Gheorghiță Alexandru

Grupa 2

dOCUMENTAȚIE PROEICET

**Unitate de preprocesare a pachetelor**

Detalii Proiect

* Acest proiect descrie implementarea unei unități de preprocesare a pachetelor cu anumite specificații. Unitatea va avea patru porturi de intrare: clk (un bit), pkg (32 de biți), rst (un bit) și en (un bit) și două porturi de ieșire: data\_out (512 de biți) și data\_avail (un bit).
* Modulul va concatena valorile pachetelor primite prin intrarea pkg atâta timp cât intrarea en este 1. Dacă intrarea en este 0, modulul nu va continua concatenarea, dar va păstra toate concatenările precedente și va relua concatenarea atunci când intrarea en devine 1 din nou.
* Modulul va concatena un pachet la fiecare front crescător al semnalului clk. În acest timp, ieșirea data\_out va avea valoarea precedentă, iar data\_avail va avea valoarea 0
* Initial, toate ieșirile sunt 0. După 16 cicluri ale semnalului clk, ieșirea data\_out va conține concatenarea tuturor celor 16 pachetelor, în ordine inversă (ultimul pachet primit va fi cel mai la dreapta și primul pachet primit va fi cel mai la stânga). După 16 cicluri ale semnalului clk, ieșirea data\_avail va fi 1, doar pentru o perioadă a semnalului clk.
* Când intrarea rst este 0, modulul se va reseta, punând toate ieșirile pe 0 și va relua procesul de concatenare.
* Proiectul include o provocare de a utiliza un numărător pentru ciclurile de clk și un decodificator pentru a selecta care registru intern va reține valoarea pkg. Este recomandat să se folosească bistabile de tip D și intrările load și rst pentru registrele interne.

Proiectul va conține 3 module:

* Preprocesor
* Regiștri
* Decodificator

Decodificator

* Un decodificator este un circuit electronic care transformă un număr binar de intrări într-un număr binar de ieșiri. În acest proiect, decodificatorul va fi utilizat pentru a selecta registrul intern care va reține valoarea pkg, ținand cont de numărul ciclurilor de clk.
* Există mai multe tipuri de decodificatoare, cum ar fi decodificatoare binare, octale sau decimale, dar cele mai frecvente sunt decodificatoarele binare. Acestea au un număr n de intrări și 2^n ieșiri, unde n este numărul de intrări.
* Funcționarea unui decodificator binar este următoarea: pentru fiecare combinație posibilă de intrări, un singur bit al ieșirii este setat la 1 și restul biților ieșirii sunt setați la 0. Așadar, dacă avem 2 intrări, decodificatorul va avea 4 ieșiri și pentru fiecare combinație posibilă de intrări, o singură ieșire va fi setată la 1
* În cazul acestui proiect, daca se va utiliza un decodificator cu 4 intrari si 16 iesiri , va permite selectarea unui singur registru din cele 16 registre disponibile la fiecare ciclu de clk, în funcție de combinația de intrări..

Registri

Modulul de registri din acest proiect se va ocupa de stocarea valorilor pachetelor primite prin intrarea "pkg" și de transferarea acestor valori către ieșirea "data\_out" dupa un numar de cicluri ale semnalului de intrare clk.

* In functie de specificatiile proiectului, modulul poate utiliza registre interne, care pot fi bistabile de tip D sau alte tipuri de registre, care vor stoca valorile pachetelor primite prin intrarea "pkg" si vor fi utilizate pentru a forma ieșirea "data\_out" dupa un numar specific de cicluri ale semnalului de clk.
* In plus, modulul de registri poate include și o funcționalitate de load, care va permite transferul valorilor din registre către ieșirea "data\_out" și o funcționalitate de reset, care va permite reinitializarea registrelor interne la valori predefinite.
* Acest modul poate include un decodificator și un numărător pentru a selecta registrul care trebuie să rețină valoarea pkg și pentru a determina când să fie transferate valorile din registre către ieșirea data\_out.
* In general, modulul de registri va fi responsabil pentru stocarea valorilor pachetelor primite prin intrarea "pkg" și de transferarea acestor valori către ieșirea "data\_out" după un număr specific de cicluri ale semnalului de clk. Va fi responsabil pentru a gestiona aceste valori și pentru a le pregăti pentru ieșirea "data\_out", ținând cont de condițiile de operare specificate în proiect, cum ar fi ciclurile de clk necesare pentru transfer sau resetarea registrelor.

Preprocesare

Modulul de preprocesare din acest proiect se va ocupa de prelucrarea pachetelor primite prin intrarea "pkg" și de furnizarea unui flux de date prelucrat către ieșirea "data\_out".

* In functie de specificatii, modulul poate concatena valorile pachetelor primite prin intrarea "pkg" când intrarea "en" este 1. Când intrarea "en" este 0, modulul nu va continua concatenarea, dar va păstra toate concatenările precedente și va relua concatenarea atunci când intrarea "en" devine 1 din nou. Acest modul poate include și o funcționalitate de reset care va permite reinitializarea modulului la valorile implicite.
* Modulul poate include si un mecanism de numarare a ciclurilor de clk pentru a determina când să fie furnizate date prelucrate prin ieșirea "data\_out" și când să fie setat semnalul de "data\_avail" la 1.
* In general, modulul de preprocesare va fi responsabil pentru a prelucra pachetele primite prin intrarea "pkg" și pentru a furniza un flux de date prelucrat către ieșirea "data\_out" în conformitate cu specificațiile proiectului, inclusiv concatenarea pachetelor și gestionarea ciclurilor de clk.

Referinte:

OUTPUT 512 BIT BIT

REGISTRI

16 FIECARE

INPUT 32 biti

INTRARE 3

INTRARE 2

INTRARE 1

INTRARE 0

**DECODIFICATOR**

1] Sintaxa pentru front crescator sau descrescator, <https://docs.google.com/document/d/1zLWCyt6pa1UYGo3qWaocylgxtgLo-_dlxReI5XQfzN4/edit#heading=h.ufi7ra1ho1f6>

[2] Proiect numarator, <https://docs.google.com/document/d/1TKwx22QjCWmcVsYhEmYE04cUEr2_vvDpHMQll1kUQB0/edit>

[3] Decodificator, <https://docs.google.com/document/d/1USuli6aD4ie0y6K6gSdZ4_vIY0W8LLafRfGke8JQQDI/edit#heading=h.fpm65xat7rlq>

[4] Bistabile, <https://docs.google.com/document/d/1Eq7M9C-ACNcU-ZdQmThcXE-1WnBg_rN3rGWumh3tB78/edit#heading=h.azxuuyrx7h66>

[5] Mircea Vlăduțiu:  “Computer Arithmetic. Algorithms and Hardware Implementations” Springer-Verlag, Heidelgerg, New York, Dordecht, London, 2012, ISBN 978-3-642-18314-0, ISBN 978-3-642-18315-7 (<http://www.springer.com/computer/hardware/book/978-3-642-18314-0>).

[6] Mircea Vlăduţiu: „Arhitectura şi organizarea calculatoarelor” Vol.1: Aritmetica sistemelor de calcul (monografie), Editura Politehnica Timişoara, 2008 (274pagini), ISBN 978-973-625-706-3 (general), ISBN 978-973-625-709-4 (vol. 1).

https://www.youtube.com/watch?v=NaO-xw6Mlw0

Referinte externe

Testbench public

[***https://drive.google.com/file/d/1MggQ9\_qhqa9iwKI7XKl59QiQcj7BGgFh/view?usp=sharing***](https://drive.google.com/file/d/1MggQ9_qhqa9iwKI7XKl59QiQcj7BGgFh/view?usp=sharing)