

**Universitatea  
Transilvania  
din Braşov**

**FACULTATEA DE INGINERIE ELECTRICĂ  
ŞI ŞTIINŢA CALCULATOARELOR**

# Arbitru de bus

**Student:**

**Gheorghe Ştefan**

**BRAŞOV, 2020**

## Cuprins

1. Specificații.....	3
2. Detalii de implementare.....	3
1. Privire de ansamblu .....	3
2. Descrierea interfețelor .....	4
3. Modulul strict_priority_logic .....	5
4. Modulul round_robin_logic.....	7
5. Modulul client.....	8
3. Precizări finale .....	9

## 1. Specificații

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

- Sistem sincron, frecvența 50MHz oferită de placa Spartan 3E.
- Reset asincron, activ 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.

## 2. Detalii de implementare

### 1. 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.

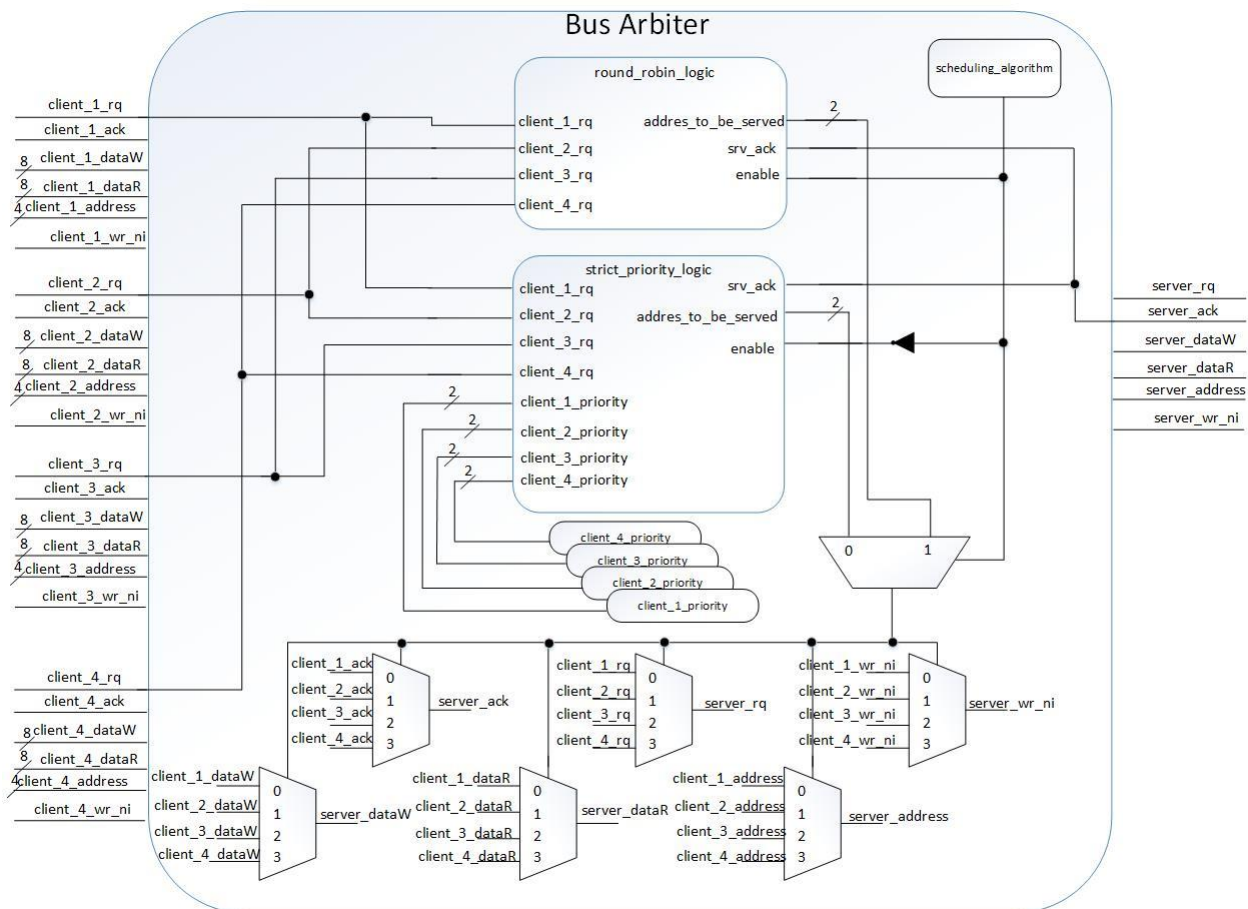


Figura 1: Arhitectură Bus Arbiter

## 2. 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 menționate 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	0	1	Semnalul de scriere/citire transmis către server.
server_dataW	0	8	Datele ce trebuie scrise în memorie.
server_dataR	1	8	Datele citite din memorie.
client_X_address	1	4	Adresa de memorie transmisă de către client pe portul X.
client_X_rq	1	1	Semnalul de request transmis de către client pe portul X.
client_X_ack	0	1	Semnalul de acknowledge transmis spre client pe portul X.
client_X_wr_ni	1	1	Semnalul de scriere/citire transmis de către client pe portul X.
client_X_dataW	1	8	Datele ce trebuie scrise în memorie de la clientul X.
client_X_dataR	0	8	Datele citite din memorie în urma cererii clientului X.

Tabelul 1: Descrierea interfețelor modulului

### 3. Modulul strict\_priority\_logic

Acest modul, prezentat în Figura 2, implementează algoritmul Strict Priority. Modulul este parametrizabil, fiind posibilă asignarea de către utilizator a priorității fiecărui canal în parte.

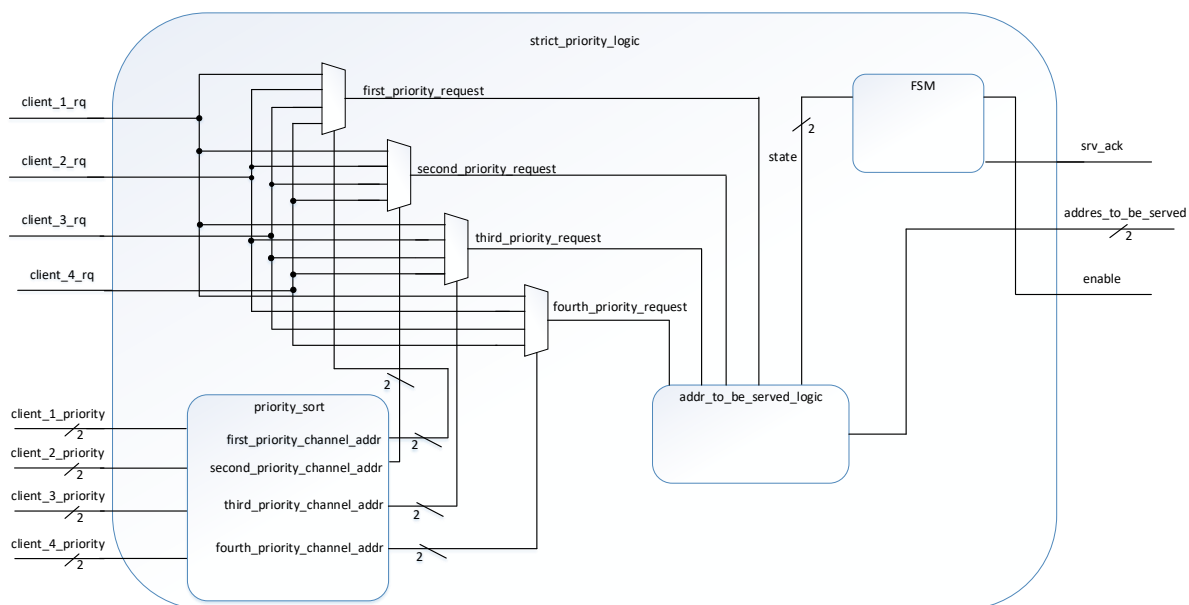


Figura 2: Modulul strict\_priority\_logic

În Figura 3 este prezentat graful de tranziție a stărilor pentru modulul menționat anterior, iar Figura 4 reprezintă formele de undă obținute în urma simulării. Toate formele de undă prezentate în acest document au fost obținute prin simularea modulului folosind următorii parametri:

```
localparam CLIENT_1_PRIORITY = 1;
localparam CLIENT_2_PRIORITY = 2;
localparam CLIENT_3_PRIORITY = 3;
localparam CLIENT_4_PRIORITY = 0;
localparam NR_OF_CLIENTS    = 4;
```

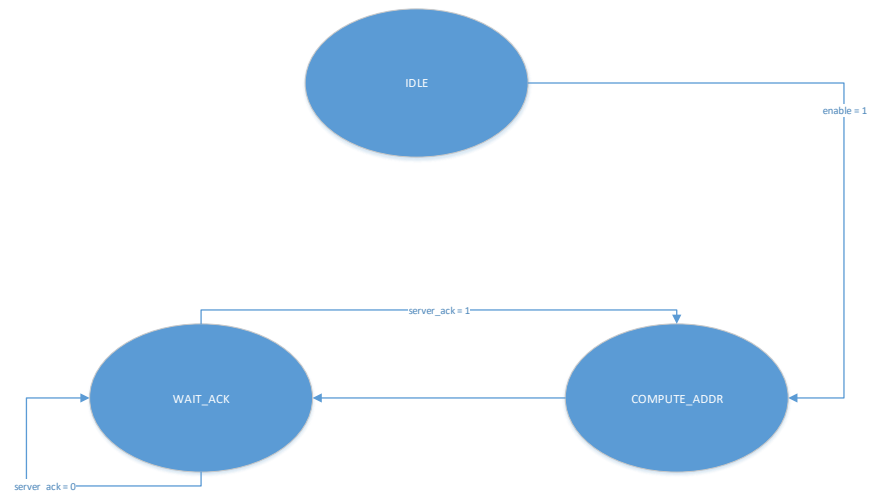


Figura 3: Graful de tranziții pentru modulul strict\_priority\_logic

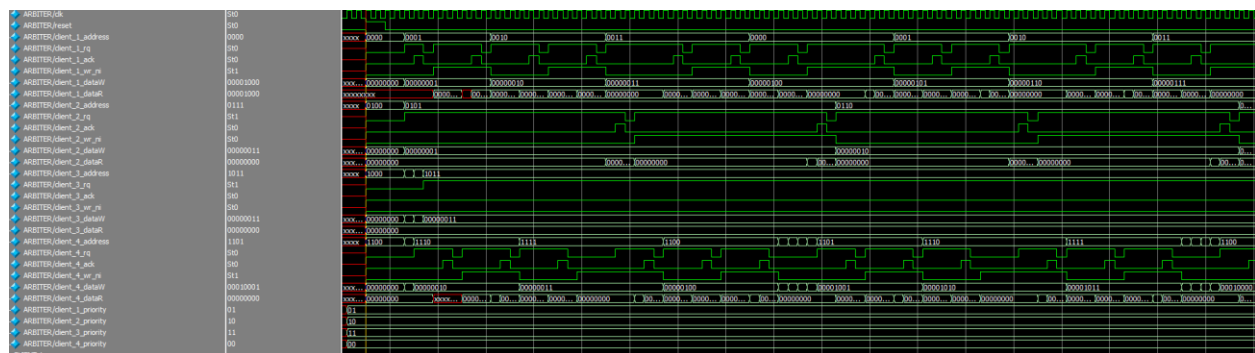


Figura 4: Formele de undă obținute în urma simulării modulului strict\_priority\_logic

#### 4. Modulul round\_robin\_logic

Acest modul, așa cum este prezentat în Figura 5, este alcătuit din patru module strict\_priority\_logic și un numărator circular. Acesta este folosit pentru a activea fiecare modul strict\_priority\_logic în parte. Figura 6 reprezintă formele de undă obținute în urma simulării.

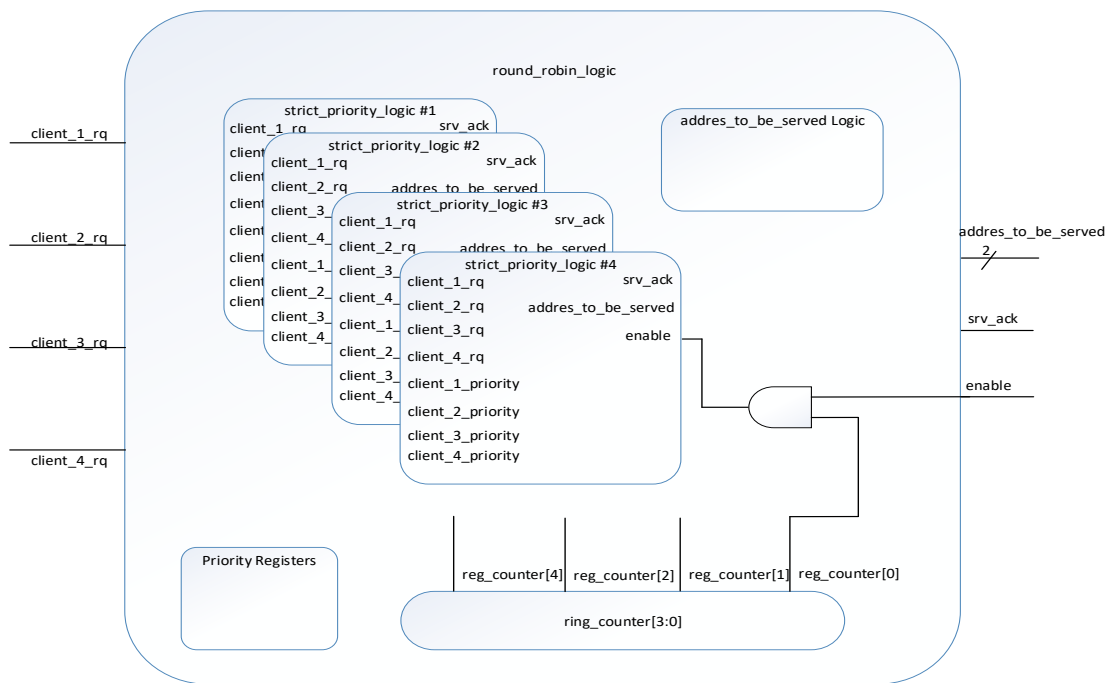


Figura 5: Arhitectura modulului round\_robin\_logic

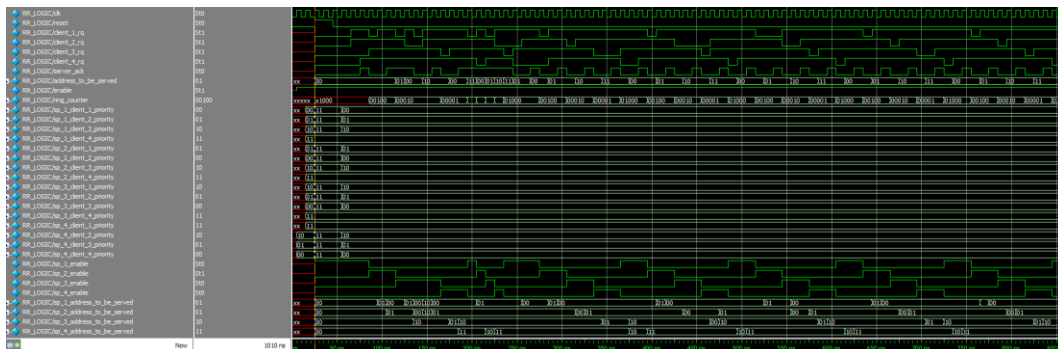


Figura 6: Formele de undă obținute în urma simulării modulului round\_robin\_logic

## 5. Modulul client

Modulul client este folosit pentru testarea arbitrului de bus. Acesta a fost implementat pentru a genera cereri către server în mod aleator. Pentru aceasta, modulul utilizează un LFSR și primește ca parametru un seed. Acesta poate fi diferit pentru fiecare instanță a modulului client.

Fiecărei instanțe a modulului client i se poate atribui un spațiu de adrese. Modulul va trimite cereri către server doar cu adrese cuprinse în spațiul specificat.

Adresele sunt generate de un numărător, la fel ca și datele transmise pentru scriere. Aspectul aleator al cererilor face referire la intervalele de timp la care sunt transmise cererile.

Pentru verificarea funcționalității modulului bus arbiter, cererile transmise către server sunt generate în felul următor: se generează o cerere de scriere la o anumită adresă, se așteaptă soluționarea cererii, apoi se transmite o cerere de citire la aceeași adresă.

Arhitectura modulului client, precum și graful de tranziție a stărilor sunt prezentate în Figura 7, respective Figura 8.

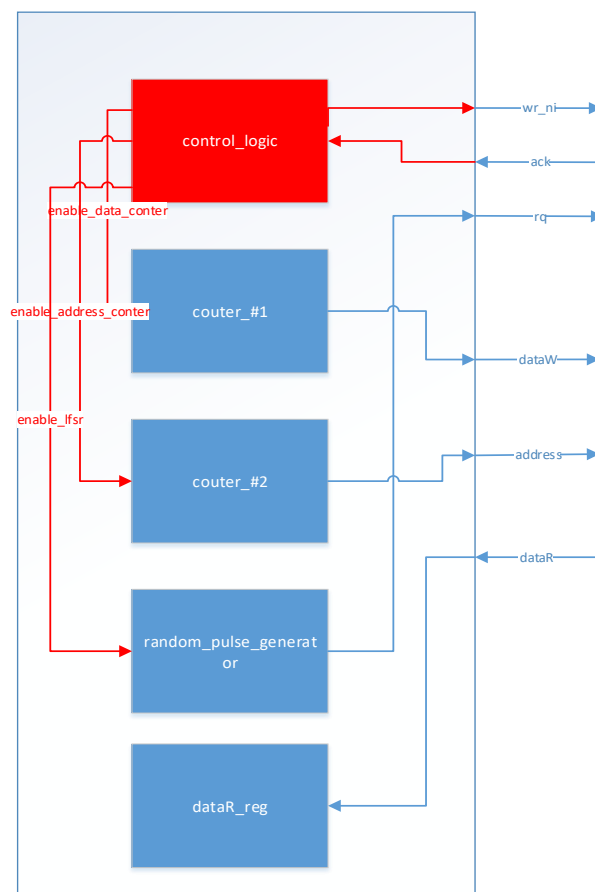


Figura 7: Arhitectura modulului client



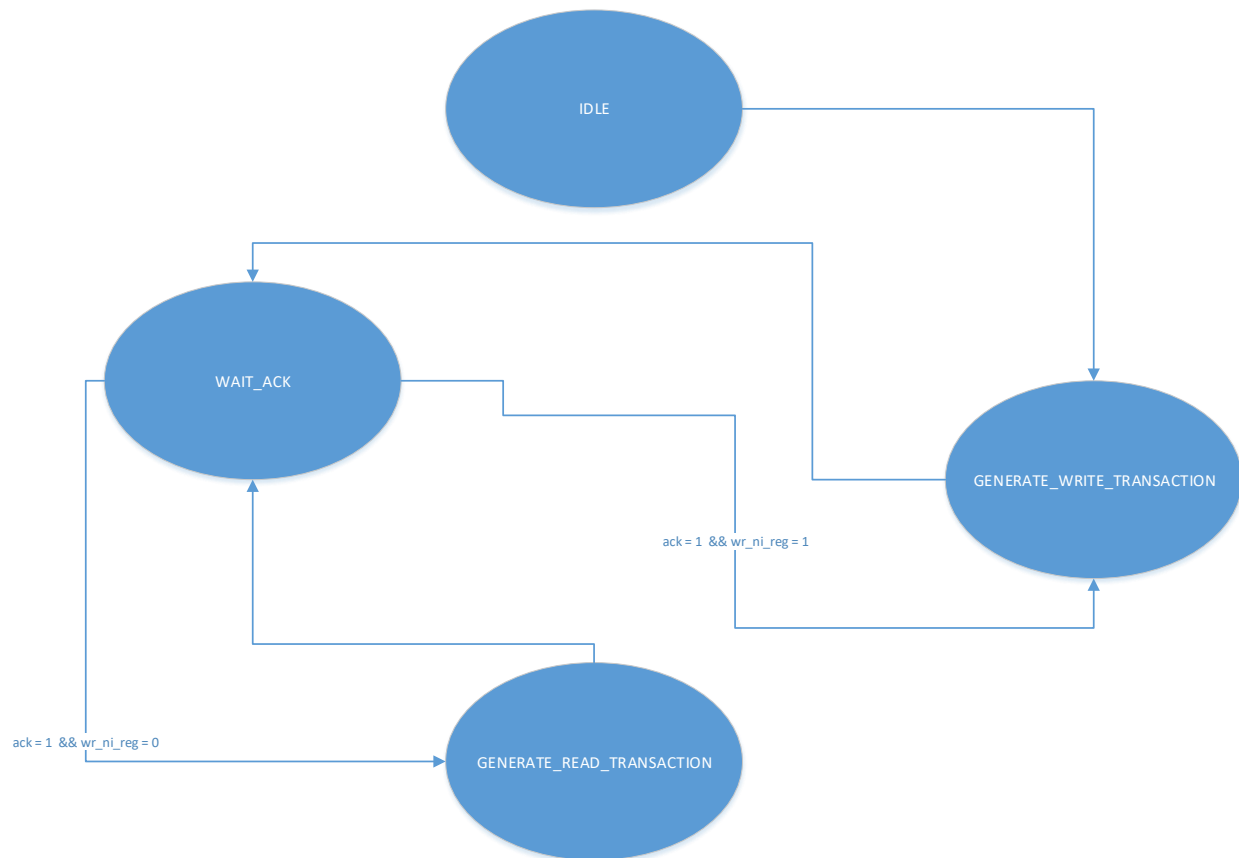


Figura 8: Graful de tranziție a stărilor pentru modulul client

### 3. Precizări finale

Împreună cu documentația prezentă, arhiva conține și codul sursă al proiectului, precum și două script-uri pentru simulare ce se găsesc în fișierul `sim`. Fișierul `run_client_memory_sim.do` rulează o simulare fără arbitru de bus, cu un singur client și memoria, în timp ce `run_system_test.do` simulează întregul sistem.

Fișierul `system-test.v` reprezintă mediul de testare. De aici se pot modifica parametrii testului (prioritatea clienților, numărul lor, algoritmul folosit, spațiul de adrese etc.).