítják a periféria felé az adatkommunikációt.



Előnye:

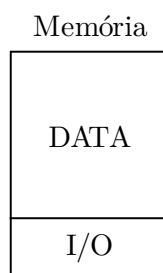
- egyszerű
- olcsó megvalósítás

Hátránya:

- terheli a CPU-t
- plusz utasítások az I/O kezeléshez

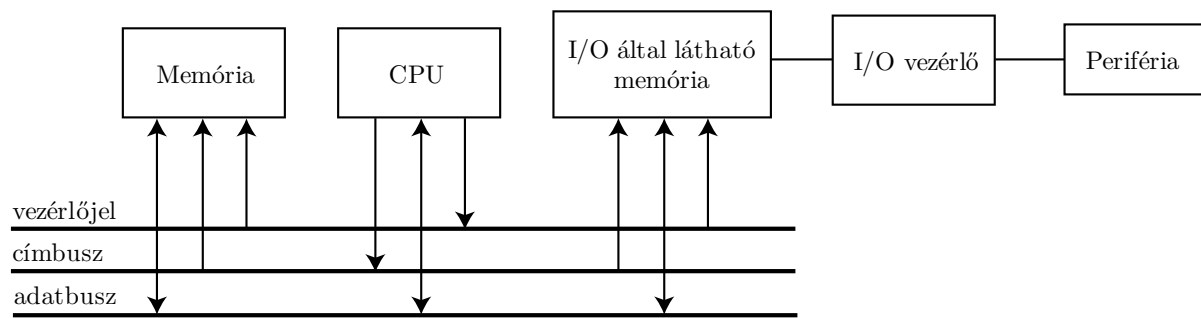
1.3.3. Memóriában leképzett I/O

Annyiban különbözik a különálló I/O címtértől, hogy a perifériák számára a memóriában lefoglal a CPU egy területet. A vezérlőjelen biztosítja a processzor a vezérlő információkat (írás, olvasás, M/IO). Egy memória hivatkozás akkor válik I/O utasítássá, hogyha a címbe az I/O által látható memóriacím kerül.



Előnye:

- LOAD/STORE utasítással történik az adatok mozgatása, nem kell külön utasítás
- I/O vezérlő hozzáfér a rendszerbuszhoz → gyorsabb az átvitel



Például ilyen periféria a kijelző. A legtöbb grafikus rendszer megengedi a CPU számára, hogy közvetlenül címezze azt a frame buffert, amely a képernyőn az adott pillanatban megjelenő képet tartalmazza.

1.3.4. Működés

Átviteli módszerek alapján:

- feltétel nélküli adatátvitel (direkt programozott)
- feltételes adatátvitel

Feltétel nélküli előfeltételei:

- a perifériának mindig adatátvitelre alkalmas állapotban kell lennie
- ellenőrzésre sem az adatátvitel előtt, sem utána nincs szükség
- semmilyen szinkronizálás nincs a CPU és a periféria között

Például a kijelző.

Hátránya: nincs visszacsatolás

Feltételes adatátvitel: valamilyen feltétel teljesülése esetén adatátvitel

Típusai:

- lekérdezéses adatátvitel: a processzor beírja a kívánságát az I/O vezérlőbe és várja a választ az által, hogy folyamatosan vizsgálja az állapotregiszter tartalmát.
Ez egy nagyon pazarló működés, mert a periféria nagyságrendekkel lassabb, azért találták a megszakításos adatátvitelt.
- megszakításos adatátvitel: a processzor beírja a kívánságát az I/O vezérlőbe és elkezd mással foglalkozni. A periféria vezérlő rendelkezik egy megszakítás rendszerrel. Ha megtörténik az adat beolvasás, akkor megszakításkérést küld a processzornak és az a következő utasítás töréspontban lekérdezi az állapotregisztert (és ha Ready, akkor megtörténik az adatátvitel).

Például nyomtató.

Előnye: sokkal gyorsabb

Hátrány:

- nagy mennyiségű adat esetén lassú és sok megszakítással jár → HDD és SSD esetén nem elég
- mindent a CPU vezérel

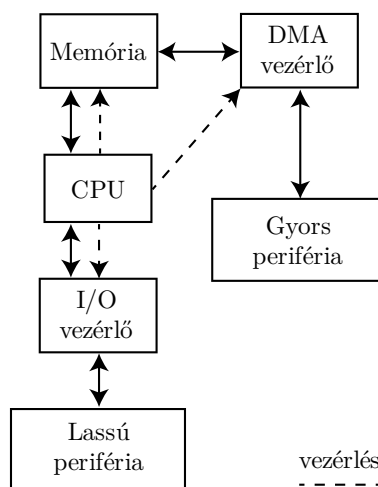
1.4. DMA

A Direct Memory Access tovább gyorsítja az adatátvitelt a perifériákról és még inkább leveszi a terhet a processzor válláról.

Fontos tulajdonságai:

- nagy mennyiségű adat esetén (általában blokkos adatátvitellel)
- gyors perifériáknál érdemes használni
- CPU közreműködése nélkül végzi az adatátvitelt

A DMA vezérlő a memórián keresztül a nagy mennyiségű adatátvitelt biztosítja a gyors periféria számára, a CPU csak a vezérlést biztosítja. A hagyományos I/O vezérlő a lassú periféria számára biztosítja az adatátvitelt a processzor közreműködésével.



Viszonylag szerény mértékű komplexitás növekedés segítségével valósítható meg a közvetlen adatátvitel.

Feltétele, hogy címet kell tudni generálni a közvetlen tárhozzáféréshez, valamint busz vezérlési funkciókkal is el kell látni.

Előnye a jóval kevesebb megszakítás.

1.4.1. Működése

A DMA el van látva regiszterekkel: DC, I/O címregiszter, I/O adatregiszter, belső transzparens regiszterek. A működés mind blokkos, mind cikluslopásos esetén felparaméterezéssel kezdődik:

Programozott I/O-val átvisszük a processzorból a DMA vezérlőbe az átvitelhez szükséges információkat

1. felhelyezzük az adatokat az adatbuszra
2. a DC-be beírjuk az átvinni kívánt adategységek számát
3. I/O AR-be beírjuk az adat leendő memóriabeli kezdőcímét

Paraméterek lehetnek:

- írás vagy olvasási művelet (tulajdonképpen átvitel iránya)
- I/O egység címe
- memória cím kezdőértéke (I/O AR-be)
- átvivendő adat jellege (byte, félszó, szó)
- olvasandó/írandó egységek száma (DC-be)
- átvitel módja (blokkos vagy cikluslopásos)
- DMA csatornához prioritási értékeket rendelhetünk
- résztvevő egységek (memória-memória, I/O-memória, I/O-I/O)

1.4.2. Blokkos átvitel

1-3. CPU felparaméterezi a DMA vezérlőt

4. A DMA request jellel kéri a rendszerbusz használati jogát

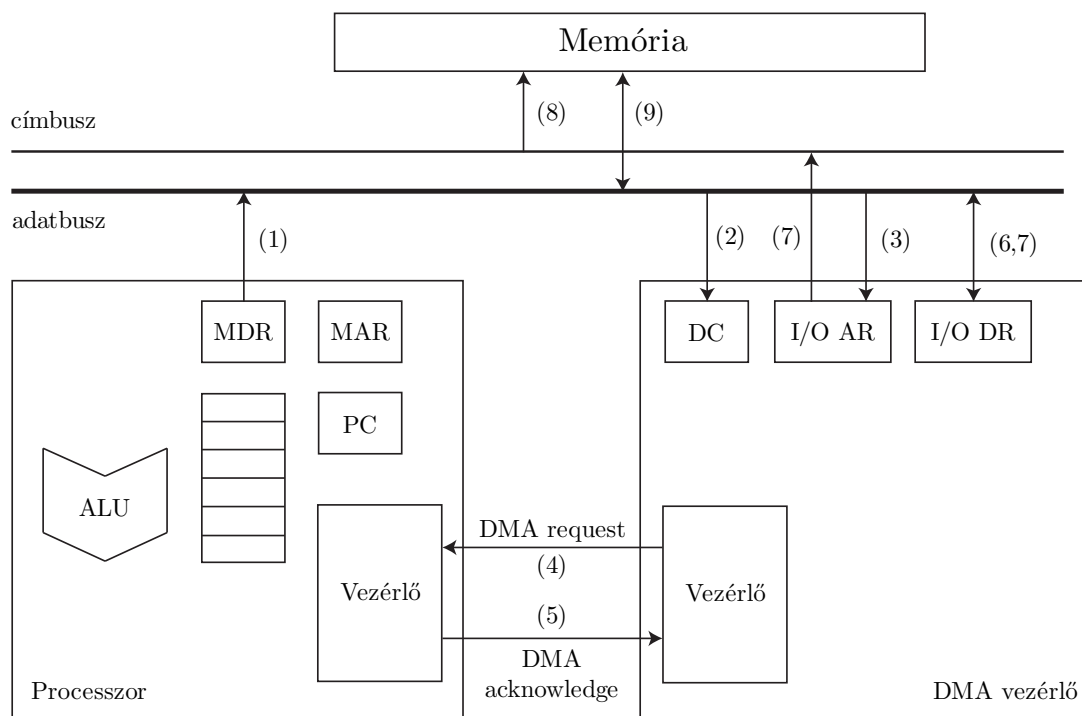
5. A processzor DMA acknowledge jelzéssel lemond a rendszerbusz használati jogáról

6. A DMA vezérlő a kapott adatok alapján a perifériáról beírja az első átvinni kívánt adatot az I/O DR-be

7-9. A DMA vezérlő az I/O DR-ben lévő adatot a rendszerbuszon keresztül beírja az I/O AR által meghatározott memóriacímre

10. A DMA vezérlő dekrementálja a DC-ben tárolt értéket, és inkrementálja az I/O AR -ben tárolt értéket (egy adategységgel növel – byte, félszó, szó)

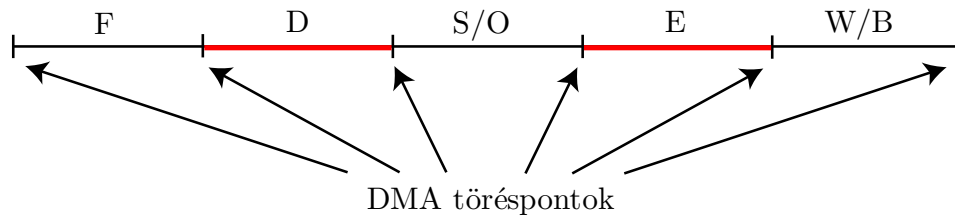
11. DMA ellenőrzi a DC tartalmát. Ha nem 0, vissza a (6) -ra. Ha igen, megszakításkérésrel jelzi a CPU felé, hogy befejeződött egy blokk átvitele. A CPU ellenőrzi az adatvitelt, megszünteti a buszhasználat engedélyezést és átveszi a vezérlést.



1.4.3. Cikluslopásos átvitel

Cikluslopás esetén nem a teljes időtartamra foglalja le a DMA vezérlő a busz használatát, hanem csak 1-1 adat átvitelének időtartamára. Ezt kevesebb adat esetén használják (például gyors nyomtatók).

Egy utasítás végrehajtásának felbontása:



F - fetch
D - decode
S/O - forrás operandus beolvasás
E - execute
W/B - writeback

Ezek közül a processzor FETCH, S/O és W/B esetén használja a rendszerbuszt. DECODE és EXECUTE esetén a CPU le tud mondani a rendszerbusz használatának jogáról és a DMA 1-1 adatot át tud irányítani a perifériáról → DMA szubrutin szerű működést végez:

1. Felparaméterezés
2. DMA request vonal aktiválása
3. A DMA töréspontban a CPU lemond a buszhasználatról és ezt a DMA acknowledge jel aktiválásával jelzi
4. Adatátvitel I/O és memória között
5. Ha a DC nem 0, de a periféria nincs Ready állapotban, akkor a DMA visszaadja a vezérlést a CPU-nak. Ha DC 0, a DMA megszakításjelzést küld és befejezi az adatátvitelt.

Hátránya: manapság a párhuzamos működés során a CPU közel minden óraciklusban használja a rendszerbuszt.

1.5. I/O csatorna

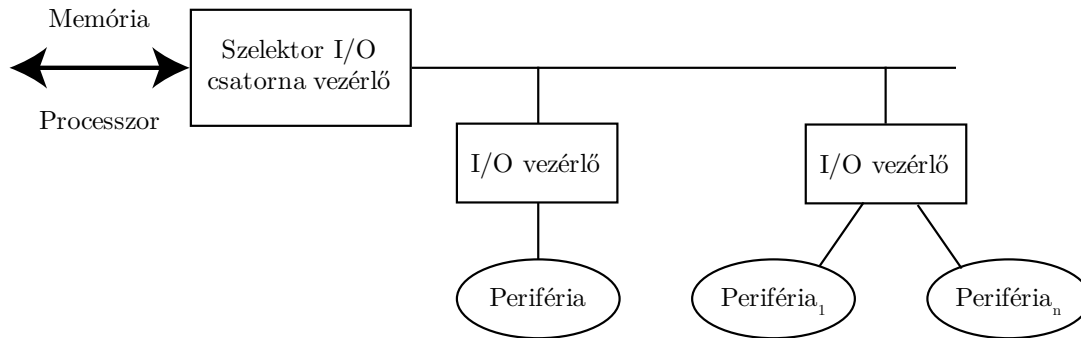
A DMA kiterjesztése a lassabb perifériákra. Fejlettebb, mint a programozott I/O, de egyszerűbb, mint a DMA. Ilyenkor a CPU nem hajt végre I/O műveletet. Az utasítások az I/O csatorna vezérlő számára a memóriában vannak tárolva. A CPU kezdeményezi az átvitelt, viszont az I/O csatorna hajtja végre a memóriában tárolt programot, ami specifikálja az átvitelt (nincs felparaméterezés).

Típusai:

- szelektor csatorna
- multiplexer csatorna

1.5.1. Szelektor csatorna

A lassabb eszközök közül is a nagyobb sebességű I/O eszközök számára. A szelektor I/O csatorna vezérlő egyetlen I/O vonallal rendelkezik, amire több I/O vezérlő is rácsatlakozhat különböző perifériákkal. A szelektor I/O csatorna vezérlő kommunikál a CPU-val és memóriával. Egyszerre csak egy egység kommunikálhat a vezérlővel és azon keresztül a memóriával. Azért szelektor, mert az adatátvitel gyorsan történik, ezért megoldható hogy-hogy más-más perifériáról történjen.



1.5.2. Multiplexer csatorna

Kifejezetten lassú perifériákhoz találták ki, amik folyamatosan küldik az információt, de viszonylag nagy az időkülönbség az adatok között. A csatorna vezérlőre több I/O vezérlő csatlakozhat párhuzamosan. Elég lassúak a perifériák ahhoz, hogy a vezérlő egy időben több perifériával kommunikálhat.

Két típusa van:

- byte multiplexer: bytonként küldi az adatokat
- blokk multiplexer: kisebb-nagyobb blokkonként küldi az adatokat

Cél az átvitel maximalizálása.

