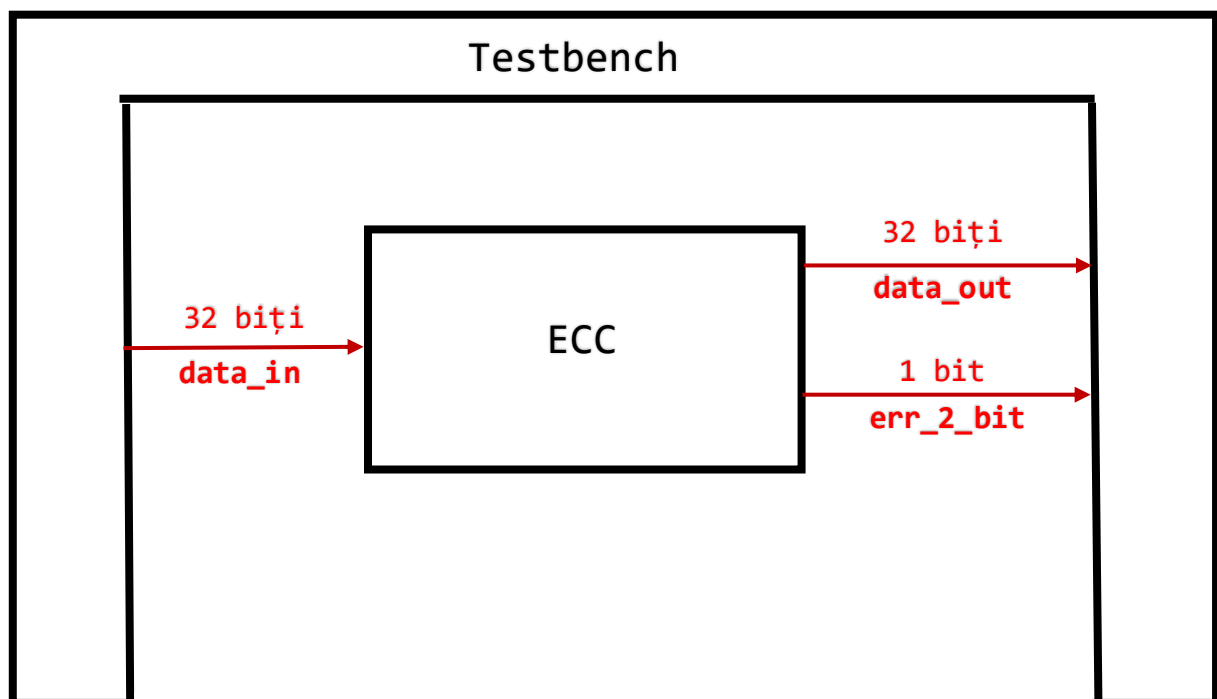


Hamming Code - ECC

Descrierea programului:

Programul este alcătuit dintr-un singur modul numit ECC, cu o intrare pe 32 de biți, și două ieșiri, pe 32 de biți, respectiv pe 1 bit. Modulul primește prin intrarea de 32 de biți un șir binar codificat conform codului Hamming(32, 26), reprezentând o implementare a codului Hamming extins (SECDED). Modulul are o ieșire pe 32 de biți prin care returnează șirul corectat, în cazul în care un singur bit a fost inversat, sau șirul original, în cazul în care doi biți au fost inversați, și o ieșire pe 1 bit prin care furnizează rezultatul evaluării intrării - valoare 0 dacă au fost 0 sau 1 biți inversați, respectiv valoarea 1 dacă au fost 2 biți inversați. Conform specificațiilor de realizare a modulului, se exclude posibilitatea ca intrarea să aibă mai mult de doi biți inversați.

Schemă bloc:



Implementarea modulului:

Inițial, modulul parcurge șirul de intrare începând cu a doua poziție, salvând poziția fiecărui bit cu valoarea 1, într-un format pe 5 biți, într-un registru de 155 de biți, acest număr reprezentând numărul maxim de elemente de 1, cu poziția reprezentată pe 5 biți, într-un șir de 31 de biți, bitul de pe prima poziție fiind ignorat, acesta având rol de determinare a parității celor 31 de biți rămași.

Ulterior se realizează operația XOR la nivel de bit între indicii memorați pe 5 biți la pasul anterior, obținându-se o singură valoare pe 5 biți în urma acestor operații. Această valoare va reprezenta poziția bitului inversat, în cazul în care doar unul se află în această situație.

În continuare se va verifica dacă paritatea numărului de biți cu valoarea 1 este egală cu paritatea bitului de pe prima poziție din șirul de intrare. Dacă această paritate este egală înseamnă că fie nu există nici o eroare, fie există două, acest lucru determinându-se prin verificarea valorii obținute în urma operației XOR descrise în paragraful de mai sus. Prin urmare, dacă aceasta are valoare 0 nu există eroare, iar dacă are o valoare diferită de 0, există două erori.

În cazul în care paritatea primului bit nu corespunde cu cea a numărului de valori de 1 din șir, se poate concluziona că există un singur bit inversat, iar acesta se va afla la poziția cu indicele egal cu valoarea obținută în urma operației XOR între indicii pozițiilor cu valoarea 1.

Probleme întâlnite inițial:

În versiunea inițială a implementării a fost folosită indexarea descrescătoare a șirurilor de intrare și de ieșire. Acest lucru furniza rezultate greșite pentru unele cazuri de test deoarece bitul global de paritate nu se

afla pe poziția 0. Acest lucru a fost inițial remediat printr-o branșare determinată de valoarea bitului global de paritate. Astfel, dacă acesta era 1 se compara valoarea operației XOR dintre pozițiile cu valoarea 1 – enunțată mai sus – cu numărul 31, poziția primului bit. În caz contrar, acea valoare se compara cu numărul 0, poziția ultimului bit.

```
42     if((nr_of_1[0] & 1) == data_in[31]) begin
43         data_out[31:0] = data_in[31:0];
44         if(data_in[31]==1) begin
45             if(error_loc != 31) err_2_bit = 1;
46             else err_2_bit = 0;
47         end
48         else begin
49             if(error_loc != 0) err_2_bit = 1;
50             else err_2_bit = 0;
51         end
52     end
53     else begin
54         data_out[31:0] = data_in[31:0];
55         err_2_bit = 0;
56         //data_out[31 - error_loc] = ~data_out[31 - error_loc];
57         if(data_out[31] == 0) data_out[error_loc] = ~data_out[error_loc];
58         else data_out[31 - error_loc] = ~data_out[31 - error_loc];
59     end
```

În versiunea finală, această problemă a fost remediată mai simplu, prin inversarea ordinii indexării șirurilor de intrare, respectiv ieșire.

```
module ECC(input[0:31] data_in, output reg[0:31] data_out, output reg err_2_bit);
```

Referințe:

- <https://www.youtube.com/watch?v=X8jsijh1lIA>
- https://www.youtube.com/watch?v=b3NxrZOu_CE&
- https://en.wikipedia.org/wiki/Hamming_code