Architektura Systemów Komputerowych

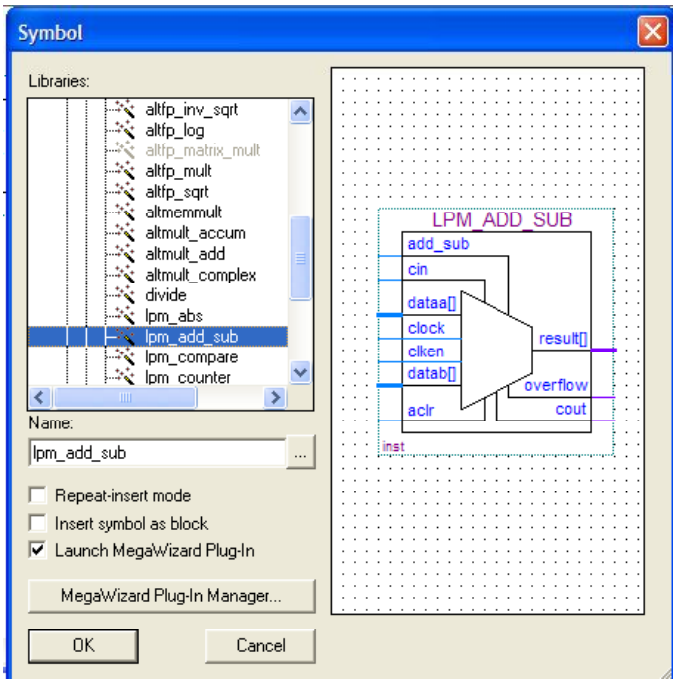
Laboratorium 5

Przykład prostego ALU

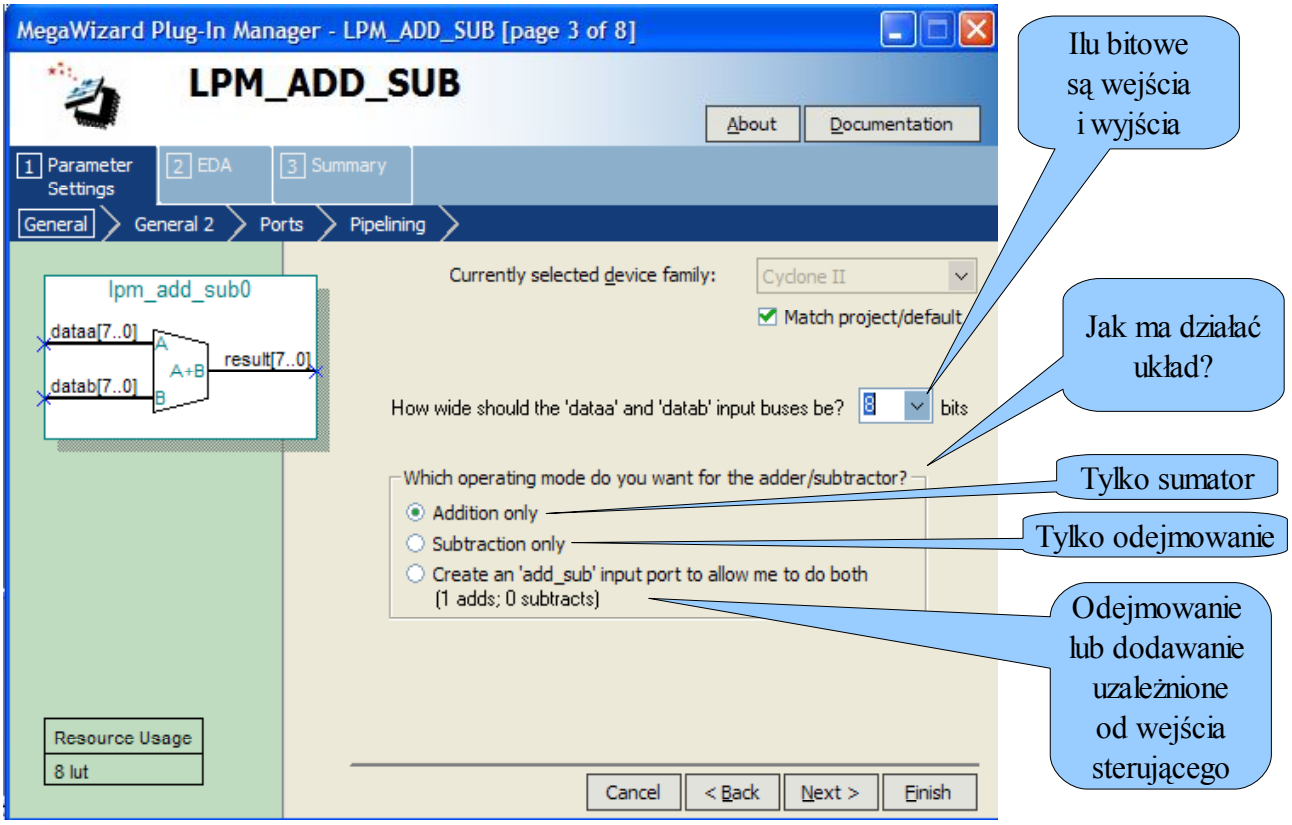
mgr inż. Leszek Ciopiński

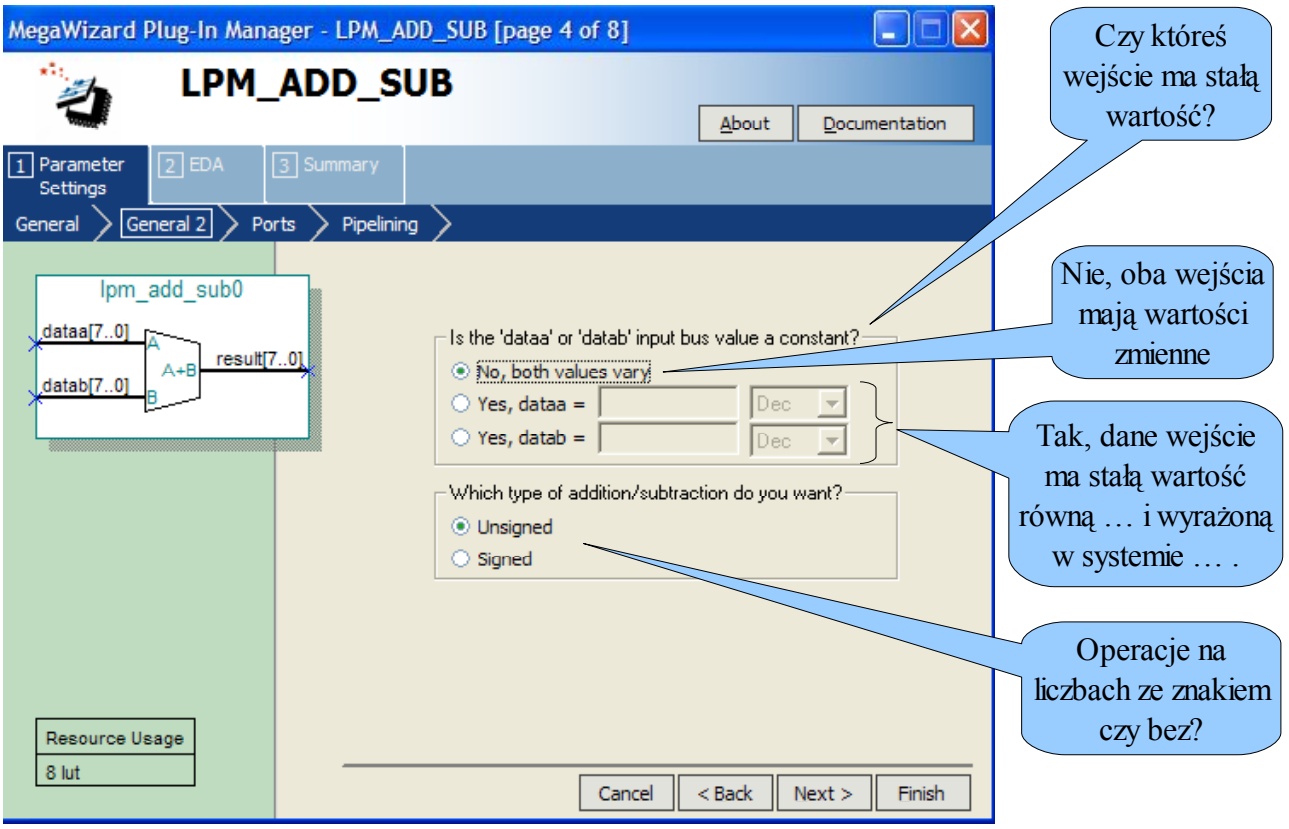
Wybrane układy z biblioteki standardowej – Altera Quartus II

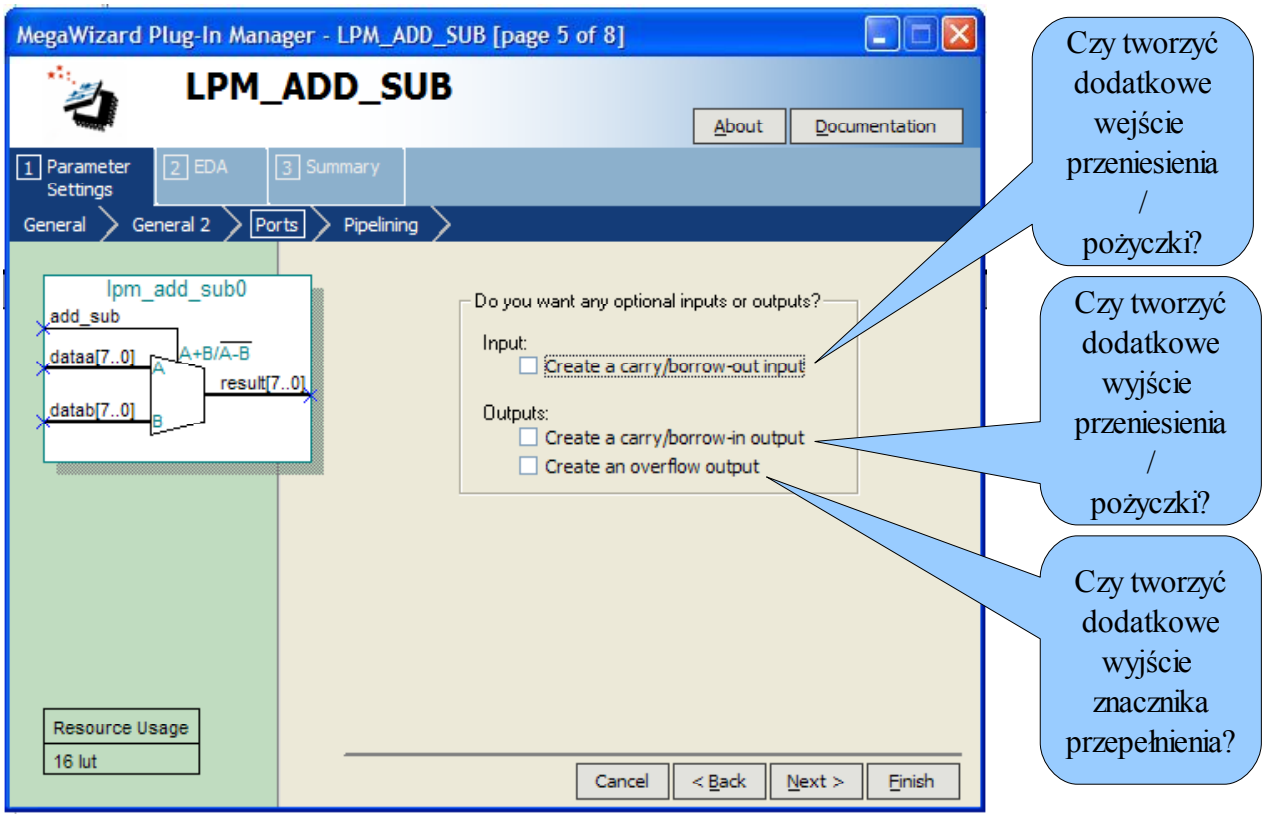
lpm_add_sub

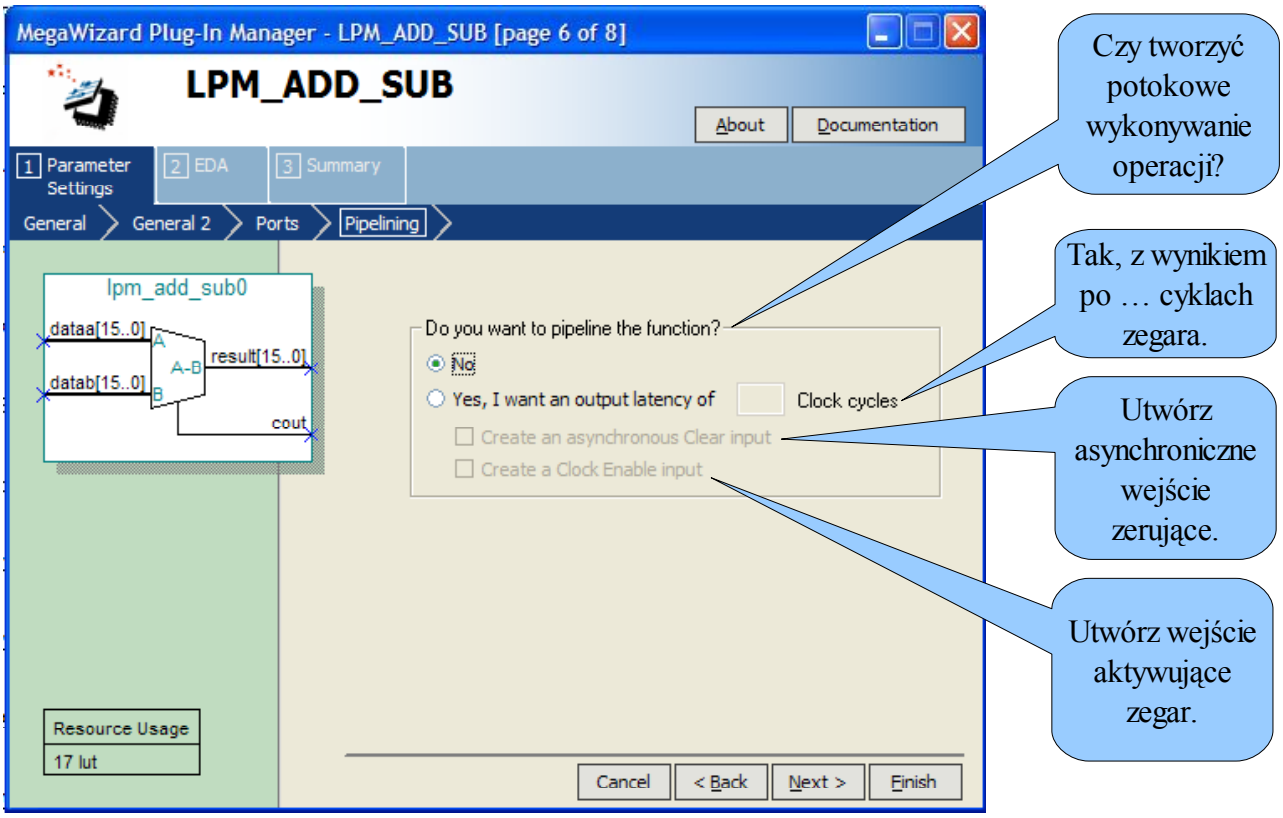


Układ ten należy do grupy MegaFunctions (dokładna lokalizacja to: megafunctions/arithmetic/lpm_add_sub). Oznacza to, że dokładne właściwości układu można dostosować w zależności od potrzeb projektanta. Po wybraniu tego układu otworzone zostanie okno MegaWizard, w którym należy dokonać wyboru języka w jakim opisany zostanie układ (proszę wybierać język VHDL) oraz miejsce i nazwę pod jaką układ ma zostać zapisany. Poniżej przedstawiono opis najważniejszych ustawień:


MegaWizard Plug-In Manager - LPM_ADD_SUB [page 3 of 8]
LPM_ADD_SUB
About
Documentation
1 Parameter Settings
2 EDA
3 Summary
General
General 2
Ports
Pipelining
lpm_add_sub0
dataa[7..0]
datab[7..0]
A+B
result[7..0]
Currently selected device family:
Cyclone II
Match project/default
How wide should the 'dataa' and 'datab' input buses be?
8
bits
Which operating mode do you want for the adder/subtractor?
Addition only
Subtraction only
Create an 'add_sub' input port to allow me to do both
(1 adds; 0 subtracts)
Resource Usage
8 lut
Cancel
< Back
Next >
Finish
Ilu bitowe są wejścia i wyjścia
Jak ma działać układ?
Tylko sumator
Tylko odejmowanie
Odejmowanie lub dodawanie uzależnione od wejścia sterującego

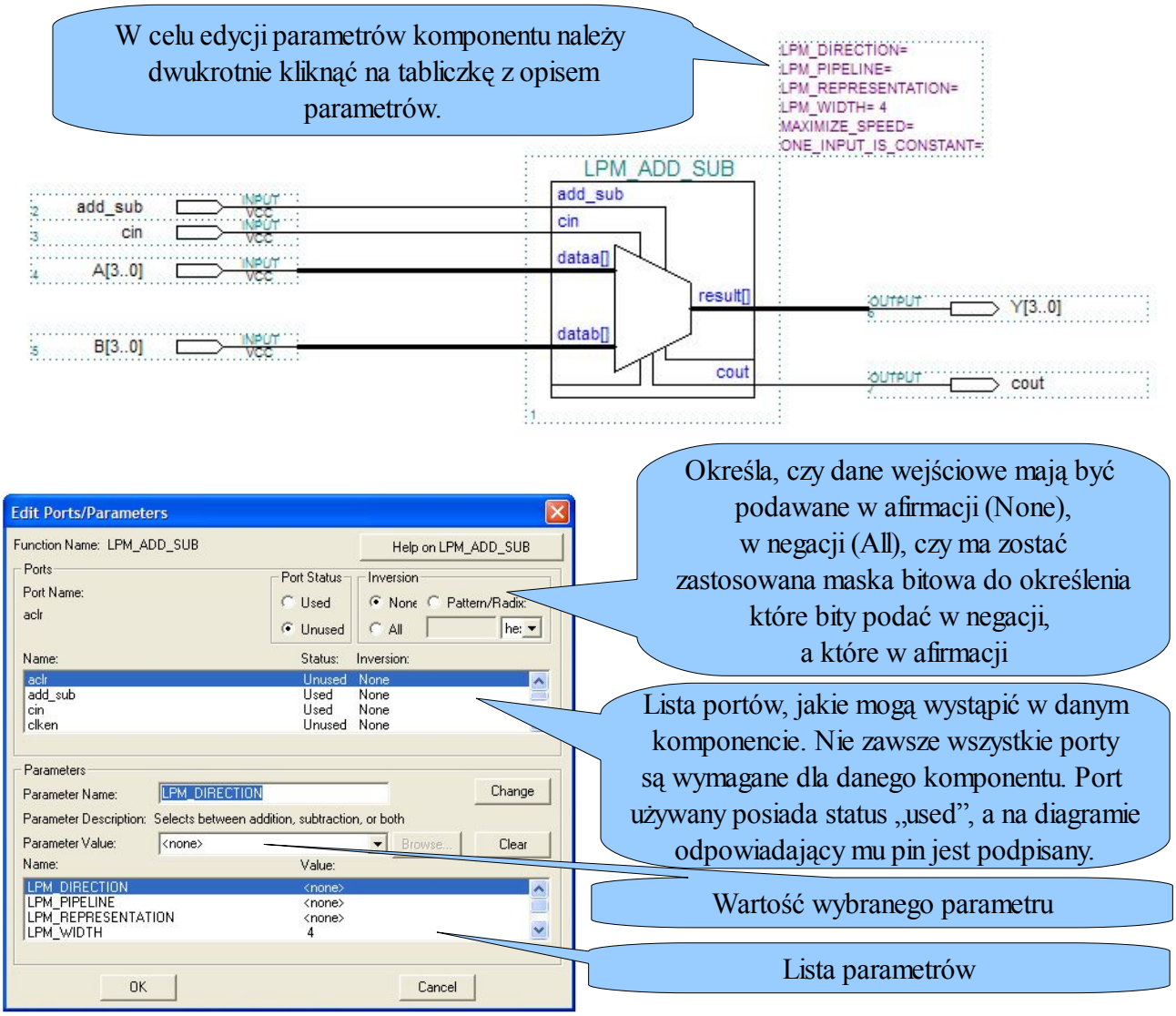

MegaWizard Plug-In Manager - LPM_ADD_SUB [page 4 of 8]
LPM_ADD_SUB
About
Documentation
1 Parameter Settings
2 EDA
3 Summary
General
General 2
Ports
Pipelining
lpm_add_sub0
dataa[7..0]
datab[7..0]
A+B
result[7..0]
Is the 'dataa' or 'datab' input bus value a constant?
No, both values vary
Yes, dataa =
Yes, datab =
Dec
Which type of addition/subtraction do you want?
Unsigned
Signed
Resource Usage
8 lut
Cancel
< Back
Next >
Finish
Czy któreś wejście ma stałą wartość?
Nie, oba wejścia mają wartości zmienne
Tak, dane wejście ma stałą wartość równą … i wyrażoną w systemie … .
Operacje na liczbach ze znakiem czy bez?


MegaWizard Plug-In Manager - LPM_ADD_SUB [page 5 of 8]
LPM_ADD_SUB
About
Documentation
1 Parameter Settings
2 EDA
3 Summary
General
General 2
Ports
Pipelining
lpm_add_sub0
add_sub
dataa[7..0]
datab[7..0]
A+B/A-B
result[7..0]
Do you want any optional inputs or outputs?
Input:
Create a carry/borrow-out input
Outputs:
Create a carry/borrow-in output
Create an overflow output
Resource Usage
16 lut
Cancel
< Back
Next >
Finish
Czy tworzyć dodatkowe wejście przeniesienia / pożyczki?
Czy tworzyć dodatkowe wyjście przeniesienia / pożyczki?
Czy tworzyć dodatkowe wyjście znacznika przepełnienia?


MegaWizard Plug-In Manager - LPM_ADD_SUB [page 6 of 8]
LPM_ADD_SUB
About
Documentation
1 Parameter Settings
2 EDA
3 Summary
General
General 2
Ports
Pipelining
lpm_add_sub0
dataa[15..0]
datab[15..0]
A-B
result[15..0]
cout
Do you want to pipeline the function?
No
Yes, I want an output latency of
Clock cycles
Create an asynchronous Clear input
Create a Clock Enable input
Resource Usage
17 lut
Cancel
< Back
Next >
Finish
Czy tworzyć potokowe wykonywanie operacji?
Tak, z wynikiem po … cyklach zegara.
Utwórz asynchroniczne wejście zerujące.
Utwórz wejście aktywujące zegar.

Wybrane układy z biblioteki standardowej – Altera MAX+plus II

lpm_add_sub – układ dodająco-odejmujący



Czy dany port jest wymagany

Przy pominięciu, wartość wynosi 0.

INPUTS

Port Name	Required	Description	Comments
cin	No	Carry-in to the low-order bit.	If omitted, the default is 0.
dataa[]	Yes	Augend/Minuend.	Input port LPM_WIDTH wide.
datab[]	Yes	Addend/Subtrahend.	Input port LPM_WIDTH wide.
add_sub	No	If the signal is high, the operation = dataa[]+datab[] +cin. If the signal is low, the operation = dataa[]–datab[] +cin–1.	If the LPM_DIRECTION parameter is used, add_sub cannot be used. If omitted, the default is "ADD". Altera recommends that you use the LPM_DIRECTION parameter to specify the operation of the lpm_add_sub function, rather than assigning a constant to the add_sub port.
clock	No	Clock for pipelined usage.	The clock port provides pipelined operation for the lpm_add_sub function. For LPM_PIPELINE values other than 0 (default value), the clock port must be connected.
clken	No	Clock enable for pipelined usage.	If omitted, the default is 1.
aclr	No	Asynchronous Clear for pipelined usage.	The pipeline initializes to an undefined (X) logic level. The aclr port can be used at any time to reset the pipeline to all 0's, asynchronously to the clock signal.

OUTPUTS

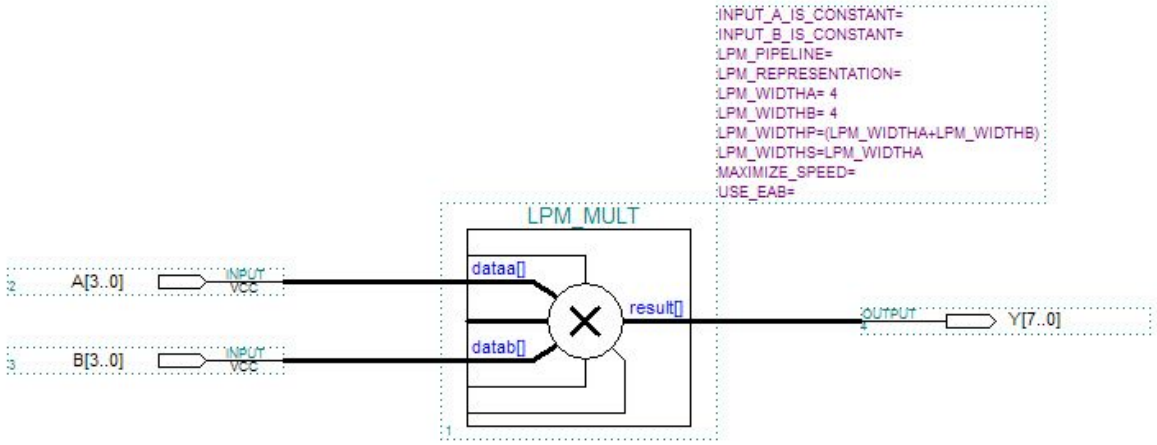
Port Name	Required	Description	Comments
result[]	Yes	dataa[] +datab[] +cin or dataa[] –datab[] +cin–1.	Output port LPM_WIDTH wide.
cout	No	Carry-out (borrow-in) of the MSB. *Note 1*	If overflow is used, cout cannot be used. The cout port has a physical interpretation as the carry-out (borrow-in) of the MSB. cout is most meaningful for detecting overflow in "UNSIGNED" operations.
overflow	No	Result exceeds available precision. *Note 2*	If overflow is used, cout cannot be used. The overflow port has a physical interpretation as the XOR of the carry-in to the MSB with the carry-out of the MSB. overflow is meaningful only when the LPM_REPRESENTATION parameter value is "SIGNED".

Parametr określający szerokość (ilość bitów) danych wejściowych i wyjściowych

Parameters:

Parameter	Type	Required	Description
LPM_WIDTH	Integer	Yes	Width of the dataa[], datab[], and result[] ports.
LPM_DIRECTION	String	No	Values are "ADD", "SUB", and "UNUSED". If omitted, the default is "DEFAULT", which directs the parameter to take its value from the add_sub port. The add_sub port cannot be used if LPM_DIRECTION is used. Altera recommends that you use the LPM_DIRECTION parameter to specify the operation of the lpm_add_sub function, rather than assigning a constant to the add_sub port.
LPM_REPRESENTATION	String	No	Type of addition performed: "SIGNED", "UNSIGNED", or "UNUSED". If omitted, the default is "SIGNED".
LPM_PIPELINE	Integer	No	Specifies the number of Clock cycles of latency associated with the result[] output. A value of zero (0) indicates that no latency exists, and that a purely combinatorial function will be instantiated. If omitted, the default is 0 (non-pipelined).
LPM_HINT	String	No	Allows you to specify Altera-specific parameters in VHDL Design Files. The default is "UNUSED".
LPM_TYPE	String	No	Identifies the LPM entity name in VHDL Design Files.
ONE_INPUT_IS_CONSTANT	String	No	Altera-specific parameter. Values are "YES", "NO", and "UNUSED". Provides greater optimization if one input is constant. If omitted, the default is "NO".
MAXIMIZE_SPEED	Integer	No	Altera-specific parameter. You can specify a value between 0 and 10. If used, MAX+PLUS II attempts to optimize a specific instance of the lpm_add_sub function for speed rather than area, and overrides the setting of the *Optimize* option in the **Global Project Logic Synthesis** dialog box (Assign menu). If MAXIMIZE_SPEED is unused, the value of the *Optimize* option is used instead. If the setting for MAXIMIZE_SPEED is 6 or higher, the Compiler will optimize lpm_add_sub megafunctions for higher speed; if the setting is 5 or less, the Compiler will optimize for smaller area.

lpm_mult – układ mnożący



INPUTS

Port Name	Required	Description	Comments
dataa[]	Yes	Multiplicand.	Input port LPM_WIDTHA wide.
datab[]	Yes	Multiplier.	Input port LPM_WIDTHB wide.
sum[]	No	Partial sum.	Input port LPM_WIDTHS wide.
clock	No	Clock for pipelined usage.	The clock port provides pipelined operation for the lpm_mult function. For LPM_PIPELINE values other than 0 (default value), the clock port must be connected.
clken	No	Clock enable for pipelined usage.	Available for VHDL only.
aclr	No	Asynchronous Clear for pipelined usage.	The pipeline initializes to an undefined (X) logic level. The aclr port can be used at any time to reset the pipeline to all 0's, asynchronously to the clock signal.

OUTPUTS

Port Name	Required	Description	Comments
result[]	Yes	result = dataa[] * datab[] + sum. The product LSB is aligned with the sum LSB.	Output port LPM_WIDTHP wide. If LPM_WIDTHP < max (LPM_WIDTHA + LPM_WIDTHB, LPM_WIDTHS) or (LPM_WIDTHA + LPM_WIDTHS), only the LPM_WIDTHP MSBs are present.

Parameters:

Parameter	Type	Required	Description
LPM_WIDTHA	Integer	Yes	Width of the dataa[] port.
LPM_WIDTHB	Integer	Yes	Width of the datab[] port.
LPM_WIDTHP	Integer	Yes	Width of the result[] port.
LPM_WIDTHS	Integer	Yes	Width of the sum[] port. Required even if the sum port is not used.
LPM_REPRESENTATION	String	No	Type of multiplication performed: "SIGNED", "UNSIGNED", or "UNUSED". If omitted, the default is "UNSIGNED".
LPM_PIPELINE	Integer	No	Specifies the number of Clock cycles of latency associated with the result[] output. A value of zero (0) indicates that no latency exists, and that a purely combinatorial function will be instantiated. If omitted, the default is 0 (non-pipelined).

Parameter	Type	Required	Description
LPM_WIDTHA	Integer	Yes	Width of the dataa[] port.
LPM_WIDTHB	Integer	Yes	Width of the datab[] port.
LPM_WIDTHP	Integer	Yes	Width of the result[] port.
LPM_WIDTHS	Integer	Yes	Width of the sum[] port. Required even if the sum port is not used.
LPM_REPRESENTATION	String	No	Type of multiplication performed: "SIGNED", "UNSIGNED", or "UNUSED". If omitted, the default is "UNSIGNED".
LPM_PIPELINE	Integer	No	Specifies the number of Clock cycles of latency associated with the result[] output. A value of zero (0) indicates that no latency exists, and that a purely combinatorial function will be instantiated. If omitted, the default is 0 (non-pipelined).
LPM_HINT	String	No	Allows you to assign Altera-specific parameters in VHDL Design Files. The default is "UNUSED".
LPM_TYPE	String	No	Identifies the LPM entity name in VHDL Design Files.
INPUT_A_IS_CONSTANT	String	No	Altera-specific parameter. Values are "YES", "NO", and "UNUSED". If dataa[] is connected to a constant value, setting INPUT_A_IS_CONSTANT to "YES" optimizes the multiplier for resource usage and speed. If omitted, the default is "NO".
INPUT_B_IS_CONSTANT	String	No	Altera-specific parameter. Values are "YES", "NO", and "UNUSED". If datab[] is connected to a constant value, setting INPUT_B_IS_CONSTANT to "YES" optimizes the multiplier for resource usage and speed. The default is "NO".
USE_EAB	String	No	Altera-specific parameter. Values are "ON", "OFF", and "UNUSED". Setting the USE_EAB parameter to "ON" allows MAX+PLUS II to use EABs to implement 4 x 4 or (8 x const value) building blocks in ACEX 1K and FLEX 10K devices. Altera recommends that you set USE_EAB to "ON" only when LCELLS are in short supply. If you wish to use this parameter when you instantiate the function in a GDF, you must specify it by entering the parameter name and value manually with the **Edit Ports/Parameters** dialog box (Symbol menu). You can also use this parameter name in a TDF or a Verilog Design File. You must use the LPM_HINT parameter to specify the USE_EAB parameter in VHDL Design Files.
LATENCY	Integer	No	Altera-specific parameter. Same as LPM_PIPELINE. (This parameter is provided only for backward compatibility with MAX+PLUS II pre-version 7.0 designs. For all new designs, you should use the LPM_PIPELINE parameter instead.)
MAXIMIZE_SPEED	Integer	No	Altera-specific parameter. You can specify a value between 0 and 10. If used, MAX+PLUS II attempts to optimize a specific instance of the lpm_mult function for speed rather than area, and overrides the setting of the *Optimize* option in the **Global Project Logic Synthesis** dialog box (Assign menu). If MAXIMIZE_SPEED is unused, the value of the *Optimize* option is used instead. If the setting for MAXIMIZE_SPEED is 6 or higher, the Compiler will optimize lpm_mult megafunctions for higher speed; if the setting is 5 or less, the Compiler will optimize for smaller area.

Zadania:

1A. Zaprojektować i skompilować układ AND czterobitowy. Cechy układu:

- Układ ma dwa wejścia, każde po 4 bity.
- Układ ma jedno wyjście czterobitowe
- Operacja AND wykonywana jest na odpowiadających sobie bitach (np. drugi z drugim) *(1 punkt)*

1B. Zaprojektować i skompilować układ OR czterobitowy. Cechy układu:

- Układ ma dwa wejścia, każde po 4 bity.
- Układ ma jedno wyjście czterobitowe
- Operacja OR wykonywana jest na odpowiadających sobie bitach (np. drugi z drugim) *(1 punkt)*

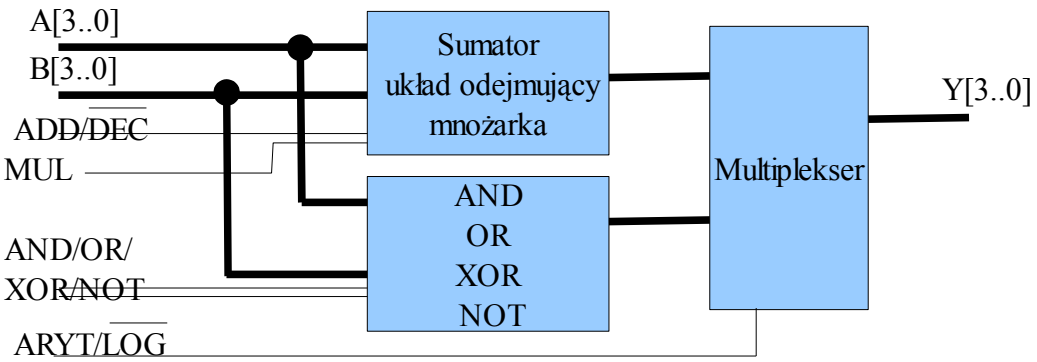
1C. Zaprojektować i skompilować układ XOR czterobitowy. Cechy układu:

- Układ ma dwa wejścia, każde po 4 bity.
- Układ ma jedno wyjście czterobitowe
- Operacja XOR wykonywana jest na odpowiadających sobie bitach (np. drugi z drugim)
  (1 punkt)

1D. Zaprojektować i skompilować układ NOT czterobitowy. Cechy układu:

- Układ ma jedno czterobitowe wejście
- Układ ma jedno wyjście czterobitowe
- Operacja NOT wykonywana jest na każdym bicie.
  (1 punkt)

2. Przetestować układy z zadania 1. Należy przedstawić dwie dowolne kombinacje. Dodatkowo na kartce należy wykonać operację AND, OR, XOR i NOT dla każdej testowanej pary wartości, potwierdzającej prawidłowość wykonanej symulacji.
   (2 punkty)
3. Zaprojektować i skompilować układ uproszczonego ALU, którego schemat ideowy przedstawiono poniżej:



   Układ logiczny musi być symbolem utworzonym z układów z pierwszego zadania.
   Układ arytmetyczny musi być dołączony do projektu w postaci symbolu i utworzony z komponentów lpm_add_sub i lpm_mult.
   Wejścia sterujące (wybór układu i operacji arytmetycznej lub logicznej) muszą być połączone w jedną szynę sterującą (oczywiście 3-bitową)
   (5 punktów)
4. Przetestować układ z zadania 3 w sposób umożliwiający weryfikację jego poprawnego zaimplementowania.
   (4 punkty)