

Arhitekture i algoritmi DSP 2



Auditorne vežbe AU-4[0]
Audio obrada u realnom vremenu



Odsek za računarsku tehniku i računarske komunikacije

Standalone audio obrada



PC (radna stanica)

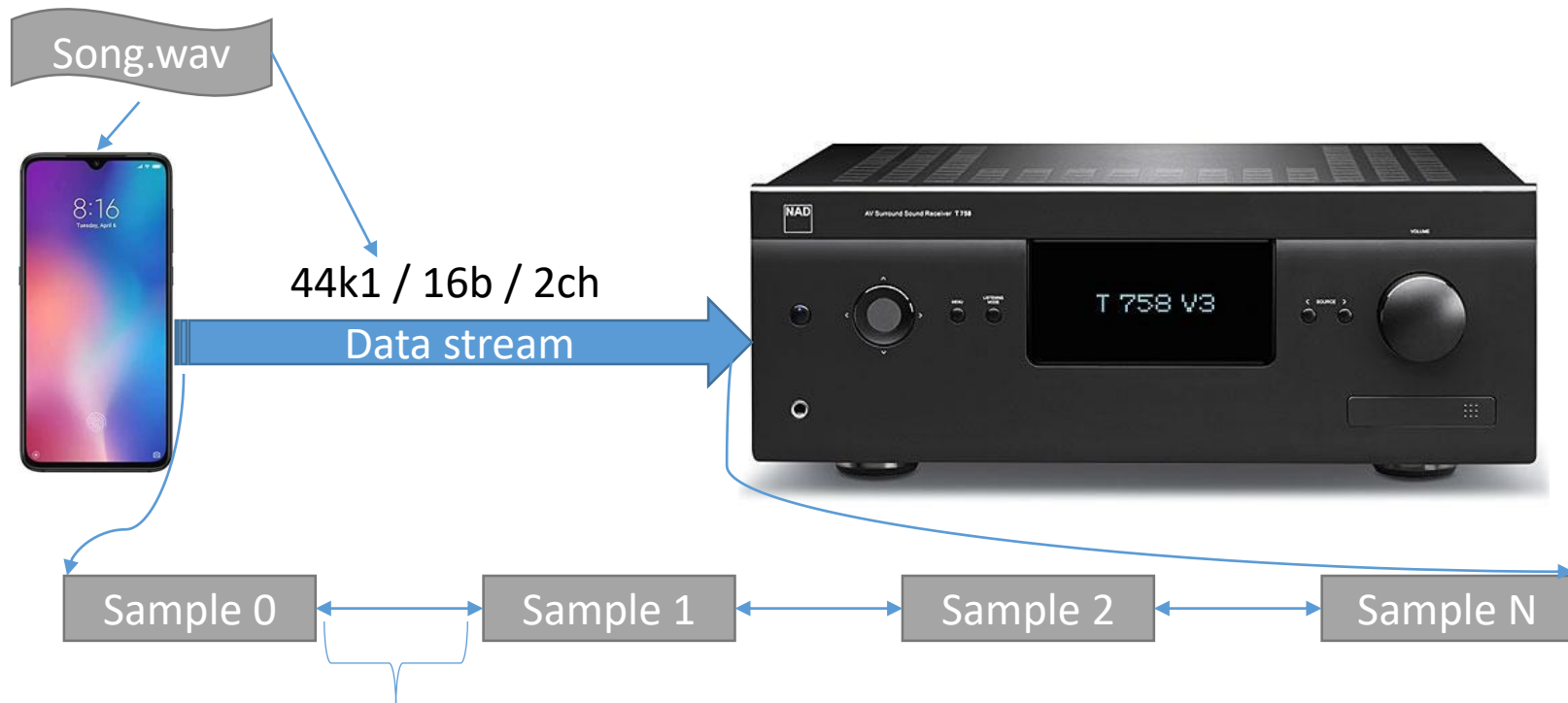
Standalone:

- Simulacija input/output mehanizma (programska podrška za čitanje iz i pisanje u wav fajlove)
- Mi definišemo brzinu pristizanja podataka u sistem kroz fajl IO
- Obrada jednog bloka podataka teorijski može da traje beskonačno dugo!

Song.wav

Output wave

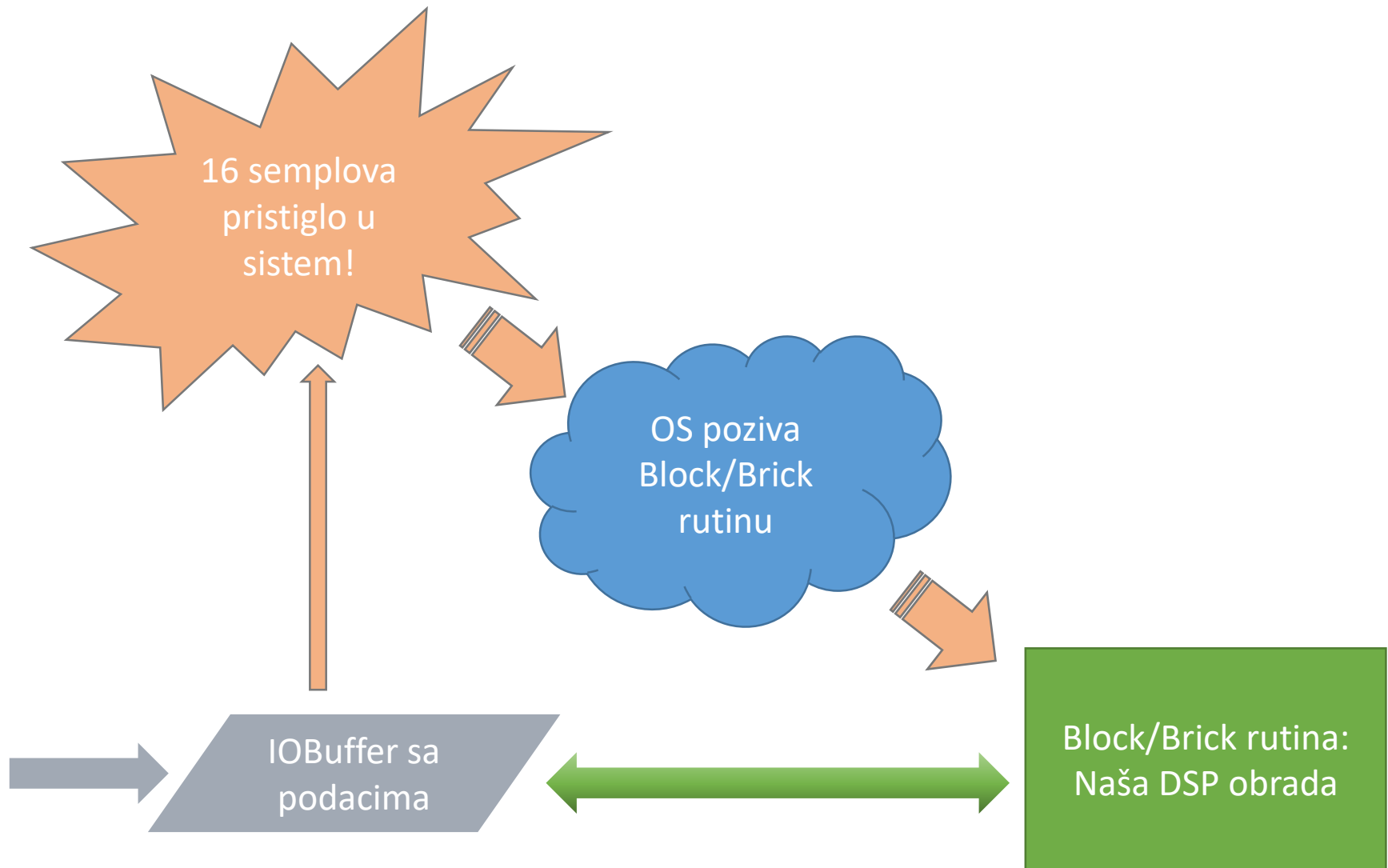
Real-time audio sistem



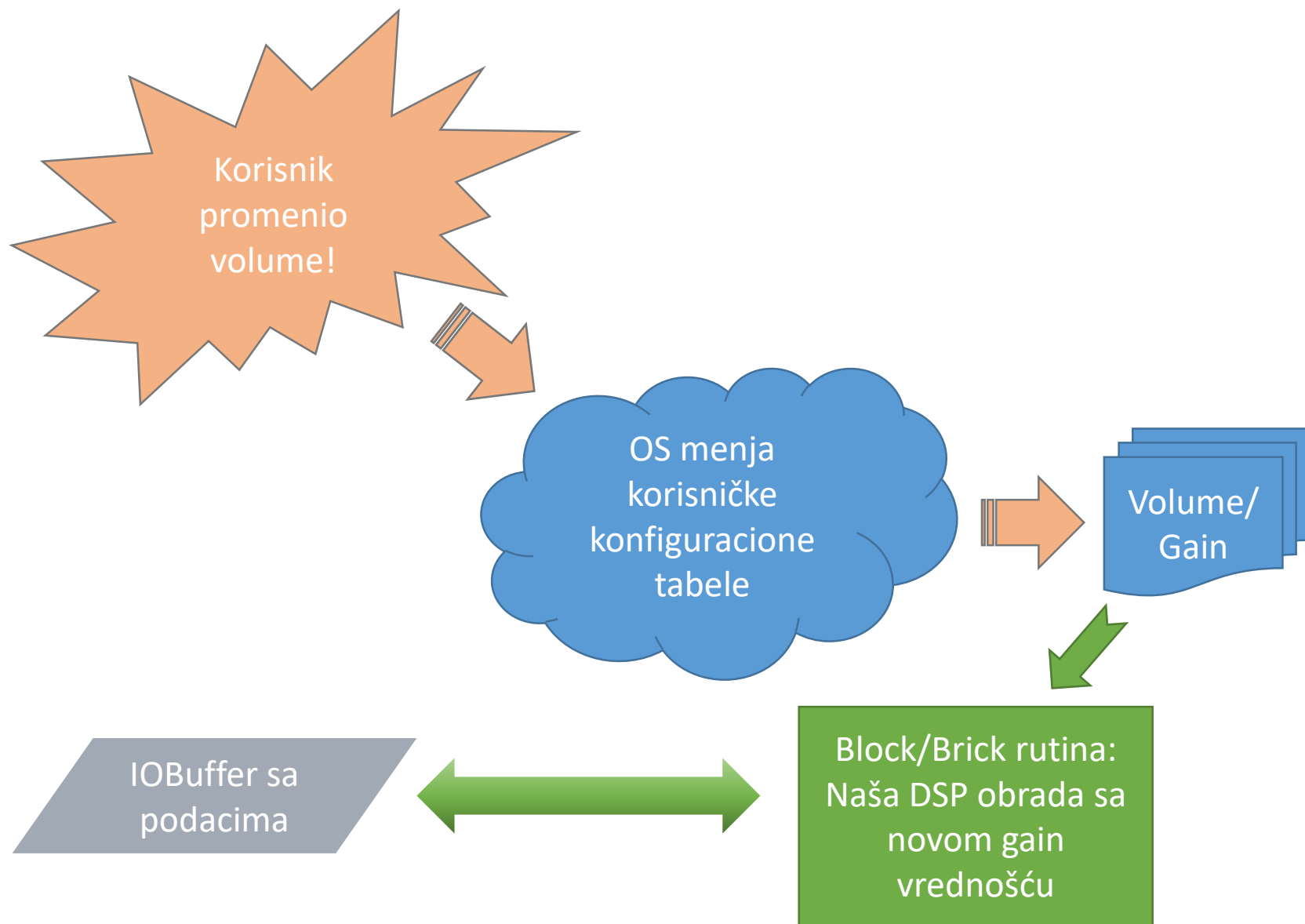
- $1/44100 = 2.2675e-5 \sim 23\text{micro seconds}$ (između 2 sempla)
- Blok obrade = 16 semplova, $16 \cdot 23$ mikro sekunde za procesing
- Šta se dogodi ako ne stignemo u tom periodu da završimo obradu?

- ❑ **Spoljni** faktori definišu i ograničavaju brzinu audio obrade!
 - Brzina pristizanja podataka u sistem
 - Promene kontrola od strane krajnjeg korisnika
 - Drugi spoljni događaji: inicijalizacija sistema, tajmeri...
- ❑ DSP operativni sistemi najčešće obezbeđuju rutine (prekide) kao način da nas obaveste o spoljnim promenama u sistemu!

Rutina za blok podataka



Promena korisnički kontrola



Profiling



- ❑ Kako biti siguran da je obrada vremenski usklađena?
- ❑ Merenje MIPS-a (tzv. „profiling“)!

Arhitekture i algoritmi DSP 2

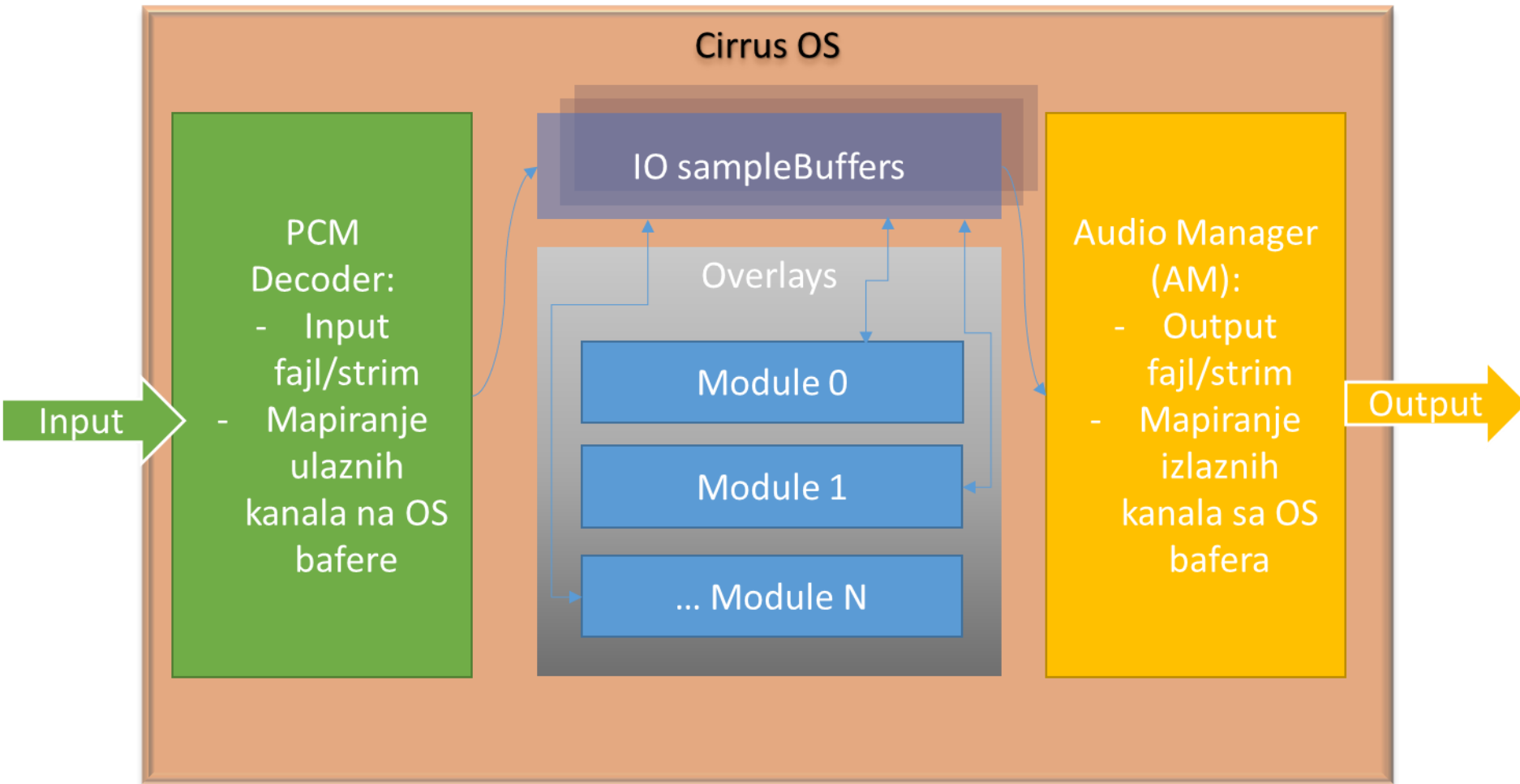


Auditorne vežbe AU-4[1]
Operativni sistem DSP-a CS497xx

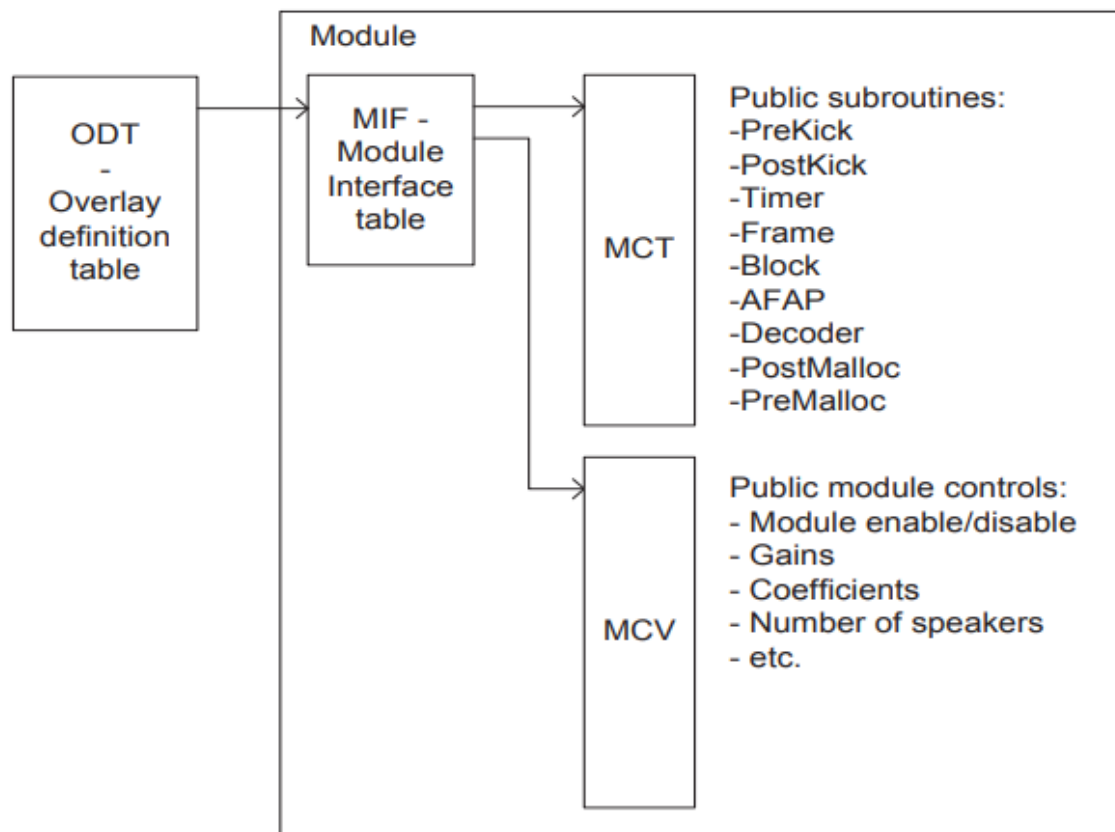


Odsek za računarsku tehniku i računarske komunikacije

Osnovna struktura OS-a (podsećanje)

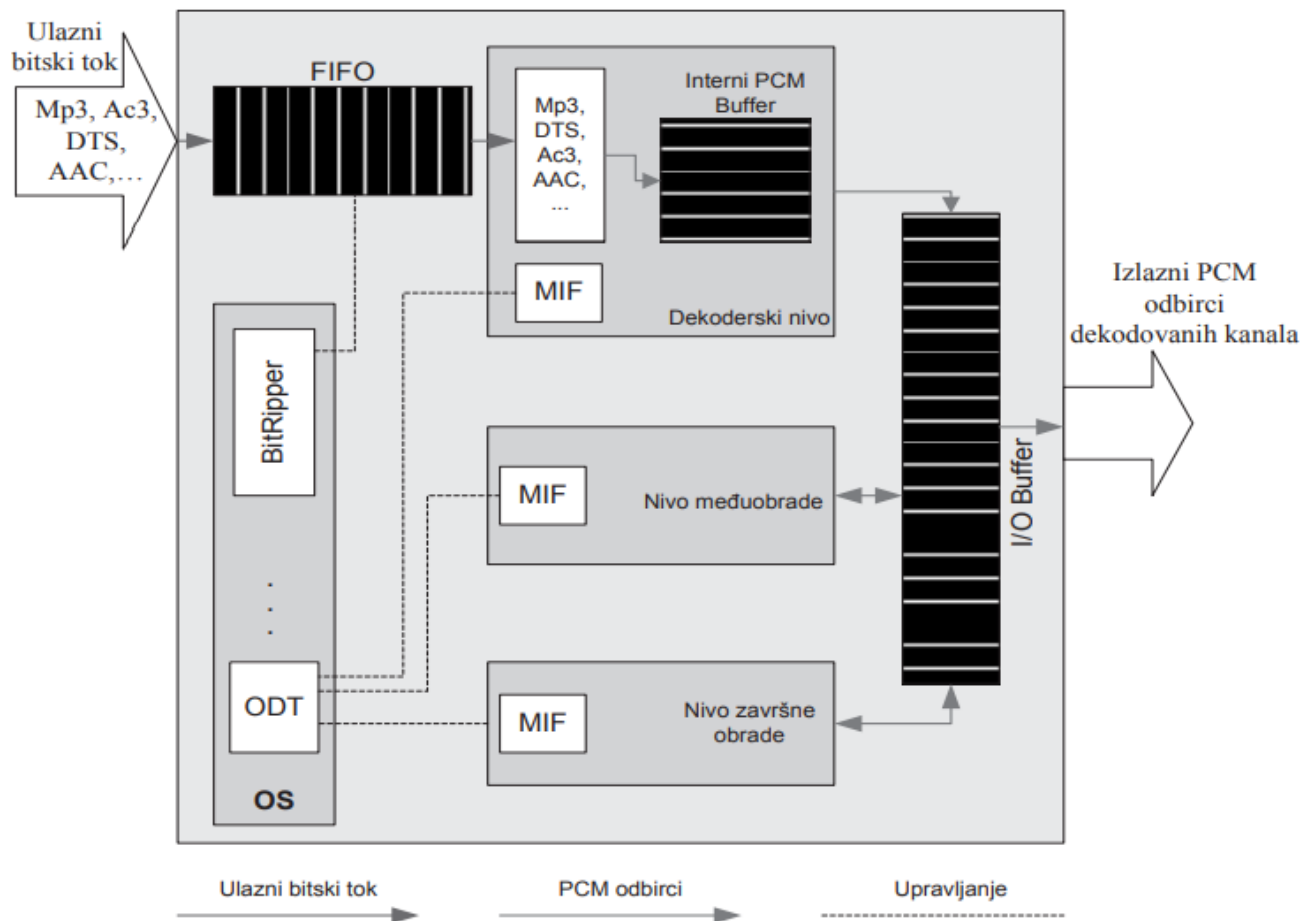


Sprega modula sa OS-om



Slika 7.2 – Blok dijagram sprege modula sa operativnim sistemom

Osnovna struktura OS-a (sa spregama)



Slika 7.6 - Sprega dekodera sa ulazom izlazom

Tri pitanja za integraciju modula



- ☐ 1. Kako znati da je došlo do određenog spoljnog događaja tj. gde ubaciti našu obradu, a gde ubaciti inicijalizaciju?
- ☐ 2. Kako pristupiti sistemskim IO baferima?
- ☐ 3. Kako reagovati na promene kontrola krajnjeg korisnika?

1. Spoljni događaji - rutine



- ❑ BRICK/BLOCK rutina – 16 pristiglih semplova u IO bafere, **tu ubaciti obradu podataka!**
- ❑ POST-KICK START rutina – poziva se jednom, nakon inicijalizacije OS-a, **tu ubaciti inicijalizaciju modula!**
- ❑ BACKGROUND rutina – izvršava se automatski kada se ni jedna druga rutina ne poziva, pogodna za ažuriranje MCV vrednosti!

2. Sprega sa IO baferima



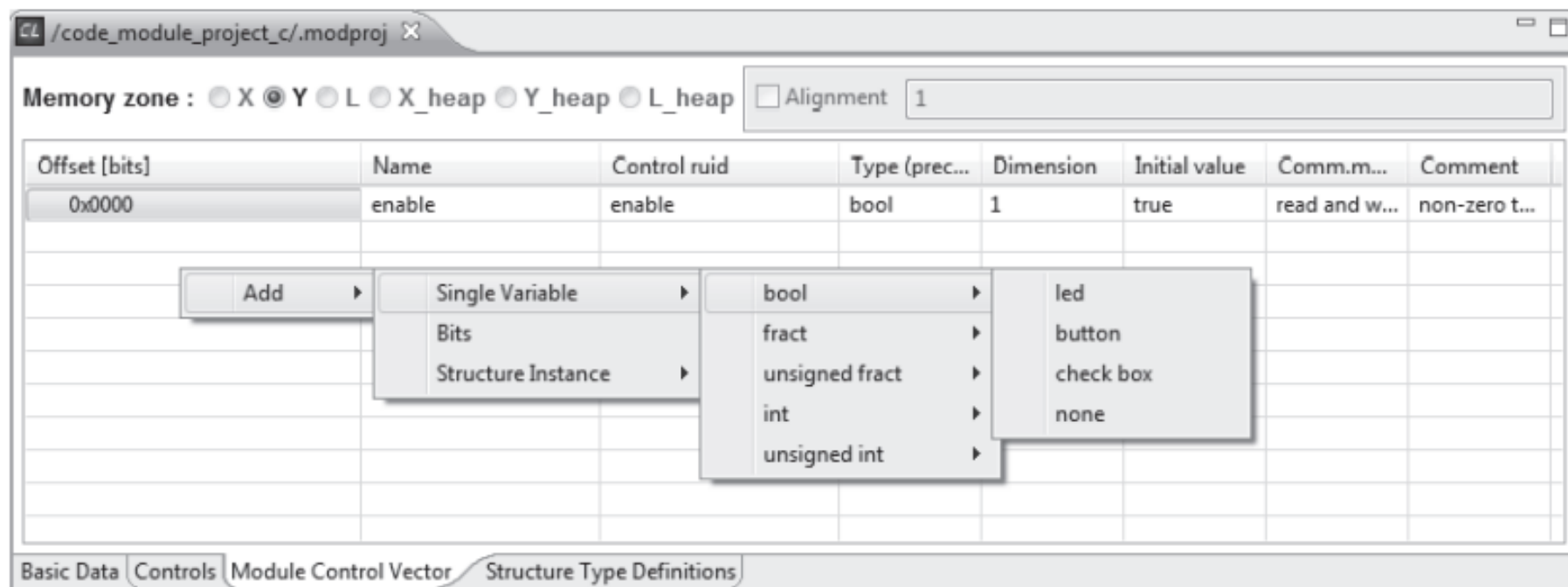
Tabela 7.2 – Pokazivači na ulazno-izlaznu sprežnu memoriju

| Rastojanje od prve lokacije | Ime pokazivača | Opis |
|-----------------------------|---|-------------------------------|
| 0 | X_BY_IOBuffer_Ptrs X_VY_IOBuffer_0_Ptr | Left Channel IO Buffer |
| 1 | X_VY_IOBuffer_1_Ptr | Center Channel IO Buffer |
| 2 | X_VY_IOBuffer_2_Ptr | Right Channel IO Buffer |
| 3 | X_VY_IOBuffer_3_Ptr | Left Surround IO Buffer |
| 4 | X_VY_IOBuffer_4_Ptr | Right Surround IO Buffer |
| 5 | X_VY_IOBuffer_5_Ptr | Surround Back Left IO Buffer |
| 6 | X_VY_IOBuffer_6_Ptr | Surround Back Right IO Buffer |
| 7 | X_VY_IOBuffer_7_Ptr | LFE0 IO Buffer |
| 8 | X_VY_IOBuffer_8_Ptr | Left High IO Buffer |
| 9 | X_VY_IOBuffer_9_Ptr | Right High IO Buffer |
| 10 | X_VY_IOBuffer_10_Ptr | Left Wide IO Buffer |
| 11 | X_VY_IOBuffer_11_Ptr | Right Wide IO Buffer |
| 12 | X_VY_IOBuffer_12_Ptr | Left DualZone IO Buffer |
| 13 | X_VY_IOBuffer_13_Ptr | Right DualZone IO Buffer |
| 14 | X_VY_IOBuffer_14_Ptr | Left Auxiliary IO Buffer |
| 15 | X_VY_IOBuffer_15_Ptr | Right Auxiliary IO Buffer |

ASM ime: **__X_BY_IOBUFFER_PTRS**

Relativni offset koristiti za ostale: **__X_BY_IOBUFFER_PTRS + 1, 2, N...**

3. Sprega sa krajnjim korisnikom



Slika 7.11 – Uređivač MCV tabele (dodavanje nove promenljive)