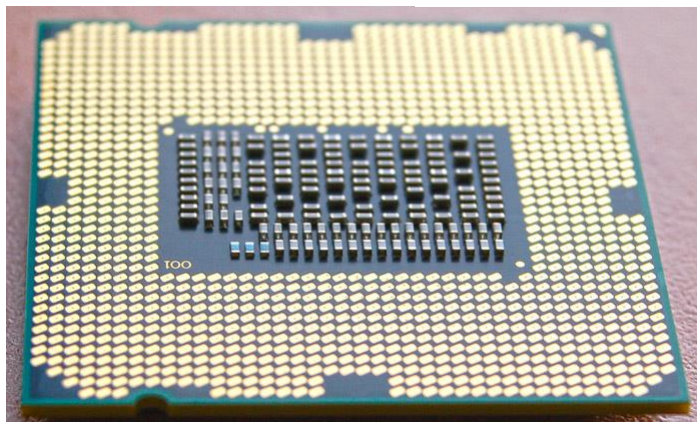
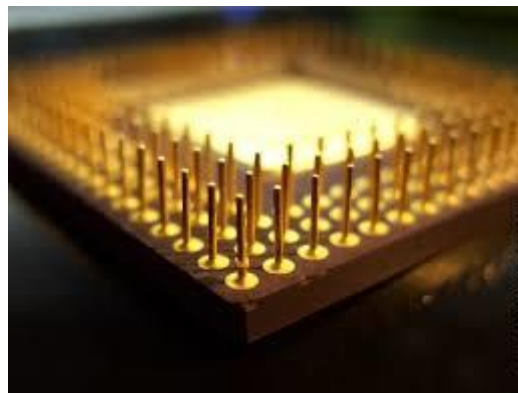
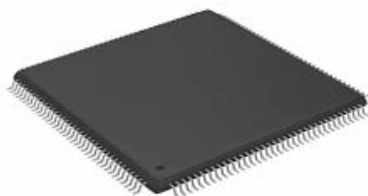


Priključci - općenito

Priključci

- Svaki procesor, memorija ili bilo koja druga komponenta ostvarena kao čip ima određen broj priključaka ili izvoda (nazivaju se još i pinovi prema engleskom izvorniku)



Priključci - podjela po namjeni

- Priključci su uobičajeno pridjeljeni pojedinim sabirnicama. Sabirnice se prema namjeni obično dijele na:
 - adresna sabirnica (address bus)
 - podatkovna sabirnica (data bus)
 - upravljačka sabirnica (control bus)
- Adresni priključci postavljaju adresu na adresnu sabirnicu, od procesora prema memoriji i vanjskim jedinicama. Procesor, kao aktivna i vodeća komponenta, adresira druge komponente zadajući im adresu s koje želi čitati ili pisati podatak
- Podatkovni priključci spojeni su na podatkovnu sabirnicu i služe za prijenos podataka između procesora i memorije ili između procesora i vanjske jedinice prilikom operacija čitanja i pisanja
- Upravljački priključci imaju razne funkcije vezane uz rad procesora i općenito služe za sinkronizaciju rada pojedinih dijelova računala

Priključci - podjela po namjeni

- Priključci koji imaju upravljačku namjenu (možemo promatrati kao da prenose logičko stanje true ili false):
 - mogu biti **aktivni u visokoj razini** i tada se nazivaju imenom bez dodatnih oznaka. Na primjer **READ** može označavati ciklus čitanja. Ako je priključak visoko, onda se trenutno obavlja ciklus čitanja, a u suprotnom se ne obavlja ciklus čitanja
 - mogu biti **aktivni u niskoj razini** i tada se označavaju imenom s „potezom”. Na primjer **IREQ** može označavati zahtjev za prekid. Ako je priključak nisko, onda znači da postoji zahtjev za prekid, a u suprotnom ne postoji.

>>>>

<<<<

- mogu označavati jedno od dva moguća stanja i tada u imenu sadrže nazive oba stanja. Na primjer **READ/WRITE** može označavati da li se trenutno izvodi ciklus čitanja ili pisanja. Ako je priključak nisko, onda se izvodi čitanje (**READ** ima potez). Ako je priključak visoko, onda se izvodi ciklus pisanja (**WRITE** nema potez)
- U pravilu, komponente koje osluškiju upravljačke priključke **aktiviraju se na brid signala** (tipično u trenutku kad signal prelazi iz neaktivnog u aktivno stanje)
- Kad se ne promatra brid, nego stanje signala, onda se stanje "očitava" u točno definiranim trenutcima

Priključci - "širina" priključaka

- Po širini priključci mogu biti:
 - Jednostruki priključci su oni koji imaju svoju samostalnu namjenu. To su obično upravljački priključci
 - Priključci grupirani u skupinu po nekoj zajedničkoj funkciji. Na primjer, mogu biti 32 adresna priključka, a pojedinačni se priključci označavaju nazivima ADR0, ADR1, ... ADR31

Priključci - smjer priključaka

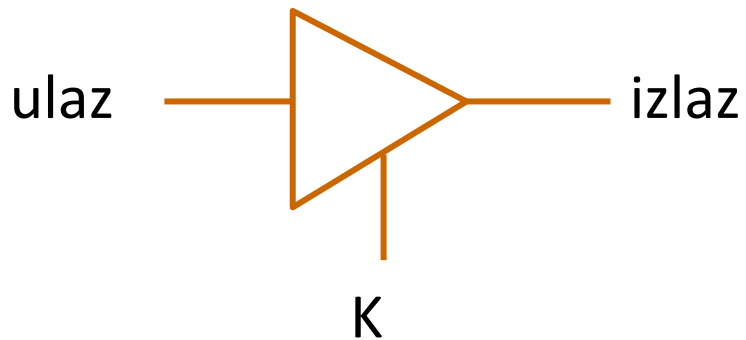
- Priključci se mogu podijeliti po smjeru signala (podataka) koji njima putuje na:
 - ulazne
 - izlazne
 - dvosmjerne
- Smjer priključka uvijek se definira u odnosu na komponentu koju promatramo

Priključci - smjer priključaka

- Komponenta upravlja svojim izlaznim priključcima, a druge komponente (spojene preko sabirnice na njih) "oslušuju" njihovo stanje
- Za ulazne priključke komponenta samo "oslušuje" stanje na njima. Sabirnicama koje su povezane na ove priključke upravljaju druge komponente
- Dvosmjerni priključci spojeni su na sabirnicu kojom u različitim trenutcima upravljaju različite komponente. Pri tome **uvijek samo jedna komponenta upravlja sabirnicom u nekom trenutku**, a priključci svih ostalih komponenata su ili **ulazni** ili u **stanju visoke impedancije**

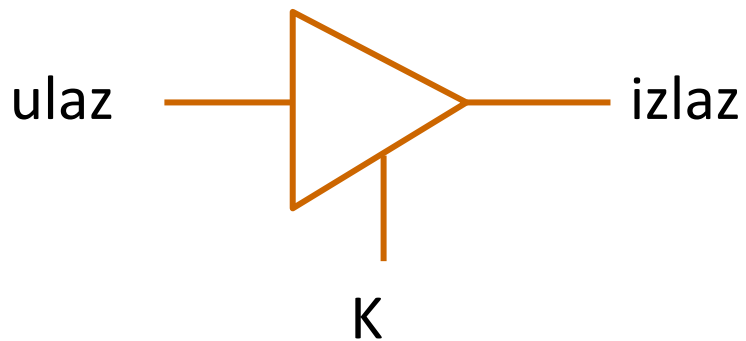
Priključci – priključci s tri stanja

- Stanje visoke impedancije omogućuje jednostavno spajanje više komponenata na istu sabirnicu (**ponoviti iz "Digitalne"**)
- Sklopove s tri stanja simbolički prikazujemo ovako:



Priključci – priključci s tri stanja

- Ako je upravljački ulaz K neaktivan (0), onda je izlaz u stanju visoke impedancije (high Z)
- Ako je upravljački ulaz K aktivan (1), onda je stanje ulaza prenosi na izlaz



K	ulaz izlaz		
0	0	Z	×
0	1	Z	×
1	0	0	→
1	1	1	→

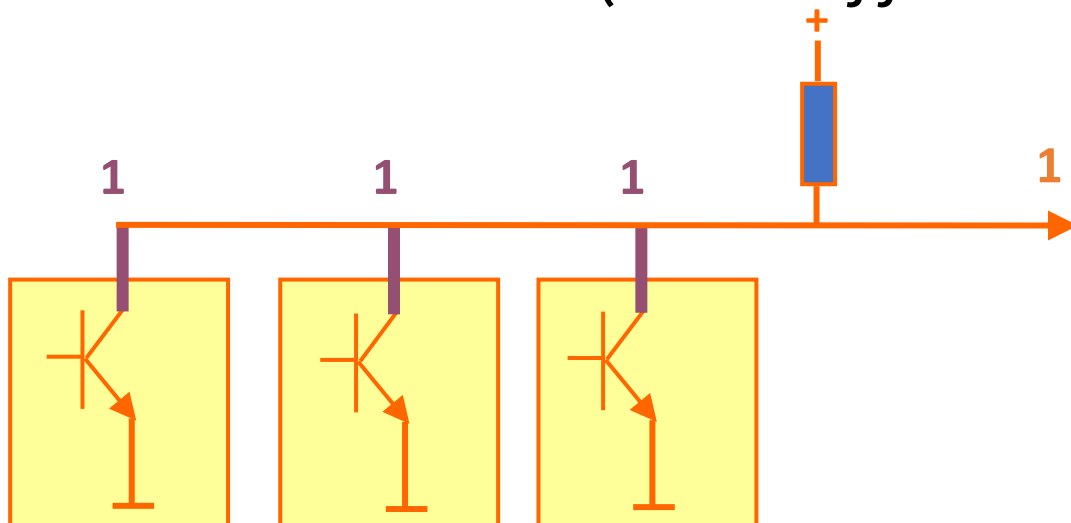
Priključci – priključci s tri stanja

- Sklop s tri stanja omogućuje spajanje više priključaka na istu sabirnicu
- Sabirnicom smije upravljati najviše jedan priključak i on određuje stanje sabirnice (0 ili 1)
- Svi ostali priključci moraju biti neaktivni i oni ne utječu na stanje na sabirnici

Priključci - spojeni I/ILI

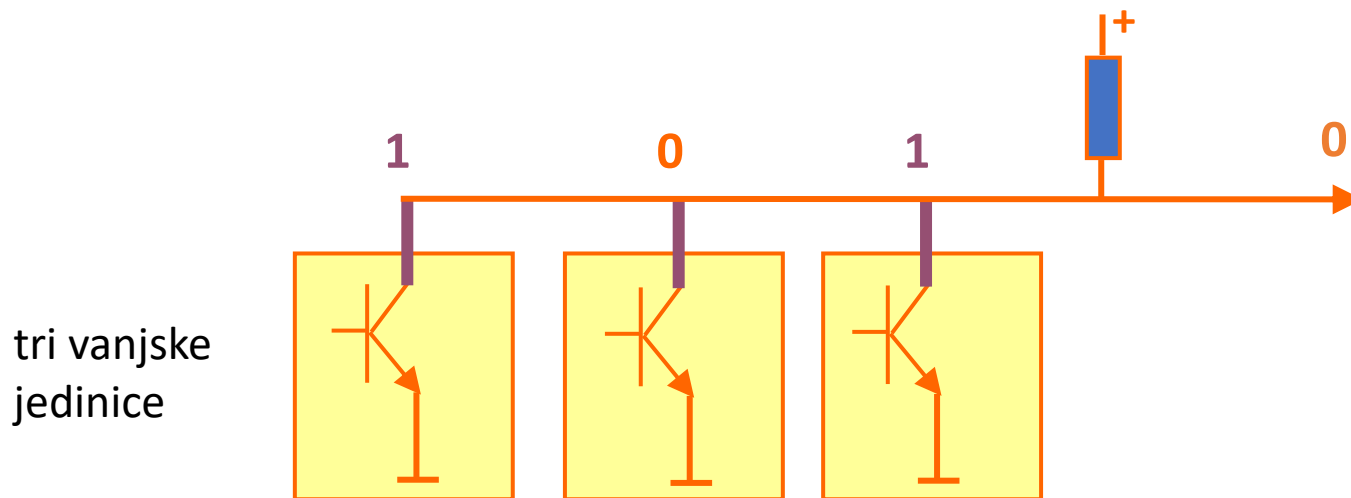
- Postoje i posebne vrste priključaka. To su tzv. *open collector* priključci (**ponoviti iz "Digitalne"**)
 - oni omogućuju da se **više izlaznih priključaka** spoji zajedno na jednu sabirnicu i da **svi zajedno njome upravljaju** u istom trenutku
 - stanje sabirnice određeno je logičkom funkcijom spojeni-I (wired-AND) između svih priključaka (naziva se još i wired-OR - za negativnu logiku)
- Ako su svi prekidni priključci **neaktivni** (u visokoj su razini), onda je cijela sabirnica **neaktivna** (u visokoj je razini)

tri vanjske
jedinice



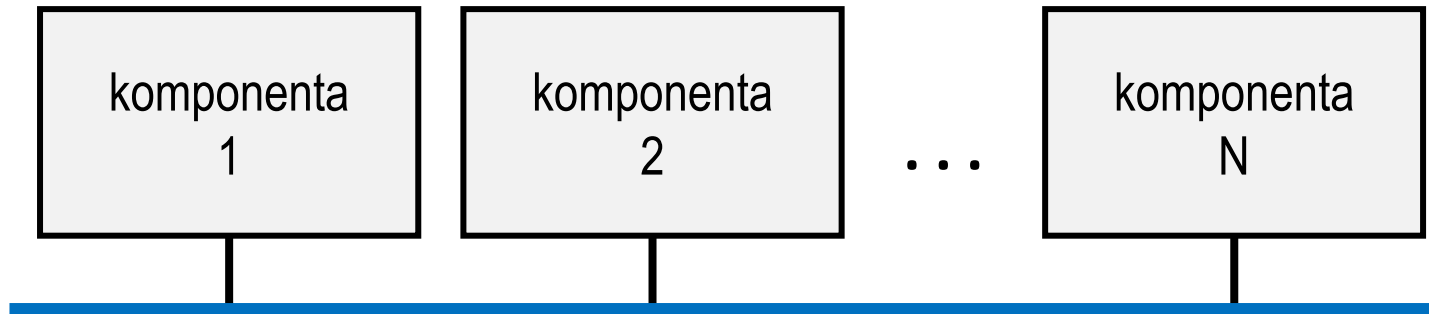
Priključci - spojeni I/ILI

- Ako se bilo koji prekidni priključak (ili više njih) **aktivira** (aktivna razina je u niskom), onda se **aktivira** cijela sabirnica, tj. prelazi u nisku razinu



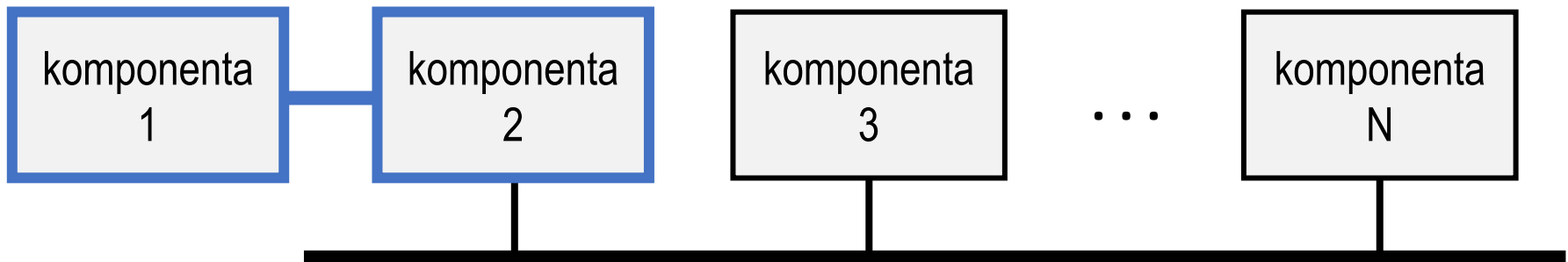
Sabirnice - Osnovno

- Sabirnica (*bus*) je spojni put koji povezuje više komponenata/djelova računalnog sustava, a sastoji se od skupa vodiča



- **Prednosti** sabirnice:
 - mala cijena (isti spojni put dijeli više komponenata)
 - prilagodljivost prilikom projektiranja i nadogradnje računala (jednostavno dodavanje komponenata)
 - standardiziranost
- **Nedostatci** sabirnice:
 - mala propusnost (samo dvije komponente komuniciraju, a ostale su neaktivne)
 - ograničena duljina sabirnice
 - ograničen broj komponenata koji se mogu spojiti na jednu sabirnicu
 - problemi zbog komponenata različite brzine

- Alternativa sabirnici je izravno spajanje dvije komponente (*point-to-point*) pomoću vlastitog spojnog puta prilagođenog upravo tim komponentama
 - Prednost je veća brzina komunikacije
 - Nedostatak je veća cijena (više spojnih putova, više priključaka na čipu)



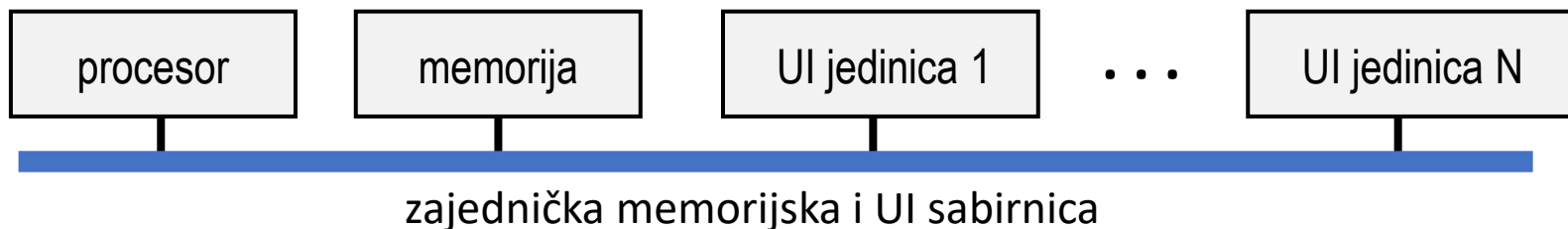
- Sabirnice se mogu dijeliti i na:
 - memorijsku sabirnicu
 - ulazno-izlaznu (U/I) sabirnicu
 - sabirnice specijalne namjene (npr. grafička)
- **Memorijska sabirnica:**
 - povezuje procesor i memoriju
 - male duljine
 - velike brzine rada
 - prilagođena brzini memorije

- **U-I sabirnica:**

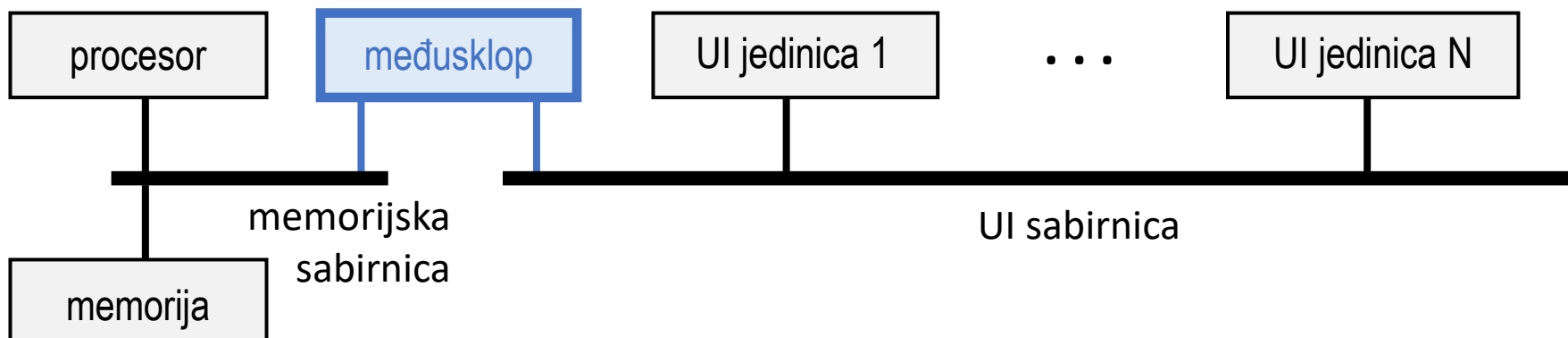
- povezuje U-I jedinice s procesorom
- velika duljina (npr. USB - do 5 metara)
- manja brzina rada nego memorijska sabirnica
- prilagodljivost različitim brzinama rada pojedinih U-I jedinica
- mogućnost spajanja velikog broja U-I jedinica (npr. 128)
- U-I sabirnica se spaja na procesor i memoriju na dva načina (vidi sljedeći slajd)

>>>>

- **Zajednička** memorijska i UI sabirnica (backplane bus):



- Spajanje memorijske i UI sabirnice pomoću posebnog **međusklopa** ili **mosta** (tj. neizravno):

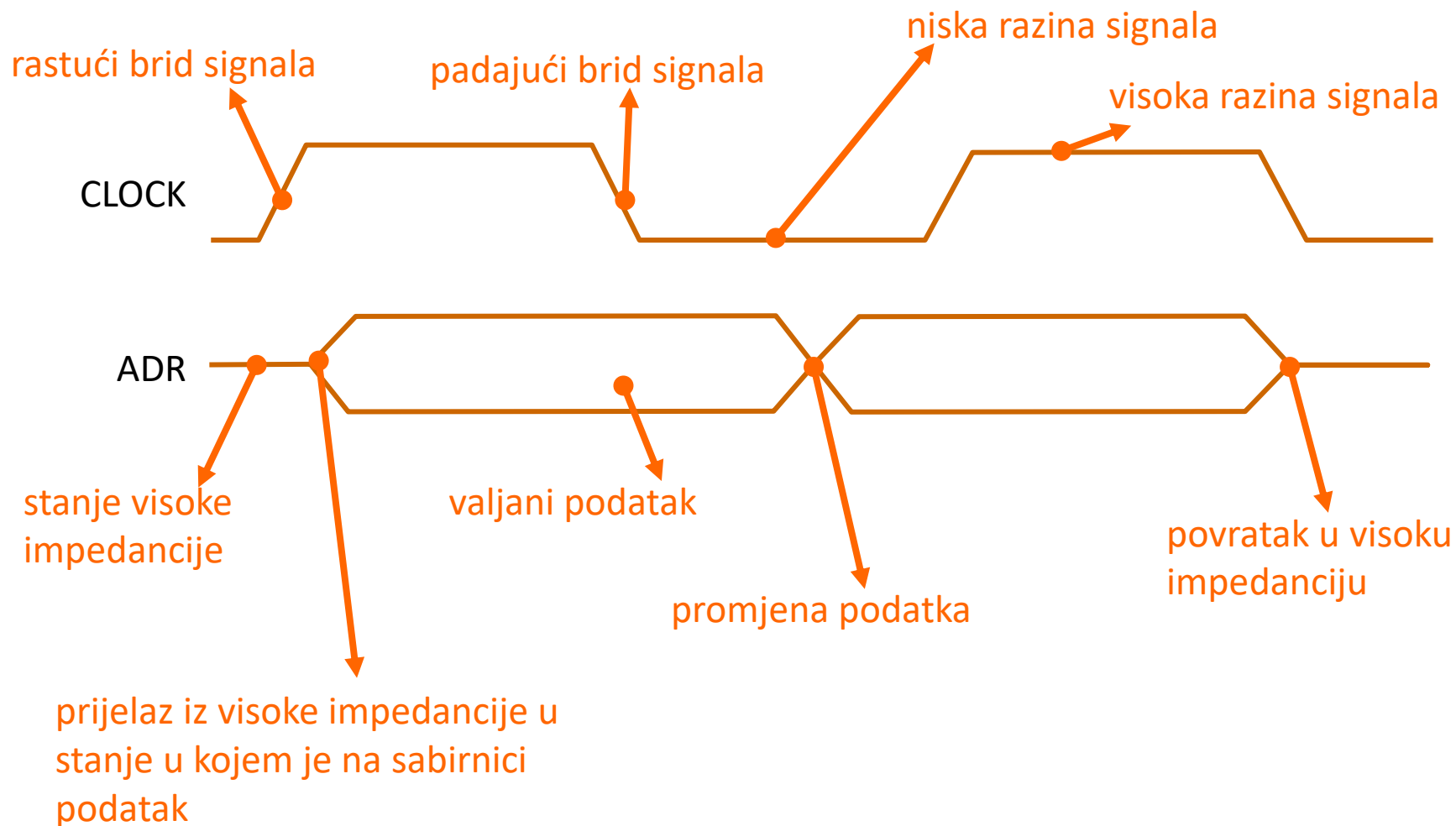


Sabirnice - sabirnički protokoli

Sabirnice - sabirnički protokoli

- Točan redoslijed svih koraka u komunikaciji naziva se **sabirnički protokol** (bus protocol)
- Sabirnička transakcija (bus transaction) je slijed koraka potrebnih da bi se na sabirnici izvela određena operacija, kao npr. operacija čitanja ili pisanja
- Sabirnička transakcija obično sadrži:
 - zahtjev (request)
 - odgovor (response)
- Unutar zahtjeva i/ili odgovora može se nalaziti podatak, adresa, naredba i sl.

Tumačenje oznaka na vrem. dijagramu:

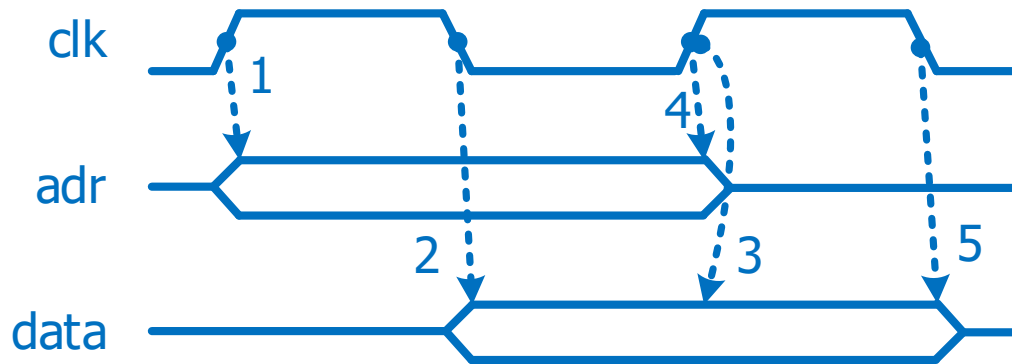


Sabirnice - sinkrone i asinkrone

- Sabirnice se prema načinu komunikacije dijele na:
 - sinkrone
 - asinkrone
- **Sinkrone sabirnice:**
 - **sve operacije su sinkronizirane s taktom sustava (tj. clockom)**
 - jednostavne su za implementaciju
 - imaju veliku brzinu rada pa zato i malu duljinu
 - bolje su prilagođene za slučaj kad svi uređaji imaju jednaku brzinu
 - imaju mogućnost prilagodbe brzine rada, ali se komunikacija većinom odvija predviđenom brzinom
 - češće se koriste za memorijske sabirnice

Sabirnice - sinkrone

- Sve operacije postavljanja adresa, čitanja podataka i uklanjanja adresa i podataka odvijaju se u vremenskim trenutcima točno definiranim u odnosu na CLOCK

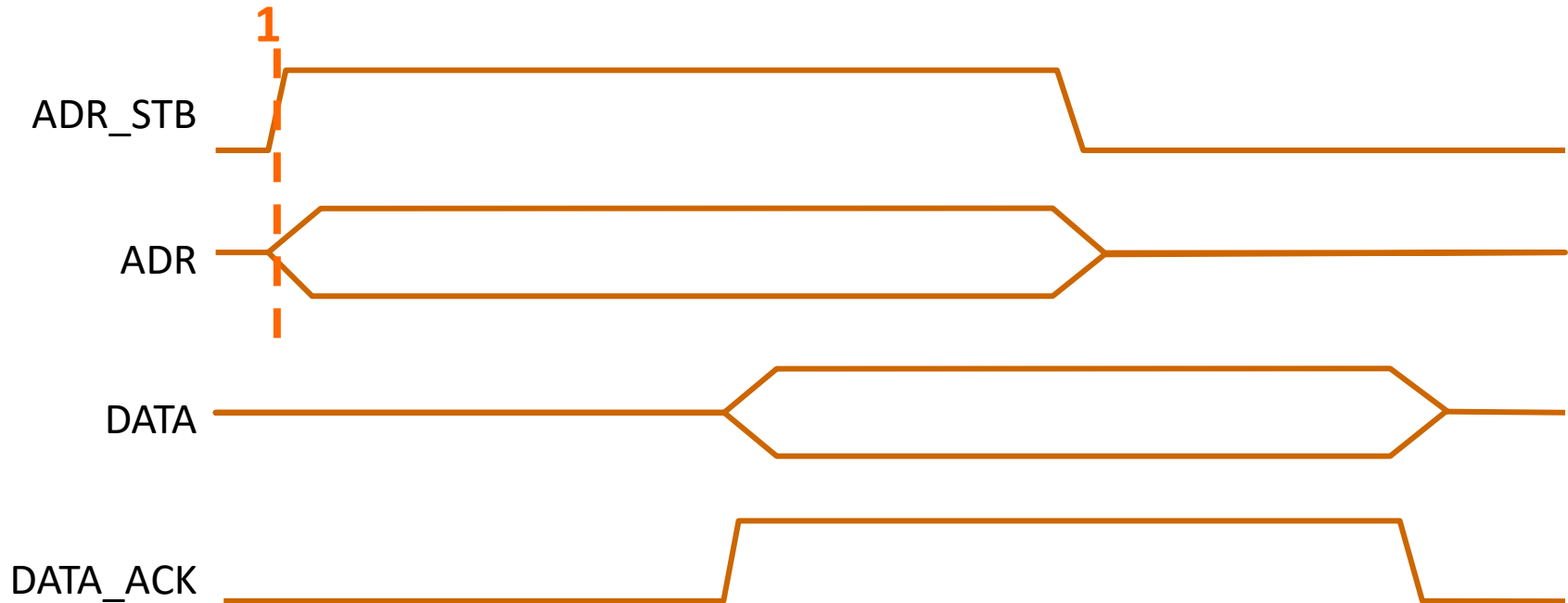


- Pojednostavljeni prikaz **čitanja**:
 - Na prvi rastući clock procesor postavlja adresu na *adr*, a memorija započinje s operacijom čitanja s te adrese (1)
 - Na prvi padajući clock memorija je dekodirala adresu i postavlja podatak na *data* (2)
 - Na drugi rastući clock
 - procesor čita podatak sa *data* (3)
 - te uklanja adresu sa *adr* jer ju memorija više ne treba (4)
 - Na drugi padajući clock memorija zna da je procesor pročitao podatak pa ga uklanja sa *data* (5)

- **Asinkrone sabirnice:**

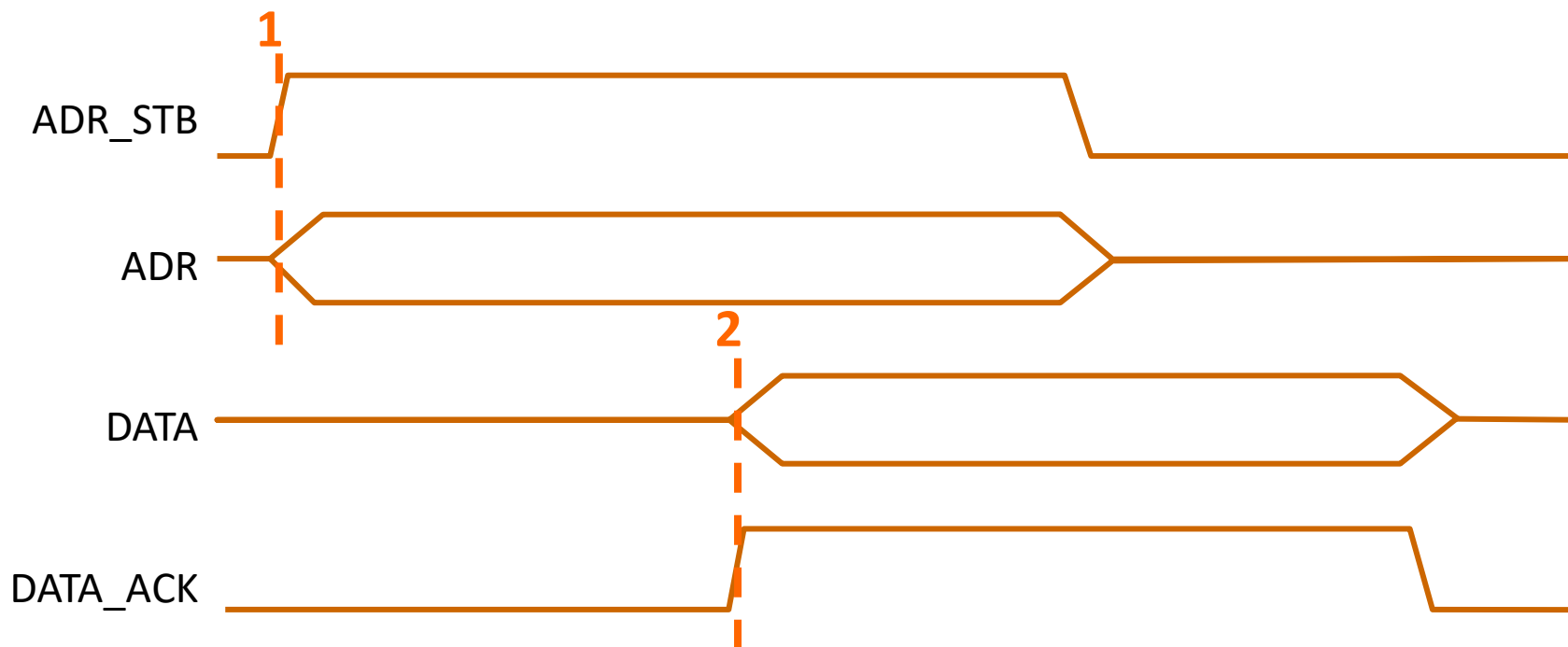
- ne koriste CLOCK za sinkronizaciju
- **uređaji se sinkroniziraju tzv. rukovanjem** (engl. handshaking protocol)
 - rukovanje je postupak u kojem strane koje komuniciraju prelaze na sljedeći korak komunikacije tek kad obje strane potvrde da je prethodni korak dovršen
- složenije su za implementaciju
- imaju manju brzinu rada
- mogu imati veliku duljinu
- bolje su prilagođene za slučaj kad uređaji imaju različite brzine
- češće se koriste za U-I sabirnice

- Pojednostavljeni prikaz asinkrone komunikacije (npr. čitanje):



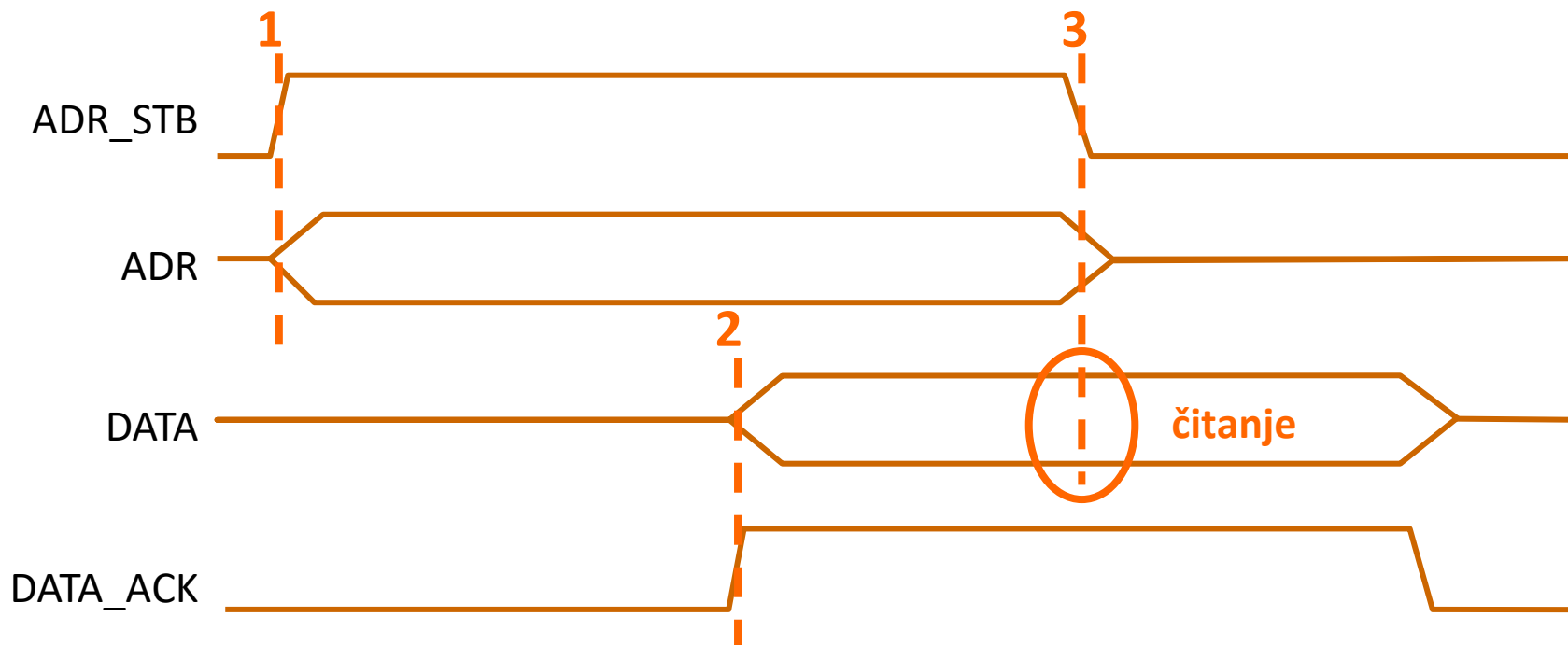
- 1) Procesor postavlja adresu na ADR i dojavljuje to memoriji aktiviranjem signala ADR_STB (address strobe)

- Pojednostavljeni prikaz asinkrone komunikacije (npr. čitanje):



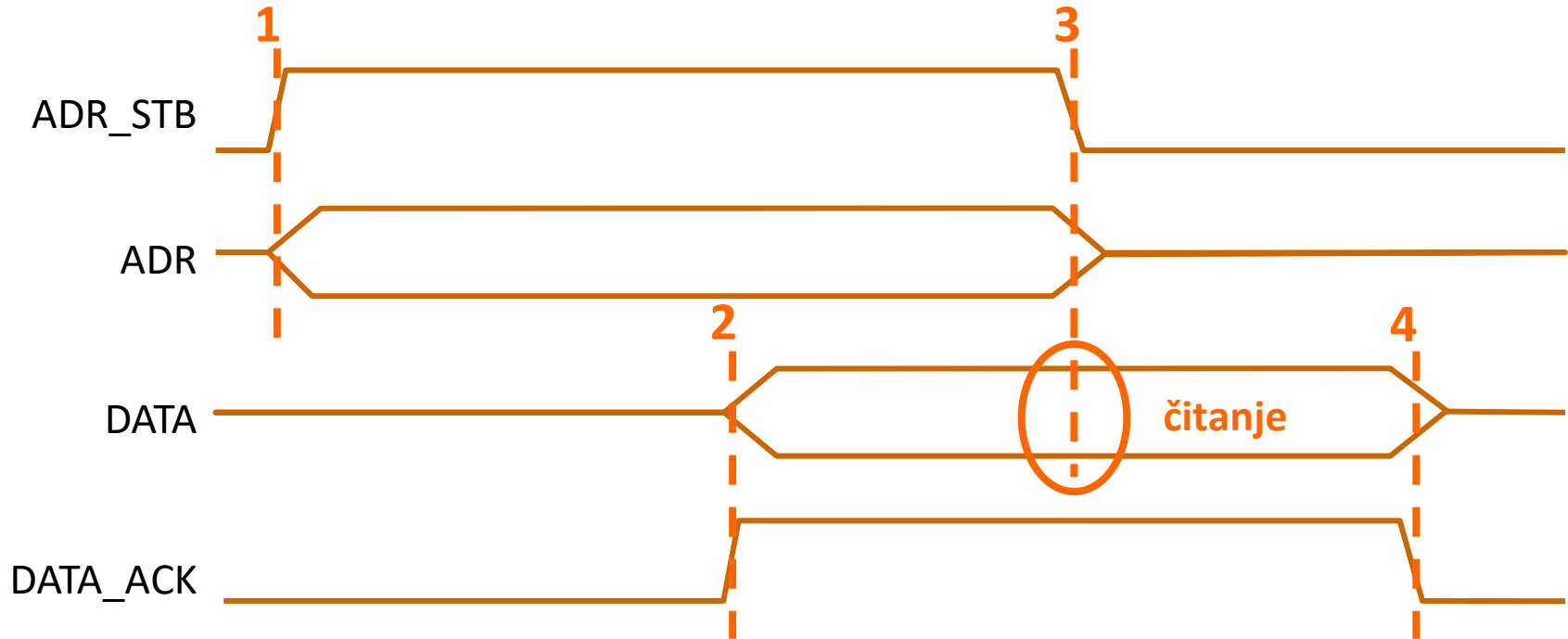
- 2) Memorija je detektirala ADR_STB i započela operaciju čitanja; nakon nekog vremena postavlja podatak na DATA i dojavljuje to procesoru aktiviranjem DATA_ACK (acknowledge)

- Pojednostavljeni prikaz asinkrone komunikacije (npr. čitanje):



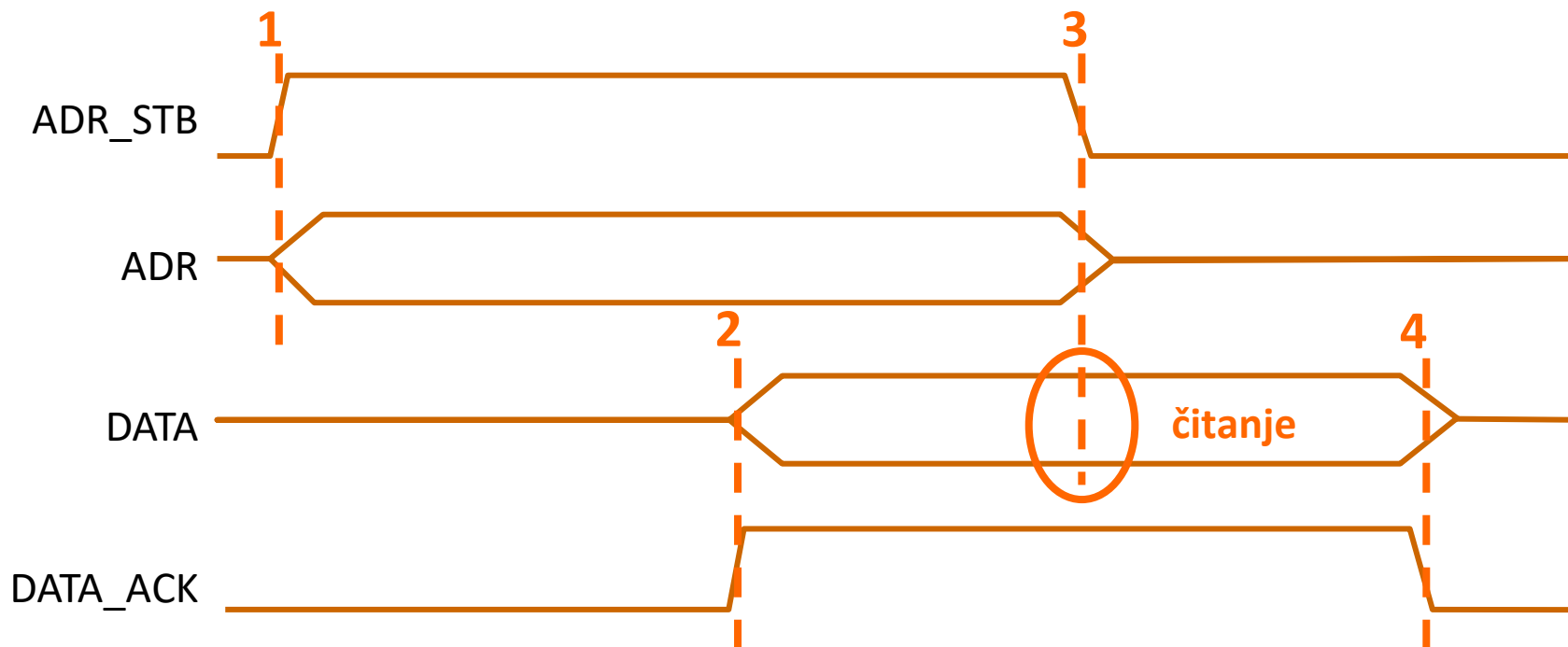
- 3) Procesor je detektirao DATA_ACK i zna da se na DATA nalazi podatak; procesor čita podatak sa DATA te uklanja adresu sa ADR i dojavljuje to memoriji deaktiviranjem ADR_STB

- Pojednostavljeni prikaz asinkrone komunikacije (npr. čitanje):



- 4) Memorija je detektirala deaktiviranje ADR_STB i zna da je procesor pročitao podatak; memorija uklanja podatak sa DATA i javlja to procesoru deaktiviranjem DATA_ACK

- Pojednostavljeni prikaz asinkrone komunikacije (npr. čitanje):



Nakon koraka 4) procesor detektira deaktiviranje DATA_ACK i može započeti novo čitanje ili pisanje podatka, tj. može ponovno započeti s korakom 1)

- Komentari:
 - I sinkrone i asinkrone sabirnice imaju mogućnost prilagodbe različitim brzinama uređaja spojenih na sabirnicu. Osnovna razlika je:
 - u tome što su asinkrone sabirnice upravo predviđene za spajanje uređaja različite brzine
 - spajanje uređaja različite brzine je više iznimka nego pravilo kod sinkronih sabirnica, ili takvi uređaji sudjeluju u manjem postotku sabirničkih transakcija
 - Asinkroni protokoli trebaju potvrdu da bi mogli nastaviti sa sljedećim koracima bez obzira koliko je vremena proteklo
 - Sinkroni protokoli nastavljaju s radom automatski kad se dosegne određeno stanje CLOCK-a, a usporeenje rada se mora izričito zahtijevati

- Komentari:
 - Neke memorijske sabirnice koriste asinkrone protokole
 - Međutim, to ne znači da te sabirnice nemaju CLOCK ili da ga uopće ne koriste
 - CLOCK se koristi u procesoru za njegov interni rad
 - Budući da interni rad procesora ovisi o njenoj komunikaciji preko sabirnica, onda su i na asinkronoj sabirnici barem neki koraci (početni) sinkronizirani sa signalom CLOCK