

Лабораторна робота № 4

Моделювання цифрових схем з використанням параметричних елементів

1. Мета роботи.

Набуття досвіду використання параметричних елементів (LPM function) в САПР QUARTUS II, експериментальне дослідження лічильників, побудованих на їх основі.

2. Завдання для виконання практичної роботи.

2.1. Виконати проектування електричної схеми синхронного лічильника з довільним модулем лічби на параметричні елементи САПР QUARTUS II, який задається викладачем згідно таблиці 2.1, або скористатися результатами виконання розрахункової роботи.

Таблиця 2.1. Варіанти завдання для проектування

№ вар.	Тип лічильника	Модуль лічби	№ вар.	Тип лічильника	Модуль лічби
1	додавальний	9	9	додавальний	10
2	віднімаючий	10	10	віднімаючий	11
3	додавальний	11	11	додавальний	12
4	віднімаючий	12	12	віднімаючий	13
5	додавальний	13	13	додавальний	14
6	віднімаючий	14	14	віднімаючий	15
7	додавальний	15	15	реверсивний	10
8	віднімаючий	9	16	реверсивний	12

2.2. Перевірити роботу схеми в сигнальному редакторі і зробити аналіз часових затримок в схемі.

2.3. Відповісти на контрольні питання, оформити звіт про виконану роботу.

3. Основні теоретичні відомості:

Застосування параметричних елементів САПР QUARTUS II для розробки проектів цифрових схем розглянемо на прикладі реалізації синхронного віднімаючого двійкового лічильника з модулем лічби $M=16$.

Для синтезу схеми лічильника на тригерах необхідно вибрати тип тригера. Наприклад, для використання тригерів JK в розділі others → maxplus2 вибираємо JKFF2, JKFFRE, або мікросхему 74113. Але це буде не параметричний синтез, тому, що параметри вказаних логічних пристроїв розробник задавати не може. Для параметричного синтезу використовуються мегафункції системи проектування, які дозволяють за допомогою майстра вибрати, або задати свої параметри пристрою.

Для параметричного синтезу створюємо новий файл графічного редактора і зберігаємо його з певним іменем (наприклад: Ivanow_lab_5) в попередньо створеному каталозі \qdesigns\Ivanow_lab_5\ . В контекстному меню (подвійне натискання правої клавіші миші) відкриваємо підменю вводу символів (Insert → Symbol), вибираємо в бібліотеці (libraries) пункт megafunction, в ній - arithmetic, а в ній - lpm_counter. Після натискання кнопки ОК з'являється вікно з вибраною мегафункцією (рис.3.1).

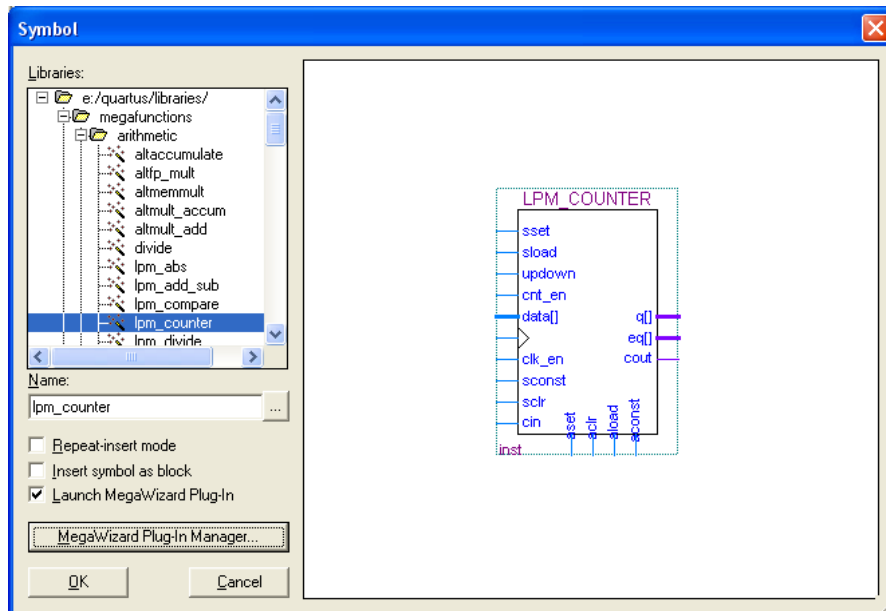


Рис.3.1. Вікно мегафункції lpm_counter.

Для запуску майстра редагування вибраної мегафункції натискаємо кнопку



Запускається майстер мегафункцій і в першому його вікні вибираємо пункт редагування існуючої мегафункції (рис.3.2.). Натискаємо кнопку Next і переходимо до наступного вікна (рис. 3.3).

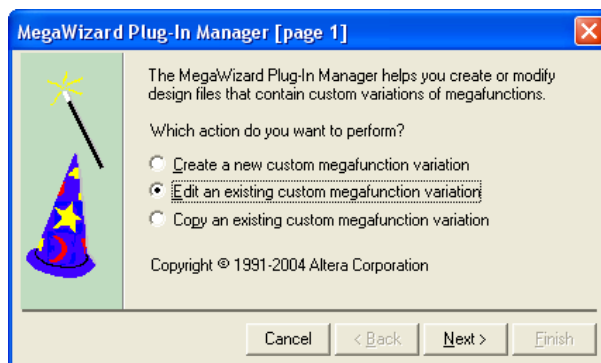


Рис. 3.2. Перше вікно майстра мегафункцій.

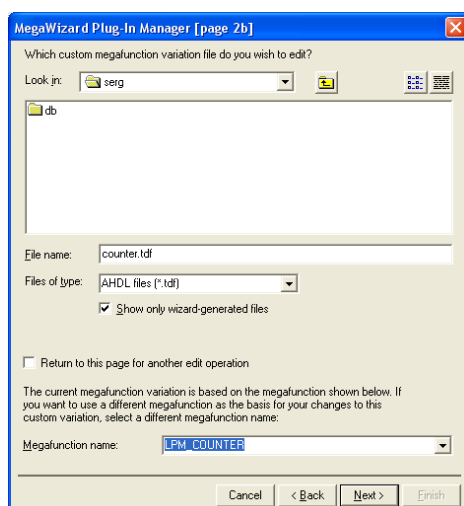


Рис. 3.3. Вікно вибору імені файла, його типу, імені мегафункції.

Задаємо ім'я файла, тип (.tdf), ім'я функції, яка редагується. Натискуємо Next.

В наступному вікні (рис.3.4) вибираємо розрядність (4 біта), напрямок лічби (в даному прикладі - віднімання), натискуємо кнопку Next.

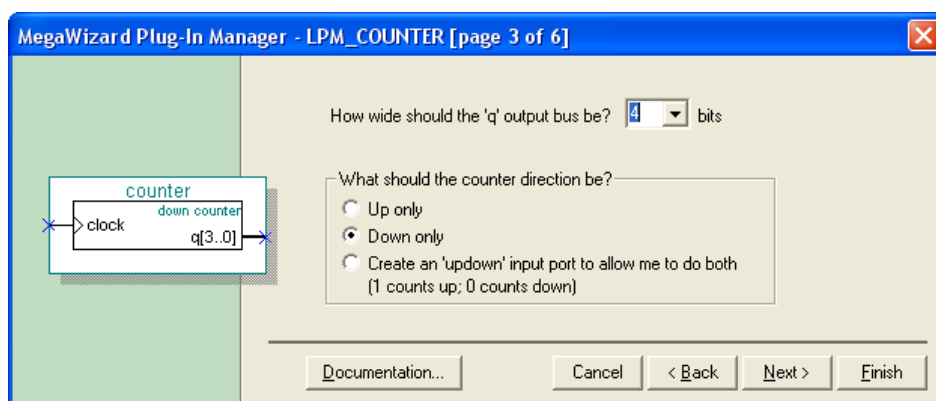


Рис.3.4. Вікно вибору розрядності лічильника, напрямку лічби

В наступному вікні (рис. 3.5.) можемо задати модуль лічби і реалізувати не лише двійковий, а й лічильник з довільним модулем лічби, додати додаткові входи керування. Натискуємо Next.

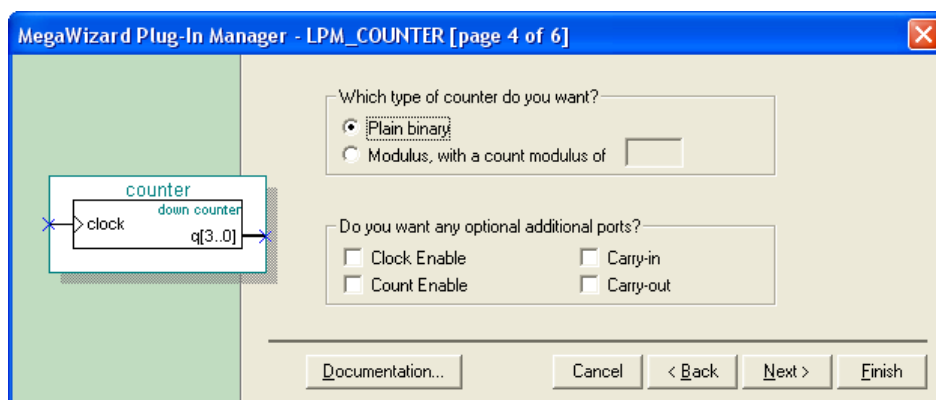


Рис. 3.5. Задання модуля лічби, додаткових входів керування

В вікні (рис. 3.6) задаємо параметри (наприклад, обнулення або встановлення лічильника, завантаження початкового значення) для синхронних і асинхронних входів. Натискуємо Next.

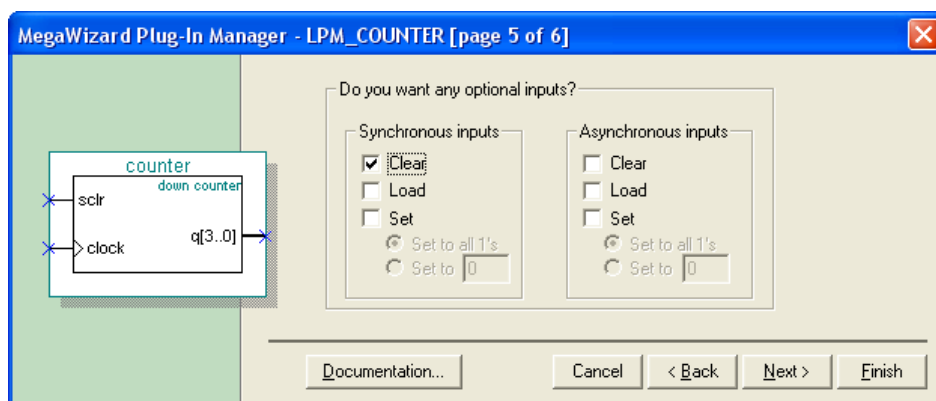


Рис.3.6. Задання параметрів лічильника.

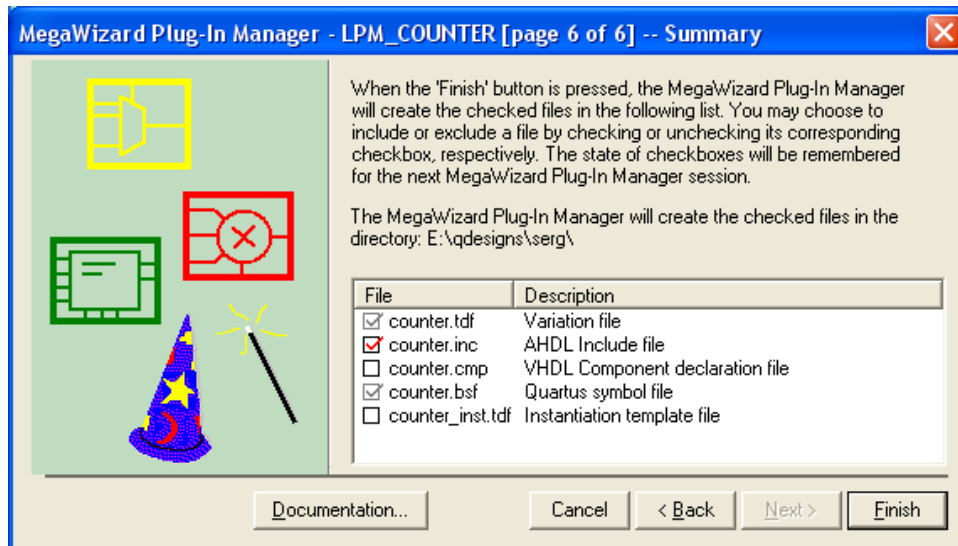


Рис. 3.7.

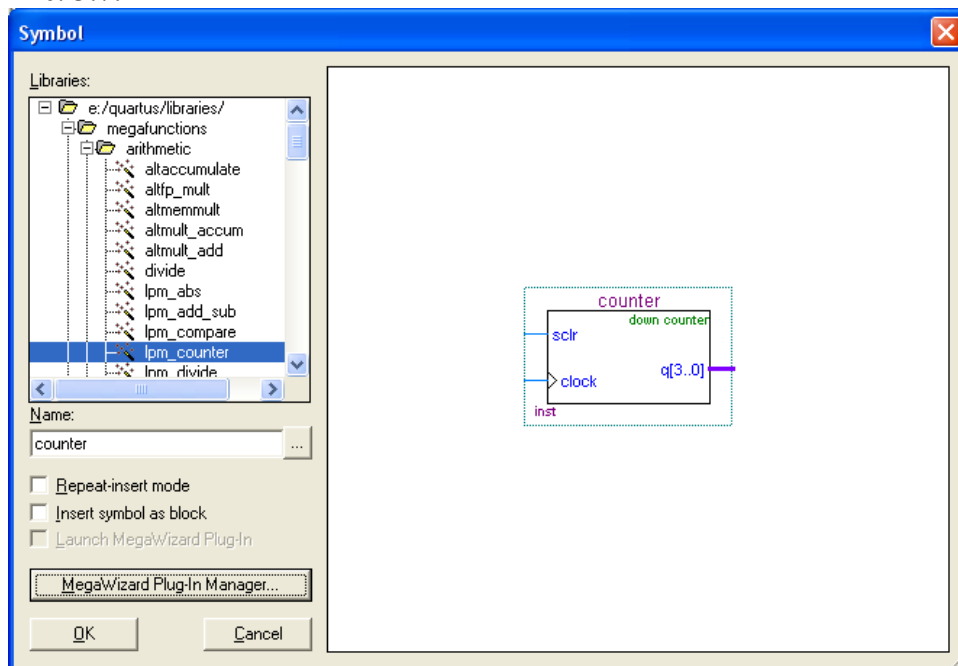


Рис. 3.8.

Далі розміщуємо вхідні і вихідні виводи схеми проекту. Після побудови схеми робимо перевірку на наявність помилок за допомогою компілятора.

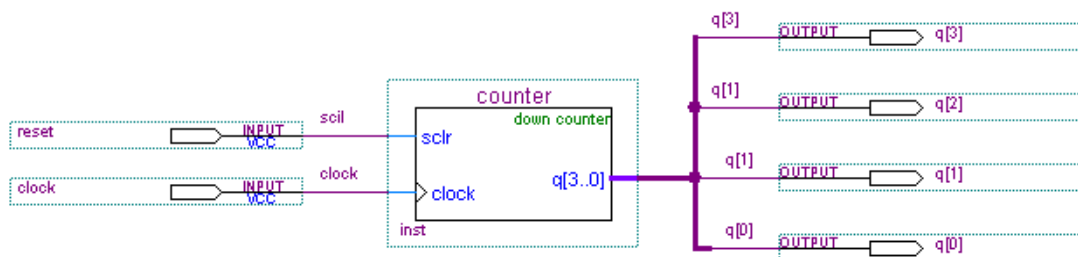


Рис.3.9.

Якщо компіляція виконана успішно, створюємо файл симулятора для аналізу роботи лічильника. В створеному файлі задаємо вхідний (clock) періодичний сигнал з періодом

імпульсів тактування 4 нс. Зберігаємо файл і запускаємо симулятор. Результатом симуляції буде діаграма роботи лічильника, приведена на рис. 3.10

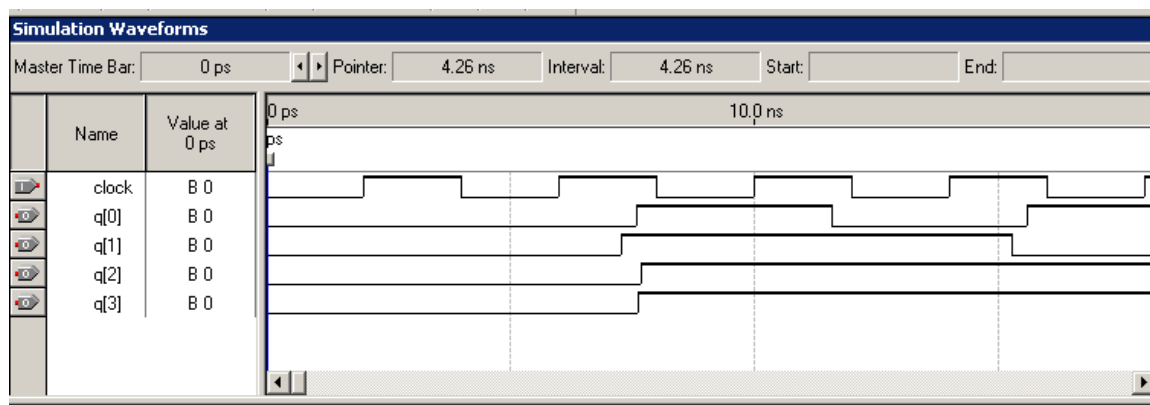


Рис. 3.10. Часові діаграми роботи лічильника

Через пункт меню Timing Analyser Tool (меню Tools) можна побачити затримку розповсюдження сигналу від входу тактування до любого із виходів лічильника (рис. 3.11). Максимальне значення становить 5,7 нс. Такі результати отримані при використанні ПЛІС із сімейства APEX20K. Якщо вибрати ПЛІС іншого сімейства, то часові затримки можуть змінитися.

The figure shows the Timing Analyser Tool window in Quartus II. It displays a table of registered performance data for the counter circuit. The table has columns for Slack, Required tco, Actual tco, From, To, and From Clock. The data is as follows:

	Slack	Required tco	Actual tco	From	To	From Clock
1	N/A	None	5.687 ns	lpm_counter:lpm_counter_component!alt_synth_counter:wysi_counter!load_path[2]	q[2]	clock
2	N/A	None	5.606 ns	lpm_counter:lpm_counter_component!alt_synth_counter:wysi_counter!load_path[3]	q[3]	clock
3	N/A	None	5.374 ns	lpm_counter:lpm_counter_component!alt_synth_counter:wysi_counter!load_path[0]	q[0]	clock
4	N/A	None	5.290 ns	lpm_counter:lpm_counter_component!alt_synth_counter:wysi_counter!load_path[1]	q[1]	clock

Рис. 3.11. Затримка розповсюдження сигналу.

Особливу увагу зверніть на те, що максимальний час затримки розповсюдження сигналу перевищує період тактуючих імпульсів. Тому необхідно зменшити частоту сигналів тактування, щоб їх період не перевищувати максимальний час затримки. Задайте період тактуючих сигналів 10 нс і виконайте симуляцію для нового значення частоти тактування. Переконайтеся, що тепер незважаючи на затримку сигнали на виході лічильника змінюються у відповідних періодах тактуючих сигналів.

Описи деяких параметричних елементів САПР QUARTUS II приведені в додатку А.

4. Контрольні питання.

- 4.1. Поясніть термін "параметричний елемент". Які параметричні елементи можна використати в САПР QUARTUS II?
- 4.2. Поясніть принцип роботи лічильника, побудованого на тригерах. Які типи лічильників існують?
- 4.3. Поясніть призначення пунктів меню.
- 4.4. Чим обмежується максимальна швидкість роботи лічильника? Яка максимальна частота роботи лічильника, зпроектованого в практичній роботі?

Параметричні елементи САПР QUARTUS II
Counter (Лічильник)

<i>Вхідні виводи</i>	
Ім'я виводу	Опис
data []	Паралельний вхід даних лічильника.
clock	Вхід лічильних імпульсів.
clk_en	Дозвіл синхронізації.
cnt_en	Дозвіл лічби.
updown	Керування напрямком лічби (1 = додавання, 0 = віднімання)
aclr	Асинхронне обнулення.
aset	Асинхронне встановлення.
aload	Асинхронне завантаження. Встановлення лічильника в значення data[].
sclr	Синхронне обнулення. Обнулення лічильника наступним тактуючим імпульсом
sset	Синхронне встановлення входів. Встановлення лічби наступним тактуючим імпульсом.
sload	Синхронне завантаження. Завантаження в лічильник значення data[] наступним тактуючим імпульсом.

<i>Вихідні виводи</i>	
Ім'я виводу	Опис
q []	Вихід лічильника.
eq[15..0]	Декодований вихід лічильника. Високий активний рівень з'являється в момент, коли лічильник досягає заданого значення.
cout	Перенесення в старший розряд.

<i>Параметри</i>	
Параметр	Опис
LPM_WIDTH	Розрядність лічильника, або вхідних значень data[] і вихідних q[].
LPM_DIRECTION	Може приймати значення "UP", "DOWN", або "UNUSED". Якщо цей параметр використовується, то вхід updown не повинен бути під'єднаний. Якщо вхід updown не під'єднаний, то значення LPM DIRECTION по замовчуванню - "UP"

LPM_MODULUS	Максимальне число, плюс один (модуль лічби M). Число унікальних станів в циклі лічильника. Якщо введене значення більше, ніж параметр LPM MODULUS, поведінка лічильника не визначена.
LPM_AVALUE	Постійне значення, яке завантажується, коли aset високий. Якщо введене значення більше, ніж <modulus>, поведінка лічильника - невизначений (X) логічний рівень, де <modulus> - LPM_MODULUS. Параметр обмежений значенням в 32 біта.
LPM_SVALUE	Постійне значення, яке завантажується по передньому фронту тактуючих імпульсів, коли sset або sconst високі. Повинен використовуватись, якщо використовується sconst.
LPM_HINT	Дозволяє задати специфічні Altera-параметри в файлах проекту VHDL.
LPM_TYPE	Ідентифікує LPM ім'я файлів проекту VHDL.

Divider (Подільник)

<i>Вхідні виводи</i>	
Ім'я виводу	Опис
numer[]	Чисельник.
denom[]	Знаменник.
clock	Вхід тактуючих імпульсів.
clken	Дозвіл використання входу тактування.
aclr	Асинхроне обнулення.

<i>Вихідні виводи</i>	
Ім'я виводу	Опис
quotient[]	Частка.
remain[]	Залишок.
<i>Параметри</i>	
Параметр	Опис
LPM_WIDTHHN	Розрядність numer[] і quotient[].
LPM_WIDTHHD	Розрядність denom[] і remain[].
LPM_NREPRESENTATION	Визначає параметр чисельника "SIGNED" або "UNSIGNED" .
LPM_DREPRESENTATION	Визначає параметр знаменника "SIGNED" або "UNSIGNED".
LPM_HINT	Дозволяє задати специфічні Altera-параметри в файлах проекту VHDL.
LPM_TYPE	Ідентифікує LPM ім'я файлів проекту VHDL.

Multiplier (Перемножувач)

<i>Вхідні виводи</i>	
Ім'я виводу	Опис
dataa []	Множене
datab []	Множник
sum[]	Часткова сума
clock	Вхід імпульсів тактування
clken	Дозвіл використання входу тактування
aclr	Асинхронне обнулення

<i>Вихідні виводи</i>	
Ім'я виводу	Опис
result[]	result = dataa[] * datab[] + sum.

<i>Параметри</i>	
Параметр	Опис
LPM_WIDTHA	Розрядність dataa[].
LPM_WIDTHB	Розрядність datab[].
LPM_WIDTHP	Розрядність result[].
LPM_WIDTHS	Розрядність sum[]. Обов'язковий, навіть якщо порт суми не використовується.
LPM_REPRESENTATION	Тип порівняння, що виконується ("SIGNED", "UNSIGNED", "UNUSED"). Якщо значення не вказане, то по замовчуванню встановлюється "UNSIGNED"
LPM_HINT	Дозволяє задати специфічні Altera-параметри в файлах проекту VHDL.
LPM_TYPE	Ідентифікує LPM ім'я файліх проекту VHDL
INPUT_A_IS_CONSTANT	Altera параметр. Приймає значення "YES", "NO" і "UNUSED". Якщо dataa [] зв'язаний з постійним значенням то, встановлюючи INPUT_A_IS_CONSTANT "YES" оптимізуємо <i>multiplier</i> по використанню ресурсів і швидкості. Якщо відсутній, то значення по замовчуванню - "NO".
INPUT_B_IS_CONSTANT	Altera параметр. Приймає значення "YES", "NO" і "UNUSED". Якщо datab [] зв'язаний з постійним значенням, встановлюючи INPUT_A_IS_CONSTANT "YES" оптимізуємо <i>multiplier</i> по використанню ресурсів і швидкості. Якщо відсутній, то значення по замовчуванню - "NO".

USE_EAB	Altera параметр. Приймає значення "ON", "OFF" і "UNUSED". Встановлюючи параметр USE_EAB "ON" дозволяємо QUARTUS II використовувати блоки додаткових атрибутів, щоб використати (4 x 4) або (8 x 8) стандартні блоки в ACEX1K и FLEX10K пристроях.
LATENCY	Altera параметр. Те ж саме, що і LPM PIPELINE. Для всіх нових проектів, використовується параметр LPM PIPELINE
MAXIMIZE_SPEED	Altera параметр. Можливі значення від 0 до 10. Якщо параметр використовується, то QUARTUS II пробує оптимізувати дану функцію lpm_mult для швидкості, а не для зменшення області, що використовується, і відміняє встановлення опції Optimize в діалоговому вікні Global Project Logic Synthesis (меню Assign). Якщо MAXIMIZE SPEED не використовується, то значення опції Optimize використовується замість нього. Якщо встановлено MAXIMIZE_SPEED - 6 або більше, то компілятор оптимізує мегафункції lpm_mult для більшої швидкості; якщо встановлено - 5 або менше - компілятор оптимізує для зменшення області, що використовується.

Adder-Subtractor (Суматор-Віднімач)

<i>Вхідні виводи</i>	
Ім'я виводу	Опис
dataa []	Перший доданок / Зменшуване
datab []	Доданок / Від'ємник
add_sub	Якщо "1" (high), операція = dataa[]+datab[] +cin. Якщо "0" (low), операція = dataa[]-datab[] +cin-1
clock	Вхід імпульсів тактування
clken	Дозвіл використання входу тактування
aclr	Асинхронне обнулення

<i>Вихідні виводи</i>	
Ім'я виводу	Опис
result[]	dataa[] +datab[] +cin або dataa[] - datab[] +cin-1.
cout	Виявляє переповнення в операціях "UNSIGNED".
overflow	Результат більший від можливої точності (переповнення)

<i>Параметри</i>	
Параметр	Опис
LPM_WIDTH	Розрядність dataa[], datab[], result[].
LPM_DIRECTION	Значення - "ADD", "SUB", і "UNUSED". якщо не вказане, то значення по замовчуванню "DEFAULT", в цьому випадку використовується значення add_sub порта. Add_sub порт не може використовуватися, якщо використовується LPM DIRECTION.
LPM_REPRESENTATION	Тип виконуваного порівняння "SIGNED", "UNSIGNED", "UNUSED". Якщо значення не вказане, то по замовчуванню встановлюється "UNSIGNED"
LPM_HINT	Дозволяє задати специфічні Altera-параметри в файлах проекту VHDL.
LPM_TYPE	Ідентифікує LPM ім'я файлів проекту VHDL
ONE_INPUT_IS_CONSTANT	Altera параметр. Приймає значення "YES", "NO" і "UNUSED". Забезпечує ефективнішу оптимізацію, якщо один вхід постійний. Якщо не вказане, то значення по замовчуванню - "NO"
MAXIMIZE_SPEED	Altera параметр. Можливі значення від 0 до 10. Якщо параметр використовується, то QUARTUS II пробує оптимізувати дану функцію lpm mult для швидкості, а не для зменшення області, що використовується, і відміння встановлення опції Optimize в діалоговому вікні Global Project Logic Synthesis (меню Assign). Якщо MAXIMIZE SPEED не використаний, то значення опції Optimize використовується замість нього. Якщо встановлено MAXIMIZE_SPEED - 6 або більше, то компілятор оптимізує мегафункції lpm_mult для більшої швидкості; якщо встановлено -5 або менше - компілятор оптимізує зменшення області, що використовується.