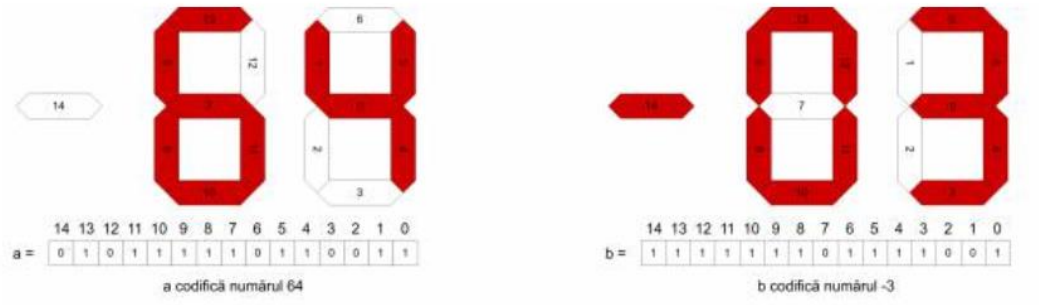
Tema 1 – Inmultirea numerelor cu semn

Pentru aceasta tema, am ales sa ma folosesc de algoritmul lui Booth pentru a realiza inmultirea cu semn a doua numere.

In primul rand, a trebuit sa decodific aceste numere care erau afisate pe un display ca in figura de mai jos:



Astfel, in modulul principal (multiplier.v), am apelat modulul decoder pentru a putea decodifica, pe rand, pe x si pe y. Acest modul l-am implementat astfel:

- Intai am impartit in 3 registrii informatii despre semnul numarului (semn_x), cifra zecilor (first_digit_x) respectiv cifra unitatilor (second_digit_x).
- Dupa ce am extras aceste informatii, am ales sa folosesc shiftarea la stanga in locul operatorului $*$. Cum $10 = 2^1 + 2^3$, a fost nevoie ca cifra zecilor (x1) sa fie shiftata la stanga o data, dupa care adunata cu shiftarea sa de trei ori la stanga pentru ca in final sa mai adunam si cifra unitatilor (x2).
- Astfel, in functie de semnul numarului am folosit urmatoarele formule (pentru semn negative a fost necesar sa calculez complementul fata de 2) :

$$Semn\ pozitiv : x = (x1 \ll 1) + (x1 \ll 3) + x2$$

$$Semn\ negativ : x = \big((x1 \ll 1) + (x1 \ll 3) + x2\big)' + 1'b1$$

Algoritmul lui Booth

Pentru implementarea algoritmului de inmultire am procedat astfel:

- Am ales ca intrari pe in1 si in2, fiecare pe cate 8 biti, iar ca iesire out, pe 16 biti.
- In registrul p (dublu ca marime), am organizat primii 8 biti pe 0, urmatorii 8 fiind ocupati de in2 iar ultimul bit initial este ales 0.
- La urmatorul pas, am verificat in care dintre urmatoarele cazuri se afla ultimii 2 biti ai lui p:

  00 sau 11: p ramane la fel, dar se shifteaza aritmetic la dreapta cu o pozitie

  00 : p este adunat la in1, dupa care se shifteza arithmetic la dreapta cu o pozitie

  10 : din p se scade in1, dupa care se shifteaza arithmetic la dreapta cu o pozitie.

- Se repeta aceste operatii de 8 ori (numarul de biti al intrarilor)
- Produsul final este p fara cel mai nesemnificativ bit (bitul 0)

Dupa apelarea modulului booth, am retinul in outputul meu, product, doar primii 14 biti care ma interesau deoarece maximul rezultatului obtinut era de 9801 sau -9801 (99*99 sau -99*99) care se afla in intervalul $(2^{13}, 2^{14})$.