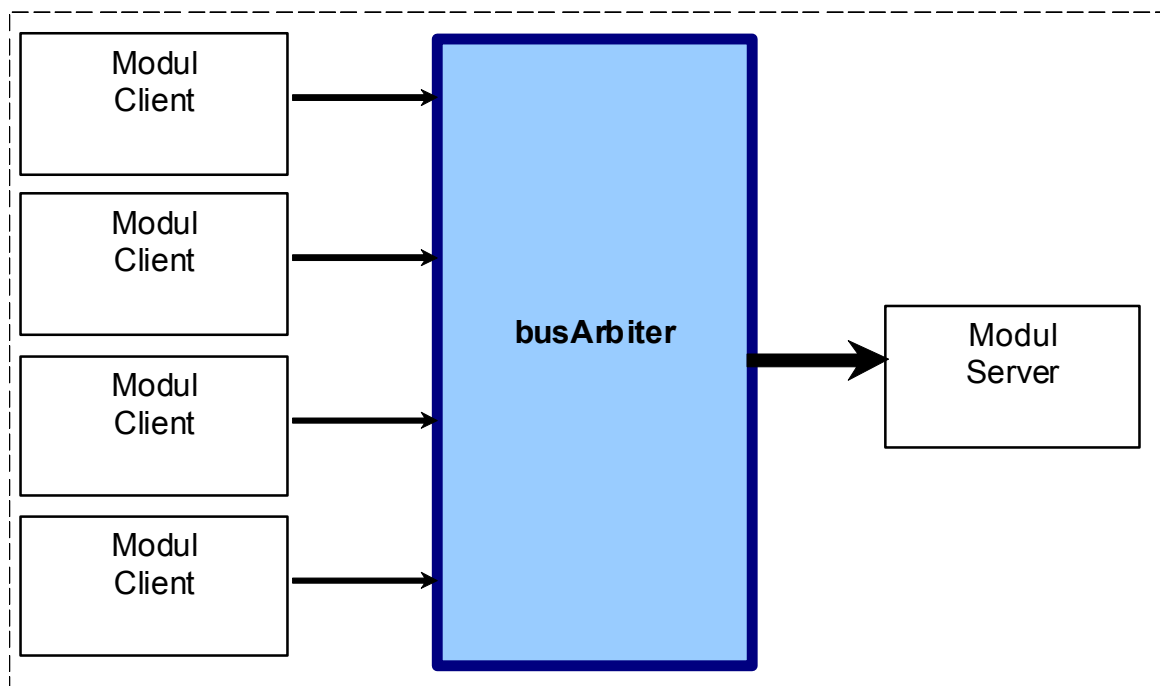


## Arbitru de bus (busArbiter)



*Arbitru de bus: poziție în cadrul sistemului.*

### Temă

- Proiectarea şi implementarea în FPGA a unui arbitru de bus ce funcţionează pe baza unui protocol de tip cerere-confirmare.

### Cerinţe

- Sistem sincron, frecvenţa 50MHz.
- Reset asincron.
- *Arbitrul* va putea gestiona până la 4 *clienţi*. Se va specifica modul de conectare a intrărilor în cazul inexistenţei unor clienţi.
- Se vor prezenta forme de undă cu scenarii de arbitrare.
- Se va testa arbitrul prin simulare în conjuncţie cu module *client* şi *server*. Opţional, se va propune un modul de verificare a protocolului pe bus. Se va asigura verificarea 100% a codului RTL.
- Se va realiza un prototip implementat pe o placă de demonstraţie cu FPGA.

## Descrierea porturilor

Port	Descriere
<b>Clock/Reset</b>	
clk_i	Ceas de frecvenţă 50MHz
reset_i	Reset activ în 1
<b>Interfaţă server</b>	
rq_o	Cerere spre server, activă în 1, stabilă până la primirea confirmării.
ack_i	Confirmare de la server, un puls de o perioadă de tact, activă în 1.
wr_no	Tipul cererii 1=citire, 0=scriere.
dataW_o[width-1:0]	Date scrise (include toate informaţiile de la client spre server, necesare pentru satisfacerea cererii).
dataR_i[width-1:0]	Date citite (include toate informaţiile de la server spre client, ca urmare a satisfacerii cererii).
<b>Interfeţe clienţi</b>	<b>&lt;X&gt; = 0, 1, 2, 3</b>
rq<X>_i	Cerere de la clientul <X>.
ack<X>_o	Confirmare pentru clientul <X>.
wr<X>_ni	Tipul cererii clientului <X> 1=citire, 0=scriere.
dataW<X>_i[width-1:0]	Date scrise preluate de la clientul <X>.
dataR<X>_o[width-1:0]	Date citite livrate către clientul <X>.

## Caracteristici *busArbiter*

- Facilitate de scriere într-un fişier extern a informaţiilor despre starea cererilor primite şi servite.

<client> <timp> <acţiune>

<acţiune> :: = <primireCerere> | <lansareCerere> | <confirmareCerere>

## Modul *client*

- Facilitate de generare de cereri deterministe şi aleatorii.
- Facilitate de preluare a datelor scrise dintr-un fişier extern şi se stocare a datelor citite în alt fişier extern.
- Generare de cereri deterministe şi aleatorii.

## Modul *server*

- Interfaţare cu un dispozitiv de stocare a datelor capabil să răspundă coerent la cereri de scriere şi citire corespunzătoare.
- Opţional, modulul are interfaţă cu o memorie bloc implementată în FPGA sau externă, existentă pe macheta de prototip.

## Predare

- Proiectul se va prezenta personal într-un dosar. Codul va fi listat cu font *Courier New*, 10pt.
- Paginile proiectului vor fi legate în dosar.
- Se va evalua profesionalismul prezentării rezolvării.

### Recomandări

- Numele alese pentru blocurile funcţionale vor sugera rolul acestora. Fiecare modul va fi descris într-un fişier separat. Fişierul va avea numele identic cu cel al modulului conţinut.
- Desenele se vor redacta ***exclusiv de mână, cu creionul***.
- Fiecare figură va conţine la bază un text explicativ.
- Testarea se va efectua iniţial prin simulare într-un mediu cu un singur client şi ulterior cu patru clienţi.

### Bibliografie

- ["Spartan-3E Starter User Guide"](#)
- ["Spartan-3E Documentation"](#)
- [http://www.xilinx.com/ise/logic\\_design\\_prod/webpack.htm](http://www.xilinx.com/ise/logic_design_prod/webpack.htm)

### Suport

- Email Dan NICULA