

Variabila *Valoarea1* memorează rezultatul comparării. Dacă bitul cel mai semnificativ al acestei variabile este 0, mantisa primului număr trebuie deplasată dreapta. Pentru a memora numărul de deplasări care trebuie efectuate, au fost alocați 8 biți (*Valoarea1*[7 : 0]).

Dacă *Valoarea1*[8] este 1, atunci mantisa asociată numărului 2 trebuie deplasată dreapta de un număr de ori egal cu valoarea lui *Valoarea1*[7 : 0]. Pentru implementarea sumatorului care apare în cadrul benzii de asamblare se recomandă utilizarea unui sumator cu transport anticipat (CARRY LOOK AHEAD).

În cadrul segmentului 3 al benzii de asamblare, cele două mantise sunt adunate sau scăzute în funcție de valoarea semnalului *OP*. Dacă *OP* = 0 atunci trebuie realizată operația de adunare și, în consecință, cele două mantise trebuie adunate. Dacă *OP* = 1, operația de scădere trebuie realizată și, în consecință, cele două mantise trebuie scăzute.

Rezultatul pasului de adunare/scădere a mantiselor se reprezintă pe 26 de biți. Bitul cel mai semnificativ, bitul 25, este utilizat pentru a specifica dacă mantisa este egală sau nu cu zero. Restul de biți este folosit pentru a reprezenta registrele *E* și *A* concatenate.

Rezultatul este normalizat în cadrul segmentului 4 al benzii de asamblare. Când apare o depășire superioară, mantisa-rezultat este deplasată dreapta și exponentul este incrementat cu o unitate. Când apare o depășire inferioară, numărul de zerouri din cadrul mantisei (pozițiile cele mai semnificative) determină numărul de deplasări stânga al mantisei, număr care trebuie scăzut din exponent. Memorarea acestui număr se face prin intermediul variabilei *Valoarea2*.

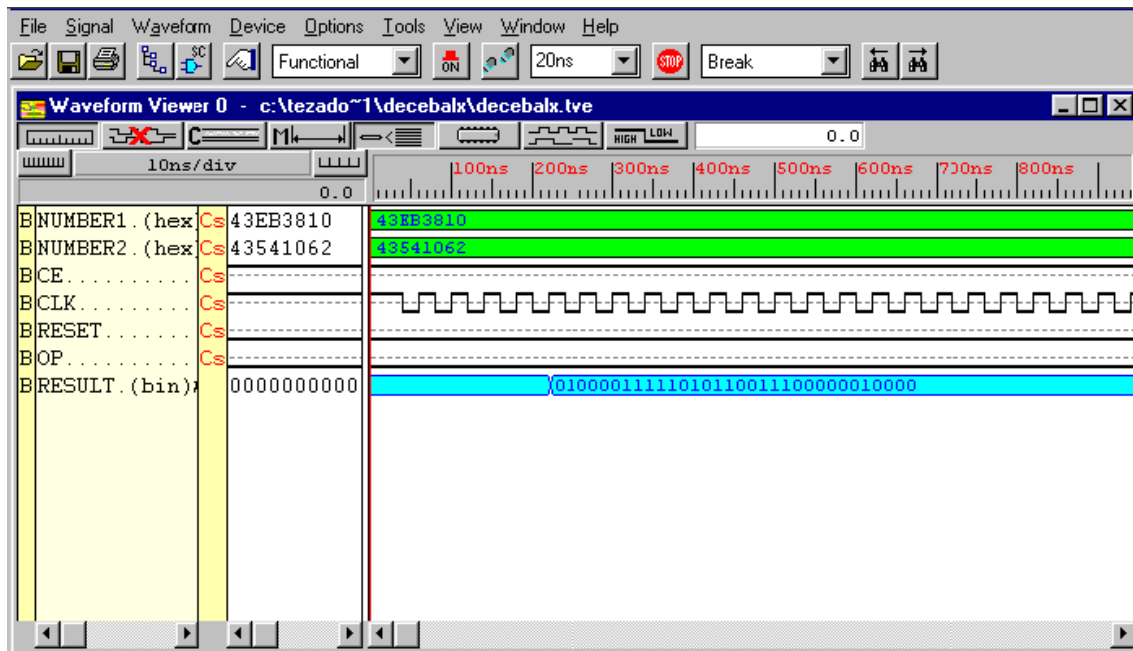
Dacă bitul cel mai semnificativ al variabilei *Valoarea1* este 0, atunci exponentul trebuie deplasat spre stânga, altfel exponentul trebuie să fie deplasat spre dreapta. Numărul de deplasări este indicat de către *Valoarea2*[7 : 0].

Rezultatele simulării acestui algoritm folosind mediul de dezvoltare Xilinx pot fi observate în figura 6.4.

## Înmulțirea în virgulă mobilă

Operația de înmulțire în virgulă mobilă presupune realizarea următorilor pași:

1. Se verifică dacă unul dintre numere este zero sau nu;
2. Se adună exponenții;
3. Se înmulțesc mantisele;
4. Se normalizează produsul.



**Figura 6.4:** Rezultatele simulării algoritmului de adunare/scădere.

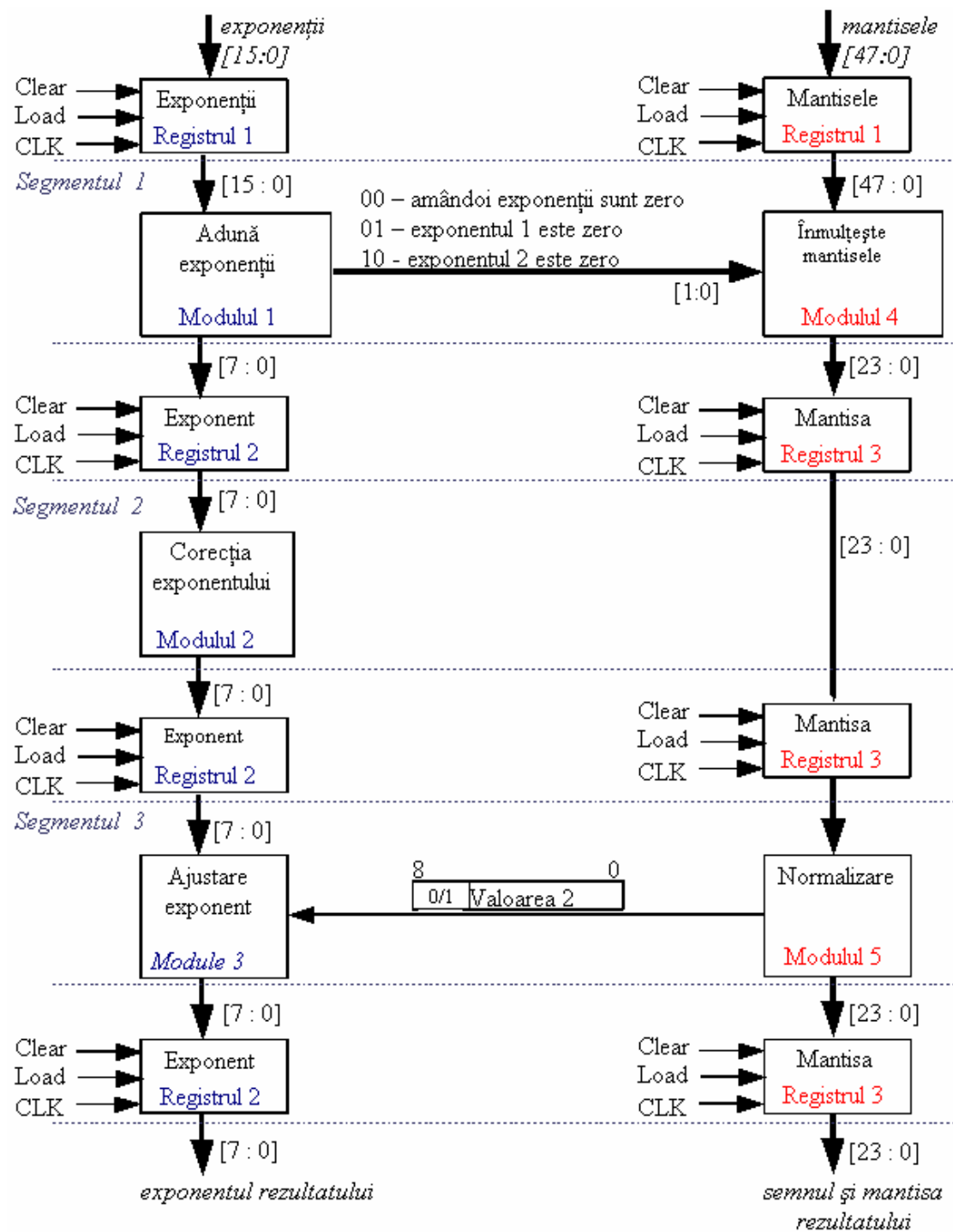
Implementarea în bandă de asamblare al acestui algoritm poate fi observată în figura 6.5.

Înmulțirea a două numere binare se realizează cu ajutorul unui circuit combinațional (denumit matrice de multiplicare) care formează toți biții produsului odată. Aceasta este cea mai rapidă metodă deoarece timpul necesar efectuării operației este egal cu timpul de propagare al semnalului luând în calcul ruta cea mai dezavantajoasă.

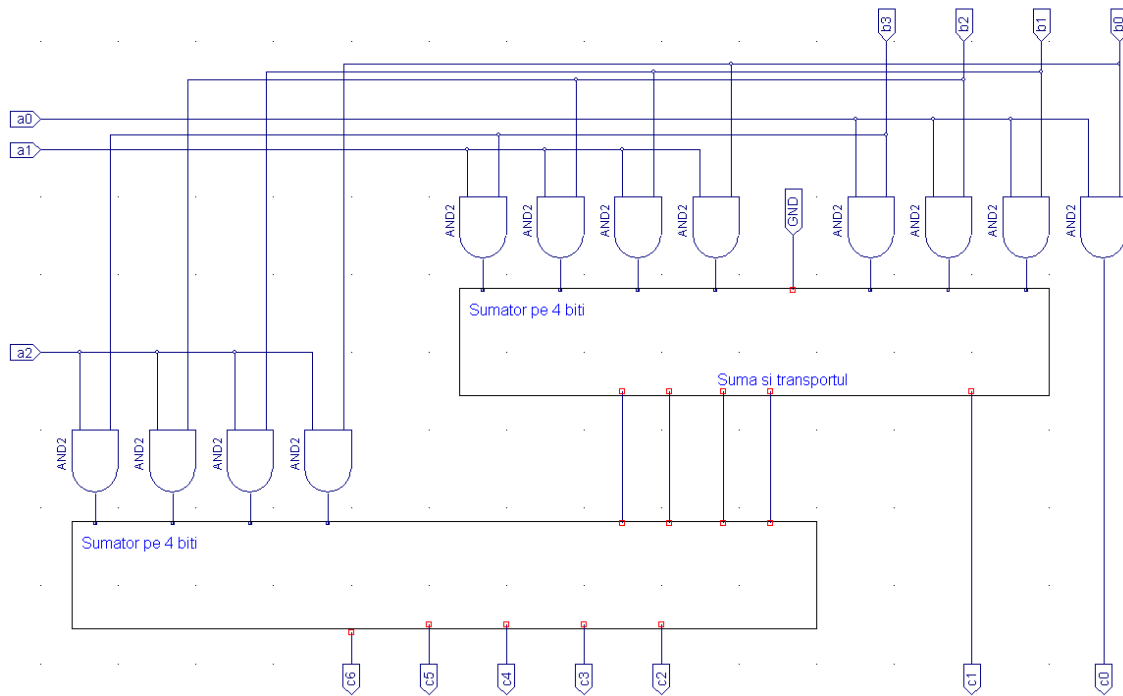
Circuitul *matrice de multiplicare* necesită un număr mare de porți, implementarea lui este neeconomică, acesta fiind principalul motiv pentru care acest circuit nu a avut o răspândire foarte mare până la apariția circuitelor integrate. Proiectarea unui asemenea circuit ale cărui intrări ar avea dimensiunile de  $j$  respectiv  $k$  biți va necesita  $j \times k$  porți ȘI și sumatoare de  $(j - 1) \cdot k$  biți pentru a obține un produs de  $(j + k)$  biți.

Un exemplu de circuit matrice de multiplicare este prezentat în figura 6.6.

Trebuie remarcat că exponentul rezultat corect este obținut prin scăderea valorii 127 din exponentul sumă.



**Figura 6.5:** Algoritmul de înmulțire a două numere în virgulă mobilă.



**Figura 6.6:** Circuitul matrice de multiplicare pentru  $k = 4$  și  $j = 3$ .

## Împărțirea în virgulă mobilă

Algoritmul de împărțire a două numere reprezentate în virgulă mobilă necesită parcurgerea următorilor pași:

1. Se verifică dacă unul dintre cei doi operanzi este zero sau nu;
2. Se inițializează registrele și se evaluează semnul;
3. Se aliniază deîmpărțitul;
4. Se scad exponenții;
5. Se împart mantisele.

Algoritmul în bandă de asamblare al acestei operații este prezentat în figura 6.7

Împărțirea a două numere în virgulă mobilă normalizate va produce întotdeauna un cât normalizat. Din acest motiv, spre deosebire de celelalte operații în virgulă mobilă, rezultatul operației de împărțire nu necesită normalizare.