

Računalni sustav s procesorom ARM



Iznimke



Obrada iznimaka

- U ARM-u se **iznimkama** nazivaju razne vrste događaja koje ne ulaze u normalno slijedno izvođenje naredaba. Na primjer: pojava prekida, dohvat neispravnog strojnog koda, resetiranje procesora, itd.
- Procesor ARM načelno obrađuje iznimke na sljedeći način:
 - ARM pohranjuje R15 (programsko brojilo) u registar LR, a zatim pohranjuje CPSR (registar trenutnog programskog stanja) u registar SPSR*
 - ARM prelazi u privilegirani način rada koji ovisi o vrsti iznimke
 - ARM skače u potprogram za obradu iznimke za dotični način rada
- Adrese potprograma za obradu iznimke su fiksno definirane za svaku pojedinu iznimku
- Ako se u isto vrijeme pojave dvije ili više iznimaka, ARM definira prioritete izvođenja

* osim za RESET

Iznimke: adrese potprograma i prioriteti

Tip iznimke	ARM se prebacuje u način rada	Adresa potprograma*	Prioritet
Reset	Supervisor	0x00000000	1
Undefined instruction	Undefined	0x00000004	6
Software interrupt (SWI)	Supervisor	0x00000008	6
Prefetch Abort (instruction fetch memory abort)	Abort	0x0000000C	5
Data Abort (data access memory abort)	Abort	0x00000010	2
IRQ (interrupt)	IRQ	0x00000018	4
FIQ (fast interrupt)	FIQ	0x0000001C	3

* Uočite da je za svaki potprogram na raspolaganju samo 4 bajta, pa se tu u pravilu stavlja naredba skoka na odsječak za obradu iznimke (osim potprograma na adresi 0x1C)

Reset

- Slijed operacija koje se obave pri pojavi pojedine iznimke vrlo je sličan za sve iznimke
- Kada se na ulazu u procesor aktivira signal Reset, procesor odmah prekida izvođenje naredbe. Nakon što se Reset deaktivira, obavi se sljedeći niz operacija:
 - R14_svc = UNPREDICTABLE value
 - SPSR_svc = UNPREDICTABLE value
 - CPSR[4:0] = %B10011 /* Enter Supervisor mode */
 - CPSR[5] = 0 /* Execute in ARM state */
 - CPSR[6] = 1 /* Disable fast interrupts */
 - CPSR[7] = 1 /* Disable normal interrupts */
 - PC = %H00000000

Reset

- Povratak iz iznimke Reset nije predviđen. U slučaju inicijalizacije procesora (npr. uključivanje napajanja) očekuje se da će vanjska logika aktivirati signal Reset kako bi procesor normalno započeo s radom.
- Prema tome, svaki program mora pretpostaviti obradu iznimke Reset i nakon toga prelazak na izvođenje izabrane aplikacije.

IRQ (prekid)

- Ako je u programu omogućeno prihvaćanje prekida (bit I u CPSR je obrisan), procesor će na kraju izvođenja svake naredbe provjeriti je li ulaz IRQ aktivan. Ako je neki vanjski sklop aktivirao signal prekida (IRQ), procesor će nakon završetka trenutne naredbe obaviti sljedeći niz operacija:

- R14_irq = address of next instruction to be executed + 4
 - SPSR_irq = CPSR
 - CPSR[4:0] = %B10010
 - CPSR[5] = 0
 - /* CPSR[6] is unchanged */
 - CPSR[7] = 1
 - PC = %H00000018
- /* Enter IRQ mode */
- /* Execute in ARM state */
- /* Disable normal interrupts */

IRQ

- Za povratak iz IRQ-potprograma treba izvesti naredbu:

SUBS PC, R14, #4

- Ova naredba* će obnoviti PC (iz R14_irq) i CPSR (iz SPSR_irq) te nastaviti izvođenje programa na mjestu gdje je prekinut

* SUB napisan s ekstenzijom S i odredišnim registrom PC, znači da treba obnoviti CPSR

FIQ (brzi prekid)

- Brzi prekid namjenjen je primjenama gdje je bitno brzo reagirati i obaviti niz operacija
- FIQ zato ima dovoljan broj 'privatnih' registara tako da ne treba obavljati operacije pohranjivanja i vraćanja konteksta na stog
- Pored toga, adresa prekidnog potprograma FIQ je namjerno zadnja na listi tako da se potprogram može napisati odmah od te adrese bez potrebe za skokom na drugo mjesto u memoriji (izbjegnuto je kašnjenje zbog takvog skoka)

FIQ

- Ako je u programu omogućeno prihvaćanje brzog prekida (bit F u CPSR je obrisan) procesor će na kraju izvođenja svake naredbe provjeriti da li je ulaz FIQ aktivan. Ako je na ulazu u procesor aktiviran signal brzog prekida (FIQ), procesor će nakon završetka trenutne naredbe obaviti sljedeći niz operacija:
 - `R14_fiq` = address of next instruction to be executed + 4
 - `SPSR_fiq` = CPSR
 - `CPSR[4:0]` = `%B10001` /* Enter FIQ mode */
 - `CPSR[5]` = 0 /* Execute in ARM state */
 - `CPSR[6]` = 1 /* Disable fast interrupts */
 - `CPSR[7]` = 1 /* Disable normal interrupts */
 - `PC` = `%H0000001C`

FIQ

- Za povratak iz FIQ-potprograma treba izvesti naredbu (isto kao za IRQ):

SUBS PC, R14,#4

- Ova naredba će obnoviti PC (iz R14_fiq) i CPSR (iz SPSR_fiq) te nastaviti izvođenje programa na mjestu gdje je prekinut

Primjer obrade iznimke

Glavni program treba povećavati registar R1. Kada dođe do prekida IRQ, treba postaviti oznaku da treba završiti glavni program.

```
ORG 0  
B GLAVNI
```

```
ORG 18  
B OBR_IRQ
```

```
GLAVNI  MRS  r0, CPSR      ; pročitaj CPSR  
        BIC  r0, r0, #80   ; pobriši bit I (za bit F koristi se maska 40)  
        MSR  CPSR_c, r0    ; upiši promjenu u CPSR (dozvoli IRQ)  
        MOV  r0, #1        ; oznaka da treba ponavljati petlju  
  
PETLJA  ADD  r1, r1, #1     ; povećavaj r1 dok ne dođe prekid  
        CMP  r0, #0        ; provjeri oznaku kraja  
        BNE  PETLJA  
        SWI  123456
```

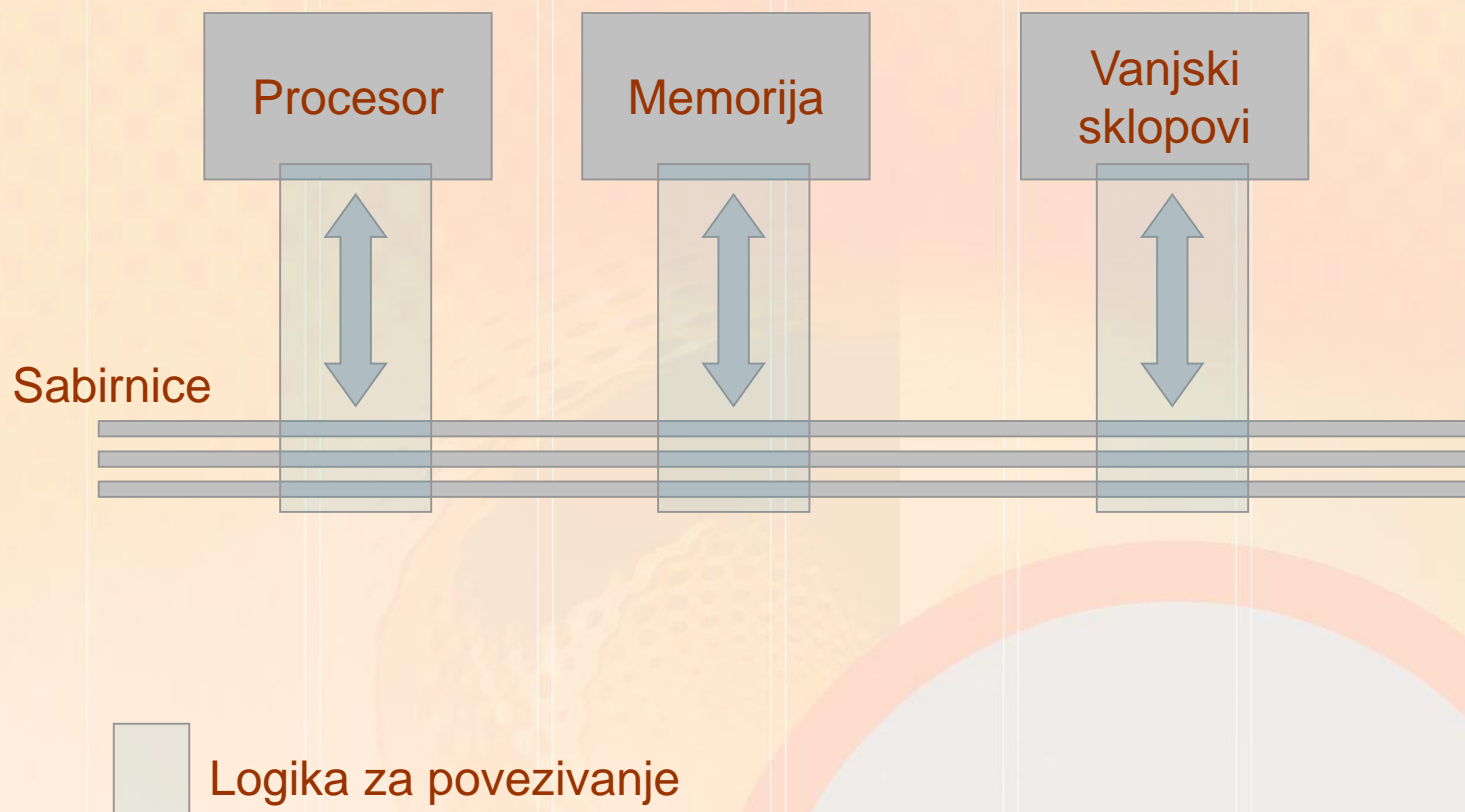
```
OBR_IRQ ; Potprogram za obradu iznimke  
        MOV  R0, #0        ; oznaka da treba završiti petlju  
        SUBS PC, R14, #4    ; povratak iz iznimke
```



Računalni sustav s procesorom ARM



Osnovni dijelovi računalnog sustava



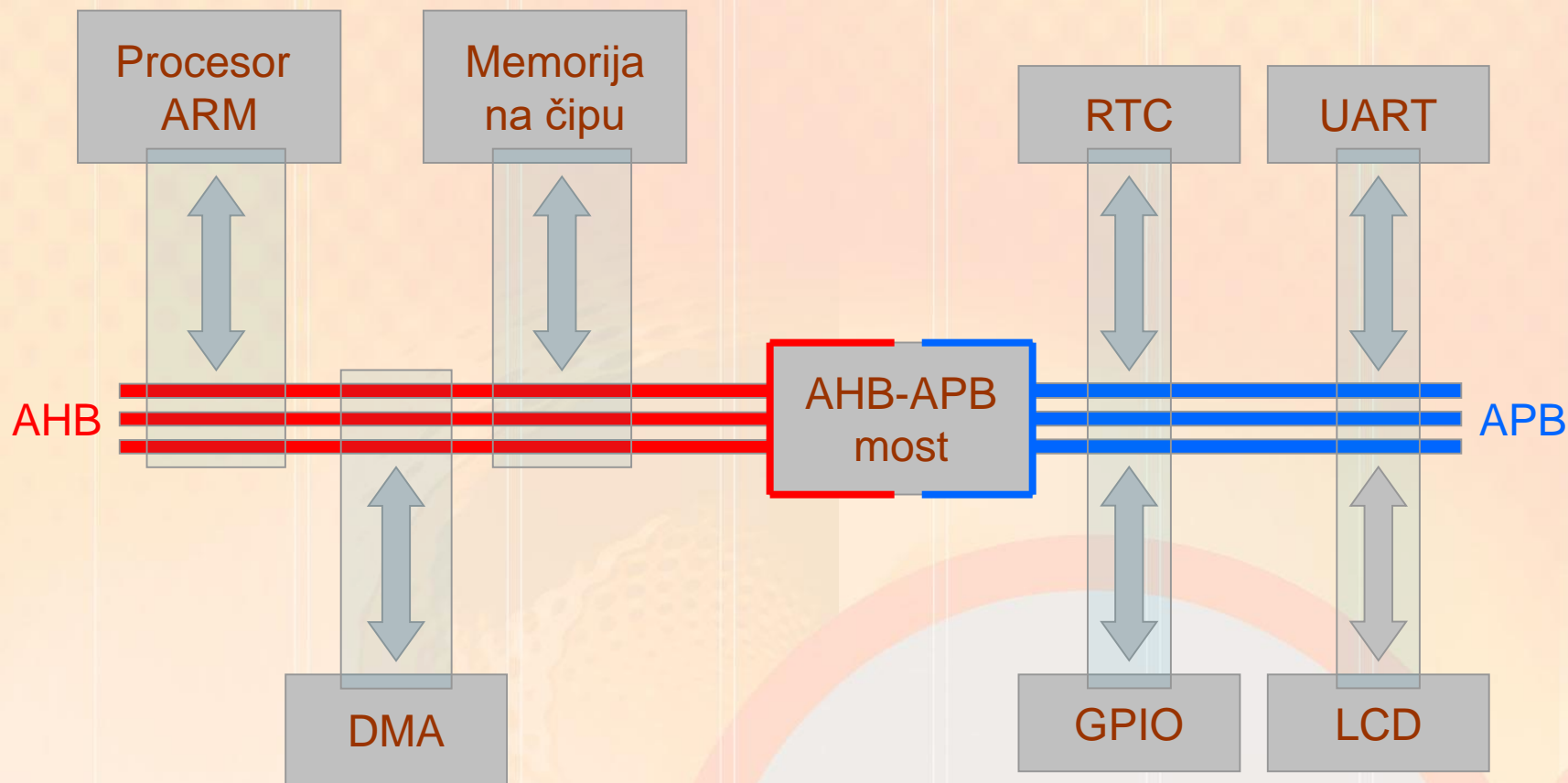
AMBA

- AMBA, kratica za Advanced Microcontroller Bus Architecture
- Trenutno aktualne specifikacije su AMBA 5
(ACE,ACE-Lite,AXI5, AXI5-Lite,AXI5-Stream,AHB,APB,ATB)
- Mi ćemo u okviru ovog predmeta proučiti samo načelno dvije sabirnice koje su definirane još u AMBA2 specifikacijama:
 - AHB (Advanced High-performance Bus)
 - APB (Advanced Peripheral Bus).

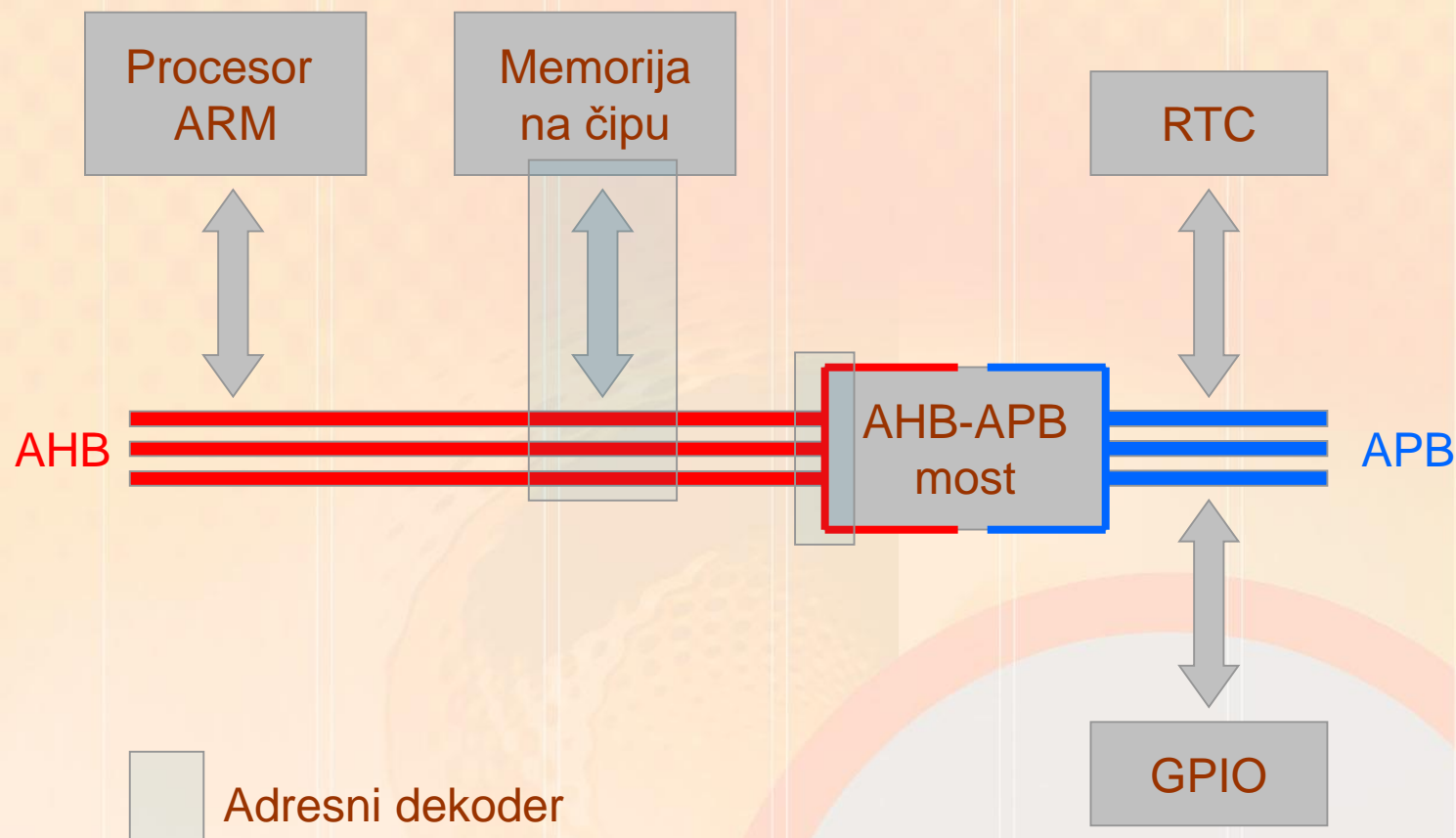
AMBA

- **AMBA AHB** je sabirnica namijenjena za sustave visokih performansi i visokih frekvencija signala vremenskog upravljanja i koristi se isključivo kao brza središnja (memorijska) sabirnica. AHB omogućuje efikasno povezivanje procesora, memorije koja se nalazi na čipu kao i vanjske memorije.
- **AMBA APB** je sabirnica za povezivanje vanjskih uređaja u sustav. Karakteristike ove sabirnice su mala potrošnja i jednostavnija izvedba u usporedbi sa središnjim sabirnicama, s ciljem što jednostavnijeg povezivanja vanjskih jedinica u cjelovit sustav. Na APB sabirnicu se povezuju vanjski uređaji koji ne zahtijevaju visoke performanse i nemaju kompleksna sučelja. Primjeri takvih uređaja su serijski kontroler (UART), LCD-kontroler, vremenski sklop (RTC) i paralelni sklop (GPIO).

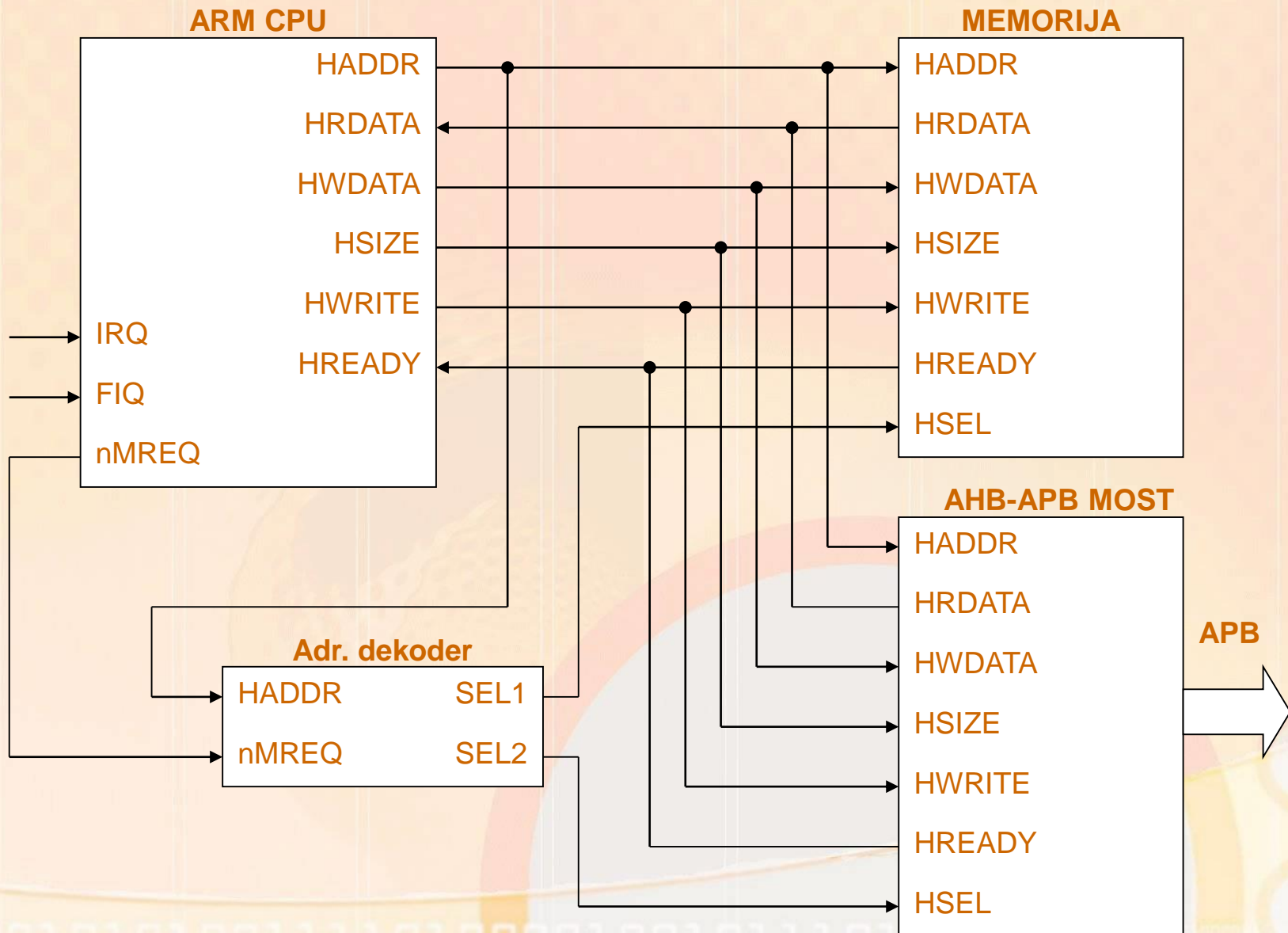
ARM sustav



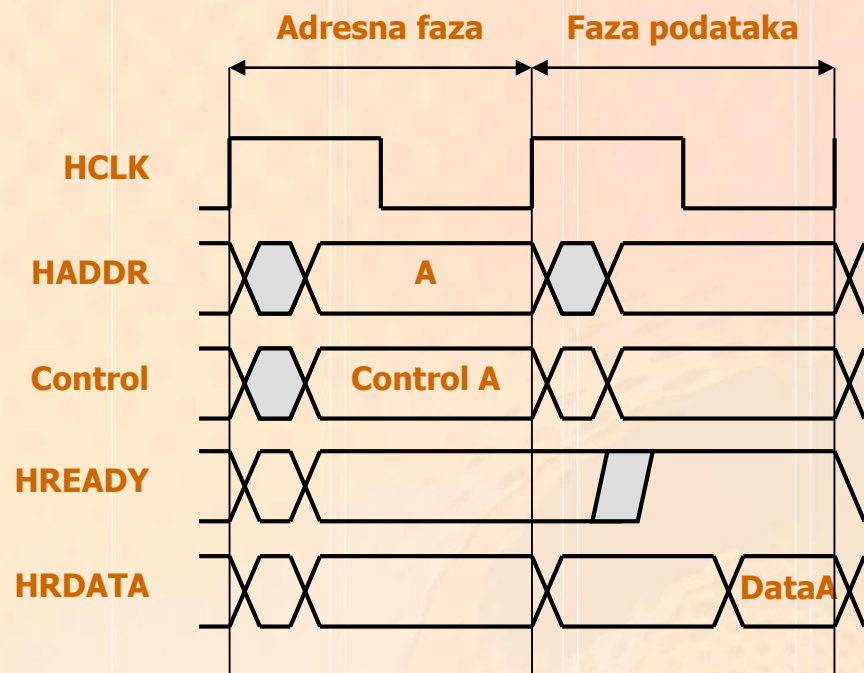
... i naš sustav u ATLAS-u



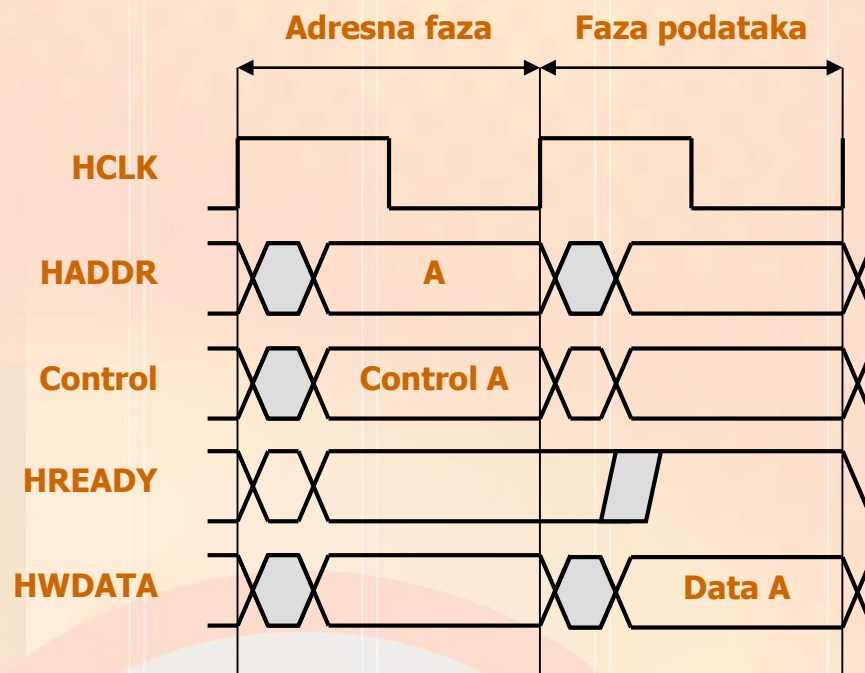
Središnji sustav sa sabirnicom AHB



Adresna i podatkovna faza na AHB

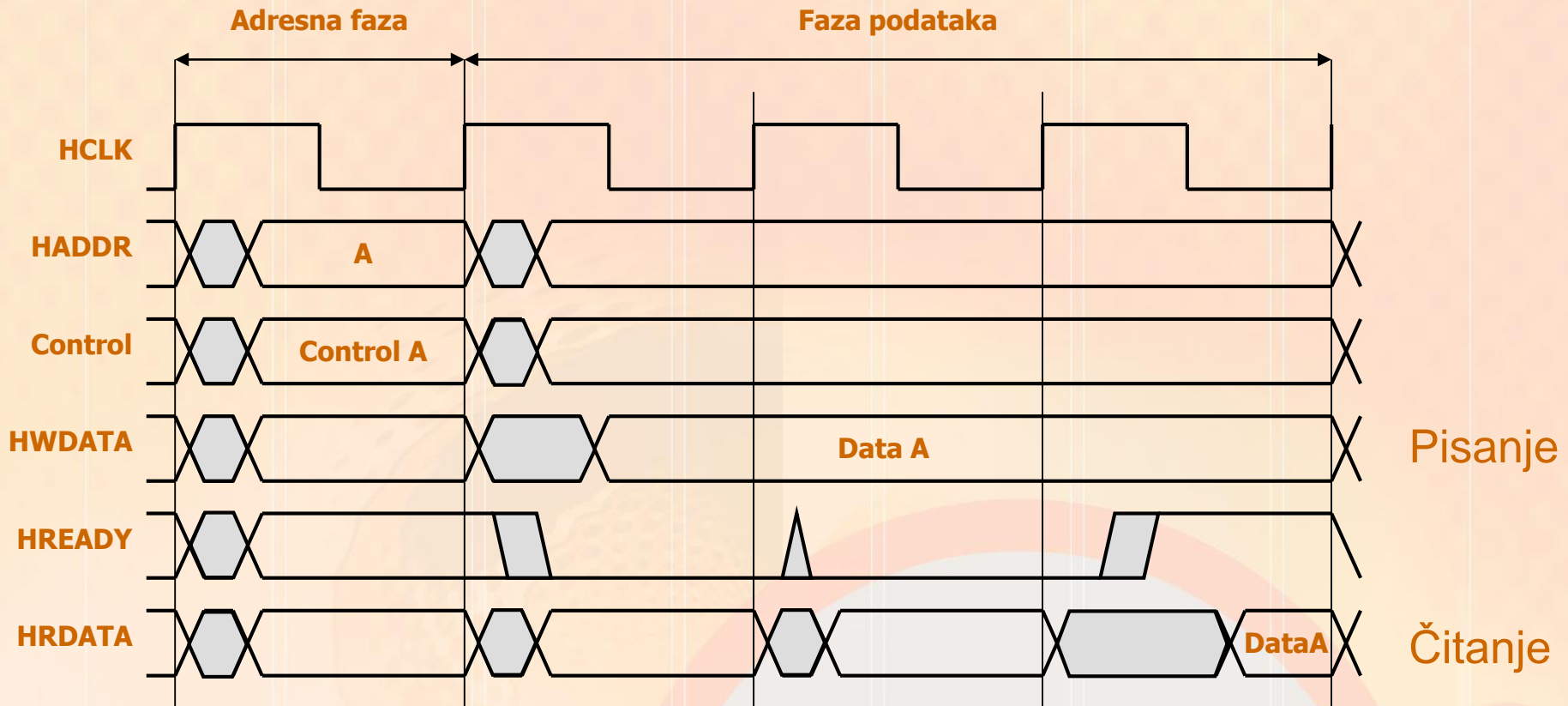


Čitanje podatka



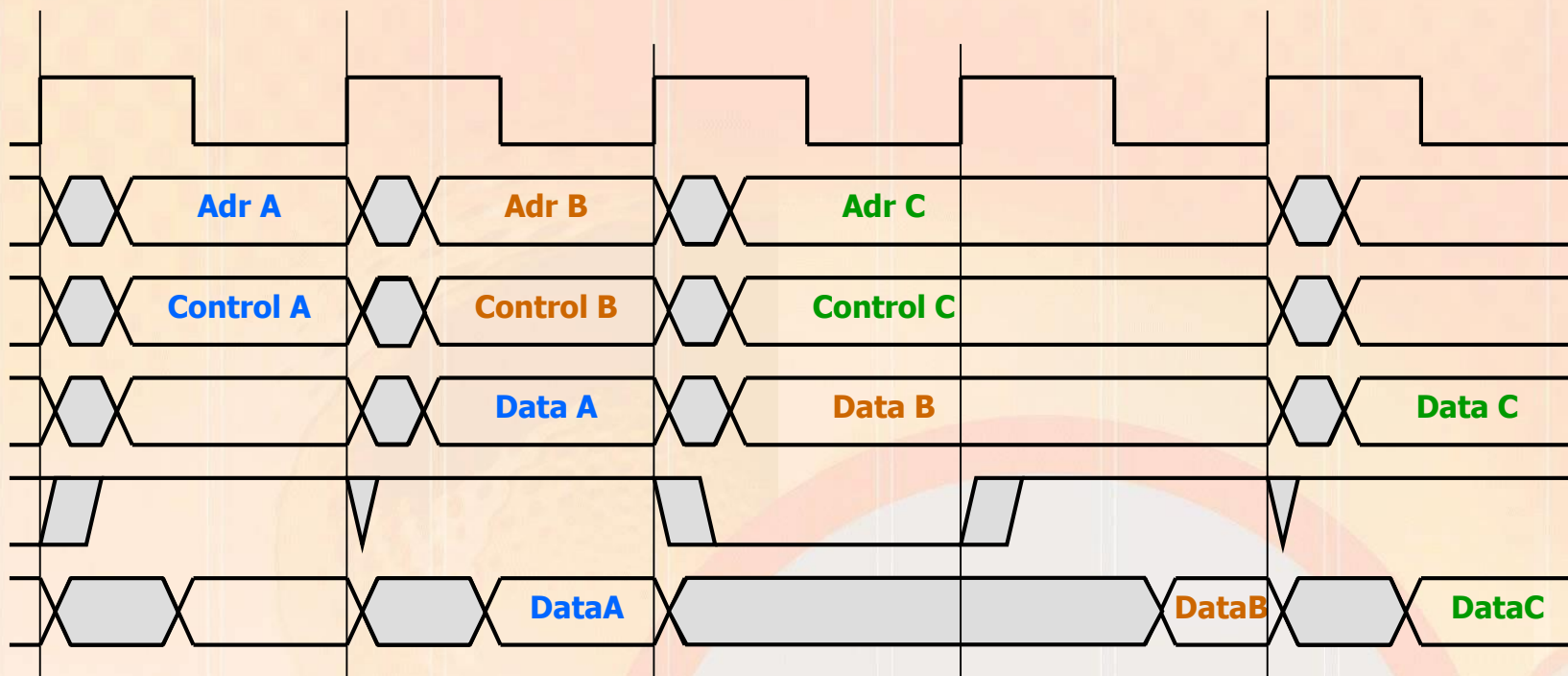
Pisanje podatka

Pristup memoriji sa stanjem čekanja na AHB



Preklapanje adresne i podatkovne faze na AHB

- Kod sabirnice AHB postoji **preklapanje** između adresne faze jednog pristupa i podatkovne faze prethodnog pristupa* čime se ubrzava komunikacija

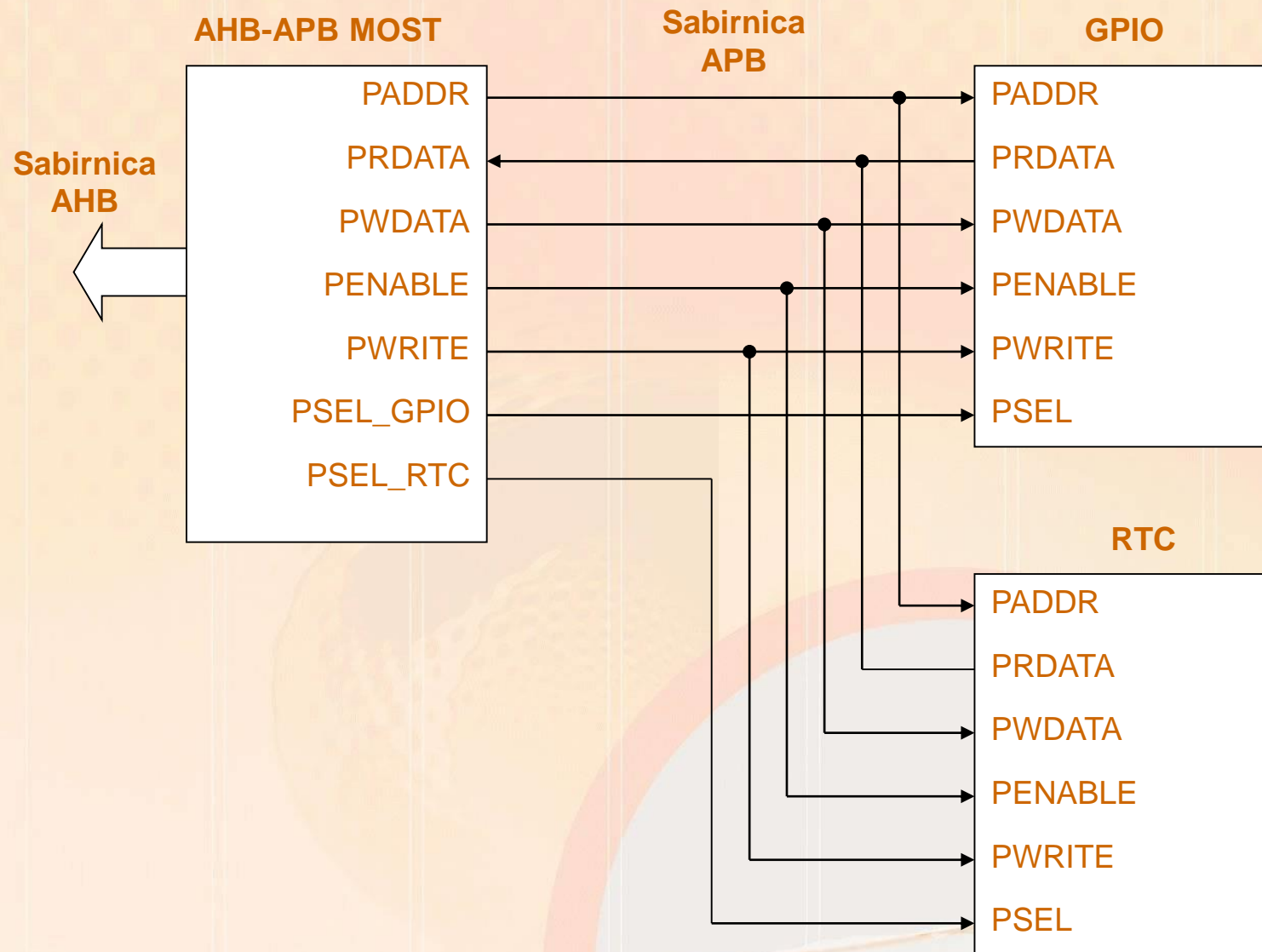


* slično ideji preklapanja faza u protočnoj strukturi

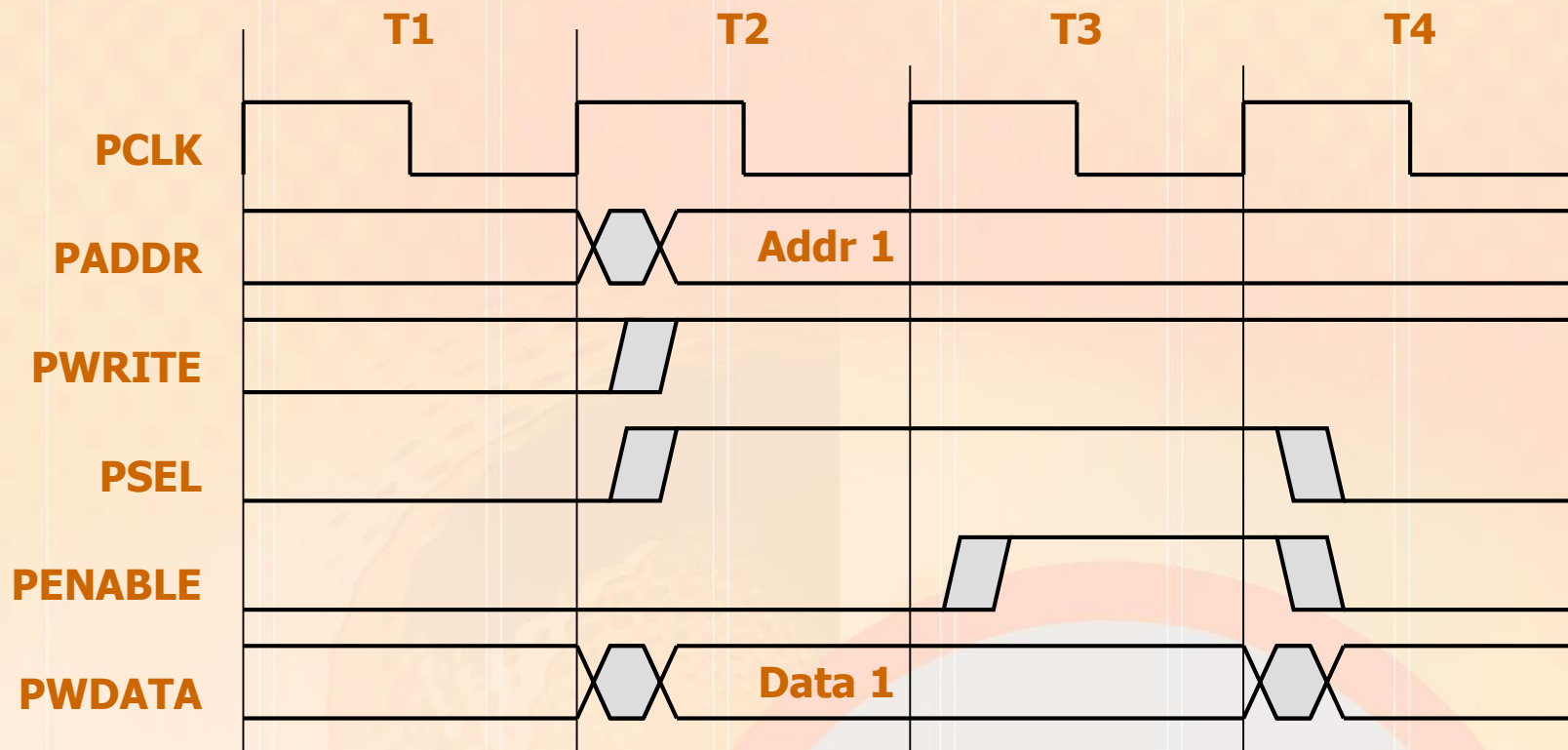
Sabirnički periodi na AHB

nMREQ	SEQ	Tip perioda	Opis
0	0	N-period	Neslijedni period (engl. Nonsequential)
0	1	S-period	Slijedni period (engl. Sequential)
1	0	I-period	Interni period (engl. Internal)
1	1	C-period	Period prijenosa sadržaja koprocesorskog registra (engl. Coprocessor register transfer)

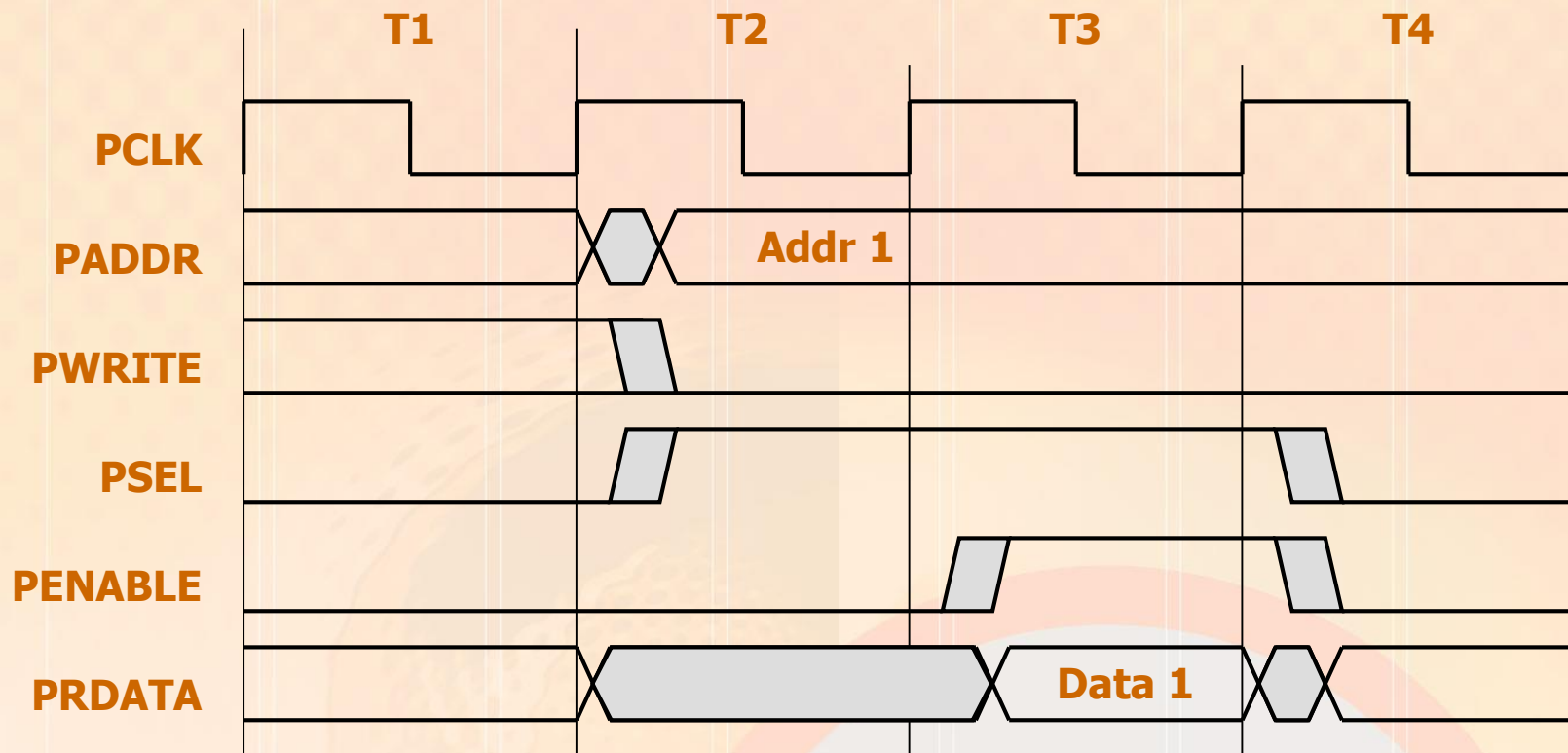
Sabirnica APB



Period pisanja na APB



Period čitanja na APB



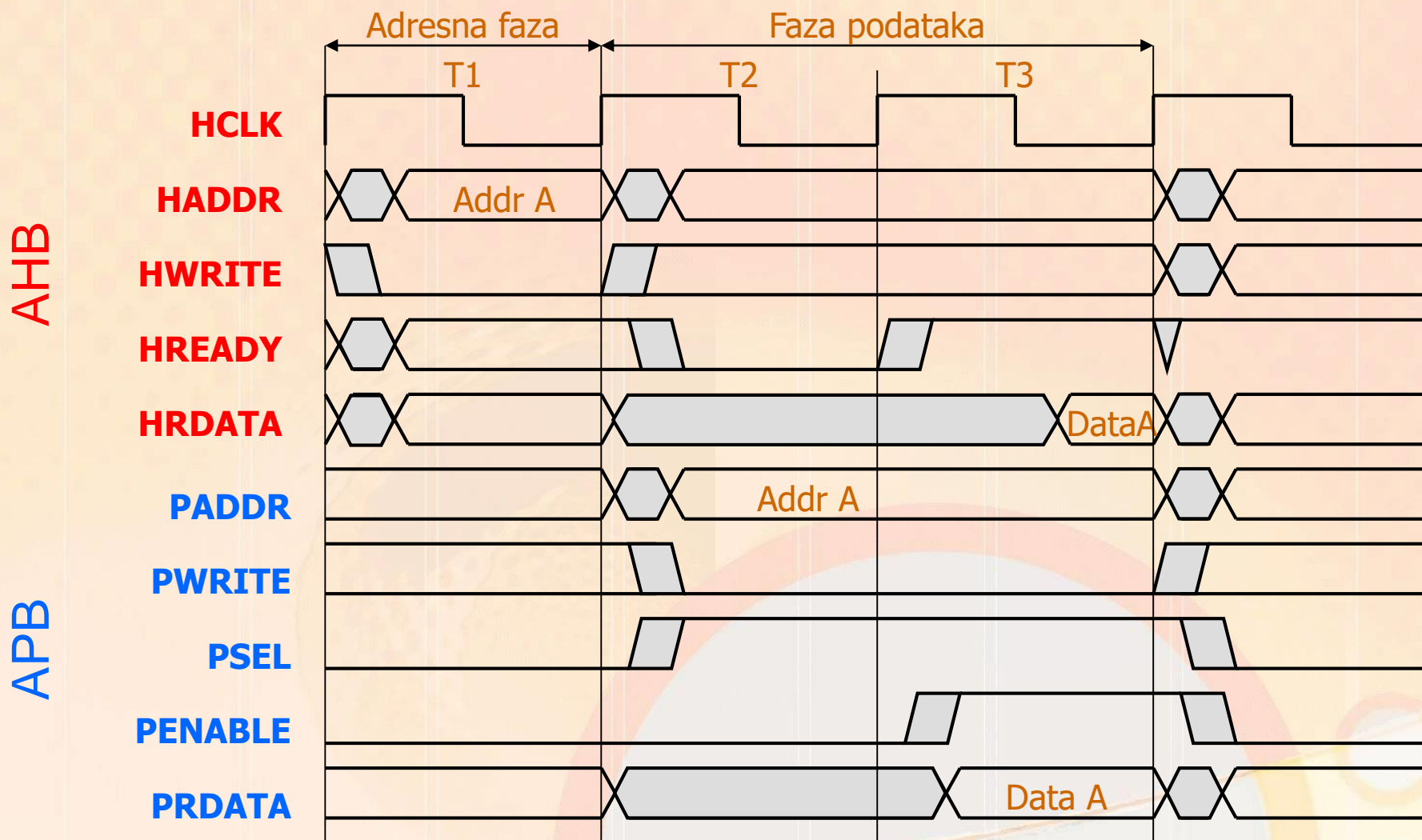
AHB APB most

- Sabirnice AHB i APB predviđene su za različite tipove sklopova:
 - AHB je središnja, brza sabirnica,
 - APB je jednostavnija sabirnica namijenjena vanjskim sklopovima koji su obično sporiji.
- Most AHB-APB je sklop koji omogućuje povezivanje ovih sabirnica i uređaja na njima u cjelovit sustav te prijenos podataka između uređaja sa sabirnicama AHB i APB.

AHB APB most

- Funkcije mosta su sljedeće:
 - uzima adresu i održava je valjanom tijekom cijelog prijenosa
 - dekodira adresu i generira signal PSEL_x kojim se izabire jedna od vanjskih jedinica kojom se izvodi prijenos podataka
 - postavlja podatke na sabirnicu APB kod perioda pisanja
 - postavlja podatke sa sabirnice APB na sabirnicu AHB tijekom perioda čitanja
 - generira vremenski signal PENABLE kojim se omogućuje prijenos

Čitanje s vanjske jedinice



Pisanje na vanjsku jedinicu

