

Semnalizarea handshake pentru transferuri de date

Probleme propuse

Oprîtoiu Flavius
flavius.opritoiu@cs.upt.ro

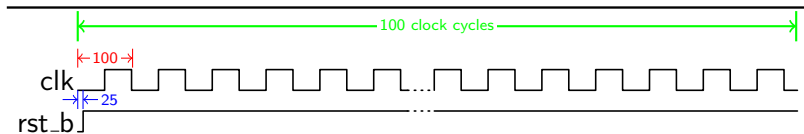
18 septembrie 2023

Problema 1

Construiți un modul numit *prod* (producător) având intrările *clk* (1-bit) și *rst_b* (1-bit) și ieșirile *val* (1-bit) și *data* (8-biți). Modulul produce valori aleatorii între 0 și 5. Ieșirea *val* (semnal handshaking *valid*) anunță când unitatea nu poate produce o nouă valoare (*val* == 0).

Modulul generează date valide cel puțin 3 și cel mult 5 cicluri de tact după care va aștepta (fără să genereze date) cel puțin 1 și cel mult 4 cicluri de tact. Construiți:

1. codul Verilog
2. fișierul script, "run_prod.txt"
3. un testbench care generează intrările ca în diagrama de timp de mai jos

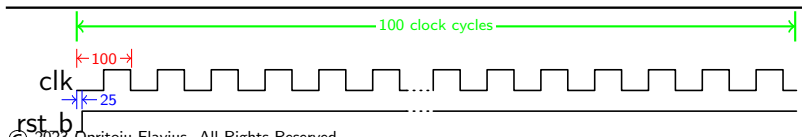


Problema 2

Construiți un consumator numit *cons* (consumator) având intrările *clk* (1-bit), *rst_b* (1-bit), *val* (1-bit) și *data* (8-biți) și având o ieșire *sum* (8-biți). Unitatea consumă date generate de producător. La ieșirea *sum*, consumatorul furnizează suma secvențelor monoton crescătoare de date valide recepționate (o dată validă este parte a unei secvențe monoton crescătoare dacă este mai mare sau egală cu anterioara dată validă). Testbench-ul va instanția producătorul de la Exercise 1 și va conecta cele 2 module.

Construiți:

1. codul Verilog
2. fișierul script, "run_cons.txt"
3. un testbench care generează intrările ca în diagrama de timp de mai jos



Problema 3

Construiți unitatea de control descrisă în Problema rezolvată a materialului din această săptămână, numită *sha2inctrl*.

Testbench-ul va instanția modulul *sha2indpath* construit săptămâna trecută, care va fi controlat de această unitate de control și va conecta cele 2 module.

Construiți:

1. codul Verilog
2. fișierul script, "run_sha2inctrl.txt"
3. un testbench care generează intrările ca în diagrama de timp de mai jos

