

## Laborator 2 -Structura și Organizarea Calculatoarelor- ~Raport tehnic~

### *Implementarea și simularea unui sumator pe 4 biți*

#### *Aspecte teoretice:*

Sumatorii reprezintă componente fundamentale ale aritmeticii binare și sunt folosiți într-o gamă largă de aplicații, cum ar fi calculatoare, procesoare și alte dispozitive electronice. Un sumator pe 4 biți este o componentă care poate aduna patru biți individuali și poate genera un rezultat de 4 biți și un bit de transport. Scopul acestui raport tehnic este de a descrie modul în care un sumator pe 4 biți poate fi implementat cu ajutorul a 4 sumatoare pe un bit și de a prezenta simularea circuitului utilizând Xilinx și ModelSim.

Pentru a construi sumatorul pe 4 biți, putem conecta în cascada cele 4 sumatoare pe un bit astfel încât bitul de transport generat( $Co$ ) de un sumator pe un bit să fie conectat la bitul de intrare de transport( $Ci$ ) al sumatorului pe un bit următor. În acest fel, putem aduna 4 biți individuali și să generăm un rezultat de 4 biți și un bit de transport final.

Un sumator binar pe 4 biți are trei intrări:

- $A_3, A_2, A_1$  și  $A_0$ : primul număr( $A$ ) în format binar reprezentat pe 4 biți
- $B_3, B_2, B_1$  și  $B_0$ : al doilea număr( $B$ ) în format binar reprezentat pe 4 biți
- $Ci$ : semnalul carry-in de la bitul anterior (valoarea sa este 0 sau 1)

Și două ieșiri:

- $S_3, S_2, S_1$  și  $S_0$ : suma binară( $S$ ) a valorilor de intrare reprezentată în sistemul binar pe 4 biți
- $Co$ (Carry-out): valoarea de transport ieșită reprezentată în sistemul binar pe un singur bit care indică dacă adunarea celor doi termeni produce o valoare care nu poate fi reprezentată pe 4 biți

În primul sumator pe un bit,  $A_0$  și  $B_0$  vor fi adunate împreună cu  $Ci$ (care inițial este 0 deoarece acesta este conectat la ground) și vor genera un bit de rezultat  $S_0$  și un bit de transport  $Co_0$ .  $S_0$  va fi conectat la ieșirea pentru bitul de sumă mai puțin semnificativ, iar  $Co_0$  va fi conectat la bitul de transport de intrare pentru sumatorul pe un bit următor.

În al doilea sumator pe un bit,  $A_1$  și  $B_1$  vor fi adunate împreună cu  $Co_0$  și vor genera un bit de rezultat  $S_1$  și un bit de transport  $Co_1$ .  $S_1$  va fi conectat la ieșirea pentru următorul bit de sumă, iar  $Co_1$  va fi conectat la bitul de transport de intrare pentru sumatorul pe un bit următor.

Acest proces va continua până la al patrulea sumator pe un bit, unde  $A_3$  și  $B_3$  vor fi adunate împreună cu  $Co_2$  și vor genera un bit de rezultat  $S_3$  și un bit de transport  $Co_3$ .  $S_3$  va fi conectat la ieșirea pentru bitul de rezultat cel mai semnificativ, iar  $Co_3$  va fi conectat la bitul de transport de ieșire final, care reprezintă bitul de transport pentru întregul sumator pe 4 biți.

### Un exemplu de intrare a sumatorului pe 4 biți:

A = 1111 și B = 1101

Vom introduce A și B în intrările A3, A2, A1, A0, respectiv B3, B2, B1, B0 ale sumatorului. Pentru a putea realiza operația de adunare, am stabilit starea inițială a Ci0=0.

A3 = 1; A2 = 1; A1 = 1; A0 = 0

B3 = 1; B2 = 1; B1 = 0; B0 = 1

Ci0=0 (deoarece este conectat la ground(GND)- 0-logic)

Mai întâi, adunăm biții de pe poziția 0,  $B_0 + A_0 + Ci_0 = 10$  (adică 0 cu o depășire de 1) => S0=0 și Co0=1 (carry-ul care este dus către poziția următoare(Ci1));

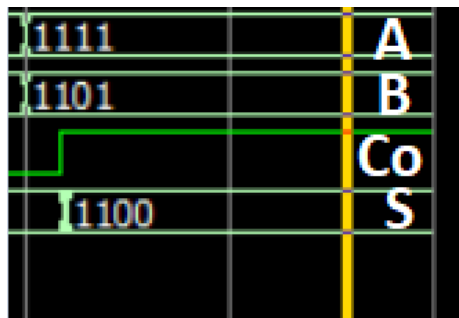
Apoi, vom aduna biții de poziția 1,  $B_1 + A_1 + Ci_1 = 10$  (adică 0 cu o depășire de 1) => S1=0 și Co1=1 (carry-ul care este dus către poziția următoare(Ci2));

Urmează să adunăm biții de pe poziția 2,  $B_2 + A_2 + Ci_2 = 11$  (adică 1 cu o depășire de 1) => S2=1 și Co2=1 (carry-ul care este dus către poziția următoare(Ci3));

Iar la final, adunăm biții de pe poziția 3,  $B_3 + A_3 + Ci_3 = 11$  (adică 1 cu o depășire de 1) => S3=1 și Co3=1

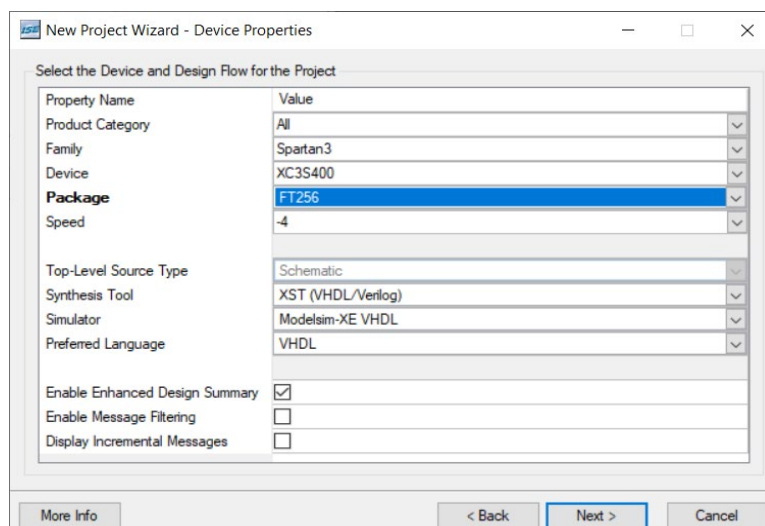
Rezultatul final: S = 1100 și Co=1

Rezultatul din Modelsim:

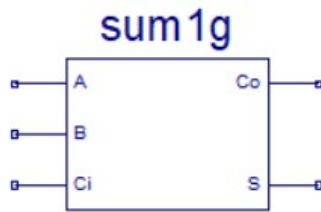


### Pașii proiectului:

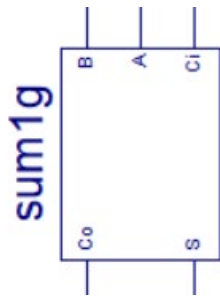
Pentru început am particularizat pentru a putea fi implementata pe placa Spartan 3.



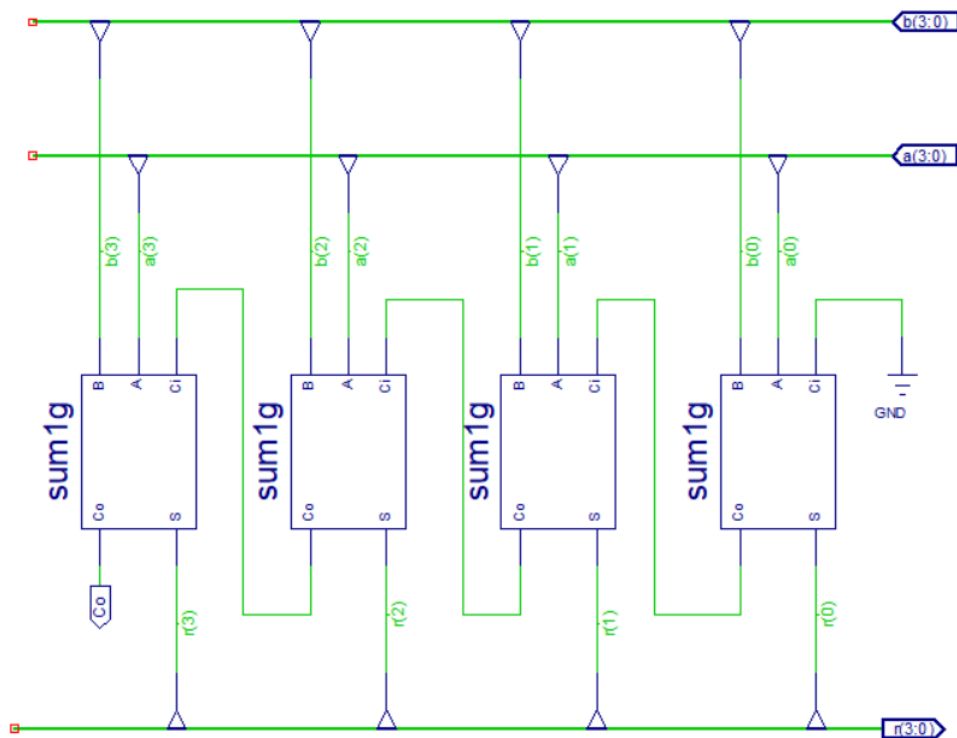
Pentru început am adăugat o copie a sursei din laboratorul precedent(sumatorul pe un bit) si am creat o schema bloc a acesteia.



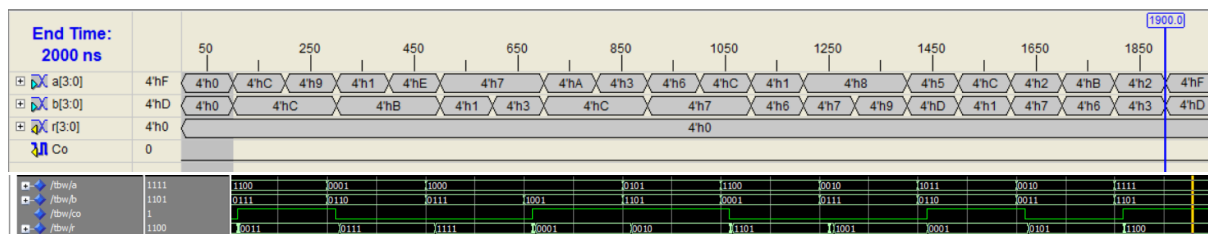
Apoi, am făcut modificările indicate pentru ca schema sa fie clară si vizibilă.



Am adăugat 4 sumatoare pe un bit. Am adăugat 3 magistrale(2 de intrare pe 4 biți a(3:0), b(3:0) si una de ieșire pe 2 biți r(3:0)-care reprezintă suma pe 4 biți). Am conectat Ci0 la 0-logic(ground), iar transportul generat de la primul sumator(Co0) l-am introdus ca intrare carry-in pentru cel de-al doilea sumator(Ci1), la fel am procedat si pentru restul. Iar apoi am adăugat marker de ieșire pentru C0(carry-out)



Corectitudinea sumatorului a fost verificată cu ajutorul Modelsimului, în Test Bench WaveForm. Testbench-ul wave generează două numere binare de 4 biți (A și B) și le aplică ca intrare în circuit. Am adăugat câteva combinații dintre cele 256(deoarece avem 9 intrări, dar una dintre ele este conectată la 0-logic=>  $2^8$ ) și le-am verificat. Am descoperit că circuitul funcționează corect și generează rezultatul așteptat.



Această metodă poate fi aplicată și pentru adunarea a două numere pe mai mult de 4 biți, prin adăugarea de sumatori pe un bit în cascadă. Totuși, această metodă este relativ lentă, deoarece bitul de carry trebuie să treacă prin toți sumatorii în lanț.