

Universitatea Tehnică din Cluj-Napoca

Unitatea Aritmetică de Tip MMX

Structura Sistemelor de Calcul

Student: Pântea Marius-Nicușor

Introducere

Context

Una dintre tehnologiile inovative introduse de Intel în anii 1990 este arhitectura MMX (MultiMedia eXtensions). Aceasta reprezintă un set de instrucțiuni SIMD (Single Instruction, Multiple Data), care permite procesarea simultană a mai multor date cu o singură instrucțiune, îmbunătățind astfel viteza de execuție a operațiilor aritmetice și logice. Inițial concepută pentru a îmbunătăți performanța aplicațiilor multimedia, arhitectura MMX și-a dovedit utilitatea și în alte domenii care necesită procesare intensă de date, cum ar fi codarea video, grafica 3D și recunoașterea imaginii.

Obiective

Acest proiect își propune să exploreze și să implementeze o unitate aritmetică de tip MMX, capabilă să execute șase operații aritmetice esențiale. Unitatea va fi proiectată în VHDL și inclusă într-un proiect Xilinx Vivado. Un banc de testare este, de asemenea, furnizat în scopuri de simulare, pentru a demonstra funcționalitatea corectă a unității.

Plan de Lucru

Întâlnirea 3- Implementare parte de analiza

Întâlnirea 4- Implementare Design

Întâlnirea 5- Implementare Unitate

Întâlnirea 6-Testarea Unități Complete

Întâlnirea 7-Prezentarea Proiectului

Bibliografie:

- x86 Assembly Language Reference Manual-Oracle
- Intel® 64 and IA-32 Architectures Software Developer's Manual ,Volume 2 (2A, 2B, 2C, & 2D):Instruction Set Reference, A-Z
- Intel ArchitectureSoftware Developer's , Manual Volume 1:Basic Architecture

Studiu Bibliografic

Ce este Tehnologia MMX?

Tehnologia Intel MMX™ cuprinde un set de extensii ale arhitecturii Intel (IA) care sunt concepute pentru a spori considerabil performanța aplicațiilor media și de comunicații avansate. Aceste extensii (care includ registre noi, tipuri de date și instrucțiuni) sunt combinate cu un model de execuție cu o singură instrucțiune, date multiple (SIMD) pentru a accelera performanța aplicațiilor precum video în mișcare, grafică combinată cu video, imagine procesare, sinteza audio, sinteza și compresia vorbirii, telefonie, videoconferința, și grafică 2D și 3D, care utilizează de obicei algoritmi de calcul intensiv pentru a efectua operații repetitive pe rețele mari de elemente de date simple, native.

MMX oferă programatorului 8 registre de uz general pe 64 de biți. Aceste registre, numite MM0 - MM7, pot fi utilizate în mai multe moduri. Ele pot fi utilizate ca valori unice pe 64 de biți, cantități duble de 32 de biți, 4 cantități de 16 biți sau 8 cantități de 8 biți. Când se întreprinde orice acțiune asupra unui registru MMX, aceasta se aplică tuturor elementelor registrului în același timp. Acest lucru permite software-ului să funcționeze de până la 8 ori mai rapid.

63	0
MM7	
MM6	
MM5	
MM4	
MM3	
MM2	
MM1	
MM0	

Figură 1

Ce aduce nou MMX?

Tehnologia MMX™ suportă o nouă capacitate aritmetică cunoscută sub numele de aritmetică cu saturație. Saturația este cel mai bine definită prin contrast cu modul de înfășurare (wraparound). În modul de înfășurare, rezultatele care depășesc limitele sau sunt sub pragul minim sunt trunchiate, iar doar biții inferiori (cei mai puțin semnificativi) ai rezultatului sunt returnați; cu alte cuvinte, transportul (carry) este ignorat. Dacă există un overflow (depășire a limitei maxime) sau un underflow (depășire a limitei minime) după o operație de adunare sau scădere, rezultatul ar trebui să fie saturat la valoarea maximă sau minimă posibilă pentru tipul de date.

Reprezentare pe 8 biți fără semn (unsigned):

- Adunare normală (wraparound): $80h + 90h = 10h$ (eroare, overflow)
- Cu saturație: $80h + 90h = FFh$ (valoarea maximă)
- Scădere normală (wraparound): $80h - 90h = F0h$ (underflow)
- Cu saturație: $80h - 90h = 00h$ (valoarea minimă)

Reprezentare pe 8 biți cu semn (complement pe 2):

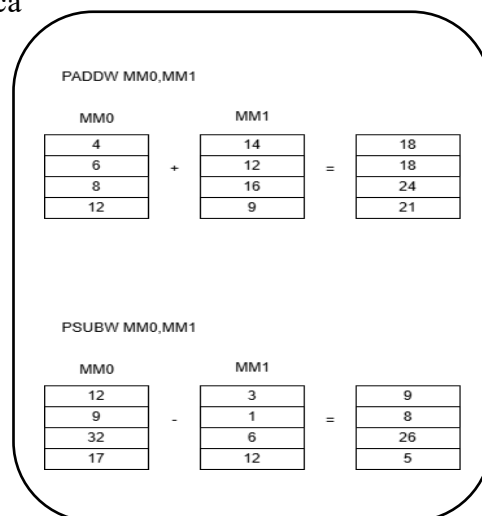
- Adunare normală (wraparound): $70h + 20h = 90h$ (eroare, rezultat negativ)
- Cu saturație: $70h + 20h = 7Fh$ (valoarea maximă)
- Scădere normală (wraparound): $80h - 20h = 60h$ (eroare, rezultat pozitiv)
- Cu saturație: $80h - 20h = 80h$ (valoarea minimă, -128)

Analiza

Specificațiile Unității Aritmetice MMX

Operațiile care vor fi implementate în această unitate aritmetică sunt:

- Adunare (PADD) – În hardware, instrucțiunea **PADD** (Packed ADD) funcționează prin realizarea de adunări paralele pe segmente de date, cum ar fi octeți sau cuvinte, într-o arhitectură SIMD (Single Instruction, Multiple Data). Această operație permite efectuarea mai multor adunări simultan într-o singură instrucțiune, procesând fiecare segment separat. Vor fi implementate 3 tipuri de adunare pe word, cu wraparound (PADDW), saturație cu semn (PADDSW), saturație fără semn (PADDUSW).

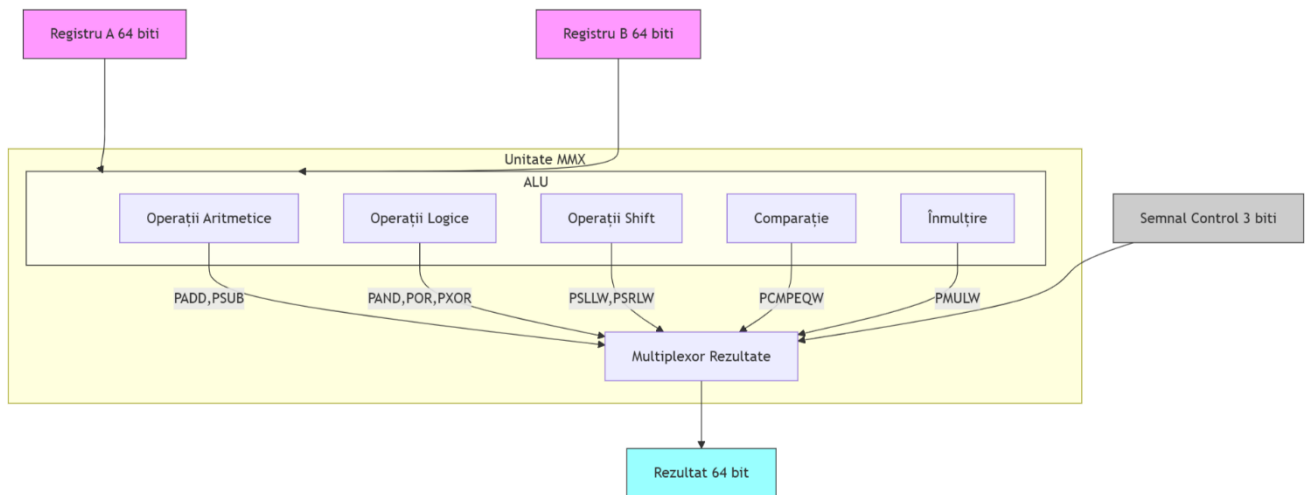


Figură 2

- Scădere (PSUB) -În hardware, instrucțiunea PSUB funcționează prin realizarea de scăderi paralele pe segmente de date, cum ar fi octeți sau cuvinte, într-o arhitectură SIMD . Această operație permite efectuarea mai multor scăderi simultan într-o singură instrucțiune, procesând fiecare segment separat. Vor fi implementate 3 tipuri de adunare pe word, cu wraparound (PSUBW), saturație cu semn (PSUBSW), saturație fără semn (PSUBUSW).
- Comparare (PCMPEQW) - compară segmente de 16 biți pentru a verifica dacă sunt egale și returnează un indicator de egalitate per segment. Dacă segmentele sunt egale, valoarea 0xFFFF este setată pe segmentul de ieșire corespunzător; altfel, se setează 0x0000.
- Shift la Stânga/Dreapta (PSLLW, PSRLW) - deplasează fiecare segment de 16 biți spre stânga/dreapta cu un număr specificat de poziții. Pentru fiecare deplasare la stânga/dreapta, biții sunt completați cu zero. Aceasta permite multiplicarea rapidă cu puteri ale lui 2 pentru fiecare segment individual.
- Operații Logice (PAND,POR,PXOR) - Aceste instrucțiuni permit manipularea bit cu bit, aplicând operațiile logice AND, OR sau XOR între segmentele de date. Fiecare segment din primul registru este combinat cu segmentul corespunzător din al doilea registru.
- Multiplicare (PMULW) - realizează multiplicarea segmentată, aplicând operația pe câte două segmente de 16 biți. Rezultatul fiecărei multiplicări este stocat într-un registru de ieșire. Pentru a evita overflow-ul, unitatea de control gestionează overflow-ul pe fiecare segment

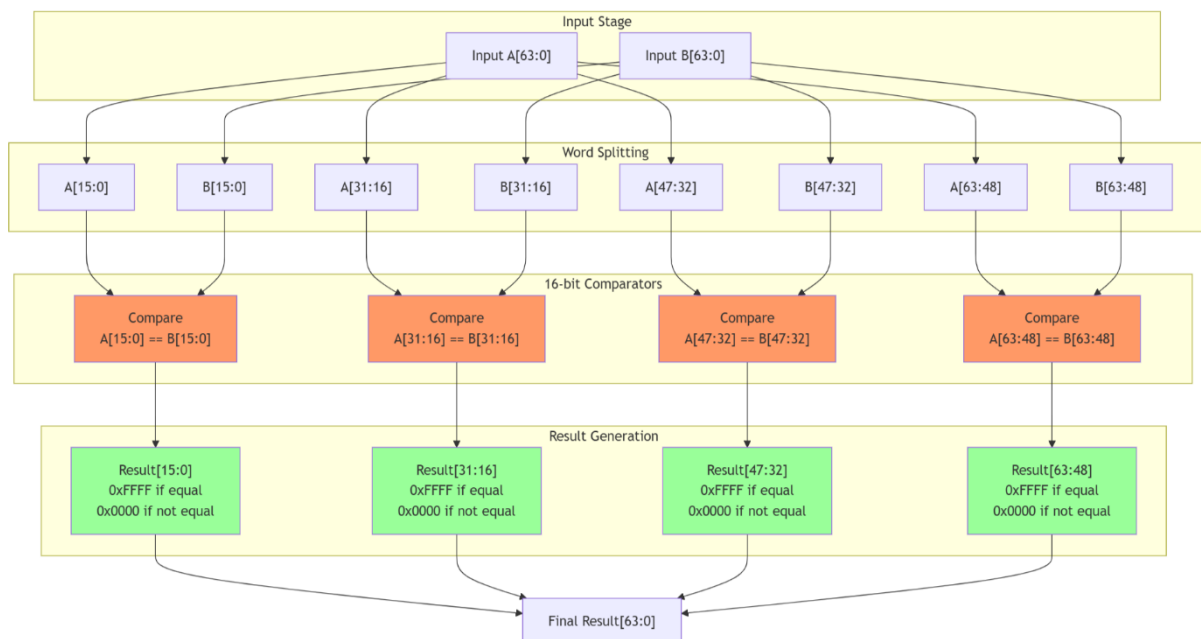
Design

Arhitectura generală



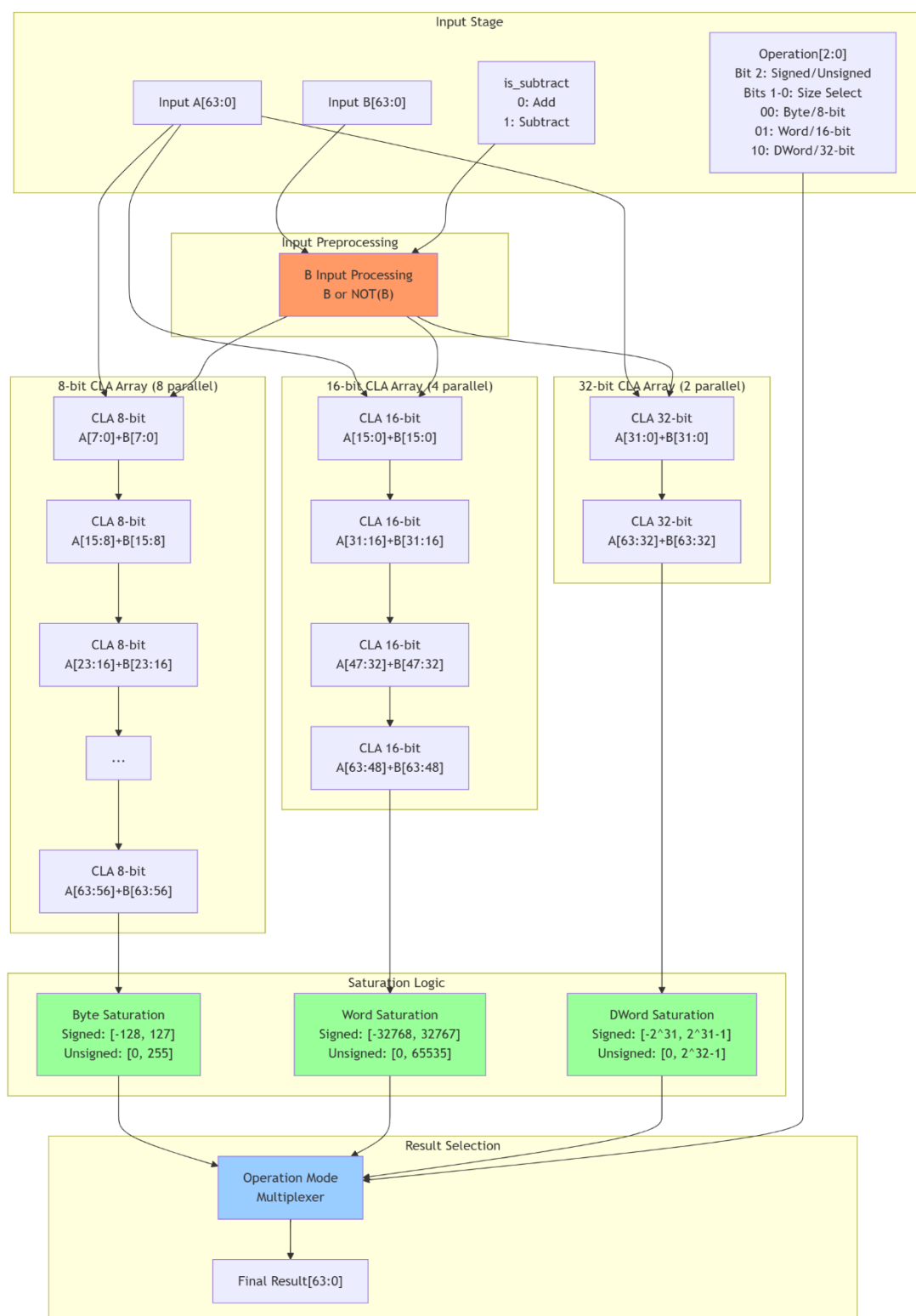
Figură 3

Instrucțiunea PCMPEQW



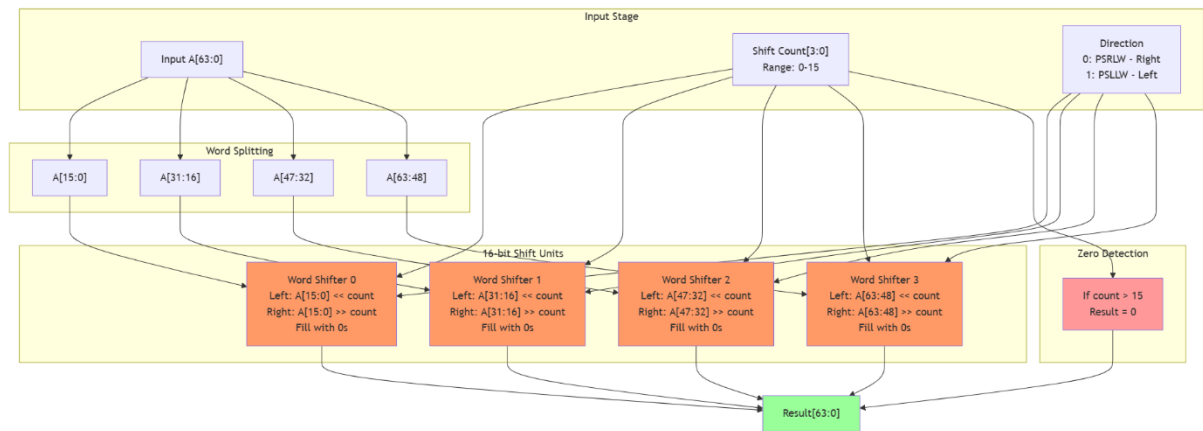
Figură 4

Instrucțiunile PADD/PSUB



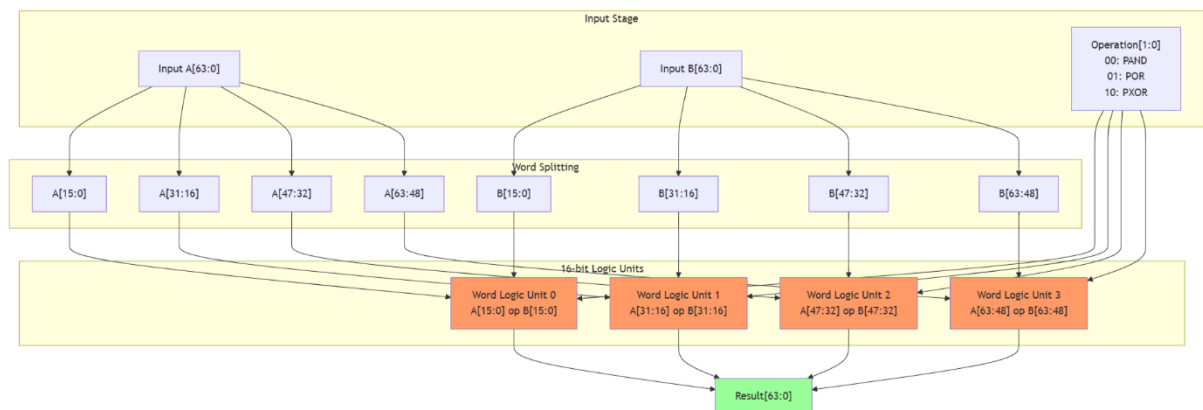
Figură 5

Instrucțiunile PSLW/PSRLW



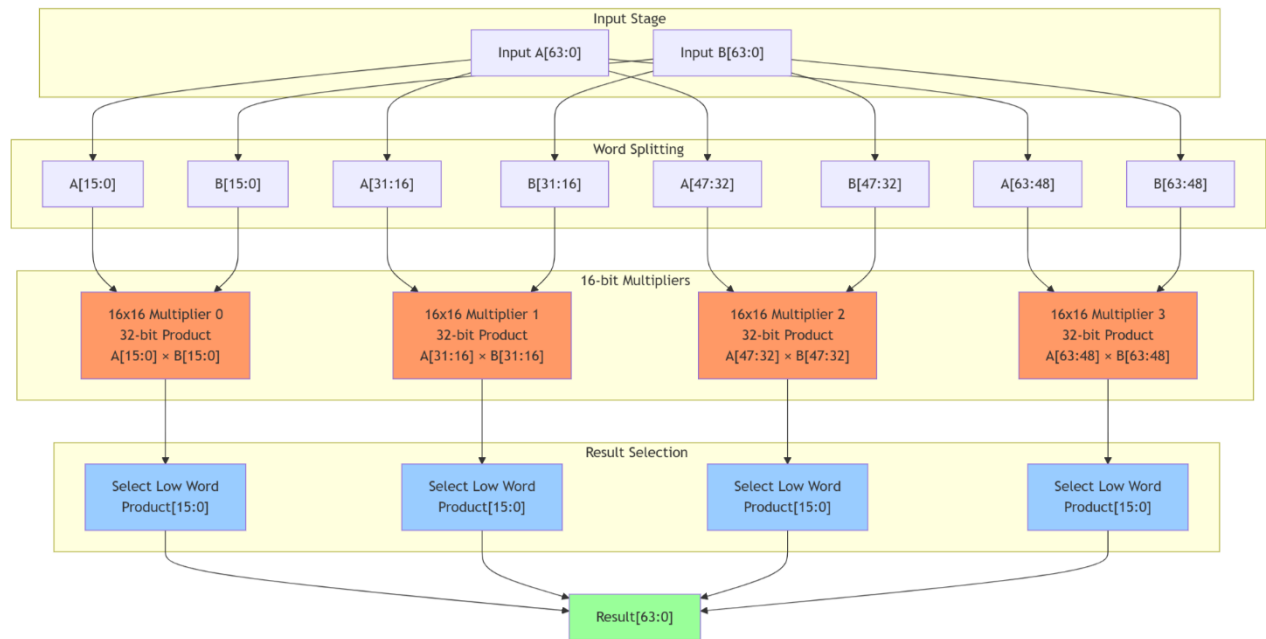
Figură 6

Instrucțiunile PAND/PXOR/POR



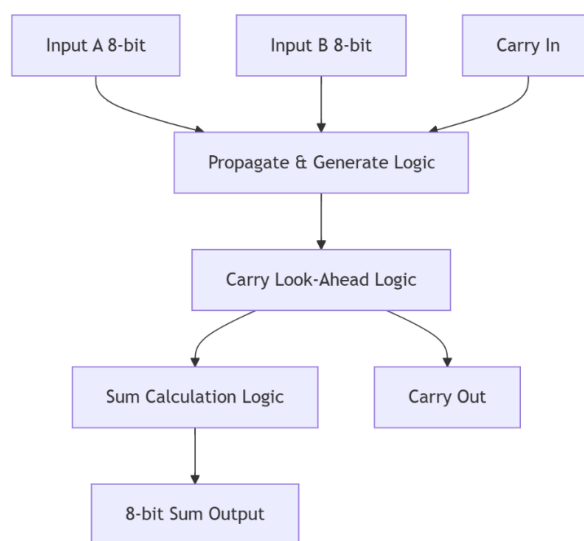
Figură 7

Instrucțiunea PMULW

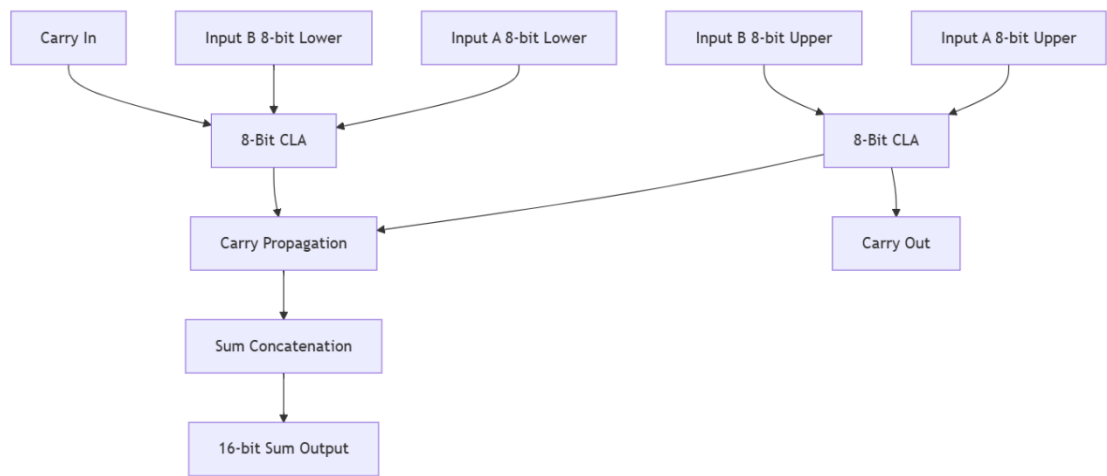


Figură 8

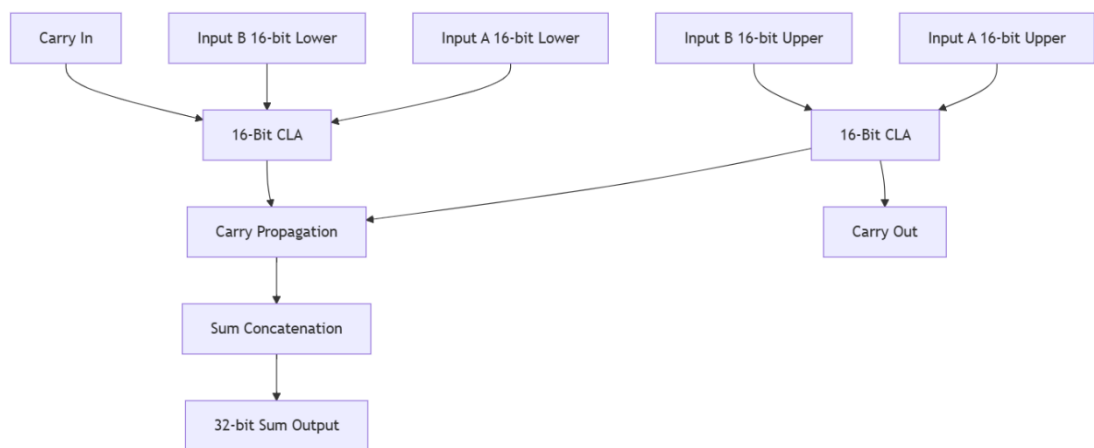
Pentru PADD/PSUB s-au folosit componente CLA_8BIT,CLA_16BIT,CLA_32BIT



Figură 9



Figură 10



Figură 11

Implementare

Implementat PADD/PSUB

Implementare PCMPEQW

Implementare PAND,POR,PXOR