

## 8. ULAZNO/IZLAZNI PODSUSTAV

Ulazno/izlazni podsustav dotaknut je djelomično i u dosadašnjim razmatranjima. Tako je spomenut prekidni sustav procesora kao i programirani prijenos podataka kod procesora MC68000. U ovom poglavlju detaljnije će se odrediti funkcije U/I podsustava te zahtjevi na sklopovlje i programsku podršku. Kao prvi korak potrebno je odrediti ulogu i svrhu U/I podsustava iz kojih slijede zahtjevi na sklopovlje i programsku podršku. Kao najjednostavnije rješenje je programski upravljani U/I prijenos podataka, dok je s motrišta iskoristivosti sklopovlja bolje rješenje U/I prijenos pomoću prekida te direktan pristup memoriji.

### 8.1 Struktura U/I podsustava

Postavlja se pitanje koji uređaji su sastavni dio U/I sustava. Jasno je da tu spada tipkovnica, miš kao ulazni uređaji te pisač, monitor kao izlazni uređaji. Ali što je s magnetskim diskom koji je interna jedinica računala? Magnetski disk je U/I uređaj ne zato što ostvaruje vezu računala s vanjskim svijetom nego zato što ima bitno različita fizička i logička svojstva od procesora i memorijskog sustava, posebice u odnosu na brzinu odziva i vremensko vođenje.

Navedeni U/I uređaji spajaju s procesorom i memorijom preko U/I podsustava, obično odgovarajućeg U/I međusklopa (*interface*). U/I međusklop osigurava prijenos različitih volumena podataka (nekoliko okteta do megaokteta) između U/I uređaja i procesora koja se odvija asinkrono s radom procesora. Sistemski takt vremenski upravlja radom procesora i memorijskog sustava, dok je dotok s i prikaz podataka na U/I uređaju sasvim neovisan o sistemskom taktu (upis znaka na tipkovnici, pomak i pritisak tipke miša, ispis na pisaču, prijem ili predaja paketa preko mreže itd.).

Temeljem svega navedenog može se zaključiti da se sljedeći faktori moraju razmotriti prilikom projektiranja U/I podsustava:

1. Odabir podataka što podrazumijeva odabir U/I uređaja i odabir lokacije u uređaju gdje su podaci pohranjeni.
2. Prijenos podataka, odnosno količina podataka koju je potrebno prenijeti, brzina i smjer prijenosa.
3. Sinkronizacija: predaja podataka samo kada je U/I uređaj spreman prihvatiti podatak, a prijem tek kada je podatak dostupan.

Ovi zahtjevi značajno se razlikuju od zahtjeva koji se postavljaju kod prijenosa podataka između memorije i procesora. Svaka memorijska lokacija jednoznačno je određena cjelobrojnom adresom. Lokacija podatka u U/I uređaju određena je adresom uređaja, a zatim adresom podatka unutar uređaja koja može biti relativno složena (npr. kod magnetskog diska lokaciju određuje broj ploče (*plate*), cilindar na ploči (*track*), sektorom na cilindru te pomakom od početka sektora. Količina podataka koja se prenosi u jednom pristupu također ovisi o U/I uređaju i može biti samo 1 bit npr. kod pritiska tipke miša pa sve do nekoliko okteta kod magnetskih diskova. S aspekta sinkronizacije prijenosa podataka, odnosno međusobno vremensko vođenje također značajno se razlikuje od vremenskog vođenja u podsustavu procesor-memorija. Tako npr. vrijeme odziva magnetskog diska prije je reda desetak ms, nakon kojeg se prenese nekoliko okteta. Kašnjenje u odazivu, kao i brzina kojom se kasnije podaci prenose je nevezana s procesorskim taktom.

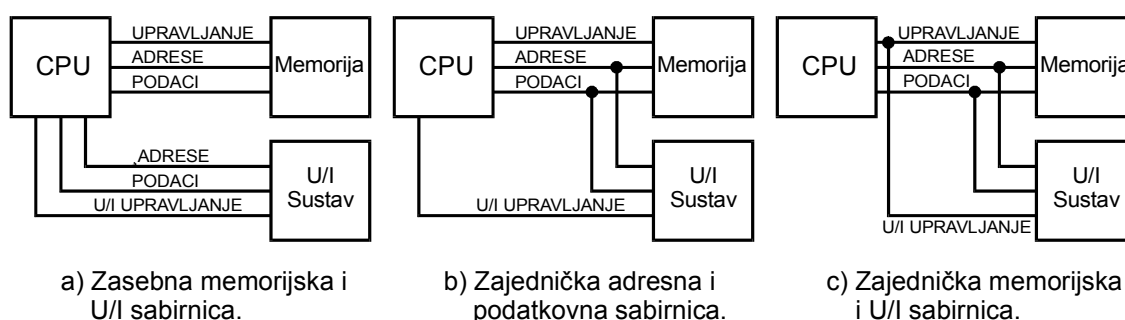
Sve navedeno vodi zaključku da je potrebno imati poseban elektronički sklop koji ostvaruje vezu između procesora i U/I uređaja tzv. U/I međusklop.

**Standardizacija U/I međusklopova.** Razmatranja vezana uz zahtjeve na sklopove koji povezuju U/I uređaj i procesor vode zaključku da su oni jako ovisni o vrsti U/I uređaja, ali ipak ih je potrebno na određeni način standardizirati definirajući strukture podataka koje oni pohranjuju. Funkcija U/I međusklopa je da prihvati s ili pripremi za U/I uređaj podatke koji mogu imati različite strukture razlažući riječi ili oktete u bitove ili skupine bitova, odnosno spajajući ih u složenije strukture podataka. Pri tome se ujedno izvodi i sinkronizacija prijenosa, otkrivanje pogrešaka u prijenosu, eventualna korekcija pogrešaka i sl. Navedene funkcije ostvaruju se pomoću spremnika U/I međusklopa koji su postavljeni (mapirani) ili u memorijski prostor procesora ili u zaseban U/I prostor. Spremnici imaju različite funkcije kao npr. prijem ili slanje podataka, pohranu statusa prijenosa. U njih se može čitati i pisati ili samo čitati ili pisati zavisno o funkciji.

Ovakva koncepcija spajanja U/I uređaja s podsustavom procesor-memorija zahtjeva i definiranje fizičkog povezivanja procesora s U/I međusklopom.

**Struktura U/I sabirnice.** Sve funkcije U/I međusklopa, određivanje lokacije podataka, prijenos podataka i sinkronizacija, mogu se sažeti u jednostavan prijenos u i iz njegovih spremnika. Procesor pristupa spremnicima U/I međusklopa naredbama sličnim ili identičnim naredbama za upis u i čitanje iz memorije. Svaki spremnik U/I međusklopa ima svoju adresu, a programer mora znati njegovu funkciju (značenje pojedinih bitova) te o kakvom se spremniku radi (samo upisni, samo ispisni ili upisno/ispisni).

Realizacija ovog prijenosa zahtjeva i fizičku strukturu za prijenos podataka, odnosno odgovarajući sabirnički sustav. Ovaj sabirnički sustav osim adresa i podataka može i mora prenositi odgovarajuće upravljačke i sinkronizacijske signale. U osnovi sabirnički sustav U/I uređaja jako je sličan memorijskom sabirničkom sustavu tako da kod različitih računalnih sustava postoje preklapanja među njima. Stupanj preklapanja različit je kod različitih procesorskih sustava. U osnovi razlikuju se tri rješenja prikazana slikom 8.1.

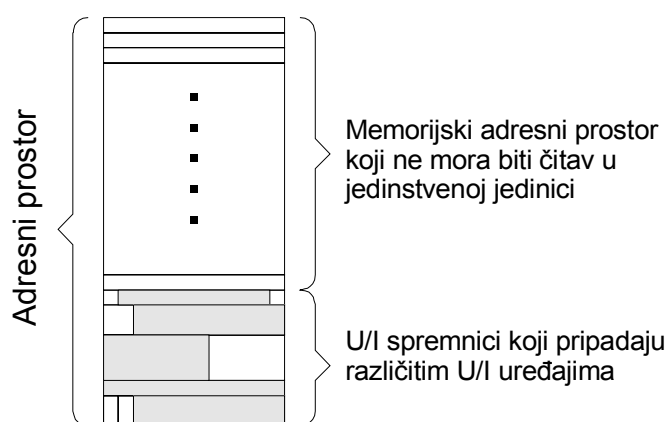


Slika 8.1 Moguća rješenja spajanja U/I sustava s procesorom.

Sustav prikazan na slici 8.1 a) ima zasebne podatkovne, adresne i upravljačke sabirnice za pristup memoriji i za pristup U/I uređajima. Ovakva arhitektura poznata je pod nazivom odvojeni (*isolated*) U/I sustav. Kod sustava prikazanog na slici 8.1 b), memorija i U/I međusklopovi dijele podatkovnu i adresnu sabirnicu dok su upravljačke sabirnice odvojene. Svaka upravljačka sabirnica nezavisno

generira svoje upravljačke signale čitaj, piši, spreman i sl. Ovakav sustav poznat je pod nazivom dijeljeni (*shared*) U/I sustav. Treće rješenje je dodijeliti dio memorijskog adresnog prostora U/I uređajima kao što je prikazano na slici 8.1 c). Ovakvo rješenje spajanja U/I sustava s procesorom naziva se memorijski preslikan (*memory mapped*) U/I sustav. Ovakvo rješenje dosta se koristi kod suvremenih računala, a osnovne prednosti su mu standardizacija prijenosa podataka i smanjenje broja linija koje procesor mora imati.

Prvobitna miniračunala i mikroračunala imala su 16- ili 24-bitovnu adresnu sabirnicu te su koristili zaseban ili dijeljeni U/I adresni prostor kako bi se povećao ukupan adresni prostor koji procesor može adresirati. Praktički svi suvremeni procesori imaju 32-bitovnu adresnu sabirnicu tako da nije problem dio adresnog prostora posvetiti U/I uređajima. Zbog navedenog će se memorijski preslikan U/I prostor uzeti kao rješenje u predstojećim razmatranjima. Podjela adresnog prostora prikazana je na slici 8.2.

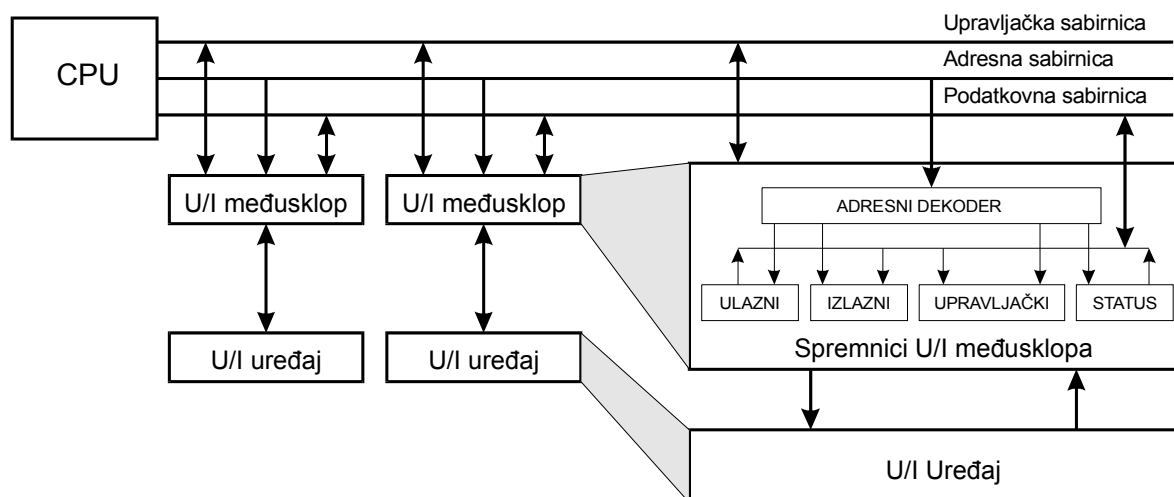


Slika 8.2 Adresni prostor podijeljen na memorijski i U/I adresni prostor.

Memorijskim modulima, kojih može biti više, dodijeljen je kontinuiran dio adresnog prostora, dok je ostatak dodijeljen spremnicima U/I uređaja. Za napomenuti je da sve lokacije U/I adresnog prostora, odnosno spremnici U/I međusklopova ne moraju koristiti sve bitove. Bitovi koji se ne koriste odbacuju se tijekom upisa u spremnik odnosno daju neodređeno stanje tijekom čitanja iz njih.

## 8.2 Programsko obavljanje U/I operacija

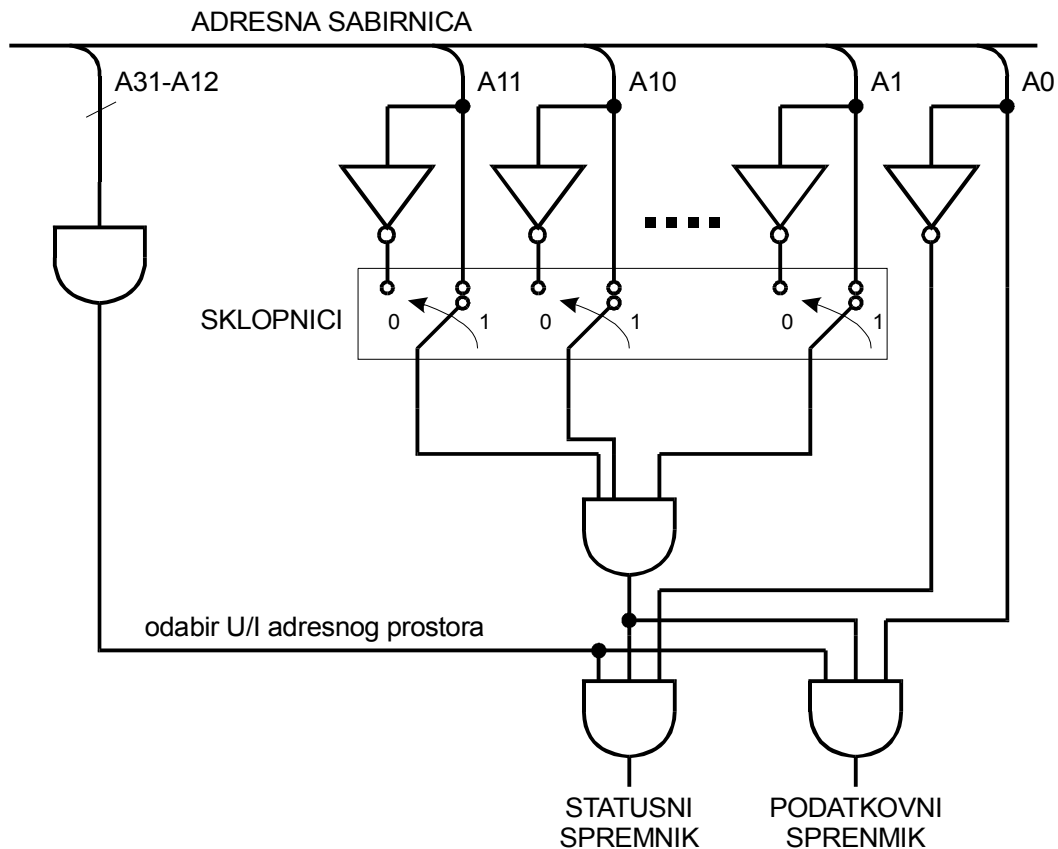
Programski upravljan U/I prijenos koristi se kod uređaja koje prenose male količine podataka, obično. jedan okteta ili riječ, a prijenos traje više procesorskih ciklusa. U ovakvim uvjetima procesor ima dovoljno vremena da pročita stanje U/I uređaja, čitajući statusni spremnik, da upiše upravljačku riječ u upravljački spremnik te da izvede prijenos podatka čitanjem iz ili pisanjem u ulazni ili izlazni spremnik. Sve ove podatke procesor razmjenjuje s U/I međusklopom koji prilagođava specifičnosti U/I uređaja jedinstvenom sabirničkom sustavu kako je prikazano slikom 8.3.



Slika 8.3. Struktura U/I međusklopa kod programski upravljano U/I prijenosu.

### 8.2.1 Sklopovska podrška programskom U/I prijenosu

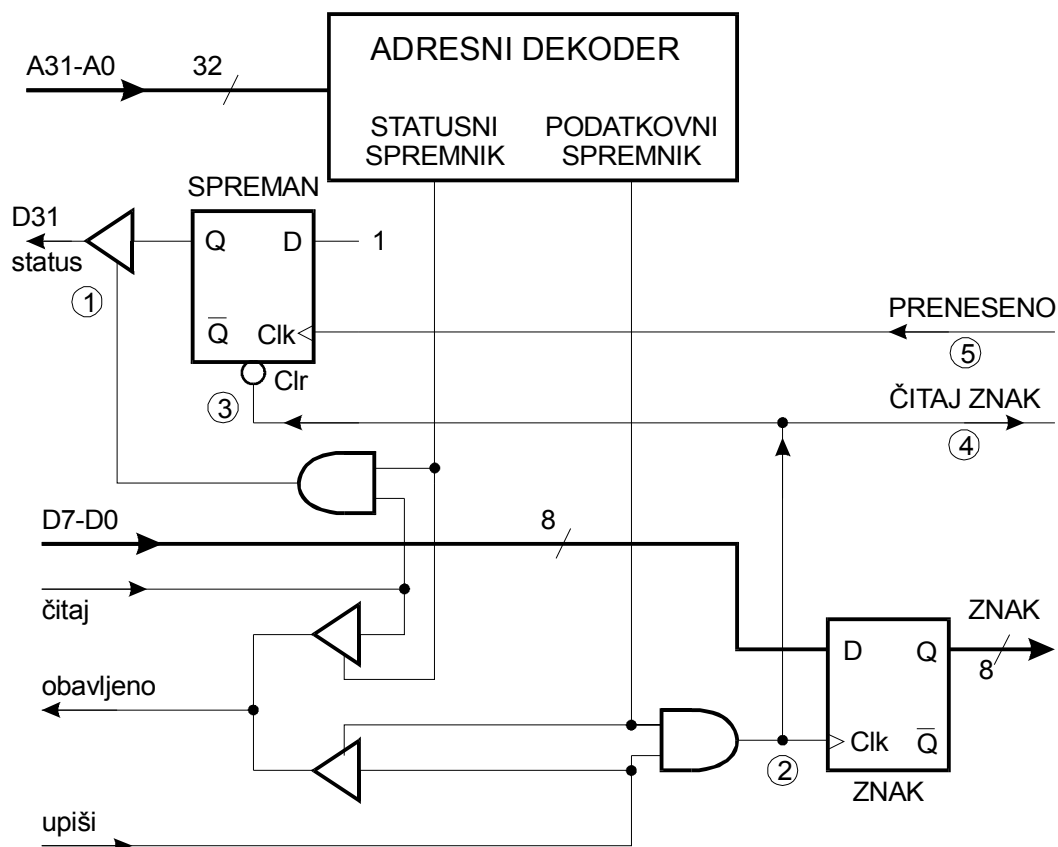
Sklopovska podrška programskom U/I prijenosu opisati će se na primjeru SRC-a razmatranog u prethodnim poglavljima. Može se podsjetiti da je SRC koristio 32-bitovnu adresnu sabirnicu kojom je moguće adresirati 4 Gokteta odnosno 1G 32-bitovna riječ. Neka se posljednjih 4 kotteta memorijskog prostora, adrese od FFFFF000H do FFFFFFFFH dodijele U/I adresnom prostoru. Poseban sklop, adresni dekoder razlučuje temeljem 20 adresnih linija većeg značenja da li se pristupa memoriji ili U/I međusklopu. Spremnici U/I međusklopova adresiraju se pomoću 12 adresnih linija manjeg značenja. Adresni dekoder prikazan je na slici 8.4.



Slika 8.4 Adresni dekodir U/I međusklopa.

Adresnim linijama A12-A31 odabire se U/I adresni prostor. Sklopnici postavljaju spremnike U/I međusklopa na odgovarajuću adresu u U/I adresnom prostoru dok adresna linija A0 odabire da li se radi o statusnom ili podatkovnom spremniku. Tako npr. ako se želi U/I međusklop postaviti na početak U/I adresnog prostora postaviti će se svi sklopnici u položaj 0. Statusni spremnik je na adresi FFFFF000H, a podatkovni na FFFFF001H.

**Sklopovlje za ispis znaka.** Kao konkretan primjer razvit će se U/I međusklop za predaju jednog znaka npr. pisaču ili terminalu za razmatrani SRC. Osnova ovog prijenosa je da procesor može upisati podatak u međusklop pisača samo kada je isti prazan. Zato je potrebno uz spremnik za predaju podatka imati i statusni spremnik, dovoljan je samo jedan bit, koji dojavljuje procesoru da je pisač podatak ispisao te da može prihvatiti novi podatak. Sklopovlje ovog međusklopa prikazano je na slici 8.5.



Slika 8.5 Međusklop za predaju znaka U/I uređaju.

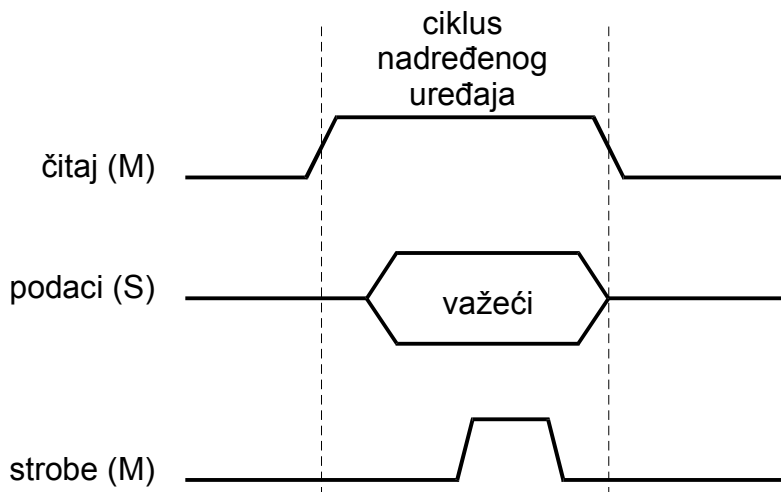
Prijenos znaka započinje čitanjem statusa međusklopa postavljanjem adrese statusnog spremnika i signala čitač (*read*) ①. Ukoliko je 31 bit prenesenog podatka jednak 1, odnosno podataka je negativan, tada je U/I uređaj spreman za prijenos. Procesor tada upisuje znak u međuspremnik ZNAK U/I međusklopa generiranjem adrese podatkovnog spremnika i signala upiši (*write*) ②. Ovi signali ujedno postavljaju spremnik SPREMAN u 0 ③. Time je signalizirano da U/I uređaj nije pročitao podatak te da prijenos novog znaka nije moguć. Također ovaj signal signalizira i U/I uređaju da je podatak upisan u spremnik ZNAK te da ga može pročitati ④. Kada je znak prenesen (pročitao) U/I uređaj to signalizira U/I međusklopu aktiviranjem signala PRENESENO koji ujedno postavlja izlaz spremnika SPREMAN u 1 ⑤ signalizirajući procesoru da je U/I uređaj spreman za prijenos novog znaka.

Procesor mora držati aktivne sve signale na sabirnici (adrese, čitač ili piši, podatak) dok U/I međusklop ili ne upiše znak ili postavi izlaz STATUS spremnika na podatkovnu sabirnicu. Kako adresni dekodier U/I međusklopa može imati relativno veliko vrijeme odziva tako je potrebno dojaviti procesoru kada je dekodiranje adresa završeno tako da on može završiti sa započetim ciklusom i preći na novi. U tu svrhu predviđen signal obavljeno (*complete*) kojim se procesoru dojavljuje da je U/I međusklop završio operaciju. Ovom linijom ostvaruje se sinkronizacija sklopova s različitim vremenima odziva.

**Sinkronizacija prijenosa podatka.** Prethodna razmatranja otvorila su problematiku sinkronizacije prijenosa podataka između komponenata sustava

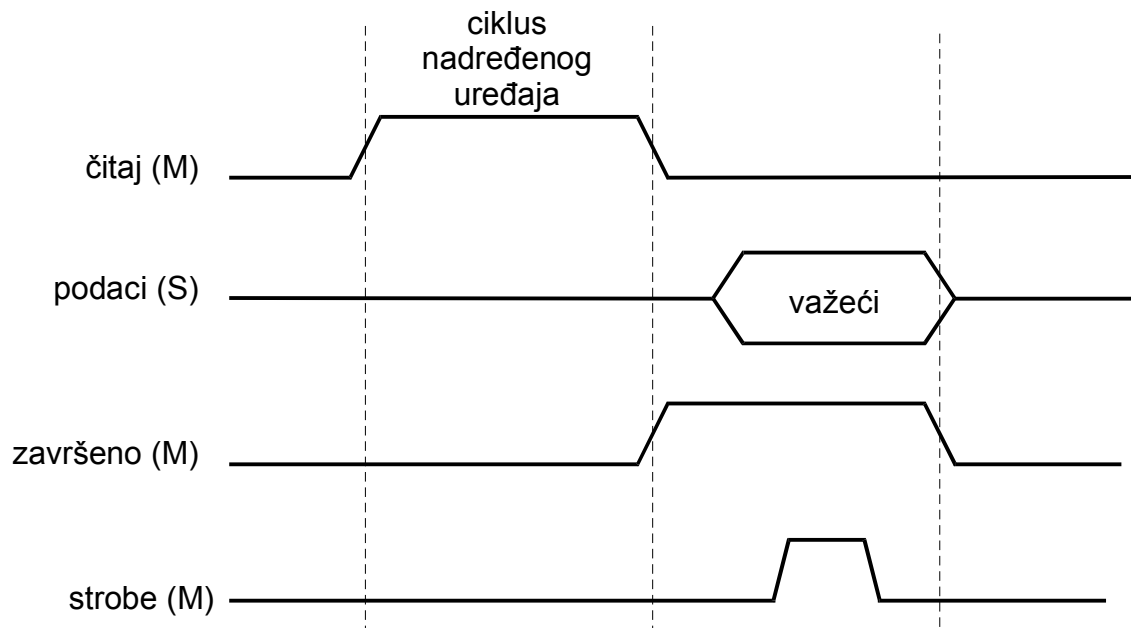
(spremnik-spremnik, spremnik-memorija, spremnik-U/I međusklop, itd.) Uređaj koji upravlja prijenosom naziva se *master*, a drugi uređaji koji sudjeluju u prijenosu *slave*. U osnovi razlikuju se tri slučaja.

*Sinkroni prijenos* koji podrazumijeva da podređeni uređaj (*slave* S) odgovara punom brzinom kojom može komunicirati nadređeni (*master* M). U ovom slučaju nisu potrebni dodatni signali za sinkronizaciju prijenosa (*handshaking*). Ovakav prijenos koristi se prilikom prijenosa podataka između spremnika procesora. Podatak se postavlja na linije na početku ciklusa, a upisuje se u odredište na kraju ciklusa. Vremenski dijagram ovakvog prijenosa prikazan je slikom 8.6 za slučaj kada se podatak iz *podređenog* prenosi nadređenom uređaju.



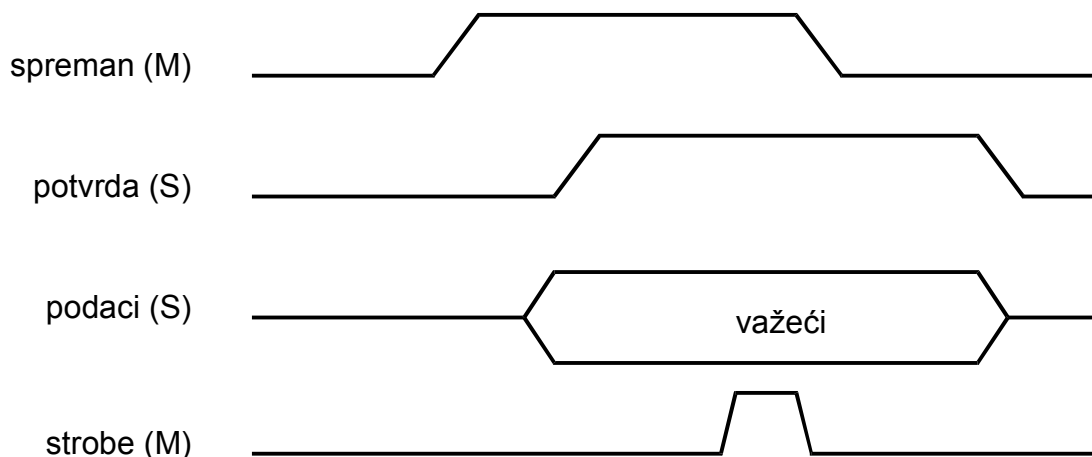
Slika 8.6 Sinkrono čitanje podataka.

*Semi-sinkroni prijenos* koristi se kada podređeni uređaj ima duže vrijeme odziva od nadređenog. Po završetku prijenosa podređeni uređaj generira signal kojim obavještava nadređeni da je prijenos obavljen. Signal koji generira podređeni uređaj sinkroniziran je s taktom nadređenog uređaja, slika 8.7. Ovakav način sinkronizacije uobičajen je između procesora i sporijih memorijskih modula.



Slika 8.7. Semi-sinkrono čitanje podatka.

*Asinkroni prijenos* pretpostavlja da podređenom uređaju je potrebno izvjesno vrijeme da prepozna da je postavljen zahtjev za prijenosom, a ne samo da odgovori na njega. Ovo posebice vrijedi za U/I prijenos gdje su kašnjenja preko sabirnice kao i vrijeme odziva sklopovlja U/I međusklopa značajno različita od brzine rada procesora. Asinkroni čitanje s podređenog uređaja može se opisati sljedećim signalima: 0-1 prijelaz signala "spreman" (*ready*) koji znači "Možeš li mi poslati podatak?". Sljedeći signal je prijelaz 0-1 signala "potvrda" (*acknowledge*) kojim podređeni uređaj signalizira: "Da možeš, podaci na sabirnici su valjani.". U trećem koraku nadređeni uređaj upisuje podatak signalom "strobe" i dojavljuje: "Primio sam podatak." postavljanje signala "spreman" iz 1-0. Podređeni uređaj odgovara: "Primio sam da je prijenos završen." postavljajući signal "potvrda" iz 1-0. Vremenski dijagram asinkronog čitanja prikazan je slikom 8.8.



Slika 8.8 Asinkrono čitanje podatka.



Asinkrono pisanje ima sličan vremenski dijagram uz razliku što nadređeni uređaj istovremeno sa signalom “spreman” postavlja podatke na sabirnicu. Signal “spreman” deaktivira se kada podređeni uređaj pročita podatak i isto dojavljuje postavljajući signal “potvrda” u 1.

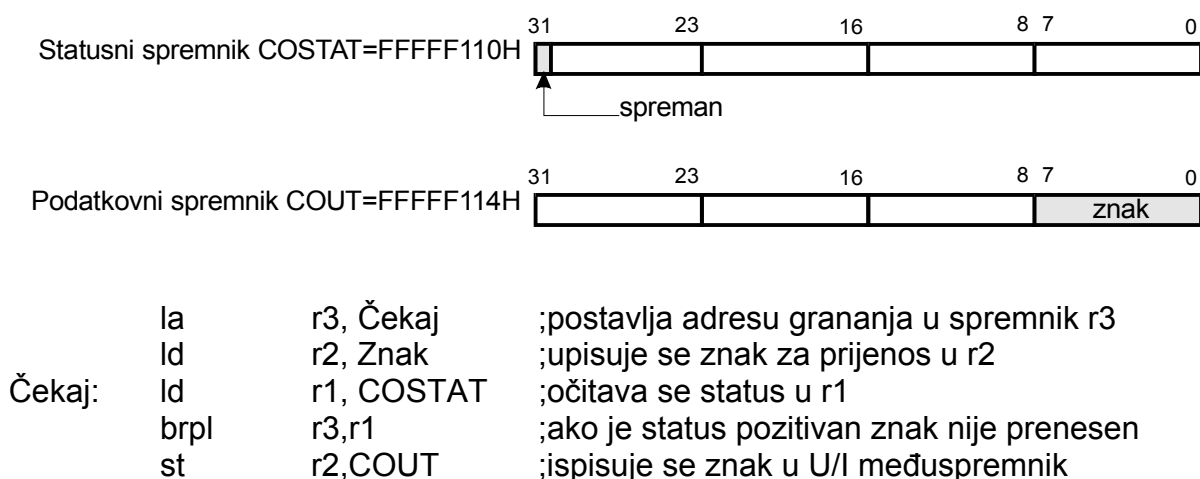
Signali “spreman” i “potvrda” kao i protokol njihovog korištenja poznati su pod nazivom sklopovsko rukovanje (*hardware handshaking*).

Konačno, može se zaključiti kako se sinkroni i semi-sinkroni prijenos koriste kod uređaja koji nemaju međusobnog vremenskog kašnjenja ili je ono relativno malo, dok se asinkrono koristi kada je vrijeme odziva relativno duže ili promjenjivo.

### 8.2.2 Programska podrška programiranom U/I prijenosu

Opisani sklopovi koriste se uz odgovarajuću programsku podršku. Ona upravlja prijenosom upravljačkih signala i podataka između procesora i U/I uređaja. Sadržava inicijalizaciju, obavljanje i okončanje prijenosa, kao i obradu posebnih stanja kao što je npr. nestanak papira kod pisača, disketa nije umetnuta, zauzeta linija kod modema i sl. Ovi programski moduli nazivaju se pogonski moduli ili tzv. *device driver* ili *device handler*.

Kao primjer napisati će se programska procedura koja šalje znak na U/I međusklop prikazan na slici 8.5. Adrese spremnika prikazane su na slici 8.9.



Slika 8.9. Programska procedura za prijenos znaka na U/I međusklop.

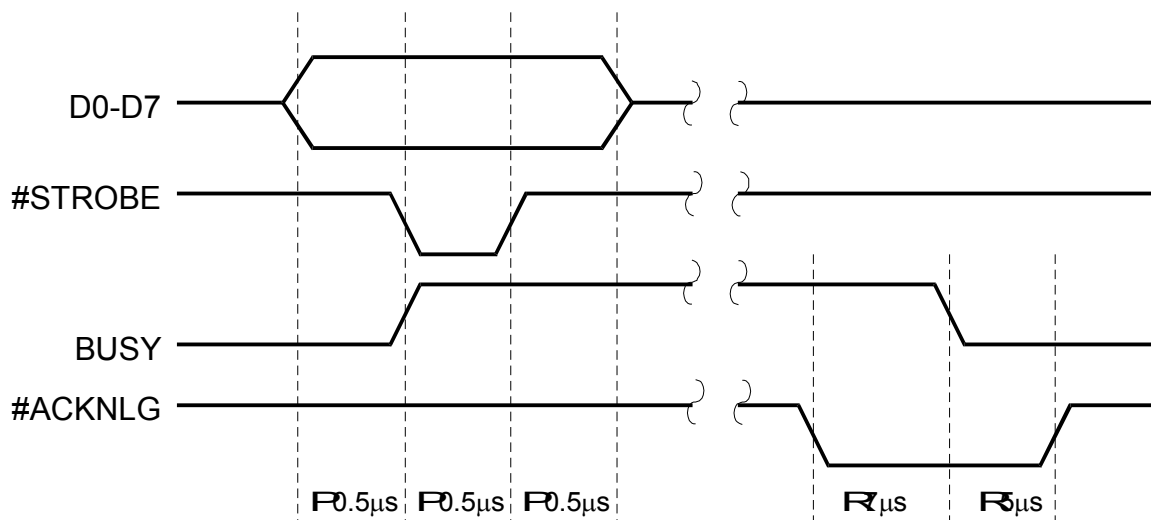
Procesor nakon što unese u spremnike r2 i r3 podatke neophodne za obavljanje U/I operacije ispituje stanje međusklopa. Kada je U/I međusklop spreman za prijenos, status je negativan, procesor predaje znak U/I međusklopu. Ukoliko je međusklop namijenjen za predaju znaka pisaču tada je potrebno i nekoliko ms da pisač ispiše znak te da bude spreman za prijem sljedećeg znaka. Za to vrijeme procesor čeka ispitujući status U/I međusklopa što iznosi desetke tisuća ciklusa. Iako ispituje status U/I međusklopa procesor praktički ne radi nikakav koristan posao, odnosno on aktivno čeka da obavi U/I operaciju.

Ukoliko je više U/I međusklopova spojeno na sabirnicu svaki statusni, upravljački i podatkovni spremnik dobiva adresu. Program ispituje statusne spremnike prema određenom redoslijedu, prioritetima ili ciklički, te određuje koji od njih zahtjeva prijenos podatka. Ovo rješenje naziva se prozivanje ili *pooling*.

Ovo podpoglavlje zaključiti će se s opisom paralelnog međusklopa Centronics koji se koristi kod IBM PC kompatibilnih računala sa spajanje vanjskih uređaja. Ovaj međusklop primarno je razvijen za Centronics pisače ali je uskoro prihvaćen kao standard za uređaje koji koriste paralelan jednosmjerni prijenos podataka. Popis signala prikazan je sljedećom tablicom:

Naziv signala	Smjer	Opis
#STROBE	Izlazni	Podaci postavljeni
D0	Izlazni	LSB podatka
D1	Izlazni	2. bit podatka
D2	Izlazni	3. bit podatka
D3	Izlazni	4. bit podatka
D4	Izlazni	5. bit podatka
D5	Izlazni	6. bit podatka
D6	Izlazni	7. bit podatka
D7	Izlazni	MSB podatka
#ACKNLG	Ulazni	Podatak pročitao
BUSY	Ulazni	Uređaj nije spreman
#PE	Ulazni	Nema papira
SLCT	Ulazni	Postavljen u 1, uređaj prisutan
#AUTO FEED XT	Izlazni	Automatski nova linija
#INIT	Izlazni	Inicijalizacija pisača
#ERROR	Ulazni	Pogreška
#SLCT IN	Izlazni	Deselektiraj protokol

Za spajanje U/I uređaja koristi se 36-pinski konektor. Sadrži osam linija za podatke standardne signale asinkrone komunikacije #STROBE i #ACKNLG, tri statusna izlaza s pisača i tri ulaza. Vremenski dijagram Centronics paralelnog međusklopa prikazan je slikom 8.10.



Slika 8.10. Vremenski dijagram Centronics paralelnog međusklopa.

### 8.3 Obavljanje U/I operacija preko zahtjeva za prekidom

Osnovni problem kod programskog obavljanja U/I operacija je što procesor aktivno čeka da U/I uređaj obavi operaciju. U tom vremenu on bi mogao raditi drugi koristan posao. Realizacija navedenog je moguća ukoliko bi U/I uređaj kada je spreman inicijalizirao U/I ciklus dojavom procesoru preko određene linije.

Detekcijom ovog signala procesor bi prema određenoj proceduri prekinuo izvođenje tekućeg programa i prebacio se na obavljanje U/I operacije. Nakon obavljene U/I operacije procesor bi nastavio s prekinutim programom. Ovakav način obavljanja U/I operacija naziva se obavljanje U/I operacija preko zahtjeva za prekidom (*I/O interrupt*).

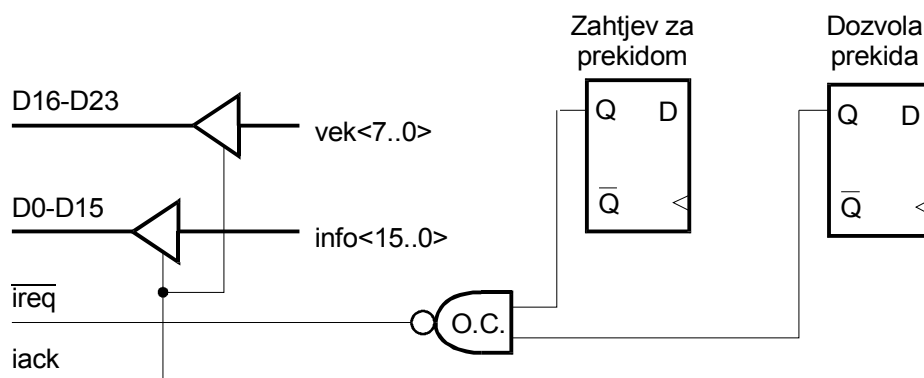
### 8.3.1 Sklopovska podrška obavljanju U/I operacija preko zahtjeva za prekidom

Zahtjev za prekidom koji U/I međusklop šalje procesoru te odgovor koji procesor šalje U/I međusklopu da prihvata zahtjev za prekidom zahtijevaju signalizaciju sličnu rukovanju kod programskog U/I prijenosa. Tako U/I međusklop postavlja zaseban signal zahtjev za prekidom (*interrupt request irq*). Primitkom ovog signala procesor obavi tekuću naredbu, te ako je prekid rada dozvoljen zapamti stanje programa kojeg izvodi i dojavljuje U/I međusklopu da je spreman za obradu prekida (*interrupt acknowledge iack*).

U stvarnom računalnom sustavu spojeno je na procesor više U/I međusklopova. Zbog uštede u broju nožica procesor obično ima samo jedan ulaz za zahtjev za prekidom. Tako se svi zahtjevi za prekidom U/I međusklopova spajaju paralelno na ulaz za prekidom procesora obično preko vezane ILI logike (*wired OR*). Iz tog razloga zahtjev za prekidom obično je aktivan u logičkoj 0, a izlaz međusklopa koji postavlja zahtjev za prekidom je sklop s otvorenim kolektorom. Navedeno rješenje otvara dodatne probleme. Naime, potrebno je da procesor odredi koji od U/I međusklopova je postavio zahtjev za prekidom.

Moguće rješenje je da procesor proziva sve U/I međusklopove, ispituje njihov statusni spremnik i temeljem njegovog sadržaja odredi koji U/I međusklop je postavio zahtjev za prekidom. Program za obradu prekida sastoji se od jedinstvene procedure koja određuje koji je uređaj postavio zahtjev za prekidom, uspoređuje zahtjev s tekućim prioritetom te se prema potrebi grana u proceduru za obradu konkretne U/I operacije.

Drugo rješenje je da U/I međusklop kada primi potvrdu da je zahtjev za prekidom prihvaćen sam pošalje određeni identifikacijski broj preko podatkovne sabirnice koje procesor čita u istom ciklusu. Identifikacijski broj može se sastojati od dva dijela: prvi koji specificira tip uređaja (pisač, tipkovnica, i sl.), a drugi koji daje dopunske informacije o stanju uređaja. Opisano rješenje zahtjeva dopunsku sklopovsku podršku koja je prikazana slikom 8.12.

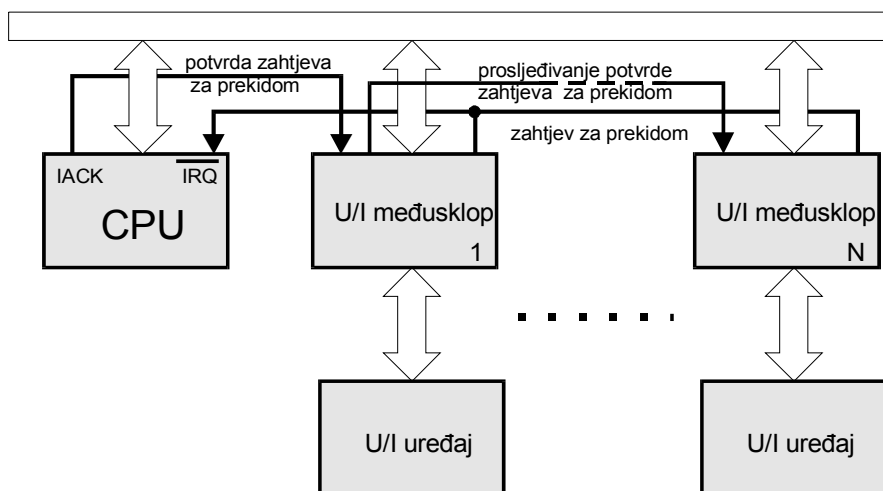


Slika 8.11. Pojednostavljeno sklopovlje za postavljanje zahtjeva za prekidom.

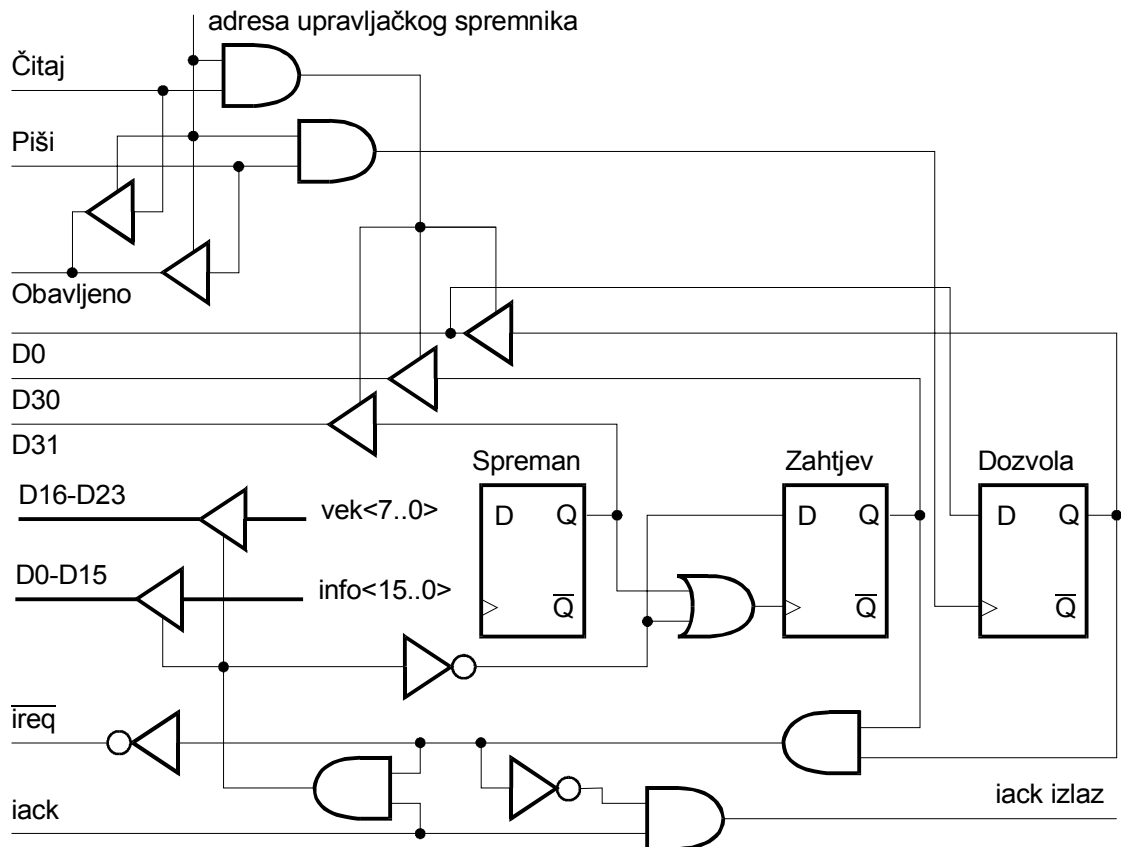
U/I uređaj postavlja zahtjev za prekidom postavljajući izlaz bistabila “zahtjev za prekidom” u 1. Ukoliko je prekidanje dozvoljeno, izlaz bistabila “dozvola prekida” je u 1, tada se preko I sklopa s koji ima izlaz s otvorenim kolektorom postavlja procesoru zahtjev za prekidom. Kada procesor potvrdi prihvaćanje zahtjeva za prekidom, postavljanjem linije iack u jedinicu na podatkovnu sabirnicu postavlja se vektor i informacija o prekidu koje procesor u istom ciklusu očitava. Stanje na adresnoj sabirnici ovdje nema utjecaja budući procesor ne zna koji je U/I međusklop postavio zahtjev za prekidom.

Spajanje više U/I međusklopova vezanom ILI logikom na ulaz za postavljanja zahtjeva za prekidom procesora omogućava da više U/I međusklopova istovremeno postave zahtjev za prekidom. U tom slučaju ukoliko bi svi ulazi za potvrdu prihvata prekida, iack, U/I međusklopova bili spojeni paralelno s istim izlazom procesora više bi U/I međusklopova postavilo istovremeno na podatkovnu sabirnicu informacije o prekidu. Zato je potrebno uvesti određenu logiku koja će razriješiti ovaj konflikt i potvrditi prihvrat prekida samo U/I uređaju koji ima veći prioritet. Sklopovski se navedeno može riješiti serijskim spajanjem iack ulaza U/I međusklopova, slika 8.12, uz odgovarajuću sklopovsku podršku. Signal prihvata zahtjeva za prekidom proslijeđuje se s jednog na drugi U/I međusklop sve dok se ne dođe do sklopa koji je postavio zahtjev za prekidom.

Sklopovlje koje obavlja spomenutu funkciju prikazano je slikom 8.13. Sklop omogućava upis u spremnik dozvola, upravljački spremnik, preko D0 linije, odnosno programsku inicijalizaciju U/I međusklopa. Također, moguće je očitati upravljački spremnik, dozvola D0, zahtjev D30 i spreman D31.



Slika 8.12. Spajanje više U/I međusklopova s procesorom.



Slika 8.13. Prekidna logika U/I međusklopa.

Ukoliko je U/I međusklop postavio zahtjev za prekidom, a taj zahtjev je potvrđen signalom iack, na podatkovnu sabirnicu se postavlja prekidni vektor i informacija o prekidu. Prekidni vektor i informacija o prekidu može biti fiksno postavljena prespojnicima (*jumpers, DIP-switch*) ili se može u tu svrhu koristiti zaseban spremnik na određenoj U/I adresi u koji procesor upisuje vrijednost prilikom inicijalizacije U/I međusklopa. Posljednje rješenje je nešto složenija ali osigurava veću fleksibilnost sustava. Ukoliko ovaj sklop nije postavio zahtjev za prekidom tada se signal iack proslijeđuje sljedećem sklopu.

### 8.3.2 Programska podrška prekidnom U/I prijenosu podataka

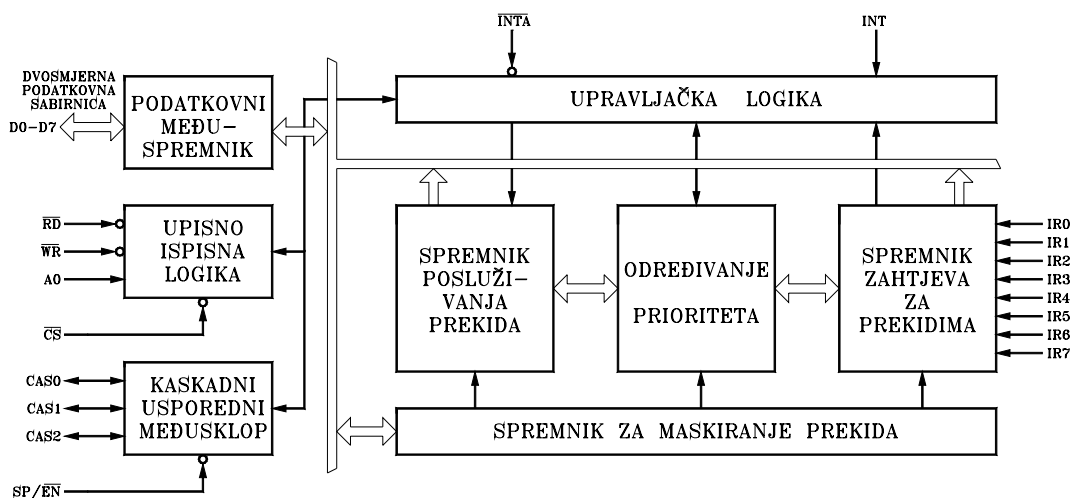
Obavljanje U/I operacija pomoću zahtjeva za prekidom zahtijeva određenu programsku podršku kao i mehanizam prekida rada procesora. Procesor ispituje irq ulaz na početku svakog ciklusa. Time je osigurano da se naredba koja je trenutno u procesu izvođenja izvede do kraja. Ukoliko je postavljen zahtjev za prekidom i dozvoljeno je prekidanje rada procesora tada procesor može prekinuti izvođenje tekućeg programa i preći na obradu prekida. Da bi nakon obrade prekida procesor mogao nesmetano nastaviti s izvođenjem prekinutog programa potrebno je zapamtiti stanje u kojem je program prekinut. To minimalno podrazumijeva pamćenje koja je sljedeća naredba koja se mora izvesti, odnosno sadržaj programskog brojila, kao i stanje procesora koji je obično upisano u statusnom spremniku. Ove informacije se automatizmom, sklopovski pohranjuju obično u memoriju, odnosno stavljaju se na stog. Slijedi generiranje iack signala i prihvata vektora i informacije o prekidu. Obično ova informacija predstavlja adresu

na kojoj se nalazi upisana adresa prve naredbe procedure za obradu tog prekida. To praktično znači da se s adrese na koju pokazuje vektor i informacija o prekidu sadržaj upisuje u programsko brojilo. Sve navedene aktivnosti provode se sklopovski, a tijekom njihovog izvođenja novi zahtjevi za prekidom nisu dozvoljeni. Kod procesora sa cjevovodom situacija je znatno složenija i ovdje neće biti opisivana.

Procedura za obradu prekida koristi neke spremnike procesora. U tim spremnicima mogu se nalaziti podaci koje koristi prekinuti program. Zato je potrebno sačuvati njihov sadržaj stavljajući ga na stog. Nakon toga pristupa se izvođenju U/I operacije. Po završetku U/I operacije obnavlja se sadržaj spremnika sa stoga. Slijedi povratak u prekinuti program izvođenjem posebne naredbe "povrat iz prekida" (reti, rfi i sl.). Ova naredba sa stoga obnavlja sadržaj statusnog spremnika i programskog brojila.

### Sklop za upravljanje prekidima Intel 8259

Za upravljanje prekidima koristi se posebni sklop 8259 (Interrupt controller). Njegova uloga je da povezuje mikroprocesor s 8 prekidnih linija. Ima svojstvo da zapamti na kojoj je liniji postavljen zahtjev za prekidom, da provjeri da li je taj prekid maskiran ili ne, zatim da uspoređi da li je prekid najveće razine, pa tek zatim da izvesti procesor da postoji zahtjev za prekidom. Kada procesor odobri prekid, preko INTA linije se obavijesti sklop 8259, koji zatim na podatkovnu sabirnicu postavlja podatak koji je ulaz postavio zahtjev za prekidom. Sklop za obradu prekida 8259 prikazan je na sljedećoj slici:



**Slika 1.:** Blokovska shema sklopa za obradu prekida 8259.



G1	G2A	G2B	C	A	B	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
1	0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	1	1	1	0	1	1
1	0	0	1	1	0	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0
0	x	x	x	x	x	1	1	1	1	1	1	1	1
x	1	x	x	x	x	1	1	1	1	1	1	1	1
x	x	1	x	x	x	1	1	1	1	1	1	1	1

**Tablica 1.:** Tablica istine sklopa 74xx138.

Prema ovoj tablici i slici 2. može se zaključiti da će sklop 8259 biti odabran (selektiran) kada je ulaz CS u logičkoj nuli, odnosno kada je izlaz Y1 aktivan, a on će biti aktivan ako je: A5 = 1, A6 = 0, A7 = 0, A8 = 0, A9 = 0. Ako se zanemare A0-A4, ovo je ulazno/izlazna adresa 0020H. Aktiviranjem/deaktiviranjem linije A0 odabire se jedan od unutrašnjih spremnika sklopa za potrebe inicijalizacije.

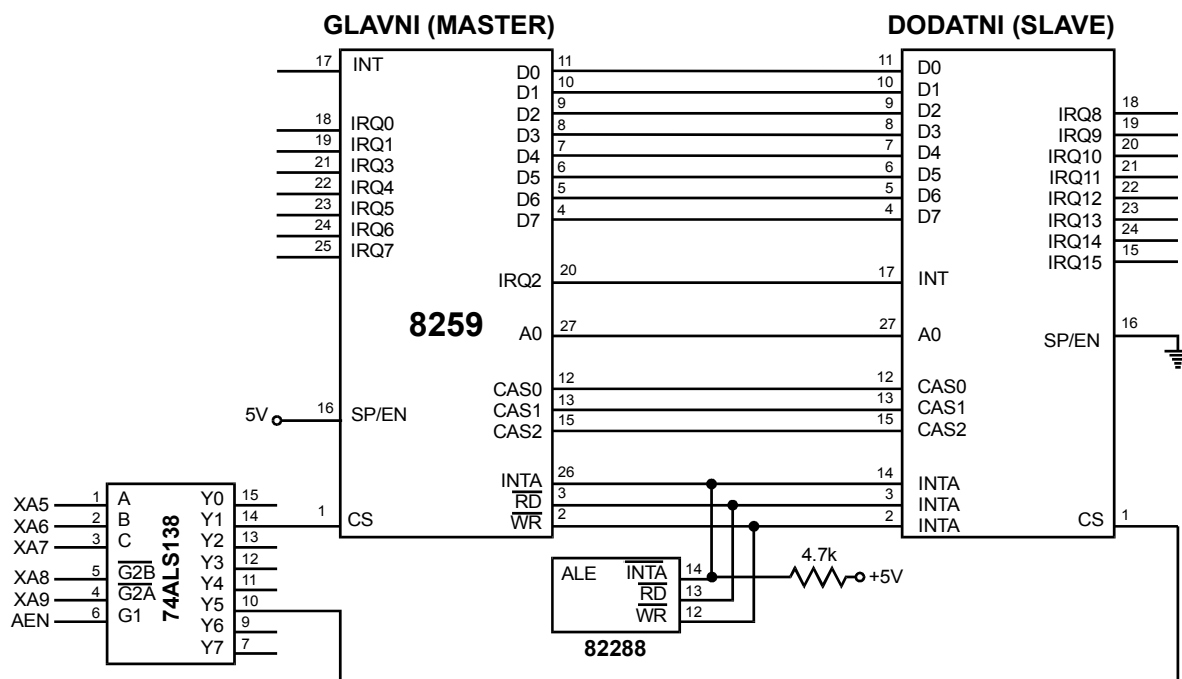
U trenutku kada je postavljen zahtjev za prekidom, te je isti i odobren od procesora aktiviranjem linije INTA preko sklopa za upravljenje sabirnicom, isti sklop postavlja IOWC signal u nisku razinu nakon čega 8259 postavlja na sabirnicu podataka (linije D0-D2) podatak o ulazu na kojem se pojavio prekid:

D2	D1	D0	prekid
0	0	0	IRQ0
0	0	1	IRQ1
0	1	0	IRQ2
0	1	1	IRQ3
1	0	0	IRQ4
1	0	1	IRQ5
1	1	0	IRQ6
1	1	1	IRQ7

Signali CAS0-CAS2 namjenjeni su za kaskadno povezivanje više sklopova za upravljanje prekidima. Glavni (master) sklop za upravljanje sabirnicom ima postavljen ulaz SP/EN u logičku jedinicu, a dodatni (slave) ima postavljenu istu liniju u logičku nulu. Zahtjev za prekid dodatnog sklopa prenosi se glavnom preko IRQ2 ulaza, te ako je mikroprocesor (INTA) a zatim i glavni sklop odobrio prekid IRQ2 tada je moguće posluživati prekide IRQ8-IRQ15. Dodatni sklop koristi linije CAS0-CAS2 da bi preko glavnog sklopa dojavio procesoru o kojem se prekidu radi. Praktički glavni sklop stanja na linijama CAS0-CAS2 prosljeđuje na linije DB0-DB2.

CAS2	CAS1	CAS0	prekid
0	0	0	IRQ8
0	0	1	IRQ9
0	1	0	IRQ10
0	1	1	IRQ11
1	0	0	IRQ12
1	0	1	IRQ13
1	1	0	IRQ14
1	1	1	IRQ15





**Slika 3.: Kaskadno spajanje dvaju sklopova 8259.**

Prekid	Uređaj	Adresa	Br. Vekt.
NMI	Greška u paritetu	08-0B	02
IRQ0	Timer	20-23	08
IRQ1	Tipkovnica	24-27	09
IRQ2	Dodatni 8259	28-2B	0A
IRQ3	Serijski ulaz 2 (COM2)	2C-2F	0B
IRQ4	Serijski ulaz 1 (COM1)	30-33	0C
IRQ5	Paralelni ulaz 2 (LPT2)	34-37	0D
IRQ6	Disketa	38-3B	0E
IRQ7	Paralelni ulaz 1 (LPT1)	3C-3F	0F
IRQ8	Sat realnog vremena	1C0-1C3	70
IRQ9	Video	1C4-1C7	71
IRQ10	Rezervirano	1C8-1CB	72
IRQ11	Rezervirano	1CC-1CF	73
IRQ12	Miš	1D0-1D3	74
IRQ13	Koprocetor	1D4-1D7	75
IRQ14	Disk	1D8-1DB	76
IRQ15	Rezervirano	1DC-1DF	77

## 8.4 Direktan pristup memoriji (DMA)

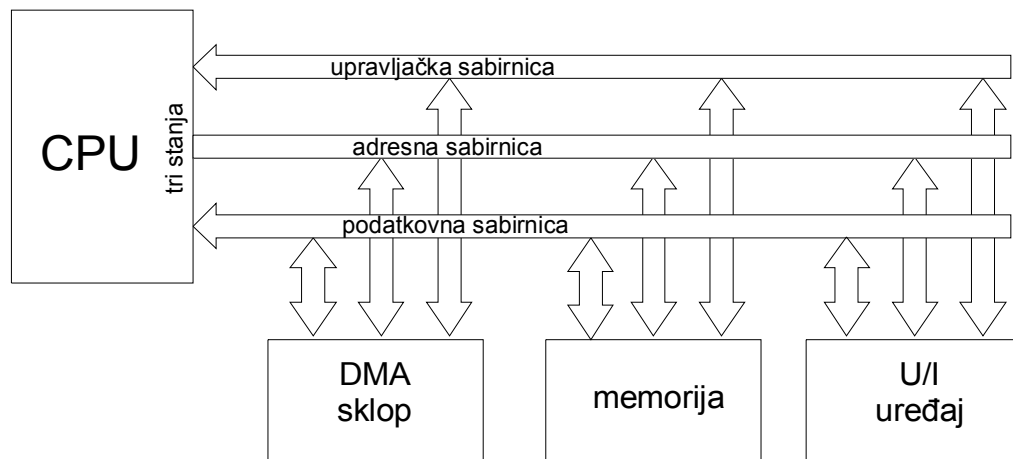
Prekidnom ulazno/izlaznom prijenosu podataka, u zavisnosti od broja vanjskih logičkih sklopova priključenih na istu prekidnu liniju te o odabranoj metodi za otkrivanje uzroka prekida, potrebno je vrijeme za posluživanje prekida, odnosno vrijeme za prijenos jednog elementa iz bloka podataka, od nekoliko  $\mu s$  do nekoliko stotina  $\mu s$ . To znači da ovaj oblik prijenosa omogućava brzinu prijenosa do nekoliko stotina tisuća podataka u sekundi.

U slučaju učestalih zahtjeva za ulazno/izlaznim prijenosom, odnosno zahtjeva za prijenosom većeg bloka podataka za očekivati je znatno usporeno i otežano izvođenje glavnog programa.

Rješenje ovog problema je u povećanju brzine ulazno/izlaznog prijenosa podataka i oslobađanju procesora od tereta ulazno/izlaznog prijenosa. Rješenje je u

ostvarenju izravnog pristupa ulazno/izlaznog sklopa memoriji. Time se ostvaruje brz prijenos podataka između ulazno/izlaznog uređaja i memorije bez sudjelovanja procesora u prijenosu i bez izmjene sadržaja njegovih spremnika. Ovakvim načinom prijenosa ostvaruju se brzine veće od nekoliko milijuna podataka u sekundi.

Primjer uporabe izravnog pristupa memoriji je pri prijenosu podataka između diska i memorije, disketne jedinice i memorije i diska i disketne jedinice.



Na slici je prikazana izvedba izravnog pristupa memoriji. Kad ne postoji zahtjev za prijenos podataka između memorije i ulazno/izlaznog uređaja DMA controler je električki odspojen od sabirnica postavljanjem svojih izlaza u stanje velike impedancije (tri stanja). Tada procesor obavlja normalne memorijske cikluse. Pri izravnom pristupu memoriji DMA sklop priključuje se i upravlja sabirnicom, a procesor se električki odvaja od sabirnice postavljajući svoje izlaze u stanje velike impedancije. DMA sklop tada upravlja prijenosom podataka.

S obzirom na način realizacije izravnog pristupa memoriji razlikuje se sljedeće tehnike:

- ❑ izravni pristup memoriji zaustavljanjem procesora,
- ❑ izravni pristup memoriji krađom procesorskih ciklusa.

Izravan pristup memoriji zaustavljanjem procesora najjednostavnija je metoda. Aktiviranjem upravljačkog signala od strane DMA sklopa procesor završava tekuću naredbu i postavlja adresnu svoje izlaze u stanje velike impedancije i predaje sabirnice upravljačkom sklopu DMA.

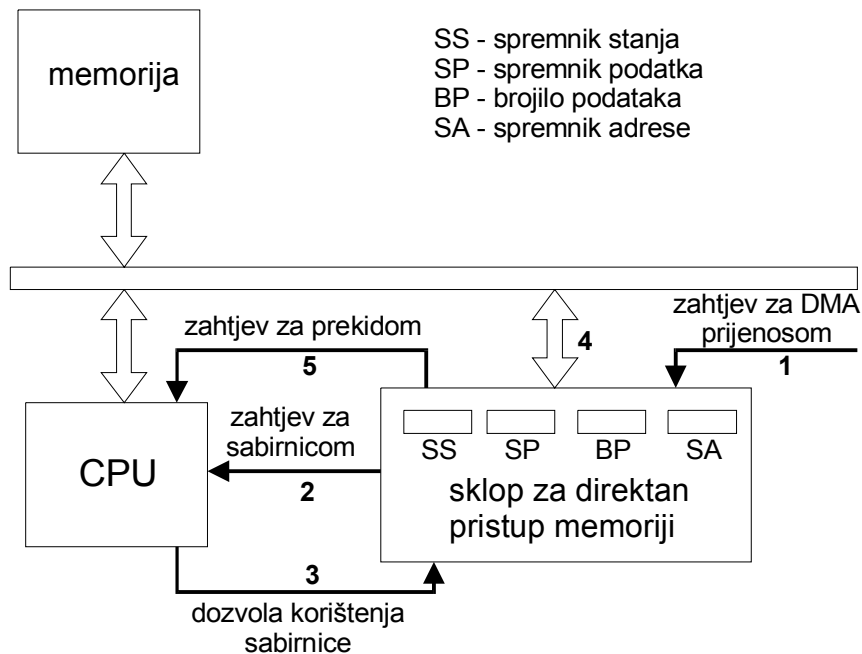
Metoda izravnog pristupa memoriji krađom procesorskih ciklusa ne zaustavlja procesor nego on samo kratkotrajno (nekoliko ciklusa) odgađa tok izvođenja naredbe, a u tom intervalu upravljački DMA sklop izvede prijenos jednog podatka. Ova metoda rezultira u povećanom vremenu obrade osnovnog programa.

Kako bi ostvario navedene funkcije upravljački DMA sklop mora ostvariti sljedeće funkcije:

- ❑ Preuzimanje upravljanja sabirnicom (podatkovnom, adresnom i upravljačkom),
- ❑ Izvoditi adresiranje memorije. Kako bi ovo ostvario DMA upravljački sklop mora imati slično procesoru adresni spremnik koji sadrži adresu sljedeće memorijske lokacije koja će sudjelovati u prijenosu.
- ❑ Brojanje prenesenih podataka. Blok podataka koji je potrebno prenijeti obično je unaprijed određene dužine. Za vrijeme prijenosa podataka broj prenesenih

podataka se mora uspoređivati sa željenom dužinom bloka. DMA prijenos završava kada je dostignuta željena duljina bloka.

- Izbor načina upravljanja. Status-upravljački spremnik upravljačkog DMA sklopa sadrži upravljačku riječ koja određuju smjer podataka pri DMA prijenosu, informaciju da li je upravljački DMA sklop aktivan, izbor tehnike prijenosa i slično.



Postupak pri DMA prijenosu može se podijeliti na sljedeće korake:

1. postavljanje zahtjeva za prijenos podataka od strane vanjskog logičkog sklopa,
2. procesoru se šalje zahtjev za dodjelu sabirnice,
3. procesor odgovara signalom potvrde prihvatanja zahtjeva,
4. adresiranje memorije i prijenos podatka između memorije i vanjskog logičkog sklopa,
5. dojava procesoru o završetku DMA prijenosa.

Primjer:

Potrebno je prenijeti N riječi iz memorije s početnom adresom M na određenu ulazno/izlaznu adresu.

procesor izvodi:

upiši u BP N  $\Rightarrow$  BP = N;  
 upiši u SA M  $\Rightarrow$  SA = M;  
 pokreni DMA prijenos;

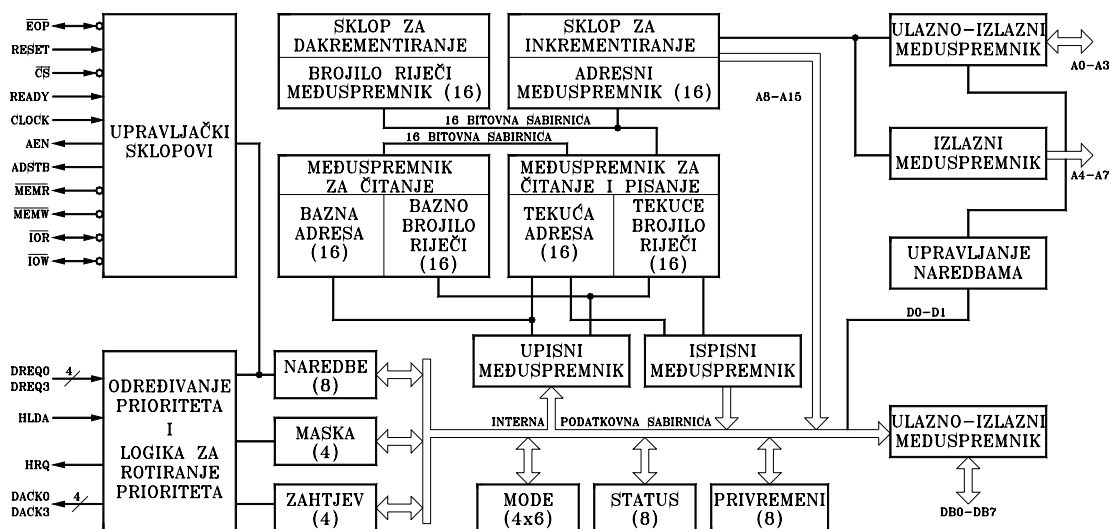
DMA upravljački sklop izvodi:

dok je BP  $\neq$  0

dohvati podatak s SA;  
 predaj podatak U/I sklopu;  
 SA = SA + 1;  
 BP = BP - 1;

izazovi prekid (prijenos završio);

Primjer ovakvog sklopa je Intel 8257 DMA upravljački sklop koji ima mogućnost upravljanje s četiri različita DMA kanala.



82C37 i poboljšana verzija 82C237 je DMA upravljački sklop namijenjen prijenosu podataka između ulazno/izlaznih uređaja i memorije te prijenosu blokova podataka između različitih memorijskih lokacija. Podaci se prenose direktno između ulazno/izlaznih uređaja i memorije bez upisivanja u međuspremnike, dok za pomicanje blokova podataka unutar memorije koriste se međuspremnici ovog sklopa.

Izlazi sklopa su izvedeni u logici s tri stanja te se on prikapča paralelno na sistemsku sabirnicu. Prijenos podataka može se započeti ili programski ili sklopovskim zahtjevom. Tako npr. ukoliko procesor treba prenijeti blok podataka iz memorije na disk prvo inicijalizira DMA kanal na koji je spojen međusklop diska (*disc controller*). Upisuje početnu adresu bloka podataka koji se prenosi kao i broj podataka koji se moraju prenijeti. U MODE spremnik upisuje se da se radi o prijenosu podataka iz memorije na ulazno/izlazni uređaj kao i vrsta prijenosa. Prijenos se započinje upisom zahtjeva o kojem se DMA kanalu radi u spremnik zahtjeva. DMA upravljački sklop zatraži od procesora pravo uporabe sabirnice aktiviranjem signala HRQ (*Hold Request*). Procesor ili sklop za upravljanje sabirnicom odgovara aktiviranjem signala HLDA (*Hold Acknowledge*). Jednom započet DMA prijenos nastavlja se postavljanjem memorijskih adresa na adresnu sabirnicu te istovremenim aktiviranjem linija  $\overline{MEMR}$  i  $\overline{IOW}$  te odabirom ulazno/izlaznog sklopa aktiviranjem linije DACK (*DMA acknowledge*). Podaci se prenose direktno iz memorije ulazno/izlaznom sklopu. Prijenos završava kad se prenese zadani broj podataka. Na sličan način ulazno/izlazni sklop preko linije DREQ (*DMA Request*) postavlja zahtjev za DMA prijenosom bloka podataka iz ulazno/izlaznog međusklopa u memoriju računala. DMA upravljački sklop može biti inicijaliziran da prenese blok podataka zadane veličine na određene lokacije u memoriji ili podaci se mogu prenositi dok ulazno/izlazni međusklop ne prekine prijenos aktiviranjem linije  $\overline{EOP}$  (*End Of Process*). Postupak izravnog prijenosa podataka iz ulazno/izlaznog međusklopa u memoriju identičan je kao i kod programski započetog prijenosa.