

Laborator 1 -Structura și Organizarea Calculatoarelor-

Descrierea sumatorului pe un bit cu porți logice

Un sumator binar pe un bit este un circuit logic care efectuează adunarea a două numere binare de un singur bit (0 sau 1) și generează un rezultat și o valoare de transport. Acest raport tehnic are ca scop prezentarea funcționării și implementării unui astfel de sumator.

Un sumator binar pe un bit are trei intrări:

- A: primul număr de adunat reprezentat în sistemul binar pe un singur bit
- B: al doilea număr de adunat reprezentat în sistemul binar pe un singur bit
- Ci(Carry-in): valoarea de transport intrată de la o etapa anterioară a operației de adunare reprezentată în sistemul binar pe un singur bit care trebuie adăugată la suma celor două numere.

Și două ieșiri:

- S: suma binară a valorilor de intrare A, B și valoarea de transport intrată Ci reprezentată în sistemul binar pe un singur bit
- 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 1 bit. Co este același lucru cu transportul, pe care îl învățăm în școala primară. De exemplu, în adunarea numerelor 7 și 8, transportul este 1, iar suma este 5

Un sumator pe un bit poate fi implementat cu ajutorul a cinci porți logice: 3 AND2, XOR3 și OR2.

Forme logice:

- $S = C_{i-1} \oplus A \oplus B$ -suma pe bitul curent este egală cu valoarea XOR între intrările A,B și carry-ul de intrare(Ci)
- $Co = AB + AC_i + BC_i$ -transportul către bitul următor este dat de o combinație de semnale AND, astfel încât să detecteze toate cazurile în care trebuie să se transporte o valoare în bitul următor.

Formele logice pot fi deduse foarte ușor cu ajutorul diagramei KV și a tabelului de adevăr(Tabel 1).

A	B	Ci	S	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0

0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1


Tabel 1

Diagrama KV conform tabelului:


Pentru suma (S):

A B		00	01	11	10
Ci	0	0	1	0	1
	1	1	0	1	0


A B		00	01	11	10
Ci	0	0	1	0	1
	1	1	0	1	0




$\bar{A}\bar{B}C_i$



$\bar{A}B\bar{C}_i$



ABC_i



$A\bar{B}\bar{C}_i$

$$\begin{aligned}
 S &= AABBC\bar{C}_i + AABBC\bar{C}_i + AABBC\bar{C}_i + AABBC\bar{C}_i \\
 &= C_i(AABB + AABB) + C\bar{C}_i(AABB + AABB) \\
 &= CCCC[(A\bar{A} + B\bar{B})(A\bar{A} + B\bar{B})] + CCCC(AABB + AABB) \\
 &= CCCC(AABB \cdot B\bar{A}\bar{A} + CCCC(AABB + AABB) \\
 &= CCCC(AABB + B\bar{A}\bar{A}) + CCCC(AABB + AABB) \\
 &= CCCC \oplus (AABB + AABB) \\
 &= CCCC \oplus AA \oplus BB
 \end{aligned}$$

Pentru carry-out(C0):

A B		00	01	11	10
Ci	0	0	0	1	0
	1	0	1	1	1

A B		00	01	11	10
Ci	0	0	0	1	0
	1	0	1	1	1

$\bar{A}BCi + ABCi$

$ABCi + ABC\bar{i}$

$ABCi + A\bar{B}Ci$

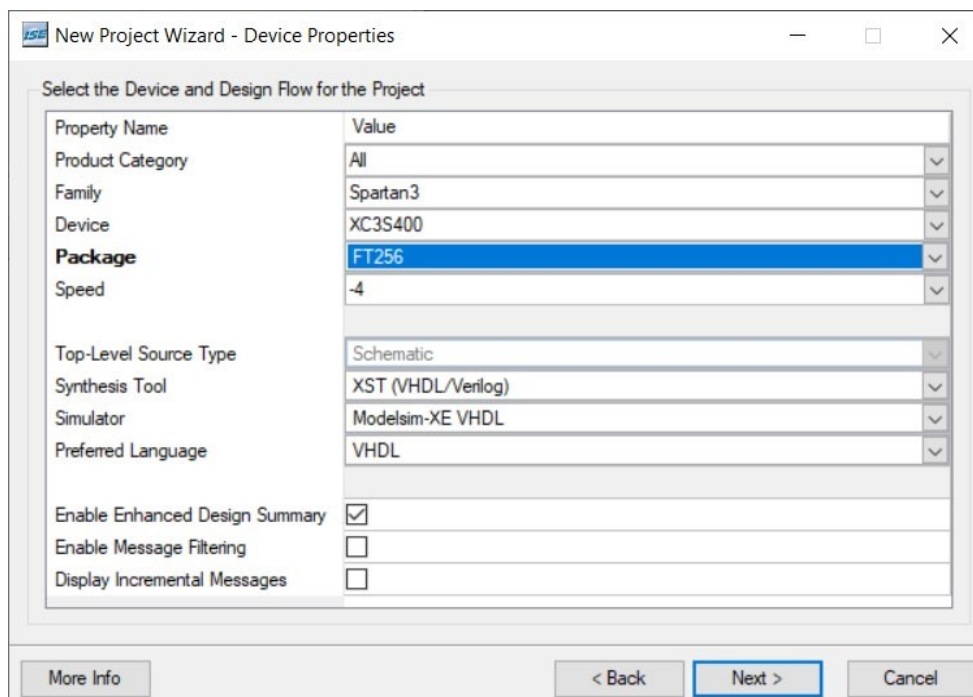
$$CCCC = AABBBCCC + AABBBCCC + AABBBCCC + AABBBCCC + AABBBCCC + AABBBCCC$$

$$= BBCCC(AA + AA) + AACCCC(BB + BB) + AAB(BCCC + CCCC)$$

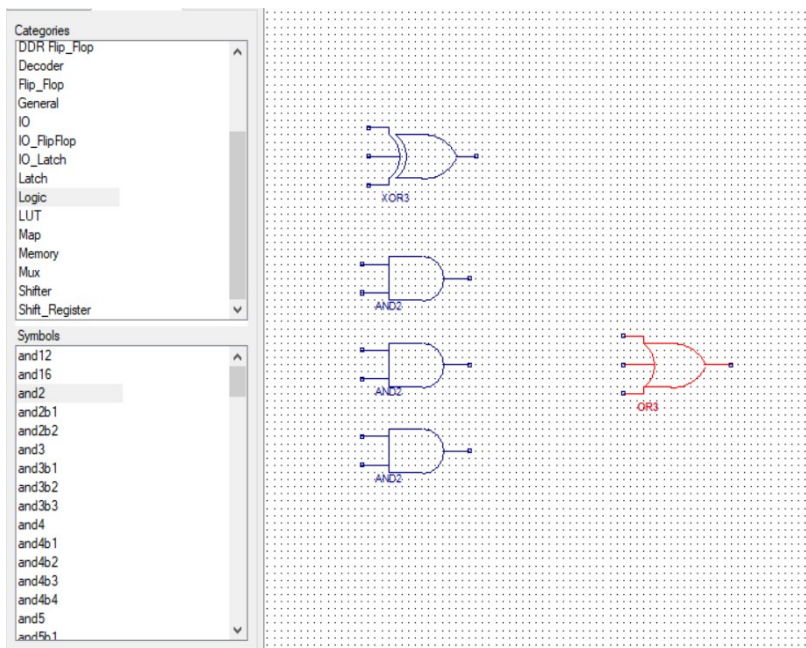
$$= BBCCC + AACCCC + AAB$$

Pașii proiectului:

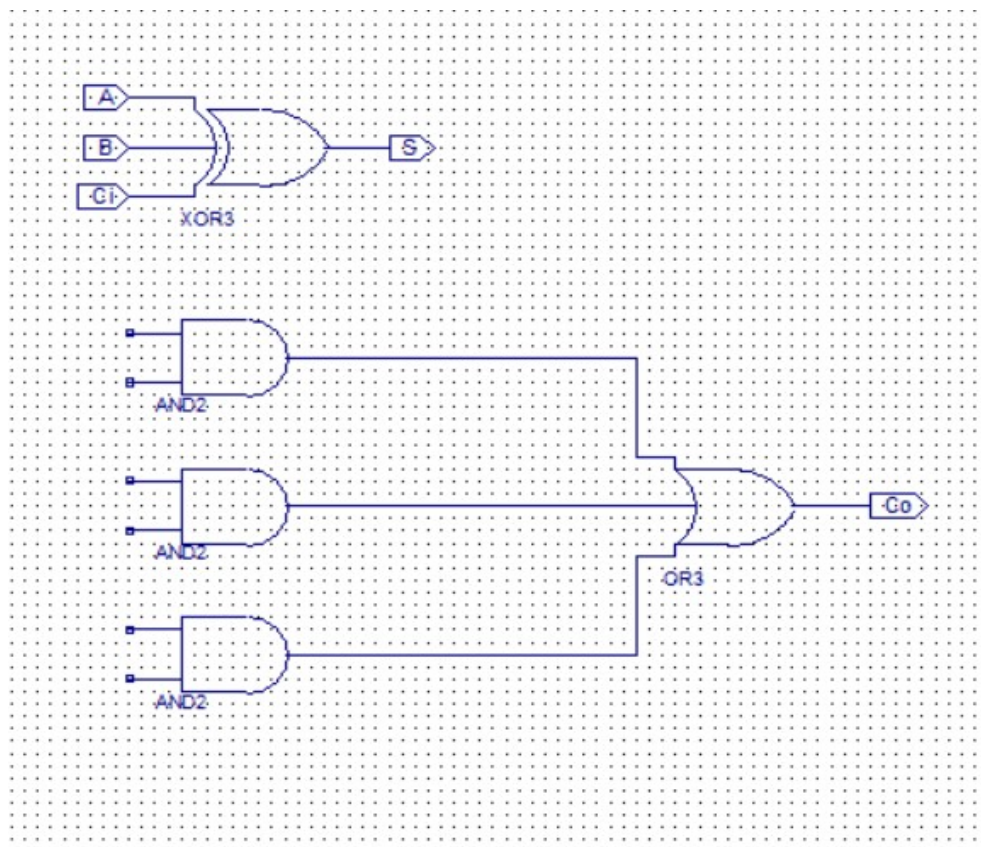
Pentru început am particularizat pentru a putea fi implementata pe placa Spartan 3.



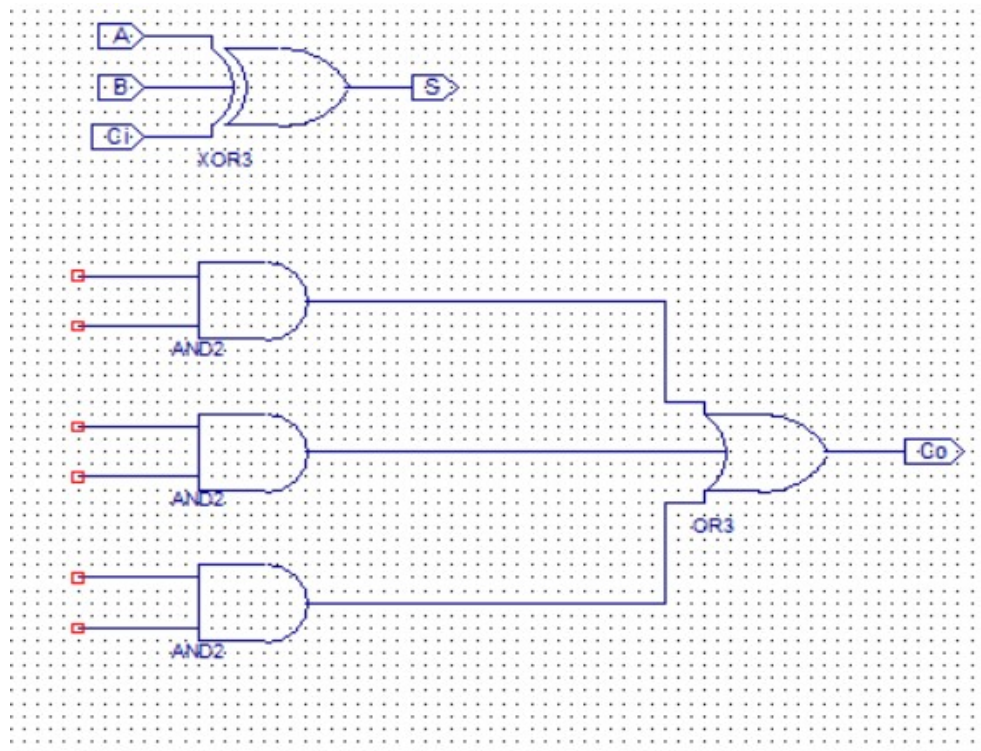
Am adăugat cele 5 porți logice (3 AND2, XOR2 SI OR3)



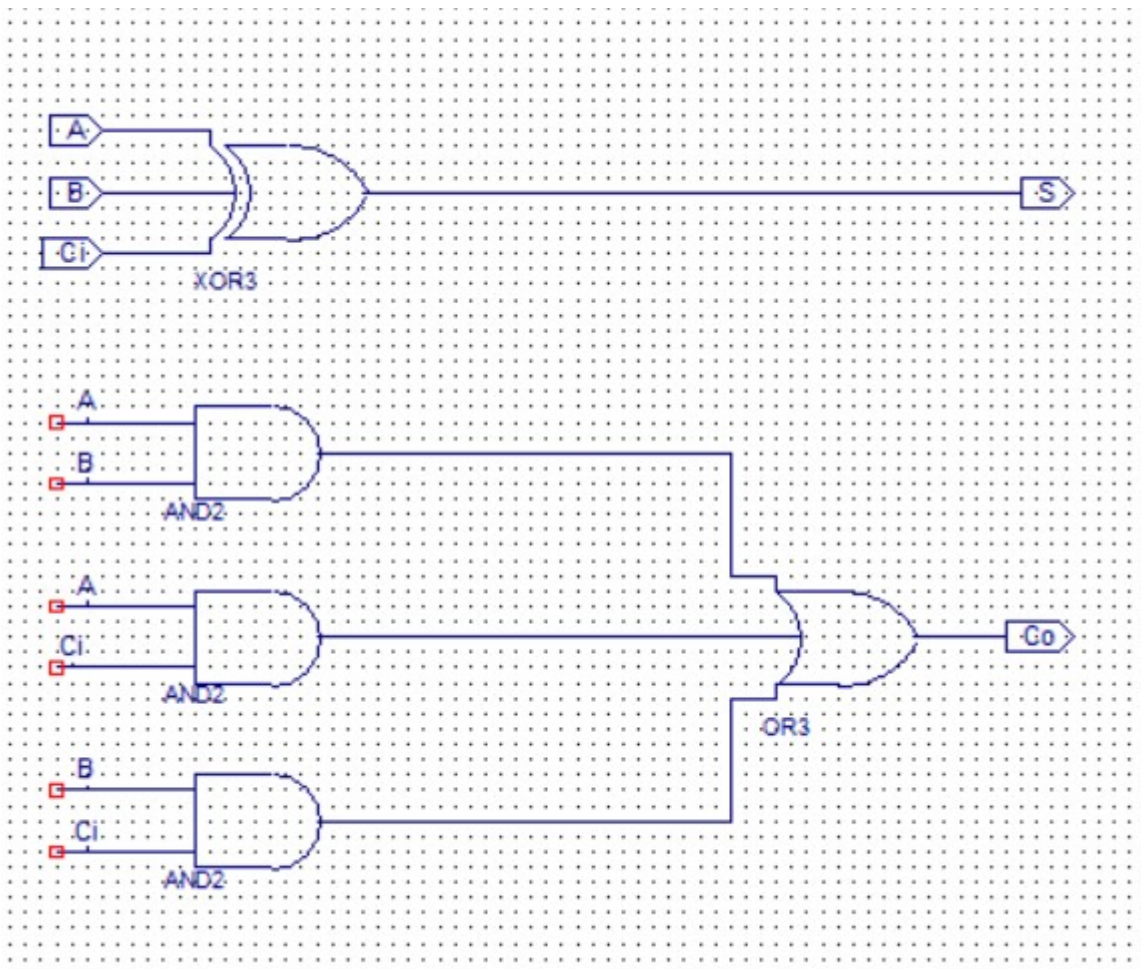
Am adăugat marcher I/O pentru poarta XOR3, urmând ulterior să le folosesc și pentru restul intrărilor.



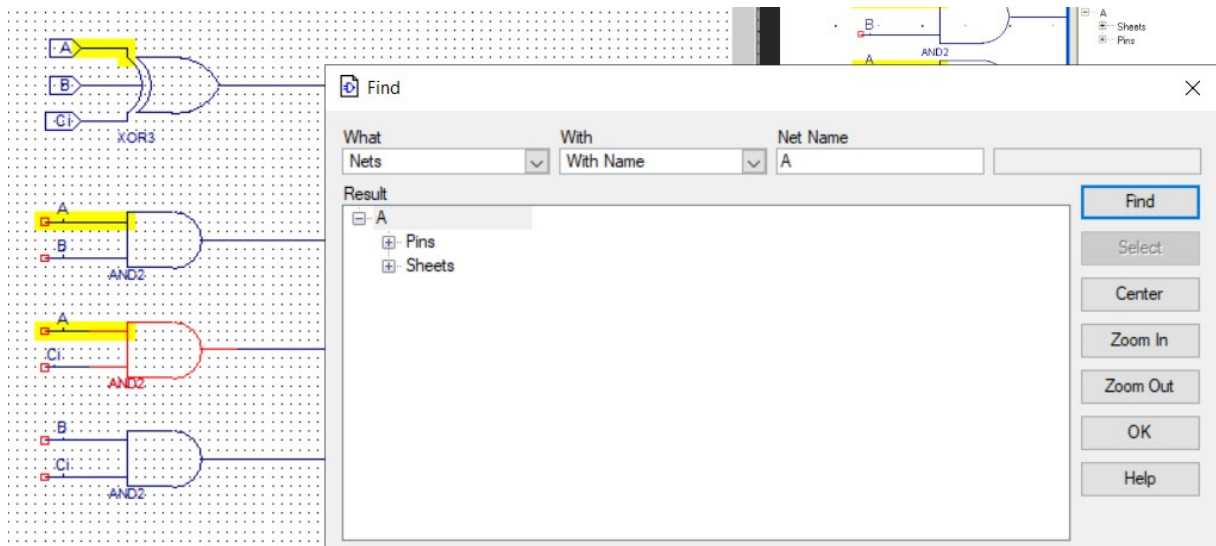
Am adăugat conexiunii terminate in aer pentru porțile AND, pentru a putea ulterior să le conectez la identificatorii de conexiune deja existenți (A, B și Ci).



Cu ajutorul funcției “Add net name” am adăugat identificatori de conexiune.

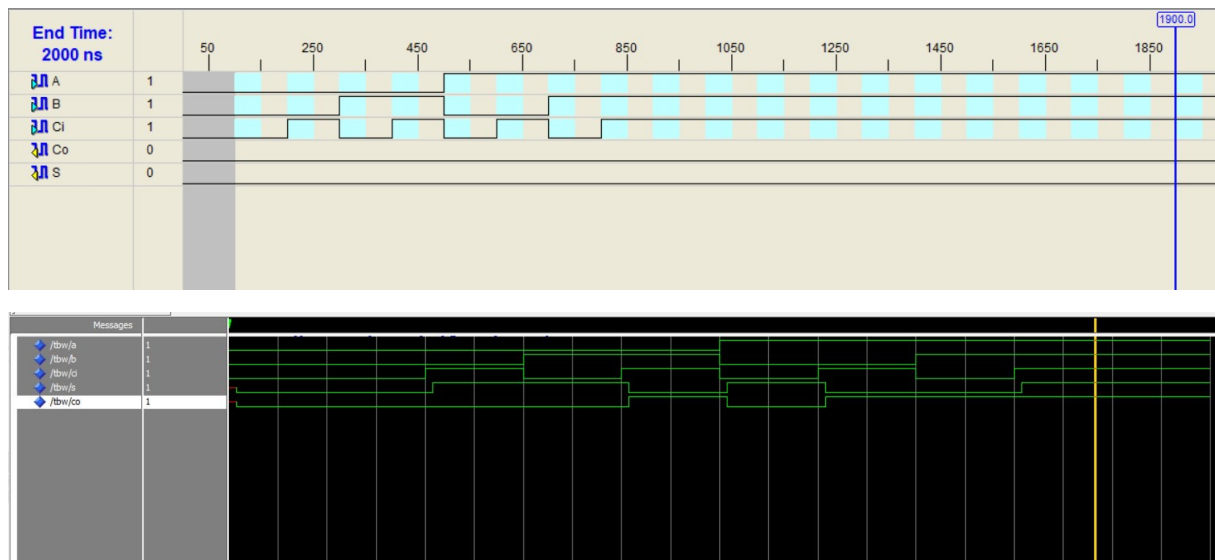


Iar apoi am verificat fiecare identificator(A, B, Ci) daca formează o conexiune.



În funcție de valorile de intrare, sumatorul pe un bit va produce o anumită valoare pentru ieșirile S și Co conform tabelului de adevăr(Tabel 1) care descrie operațiile logice care au loc în interiorul sumatorului.

Corectitudinea sumatorului a fost verificată cu ajutorul Modelsimului, in Test Bench WaveForm am adăugat toate cele 8 combinații (deoarece fiecare intrare poate avea doar doua valori posibile(1 sau 0), avem $2*2*2=8$ combinații posibile ale acestor intrări) si le-am verificat pe fiecare.



Sumatorul pe un bit este utilizat într-o varietate de aplicații electronice, cum ar fi:

- Înregistratoare de deplasare
- Contoare
- Multiplexoare
- Demultiplexoare
- Procesoare de semnal digital
- Microprocesoare

În concluzie, sumatorul pe un bit este un circuit electronic simplu, dar esențial, folosit pentru a efectua operația de adunare între două numere binare reprezentate pe un singur bit. Structura circuitului este compusă din porți logice AND, XOR și OR, iar operația sumatorului este definită prin două relații matematice. Sumatorul pe un bit este un element de bază pentru construirea altor circuite electronice mai complexe și este utilizat într-o varietate de aplicații electronice. Mai mulți sumatori pe un singur bit sunt combinați pentru a forma un sumator pe mai mulți biți, cum urmează în partea a doua a raportului. Sumatoarele pe mai mulți biți pot fi utilizate pentru a realiza operații aritmetice mai complexe, cum ar fi adunarea sau scăderea între numerele reprezentate pe mai mulți biți.

Descrierea sumatorului pe 2 biți cu porți logice

Sumatorul pe 2 biți poate efectua operații de adunare a două numere binare de câte 2 biți fiecare, precum și luarea în considerare a unui bit de transport provenit din operațiile anterioare.

Un sumator binar pe 2 biți are trei intrări:

- A1 și A0: primul număr în format binar reprezentat pe doi biți
- B1 și B0: al doilea număr în format binar reprezentat pe doi biți
- Ci: semnalul carry-in de la bitul anterior (valoarea sa este 0 sau 1)

A1 și B1 reprezintă cei mai semnificativi biți ai numerelor de intrare, iar A0 și B0 reprezintă cei mai puțin semnificativi biți.

Și două ieșiri:

- S1 și S0: suma binară a valorilor de intrare reprezentată în sistemul binar pe 2 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 2 biți

S1 și S0 reprezintă cei mai semnificativi și, respectiv, cei mai puțin semnificativi biți ai sumei numerelor de intrare.

Un sumator pe 2 biți este construit din 2 sumatoare pe un bit, care sunt conectate în cascadă. Primul sumator va aduna cei mai puțin semnificativi biți ai celor două numere de intrare și carry-in-ul, iar al doilea sumator va aduna cei mai semnificativi biți și carry-out-ul rezultat de la primul sumator.

Un exemplu de intrare a sumatorului pe 2 biți:

A = 11 și B = 11

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

A1=1; A0=1

B1=1; B0=1

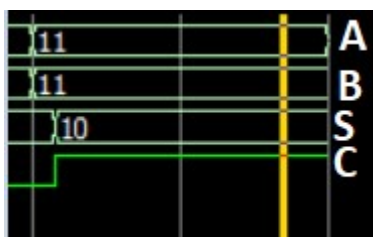
Ci0=0

Mai întâi, adunăm biții de pe poziția 0, $B_0 + A_0 + C_{i0} = 10$ (adică 0 cu o depășire de 1) => $S_0 = 0$ și $Co_0 = 1$ (carry-ul care este dus către poziția următoare(C_{i1}));

Apoi, vom aduna biții de poziția 1, $B_1 + A_1 + C_{i1} = 11$ (adică 1 cu o depășire de 1) => $S_1 = 1$ și $C_1 = 1$

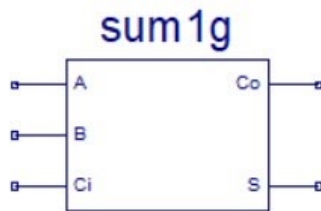
Rezultatul final: S = 10 și Co=1

Rezultatul din Modelsim:

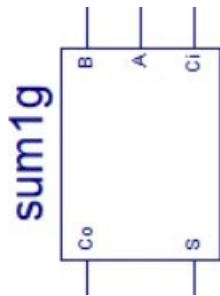


Pașii proiectului:

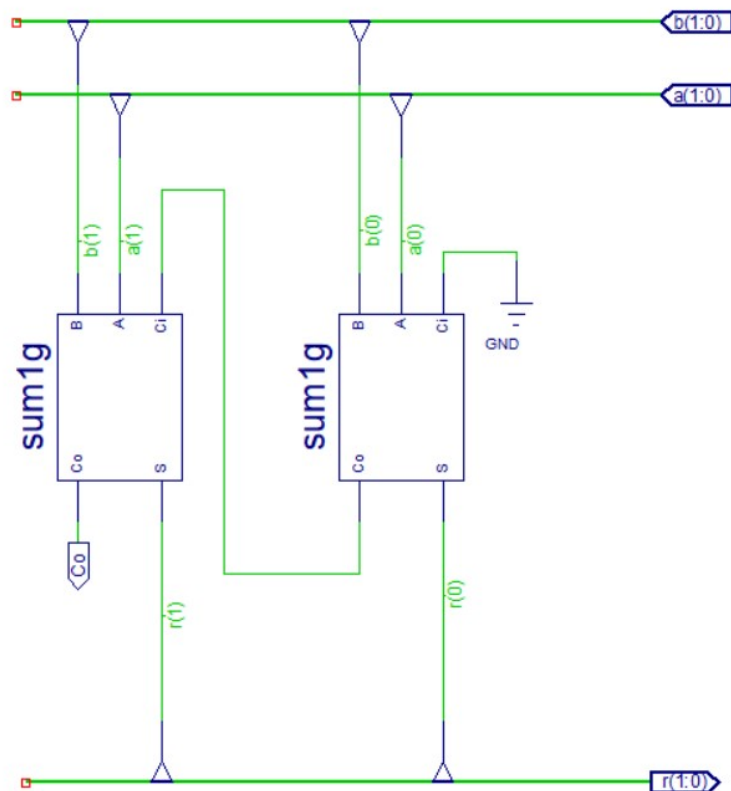
Pentru început am adăugat o copie a sursei precedente(sumatorul pe un bit) si am creat o schema bloc a acesteia.



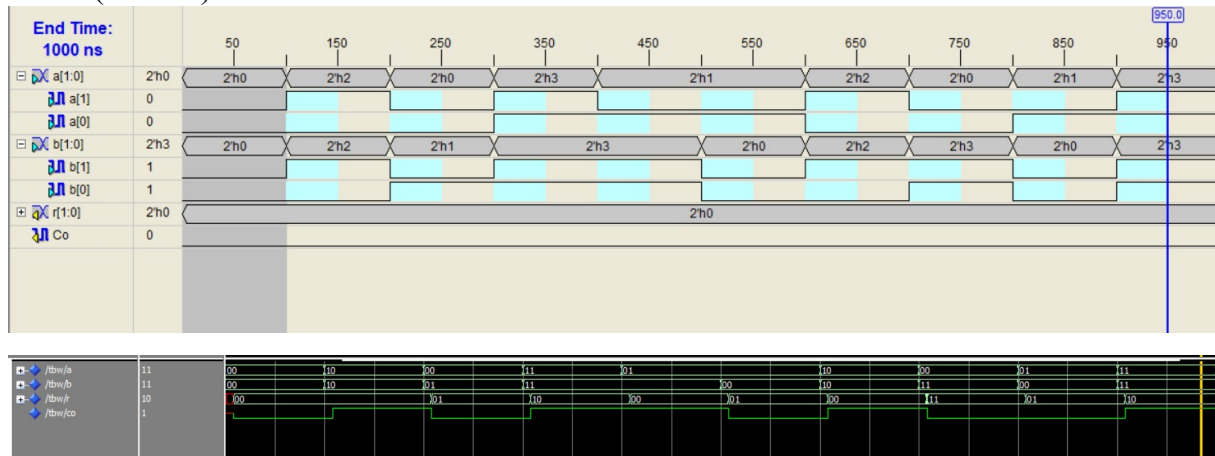
Apoi, am făcut modificările indicate pentru ca schema sa fie clară si vizibilă.



Am adăugat 2 sumatoare pe un bit. Am adăugat 3 magistrale(2 de intrare pe 2 biți a(1:0), b(1:0) si una de ieșire pe 2 biți r(1:0)-care reprezintă suma pe 2 biți). Am conectat Ci0 la 0logic, iar transportul generat de la primul sumator(Co0) l-am introdus ca intrare carry-in pentru cel de-al doilea sumator(Ci1). Iar apoi am adăugat marcher de ieșire pentru C0(carryout)



Corectitudinea sumatorului a fost verificată cu ajutorul Modelsimului, in Test Bench WaveForm am adăugat câteva combinații dintre cele 16(deoarece avem 5 intrări, dar una dintre ele este conectată la 0-logic=> 2^5) și le-am verificat cu ajutorul tabelului de adevăr(Tabel 2).



A1	A0	B1	B0	CO	S1	S0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	1
0	1	1	0	0	1	1
0	1	1	1	0	1	0
1	0	0	0	0	1	0
1	0	0	1	1	0	0
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

Tabel 2(nu l-am mai pus pe Ci0 deoarece este conectat la masa)

Această metodă poate fi aplicată și pentru adunarea a două numere pe mai mult de 2 biți, cum urmează în laboratorul 2, 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.