**8-bit Booth Multiplier**

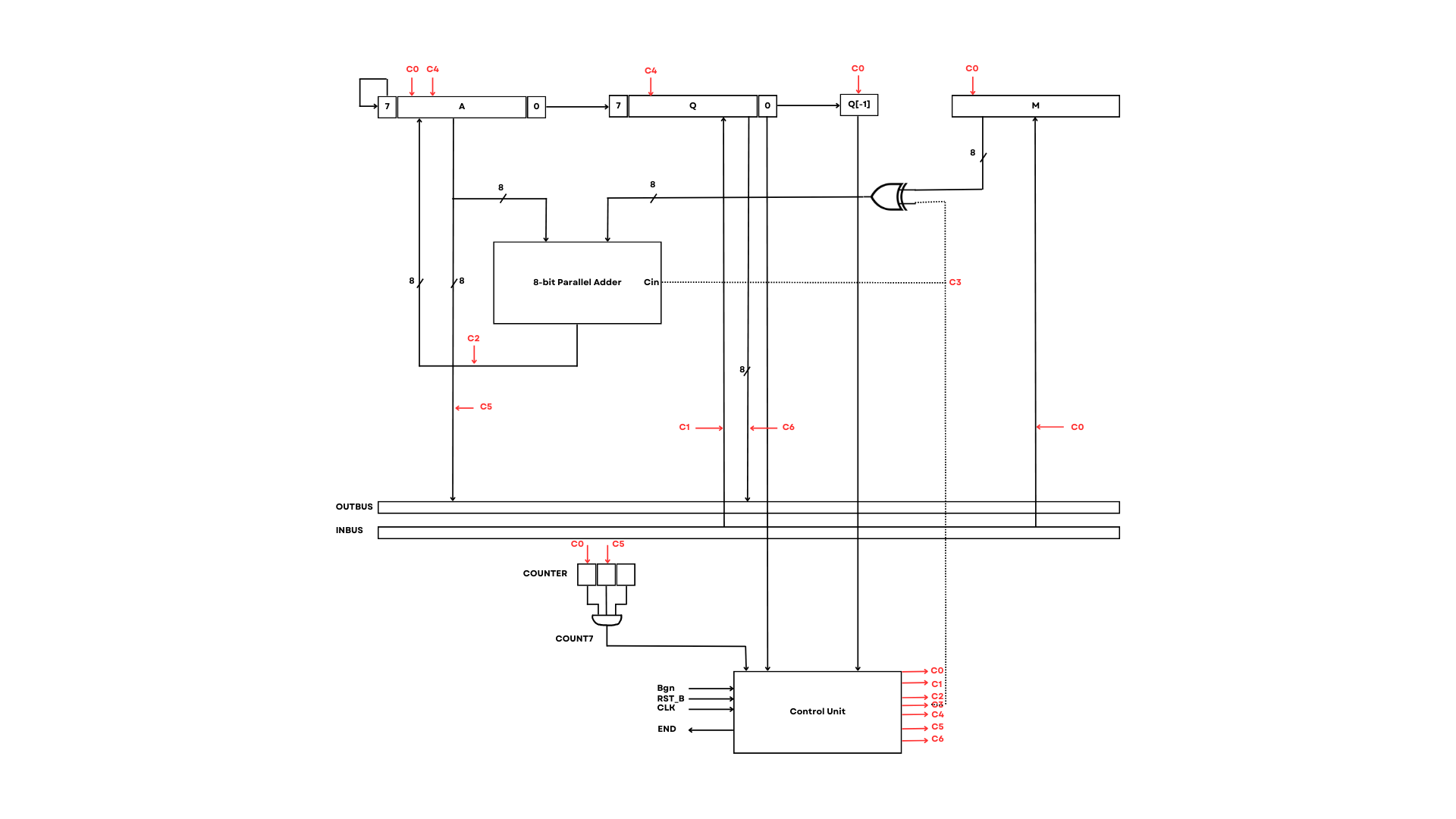
Scopul acestui proiect este de a crea un inmulțitor pe 8 biți, utilizând metoda Booth.

Intrările modulului de înmulțire sunt: semnalul de tact (clk), semnalul de reset(rst\_b), semnalul de start al algoritmului (bgn), semnalul de oprire (stop), precum și deînmulțitul și înmulțitorul ce se preiau de pe magistrala de date (inbus). Ieșirea modului este reprezentată

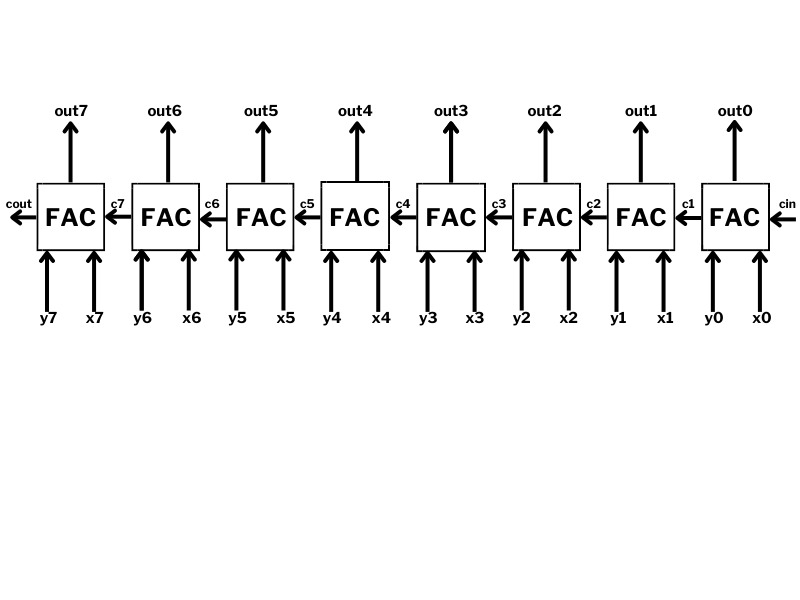
de magistrala de date pe care se descarca produsul celor doua numere (outbus).

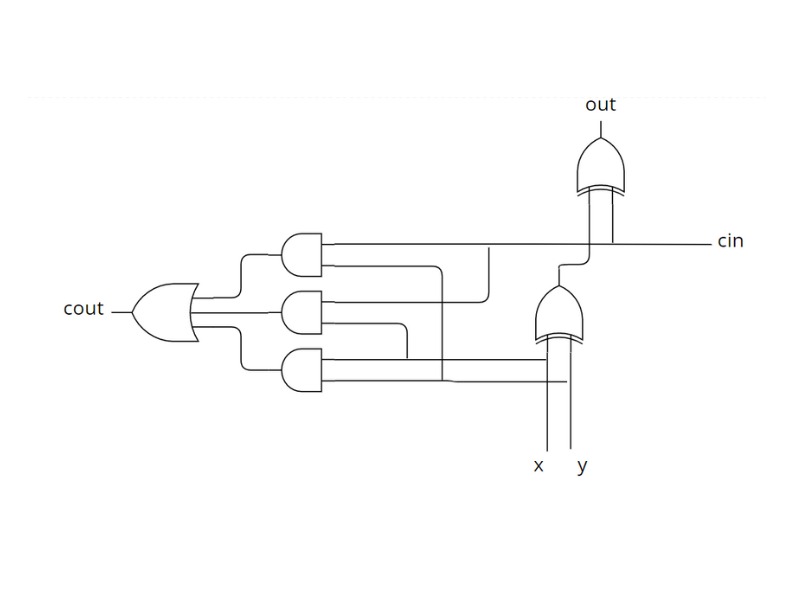
Acest modul curpinde:

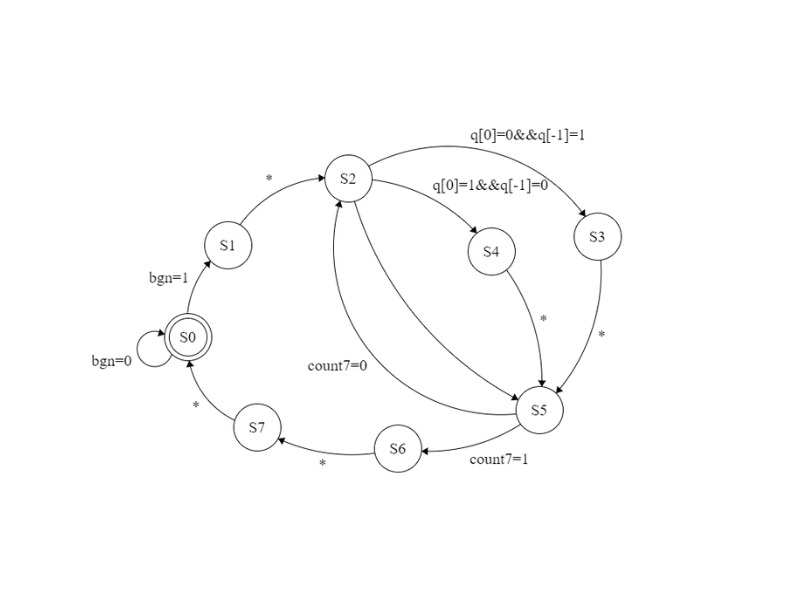
* 4 regiștrii (reg\_a, reg\_q, reg\_q\_1, reg\_m)
* O portă XOR ce realizează complementul de 1 al valorii stocate în registrul reg\_m (ulterior la care se va adăuga 1 pentru a-l transforma în complement de 2) (xor\_gate)
* Un sumator paralel pe 8 biți (parallel\_adder)
* Un counter pe 3 biți (counter)
* O unitate de control (control\_unit)

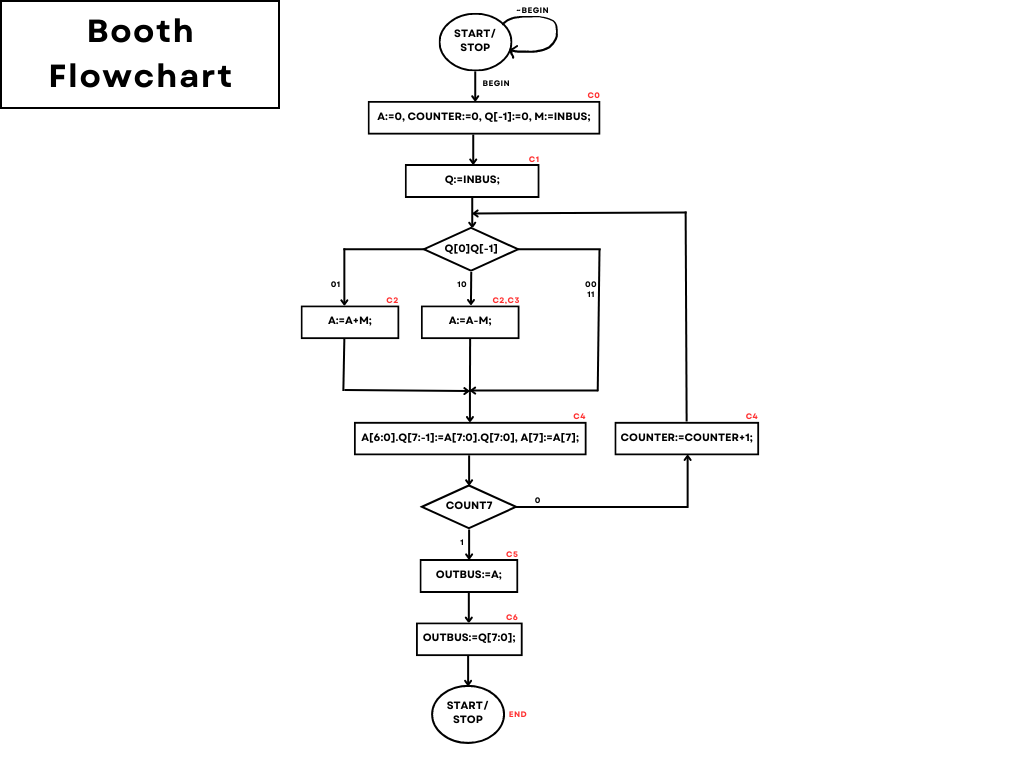
Implementarea hardware a acestui înmulțitor este:

* Registrul **reg\_a**:
  + intrări: clk, rst\_b, c0, c2, c4, c5, [7:0] sum,
  + ieșiri: a\_lsb, [7:0] obus, [7:0] q
  + se inițializează cu 0 la activarea semnalului de control c0
  + la activarea semnalului de control c2/ c2 și c3 se realizează suma/diferența dintre valoarea stocată anterior și valoarea stocată în registrul reg\_m
  + la activarea semnalului de control c4 se realizează o shiftare aritmetică la dreapta (semnul se recirculă)
  + la activarea semnalului de control c5 se descarcă pe outbus valoarea reținută
  + în cazul în care semnalul rst\_b este activ (pe zero), valoarea din registru este resetată
* Registrul **reg\_q**:
  + intrări: clk, rst\_b, c1, c4, c6, a\_lsb, [7:0] ibus,
  + ieșiri: q\_lsb, [7:0] obus
* la activarea semnalului de control c1 se încarcă de pe magistrala inbus valoarea în registru
* la activarea semnalului de control c4 se realizează o shiftare la dreapta, în urma căreia cel mai semnificativ bit al registrului reg\_q va lua valoarea celui mai puțin semnificativ bit al registrului reg\_a
* la activarea semnalului de control c6 se descarcă pe outbus valoarea reținută
* în cazul în care semnalul rst\_b este activ (pe zero), valoarea din registru este resetată
* Registrul **req\_q\_1**:
  + intrări: clk, rst\_b, c0, c4, q\_lsb,
  + ieșiri: q
  + la activarea semnalului de control c0 se inițializează conținutul registrului cu 0
  + la activarea semnalului de control c4 registrul reg\_q\_1 va lua valoarea celui mai puțin semnificativ bit al registrului reg\_q
  + în cazul în care semnalul rst\_b este activ (pe zero), valoarea din registru este resetată
* Registrul **reg\_m**:
  + intrări: clk, rst\_b, c0, [7:0] ibus,
  + ieșiri: reg [7:0] q
* la activarea semnalului de control c0 se încarcă de pe magistrala inbus valoarea în registru
* în cazul în care semnalul rst\_b este activ (pe zero), valoarea din registru este resetată
* Modulul **xor\_gate**:
  + intrări: c3, [7:0] m,
  + ieșiri: [7:0] rez
  + instanțiază 8 porți XOR (xor\_m\_c3) pentru realizarea comlementului de 1 atunci când trebuie efectuată operația de scădere, c3 fiind egal cu 1 (daca c3 este 0 atunci se face adunare, iar numărul nu trebuie complementat)
* Sumatorul **parallel\_adder**:
  + modulul are 3 intrări (x, y pe 8 biți și c3) și o ieșire (out pe 9 biți)
  + instanțiază 8 celule FAC (Full Adder Cell), cin fiind egal cu c3 (daca c3 este egal cu 1 se va realiza transformarea numărului din complement de 1 în complement de 2)



* + structura unei celule FAC:
* Registrul **counter**:
  + intrări: clk, rst\_b, c0, c4,
  + ieșiri: count7
  + la activarea semnalului de control c0 se inițializează registrul cu 0
  + la activarea semnalului de control c4 se incrementează counter-ul
  + ieșirea acestui registru este 1 atunci când count-ul are socată valoarea 710 (1112)
  + în cazul în care semnalul rst\_b este activ (pe zero), valoarea din registru este resetată
* Modulul **control\_unit**:
  + intrări: clk, rst\_b, bgn, q\_1, q0, count7
  + ieșiri: c0, c1, c2, c3, c4, c5, c6, stop
  + se realizează codificarea stărilor (de la S0 la S7)
  + se implementează un automat cu stări finite, care își modifică starea în funcție de: count7, q0 și q\_1
  + dacă q0 = 0 și q\_1 =1, atunci se activează semnalul de control c2, pentru a realiza ulterior adunare și a semnalului de control c4 pentru shiftare și pentru incrementarea counter-ului
  + dacă q0 = 1 și q\_1 = 0, atunci se activează semnalul de control c2 și c3, pentru a realiza ulterior scădere și a semnalului de control c4 pentru shiftare și pentru incrementarea counter-ului
  + dacă q0 = 0 și q\_1 =0 sau q0 = 1 și q\_1 =1 , atunci se activează semnalul de control c4, pentru a realiza ulterior shiftare și incrementarea counter-ului
  + se reiau pașii de mai sus până când counter-ul are valoarea 710(1112)
  + în cazul în care counter-ul este 710(1112) se activează pe rând semnalele de control c5 și c6 pentru descărcarea valorilor din reg\_a și req\_q pe outbus





* Modulul **booth**:
  + intrari: clk, bgn, rst\_b, ibus,
  + ieșiri: stop, [7:0] obus
  + se instanțiază legăturile modulelor descrise anterior