

**Arbitru de bus**

**Student:**

**Gheorghe Ștefan**

**BRAȘOV, 2020**

Cuprins

[1. Specificații 3](#_Toc37683568)

[2. Detalii de implementare 3](#_Toc37683569)

[1. Privire de ansamblu 3](#_Toc37683570)

[2. Modulul strict\_priority\_logic 4](#_Toc37683571)

# Specificații

Arbitrul de Bus implementat și descris în acest document prezintă următoarele specificații:

* Sistem sincron, frecvența 50MHz oferită de placa Spartan 3E.
* Reset asincron, acvtiv pe frontul pozitiv.
* Gestionarea a maximum 4 clienți; în cazul utilizării unui număr mai mic de clienți, aceștia se vor conecta începând cu portul 4 spre portul 1.
* Implementarea protocolului request-acknowledge.
* Posibilitatea de a alege între doi algoritmi de arbitrare, Strict Priority și Round Robin.
* În cazul utilizării algoritmului Strict Priority se poate seta pentru fiecare port/client nivelul de prioritate.

Pentru testarea modulului este implementat și un modul client ce poate genera cereri de scriere/citire la intervale aleatoare și de asemenea un modul de memorie ce va juca rolul serverului. Pentru fiecare instanță a modulului client trebuie specificat spațiul de adresare al acestuia.

# Detalii de implementare

## Privire de ansamblu

În Figura 1 este prezentată arhitectura modulului implementat, iar în subcapitolele următoare este prezentat în detaliu fiecare bloc component. De asemenea, va fi prezentată în detaliu și implementarea modulului client împreună cu blocurile sale componente.

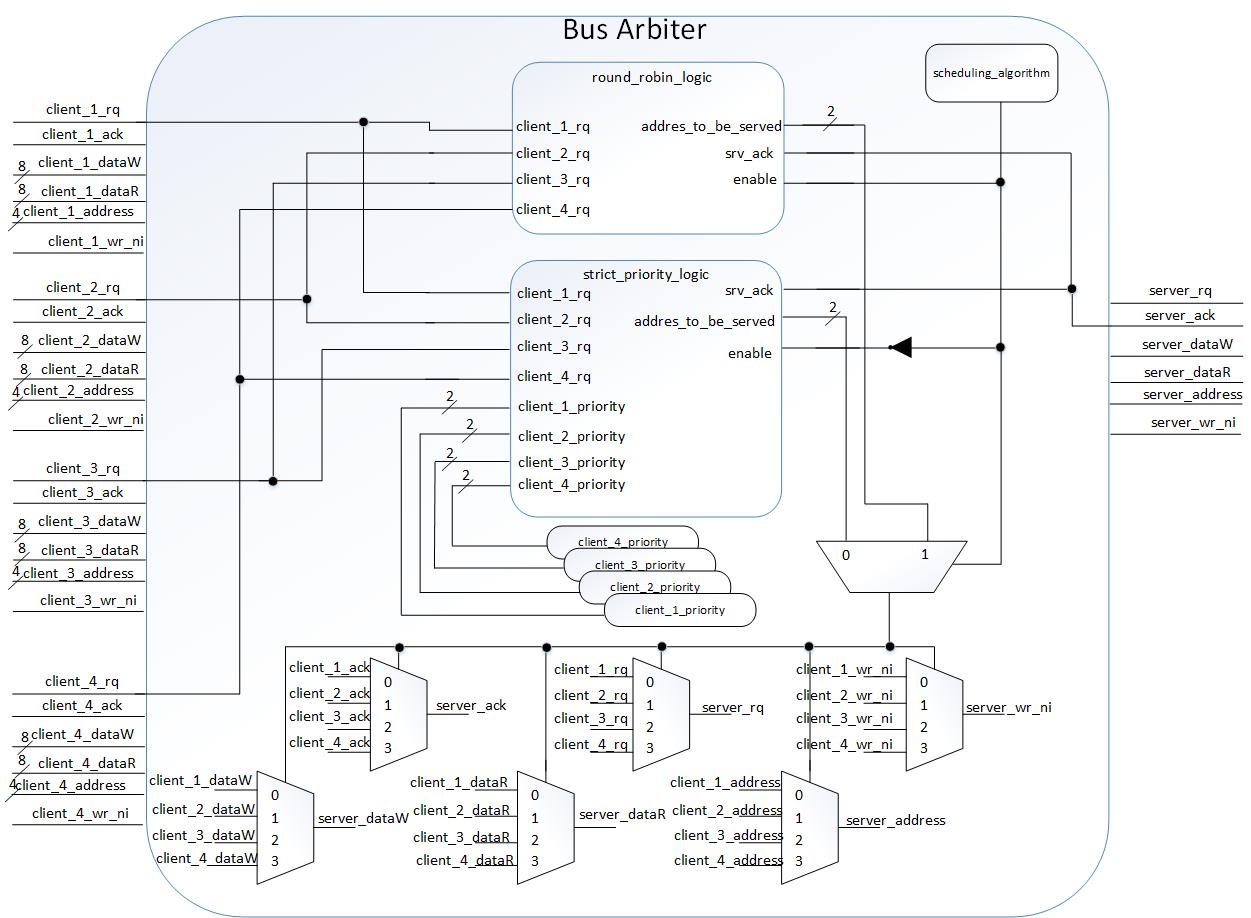


Figura 1: Arhitectură Bus Arbiter

## Descrierea interfețelor

Modulul Bus Arbiter dispune de 4 porturi la care modulul client se poate conecta și un port către server, așa cum este prezentat în Tabelul 1. Dimensiunile pentru semnalele de date/adrese mentionate în tabel sunt parametrizate, ele se pot modifica ulterior.

|  |  |  |  |
| --- | --- | --- | --- |
| Numele semnalului | Tip | Dimensiune | Descriere |
| clk | I | 1 | Ceasul sistemului. |
| reset | I | 1 | Reset activ 1. |
| server\_address | O | 4 | Adresa de memorie transmisă către server. |
| server\_rq | O | 1 | Semnalul de request transmis către server. |
| server\_ack | I | 1 | Semnalul de acknowledge transmis de la server. |
| server\_wr\_ni | O | 1 | Semnalul de scriere/citire transmis către server. |
| server\_dataW | O | 8 | Datele ce trebuie scrise în memorie. |
| server\_dataR | I | 8 | Datele citite din memorie. |
| client\_X\_address | I | 4 | Adresa de memorie transmisă de către client pe portul X. |
| client\_X \_rq | I | 1 | Semnalul de request transmis de către client pe portul X. |
| client\_X \_ack | O | 1 | Semnalul de acknowledge transmis spre client pe portul X. |
| client\_X \_wr\_ni | I | 1 | Semnalul de scriere/citire transmis de către client pe portul X. |
| client\_X \_dataW | I | 8 | Datele ce trebuie scrise în memorie de la clientul X. |
| client\_X \_dataR | O | 8 | Datele citite din memorie în urma cererii clientului X. |

Tabelul 1: Descrierea interfețelor modulului

## Modulul strict\_priority\_logic