

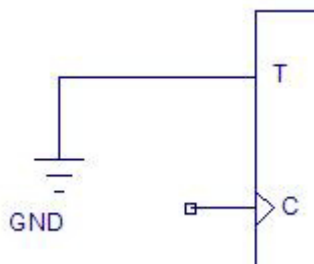
Projektowanie układów na schemacie

Przedstawione poniżej wskazówki mogą być pomocne przy projektowaniu układach na poziomie schematu.

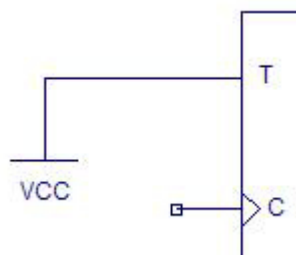
Stałe wartości logiczne

Aby podłączyć wejście do stałej wartości logicznych używamy symboli z grupy *General*:

GND – logiczne 0



VCC – logiczne 1



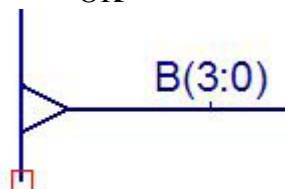
Połączenia

Przy łączeniu elementów warto zwrócić uwagę na czerwone markery, które mogą sygnalizować brak połączenia. W przypadku końca magistrali czerwony marker nie sygnalizuje nieprawidłowości:

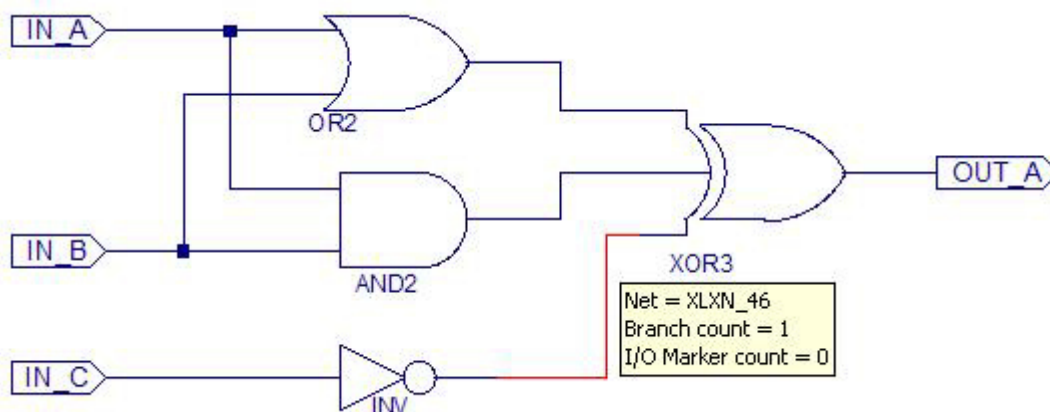
Brak połączenia



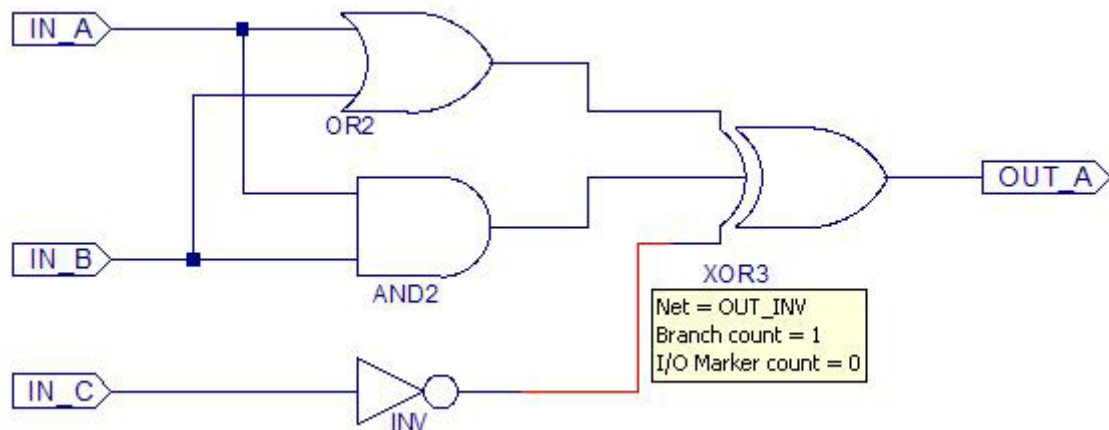
OK



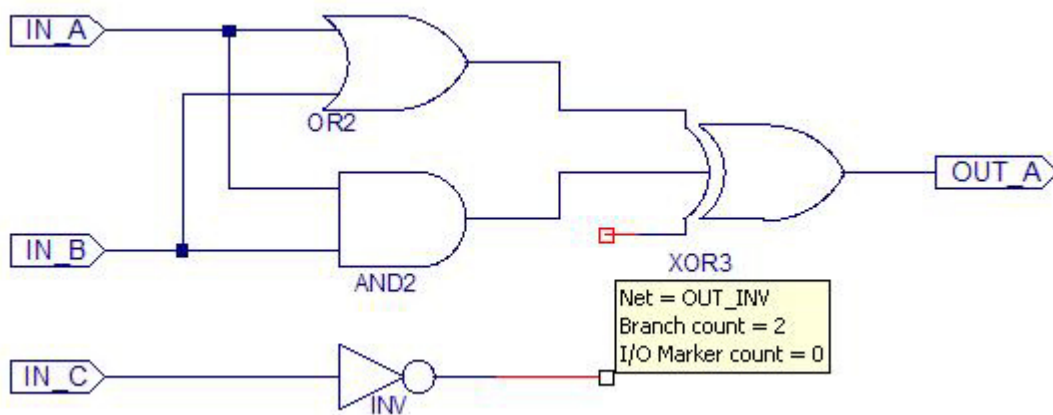
Każde połączenie ma swoją nazwę, nadawaną automatycznie na podstawie nazwy portu lub w postaci *XLXN_nn*:



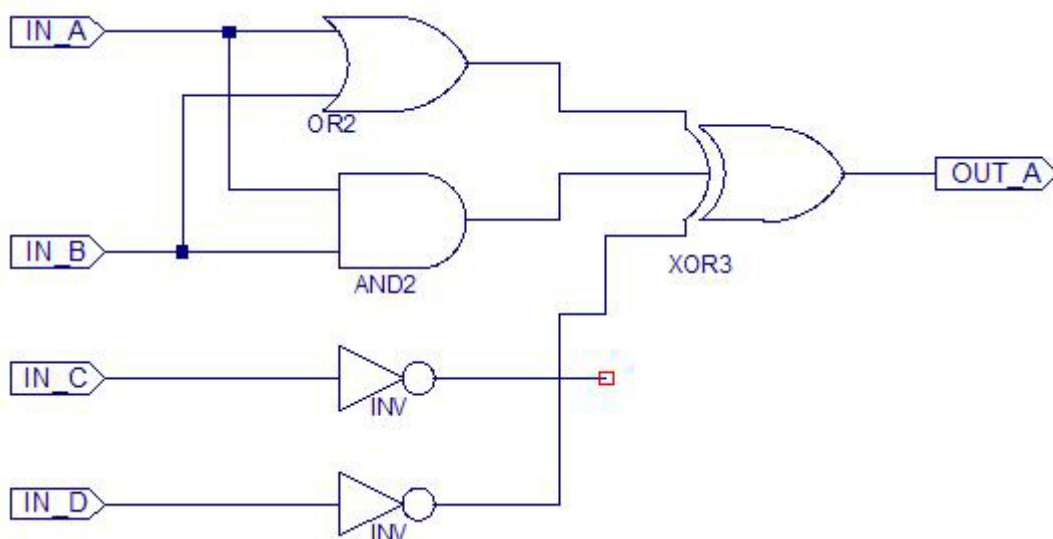
Nazwę automatyczną można zmienić, pamiętając jednak, że fragmenty połączeń o tej samej nazwie tworzą wspólną sieć nawet wówczas, gdy nie są połączone linią na schemacie:



Po usunięciu fragmentu połączenia między wyjściem negacji i wejściem bramki XOR pozostają one dalej połączone (fragmenty sieci o wspólnej nazwie *OUT_INV*):



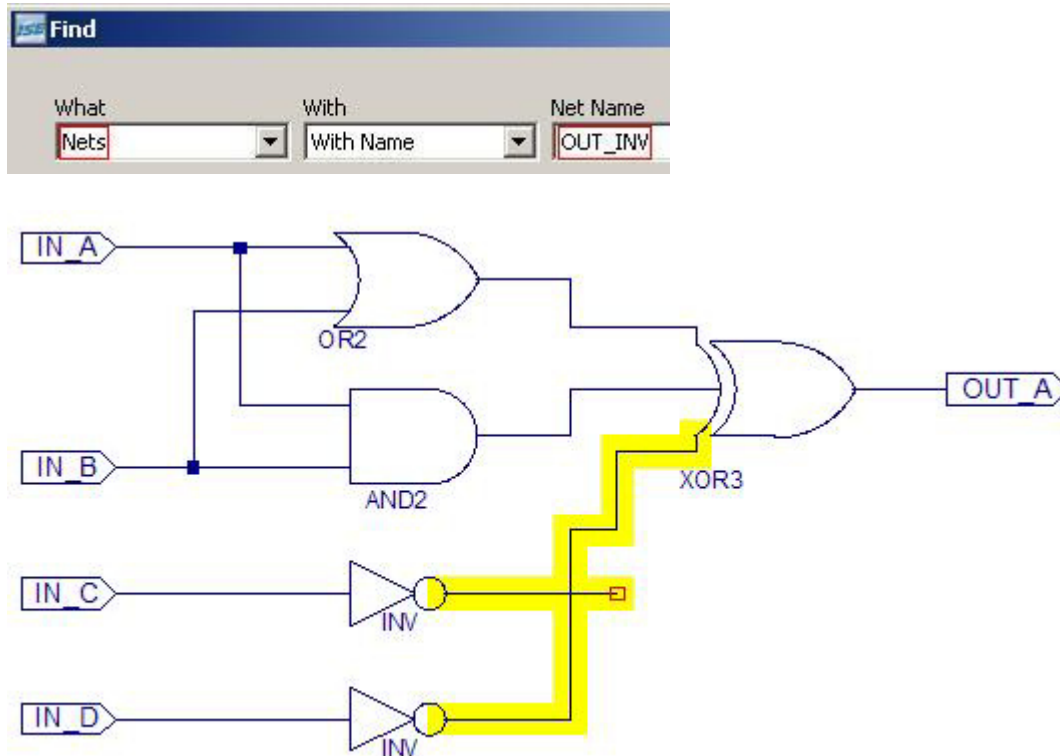
Zmodyfikujmy układ wprowadzając dodatkową negację i łącząc jej wyjście z wejściem bramki XOR:



Przy sprawdzaniu poprawności połączeń tego układu (opcja: *Check Design Rules*) pojawi się następujące ostrzeżenie:

```
WARNING:DesignEntry - Net "OUT_INV" is connected to too many source pins and/or I/O markers.
```

Oznacza ono, że sieć *OUT_INV* jest sterowana przez więcej niż jedno źródło. Tego typu problem może być trudny do zdiagnozowania, ponieważ zwarcia nie widać na schemacie. W tym przypadku można wykorzystać opcję wyszukiwania elementu lub sieci połączeń na schemacie (*Edit* → *Find*) wybierając poszukiwanie sieci (*Nets*) o nazwie *OUT_INV*:



Znaleziona i wyróżniona na schemacie sieć pozwala zdiagnozować przyczynę problemu (wystarczy przyjrzeć się dokładnie nazwom poszczególnych fragmentów sieci). Zwarcie można wyeliminować usuwając fragmenty sieci i dodając nowe, poprawne połączenia.

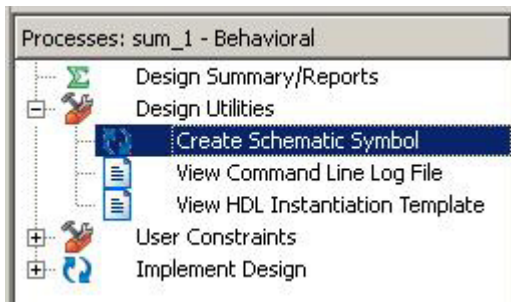
Schematy wielopoziomowe

Bardziej skomplikowane projekty (zwłaszcza w przypadku, gdy bloki funkcjonalne powtarzają się wielokrotnie) powinny być zorganizowane w postaci hierarchicznej. Dla bloków zdefiniowanych na schemacie lub opisanych w języku *VHDL* można utworzyć symbole, które będą następnie używane na schematach wyższego poziomu w identyczny sposób, jak elementy biblioteczne.

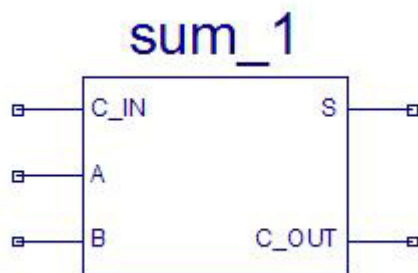
Typowym przykładem takiego projektu jest sumator wielobitowy. Załóżmy, że na schemacie (lub w module *VHDL*) został zaprojektowany sumator 1-bitowy z następującymi portami:

- wejścia: *A*, *B* (sumowane bity), *C_IN* (przeniesienie wejściowe)
- wyjścia: *S* (suma) i *C_OUT* (przeniesienie wyjściowe)

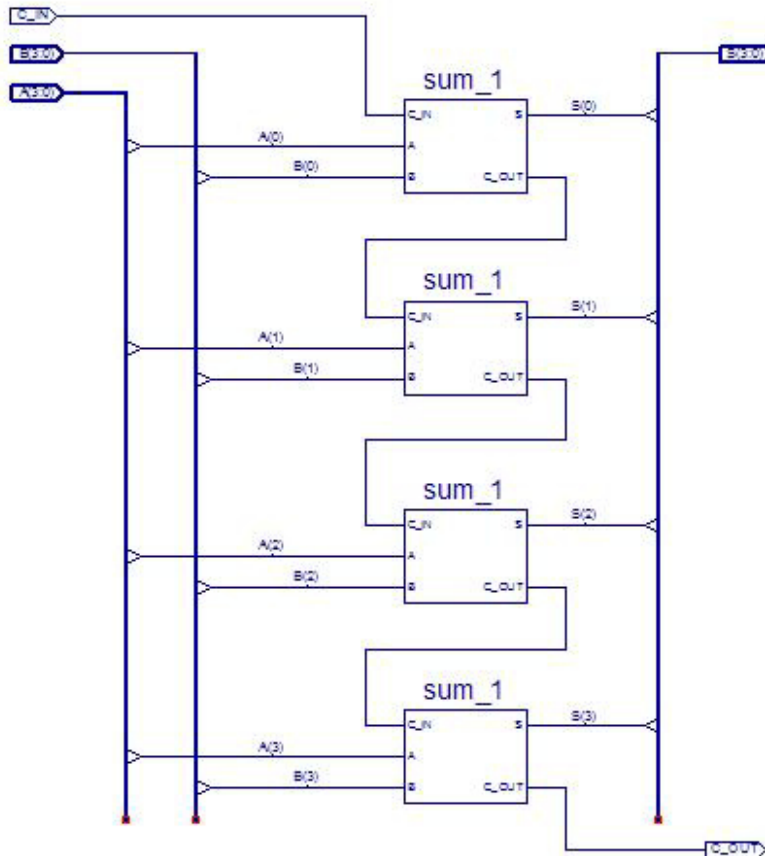
Symbol sumatora można utworzyć używając opcji *Create Schematic Symbol*:



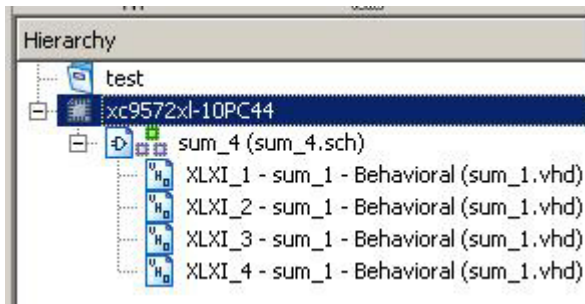
Utworzony symbol ma nazwę taką samą, jak plik zawierający definicję modułu (schemat lub *VHDL*) i jest dostępny tak samo, jak elementy biblioteczne (*Symbols*), w grupie o nazwie zgodnej ze ścieżką katalogu roboczego projektu:



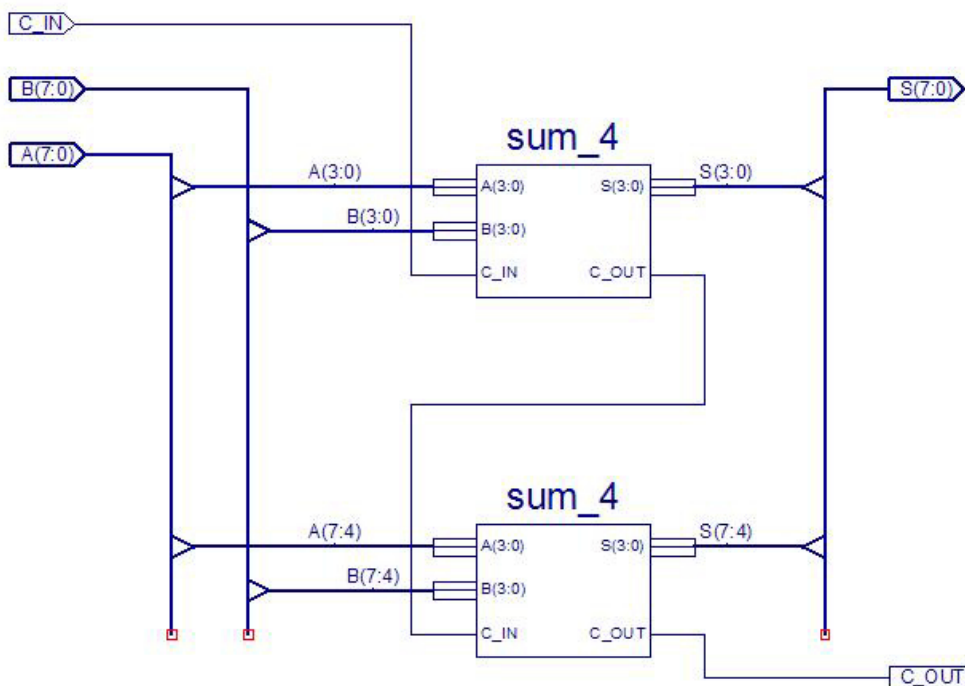
Po utworzeniu nowego schematu (*sum_4.sch*) można zdefiniować sumator 4-bitowy używając czterech symboli sumatora 1-bitowego (patrz opis wykorzystania magistrali).



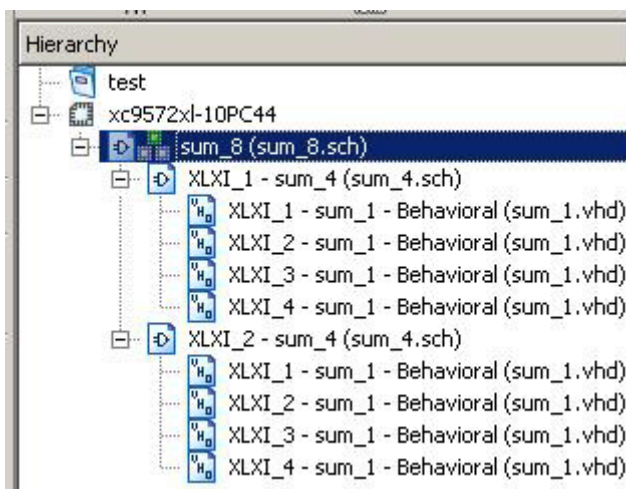
W oknie pokazującym strukturę projektu widać hierarchię zależności między modułami:



Dla sumatora 4-bitowego można również utworzyć symbol, który z kolei może być wykorzystany do zbudowania sumatora 8-bitowego (oczywiście sumator 8-bitowy mógłby być zbudowany również bezpośrednio z ośmiu sumatorów 1-bitowych):



Hierarchia zależności modułów dla sumatora 8-bitowego:



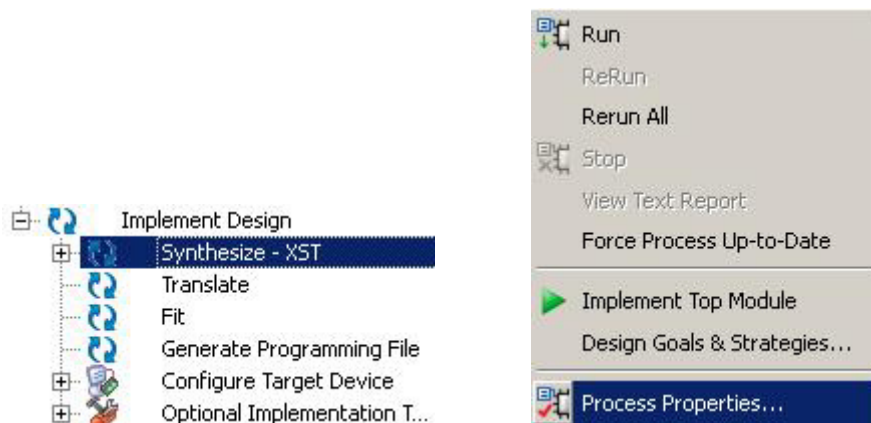
Moduły wielokrotnego użytku

Moduł zdefiniowany na schemacie lub w pliku *VHDL*, zweryfikowany w danym projekcie może być wykorzystany w kolejnych projektach. Wystarczy skopiować plik źródłowy (.sch lub .vhd), ewentualnie dodatkowe pliki z opisem symulacji, do katalogu roboczego bieżącego projektu. Można dodatkowo skopiować również plik .sym (definicja symbolu schematowego modułu), chociaż nie jest to konieczne, bo dysponując plikiem źródłowym można taki plik utworzyć (opcja: *Create Schematic Symbol*).

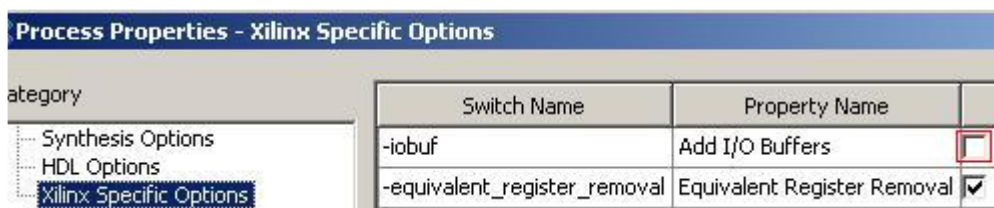
Można jednak również przygotować dla modułu plik .ngc (zawiera on wynik syntezy modułu w formacie *netlist* specyficznym dla narzędzia syntezy *XST* firmy *Xilinx*). Plik ten w połączeniu z plikiem .sym daje możliwość używania modułu, ale bez możliwości jego edycji, (co może być zaletą, bo wyklucza możliwość niezamierzonych zmian w module). Podejście to jest analogiczne jak przy używaniu gotowych modułów udostępnionych na stronie *www* przedmiotu.

Utworzenie pliku .ngc jest bardzo proste, wystarczy wykonać syntezę modułu. Istotne jest jednak w tym przypadku specjalne ustawienie opcji syntezy związanej z buforami wejściowymi i wyjściowymi.

Po wybraniu właściwości procesu syntezy:



w grupie *Xilinx Specific Options* odznaczamy pokazane poniżej pole:



Utworzony w ten sposób plik .ngc wraz z plikiem .sym stanowią kompletny opis modułu. Aby wykorzystać moduł w bieżącym projekcie należy skopiować oba pliki do katalogu roboczego, pliku .ngc nie dodajemy do projektu. Jeśli symbol modułu zostanie użyty na schemacie, to plik .ngc zostanie automatycznie wyszukany i użyty w czasie syntezy.