DIV_ALGO

Pentru algoritmul de impartire am adaptat long division gasit pe linkul de wiki de la capitolul resurse.

In general am scris cam acelasi lucru cu ce era pe Wikipedia, mai putin if-ul de la final care se datoreaza faptului ca “i” nu putea lua valoarea -1, pentru ca este initializat ca reg, asa ca am mai executat o data iteratia din for, separat. Deorece R si Q nu sunt de tipul reg nu le puteam folosi in interiorul lui always, asa ca m-am folosit de variabilele cat si rest, care au fost asignate lui Q, respectiv R.

BASE2_TO_BASE3

Am declarat toate variabilele de care aveam nevoie, dupa am creat o instanta de div_algo. Primul always reprezinta partea secventiala, iar al doilea pe cea combinationala.

In starea 0 se fac initializarile si se trece mai departed doar daca “en” este 1.

In starea 1 se executa impartirea deoarece x primeste o valoare noua.

In starea 2 se retin cei doi biti din rest in variabila auxiliara “base3_no_r” cu ajutorul explicatiei de la sfarsitul suportului pentru tema. Am nevoie de un “i” care sa se incrementeze cu cate 2 unitati de fiecare data astfel incat sa populez urmatorii 2 biti ai lui “base3_no_r”. Se trece in starea 3 doar daca prin impartire s-a ajuns la catul 0, in caz contrar programul se intoarce in starea 1.

Pentru selecția unei secțiuni dinamice de 8 biți a unei variabile folosită într-o buclă iterativă, construcția aux[i+7:i] nu este permisă. Este necesară folosirea operatorului +: sau -:, astfel: aux[i+7-:8] sau aux[i+:8]. În acest fel sunt selectați biții din intervalul i, i+7. Mai multe detalii în Standardul Verilog 2000, Indexed Vector Part Selects, din resurse

In starea 3 variabila “out” devine 1, deci in consecinta si “done” devine 1, apoi se trece din nou la starea 0 si tot asa mai departe.