

KICAD

EESCHEMA



LINUX & WINDOWS

Autor:

Jean-Pierre Charras

Wersja:

Wrzesień 2011

Spis treści

1. Wprowadzenie	strona 5
1.1. Przeznaczenie	
1.2. Opis techniczny	
2. Podstawowe polecenia	strona 6
2.1. Dostęp do poleceń	
2.2. Polecenia związane z myszą	
2.2.1. Podstawowe polecenia	
2.2.2. Operacje na blokach	
2.3. Klawisze skrótów	
2.4. Wybór rozmiaru siatki	
2.5. Wybór powiększenia - Zoom	
2.6. Wyświetlanie pozycji kursora	
2.7. Pasek menu	
2.8. Górny pasek narzędzi	
2.9. Prawy pasek narzędzi	
2.10. Lewy pasek narzędzi	
2.11. Menu kontekstowe i szybka edycja komponentów	
3. Menu główne	strona 15
3.1. Menu Plik	
3.2. Menu Ustawienia	
3.2.1. Menu podrzędne: Skróty klawiszowe	
3.2.2. Menu Ustawienia: Biblioteka	
3.2.3. Menu Ustawienia: Kolory	
3.2.4. Menu Ustawienia: Opcje	
3.2.5. Menu Ustawienia: Język	
3.3. Menu Pomoc	
4. Główny pasek narzędziowy	strona 19
4.1. Ustawienia strony	
4.2. Znajdź	
4.3. Lista sieci	
4.4. Numeracja komponentów	
4.5. Kontrola reguł projektowych - ERC	
4.5.1. Zakładka ERC	
4.5.2. Zakładka Opcje	
4.6. Lista materiałowa - BOM	
4.7. Narzędzie importu dla numeracji wstecznej	
5. Tworzenie i edycja schematu	strona 26
5.1. Definicje	
5.2. Uwagi ogólne	
5.3. Proces tworzenia	
5.4. Wstawianie i edycja komponentów	
5.4.1. Wyszukiwanie i wstawianie komponentów	
5.4.2. Porty zasilania	
5.4.3. Edycja / modyfikacja elementów (umieszczonych na schemacie)	
5.4.3.1. Modyfikacja elementów	
5.4.3.2. Modyfikacja pól tekstowych elementów	
5.5. Połączenia, Magistrale, Etykiety i Symbole zasilania	
5.5.1. Podstawy	
5.5.2. Połączenia (łącza i etykiety)	
5.5.3. Magistrale	
5.5.3.1. Składniki magistral	
5.5.3.2. Połączenia pomiędzy składnikami magistral	
5.5.3.3. Globalne połączenia pomiędzy magistralami	
5.5.4. Połączenia z symbolami zasilania	
5.5.5. Symbol „Nie połączone”	
5.6. Elementy uzupełniające	
5.6.1. Komentarze	

5.6.2. Tabelka

6. Schematy o strukturze hierarchicznej

strona 35

- [6.1. Wprowadzenie](#)
- [6.2. Nawigacja wewnątrz hierarchii](#)
- [6.3. Etykiety lokalne, hierarchiczne i globalne](#)
- [6.4. Hierarchie proste](#)
 - [6.4.1. Wprowadzenie](#)
 - [6.4.2. Arkusze podrzędne](#)
 - [6.4.3. Piny hierarchiczne](#)
 - [6.4.3.1. Wstawianie ręczne pinów hierarchicznych](#)
 - [6.4.3.2. Wstawianie półautomatyczne pinów hierarchicznych](#)
 - [6.4.4. Etykiety hierarchiczne](#)
 - [6.4.5. Etykiety, etykiety hierarchiczne, etykiety globalne oraz piny ukryte](#)
 - [6.4.5.1. Etykiety](#)
 - [6.4.5.2. Etykiety hierarchiczne](#)
 - [6.4.5.3. Ukryte piny](#)
 - [6.4.5.4. Etykiety globalne](#)
- [6.5. Hierarchia złożona](#)
- [6.6. Hierarchia płaska](#)

7. Automatyczna numeracja elementów schematu

strona 42

- [7.1. Przeznaczenie](#)
- [7.2. Dostępne opcje](#)
- [7.3. Przykłady](#)
 - [7.3.1. Zmiany porządku numeracji](#)
 - [7.3.2. Wybór numeracji](#)

8. Kontrola reguł projektowych - ERC

strona 45

- [8.1. Wprowadzenie](#)
- [8.2. Sposób użycia](#)
- [8.3. Przykładowy rezultat działania ERC](#)
- [8.4. Wyświetlanie informacji o błędzie](#)
- [8.5. Porty zasilania a flagi zasilania](#)
- [8.6. Konfiguracja](#)
- [8.7. Pliki raportów generowane przez ERC](#)

9. Generowanie list sieci

strona 48

- [9.1. Przegląd zagadnień](#)
- [9.2. Format listy sieci](#)
- [9.3. Przykłady](#)
- [9.4. Uwagi](#)
 - [9.4.1. Zalecane środki ostrożności](#)
 - [9.4.2. Listy sieci PSPICE](#)
- [9.5. Inne formaty, użycie „wtyczek”](#)
 - [9.5.1. Podstawy](#)
 - [9.5.2. Inicjalizacja okna dialogowego wtyczki](#)
 - [9.5.3. Format linii poleceń](#)
 - [9.5.4. Konwerter i arkusze stylów - Program „xslproc”](#)
 - [9.5.5. Format pliku pośredniego listy sieci](#)

10. Drukowanie i rysowanie schematów na drukarkach lub ploterach

strona 55

- [10.1. Wprowadzenie](#)
- [10.2. Polecenia wspólne](#)
- [10.3. Rysuj / Rysuj w formacie HPGL](#)
 - [10.3.1. Wybór rozmiaru arkusza](#)
 - [10.3.2. Ustawienie przesunięcia strony](#)
- [10.4. Rysuj / Rysuj w formacie PostScript](#)
- [10.5. Rysuj / Rysuj w formacie SVG](#)
- [10.6. Rysuj / Rysuj w formacie DXF](#)
- [10.7. Drukuj](#)

11. Edytor bibliotek LibEdit - Podstawy

strona 58

- [11.1. Podstawowe informacje na temat bibliotek](#)
 - [11.1.1. Czym jest biblioteka symboli ?](#)
 - [11.1.2. Polecenia związane z bibliotekami](#)
- [11.2. Elementy składowe symboli](#)
- [11.3. Odczyt symbolu w celu edycji](#)

- [11.3.1. Główny pasek menu](#)
- [11.3.2. Wybór biblioteki](#)
- [11.3.3. Wybór symbolu i sposoby jego zapisu](#)
 - [11.3.3.1. Wybór symbolu](#)
 - [11.3.3.2. Zapis symbolu](#)
 - [11.3.3.3. Przenoszenie symboli do innych bibliotek](#)
 - [11.3.3.4. Zaniechanie edycji symbolu](#)
- [11.4. Tworzenie symboli](#)
 - [11.4.1. Tworzenie nowego symbolu](#)
 - [11.4.2. Tworzenie nowego symbolu na podstawie innego](#)
 - [11.4.3. Edycja głównych właściwości symboli](#)
 - [11.4.4. Edycja symboli wieloczęściowych](#)
- [11.5. Tworzenie postaci graficznej symbolu](#)
 - [11.5.1. Elementy graficzne symbolu](#)
 - [11.5.2. Elementy geometryczne w symbolach](#)
 - [11.5.3. Tekst jako grafika w symbolu](#)
- [11.6. Ustalanie punktu zaczepienia symbolu](#)
- [11.7. Tworzenie i edycja wyprowadzeń \(pinów\)](#)
 - [11.7.1. Wyprowadzenia - Informacje podstawowe](#)
 - [11.7.2. Symbole wieloczęściowe, podwójna reprezentacja symboli](#)
 - [11.7.3. Wyprowadzenia - Opcje podstawowe](#)
 - [11.7.4. Wyprowadzenia - Określanie właściwości](#)
 - [11.7.4.1. Kształty wyprowadzeń](#)
 - [11.7.4.2. Typy elektryczne](#)
 - [11.7.5. Wyprowadzenia - Zmiany globalne](#)
 - [11.7.6. Wyprowadzenia - Symbole wieloczęściowe i podwójna reprezentacja](#)
- [11.8. Edycja pól](#)
- [11.9. Tworzenie symboli zasilania](#)

12. Edytor bibliotek LibEdit - Dodatkowe możliwości

strona 76

- [12.1. Wprowadzenie](#)
- [12.2. Aliasy](#)
- [12.3. Pola specjalne](#)
- [12.4. Dokumentowanie symboli](#)
 - [12.4.1. Słowa kluczowe](#)
 - [12.4.2. Dokumentacja symbolu](#)
 - [12.4.3. Filtrowanie modułów dla CvPcb](#)
- [12.5. Biblioteka wzorców](#)
 - [12.5.1. Eksport/Tworzenie wzorca](#)
 - [12.5.2. Importowanie wzorca](#)

13. Przeglądarka bibliotek ViewLib

strona 80

- [13.1. Przeznaczenie](#)
- [13.2. Ekran główny](#)
- [13.3. Pasek narzędzi przeglądarki bibliotek](#)

14. Dostosowywanie list sieci oraz zestawień materiałowych (BOM)

strona 82

- [14.1. Pośrednia lista sieci](#)
 - [14.1.1. Przykładowy schemat](#)
 - [14.1.2. Przykładowa pośrednia lista sieci](#)
- [14.2. Konwersja na nowy format](#)
- [14.3. Język skryptowy XSLT](#)
 - [14.3.1. Przykład tworzenia pliku z listą sieci PADS-PCB](#)
 - [14.3.2. Przykład tworzenia listy sieci programu Cadstar](#)
 - [14.3.2.1. Plik z arkuszem stylu](#)
 - [14.3.2.2. Plik wyjściowy](#)
 - [14.3.3. Przykład tworzenia listy sieci programu OrcadPCB2](#)
 - [14.3.3.1. Arkusz stylów](#)
 - [14.3.3.2. Plik wyjściowy](#)
 - [14.3.4. Używanie systemu wtyczek Eeschema](#)
 - [14.3.4.1. Inicjowanie okna dialogowego](#)
 - [14.3.4.2. Ustawienia nowych wtyczek](#)
 - [14.3.4.3. Linia poleceń](#)
 - [14.3.4.4. Format linii poleceń](#)
- [14.4. Plik pośredni listy sieci](#)
 - [14.4.1. Struktura ogólna](#)
 - [14.4.2. Sekcja nagłówka](#)
 - [14.4.3. Sekcja komponentów](#)
 - [14.4.3.1. Uwagi na temat odcisków czasowych dla komponentów](#)

[14.4.4. Sekcja elementów bibliotecznych](#)

[14.4.5. Sekcja bibliotek](#)

[14.4.6. Sekcja sieci](#)

[14.5. Więcej informacji na temat xsltproc](#)

[14.5.1. Wprowadzenie](#)

[14.5.2. Synopsis](#)

[14.5.3. Opcje linii poleceń](#)

[14.5.4. Zwracane wartości](#)

[14.5.5. Więcej informacji](#)

1. Wprowadzenie

1.1. Przeznaczenie

Eeschema to wydajne oprogramowanie przeznaczone do rysowania schematów elektronicznych, dostępne dla następujących systemów operacyjnych:

- ♦ LINUX
- ♦ WINDOWS XP/2000

Niezależnie jaki system został użyty, generowane przez program pliki są w pełni kompatybilne pomiędzy systemami.

Eeschema to oprogramowanie „zintegrowane” ponieważ wszystkie funkcje jakie oferuje: rysowanie, kontrola poprawności, układ, zarządzanie bibliotekami i dostęp do oprogramowania do tworzenia PCB są dostępne z poziomu **Eeschema**.

Pozwala on także na rysowanie w strukturach hierarchicznych, używając do tego wielu połączonych ze sobą arkuszy schematów.

Program jest przeznaczony do współpracy z oprogramowaniem do tworzenia obwodów drukowanych PCB jak np. **Pcbnew**, któremu dostarcza tzw. plików *Netlist*, które to opisują elektryczne połączenia na PCB na podstawie projektu schematu.

Eeschema posiada także zintegrowany edytor elementów pozwalający na tworzenie, edycję oraz wizualizację elementów, a również zapewnia obsługę bibliotek symboli (import, eksport, dodawanie i usuwanie elementów bibliotek).

Eeschema również zawiera następujące narzędzia dodatkowe ale stanowiące niezbędny element nowoczesnego oprogramowania do tworzenia schematów:

- ♦ Sprawdzanie poprawności projektu (*DRC*) dla zautomatyzowanego procesu kontroli poprawności połączeń, czy pozostawienia niepołączonych wejść elementów...
- ♦ Generowanie wydruków w formacie **POSTSCRIPT** lub **HPGL**.
- ♦ Generowanie wydruków na drukarce lokalnej.
- ♦ Tworzenie listy materiałowych *BOM*.
- ♦ Tworzenie list sieci połączeń (*netlisty*) dla programów tworzących PCB lub oprogramowania symulacyjnego.

1.2. Opis techniczny

To 32-bitowe oprogramowanie jest ograniczone tylko przez ilość dostępnej pamięci.

Nie posiada realnych ograniczeń co do ilości elementów, liczby wyprowadzeń, połączeń lub arkuszy...

Eeschema pozwala na budowanie prostych jak i wieloarkuszowych schematów.

W przypadku schematów wieloarkuszowych ich reprezentacja jest hierarchiczna, a dostęp do każdego z arkuszy jest niezależny.

Eeschema dla schematów wieloarkuszowych pozwala na budowanie :

- ♦ **Hierarchii prostych** (każdy schemat jest użyty tylko jeden raz),
- ♦ **Hierarchii złożonych** (niektóre ze schematów są użyte wielokrotnie),
- ♦ **Hierarchii płaskich** (niektóre schematy nie są wyraźnie połączone w głównym schemacie).

Maksymalny rozmiar rysunku jest zawsze regulowany od formatu **A4** do **A0** i od formatu **A** do **E**.

2. Podstawowe polecenia

2.1. Dostęp do poleceń

Możesz uruchomić wiele poleceń używając do tego celu:

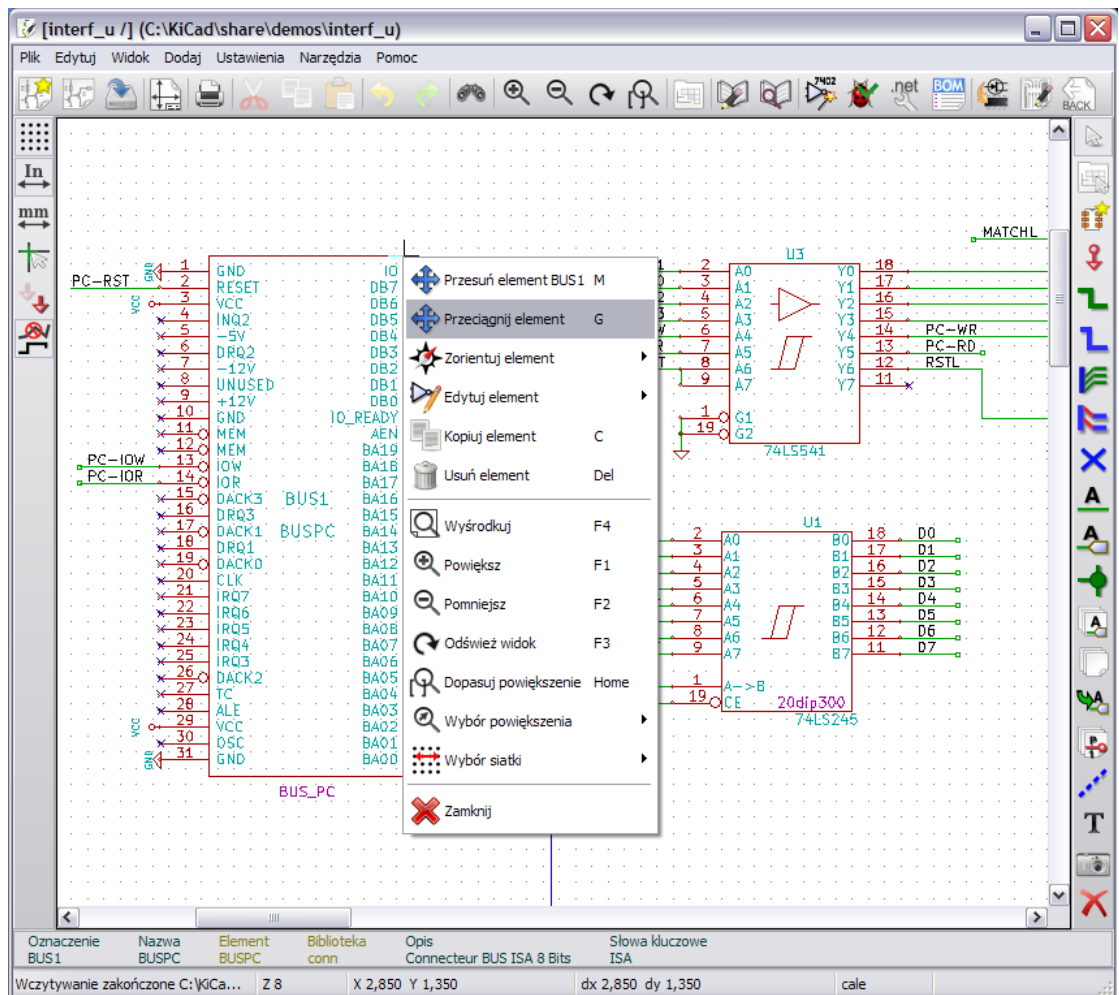
- ◆ Kliknięcia w pasek menu (na górze ekranu).
- ◆ Kliknięcia w ikony na górze ekranu (podstawowe polecenia).
- ◆ Kliknięcia w ikony na pasku z prawej strony ekranu (poszczególne polecenia lub „narzędzia”).
- ◆ Kliknięcia w ikony na pasku z lewej strony ekranu (opcje wyświetlania).
- ◆ Kliknięcia za pomocą klawisza myszy (najważniejsze polecenia); W praktyce kliknięcie prawym klawiszem otwiera menu kontekstowe, zależne od tego jaki element znajduje się na pozycji kursora (Powiększenie, Siatka oraz edycja elementów).
- ◆ Klawiszy funkcyjnych klawiatury (**F1**, **F2**, **F3**, **F4**, **Insert** oraz **spacja**).

Przykładowo:

Klawisz **Esc** pozwala na zaniechanie obecnie wykonywanego polecenia.

Klawisz **Insert** pozwala na powielenie ostatnio utworzonego elementu.

Poniżej pokazano różne możliwości dostępu do poleceń.



2.2. Polecenia związane z myszą

2.2.1. Podstawowe polecenia

Lewy klawisz:

- Pojedynczy klik: pokazuje dane charakterystyczne elementu lub tekstu znajdującego się pod kursorem.
- Podwójny klik: edycja (jeśli element można edytować) tego elementu lub tekstu.

Prawy klawisz:

- Otwarcie menu kontekstowego.

2.2.2. Operacje na blokach

Możesz przesuwać, przeciągać, kopiować oraz usuwać wybrane obszary w każdym z menu **Eeschema**.

Obszary (bloki) są wybierane za pomocą lewego klawisza myszy. Polecenie jest dokańczane z chwilą puszczenia klawisza.

Trzymając dodatkowo jeden z klawiszy **Shift**, **Ctrl**, lub oba naraz podczas zaznaczania w rezultacie wykonujesz kopiowanie, przeciąganie lub kasowanie wybranego obszaru.

Podsumowanie poleceń:

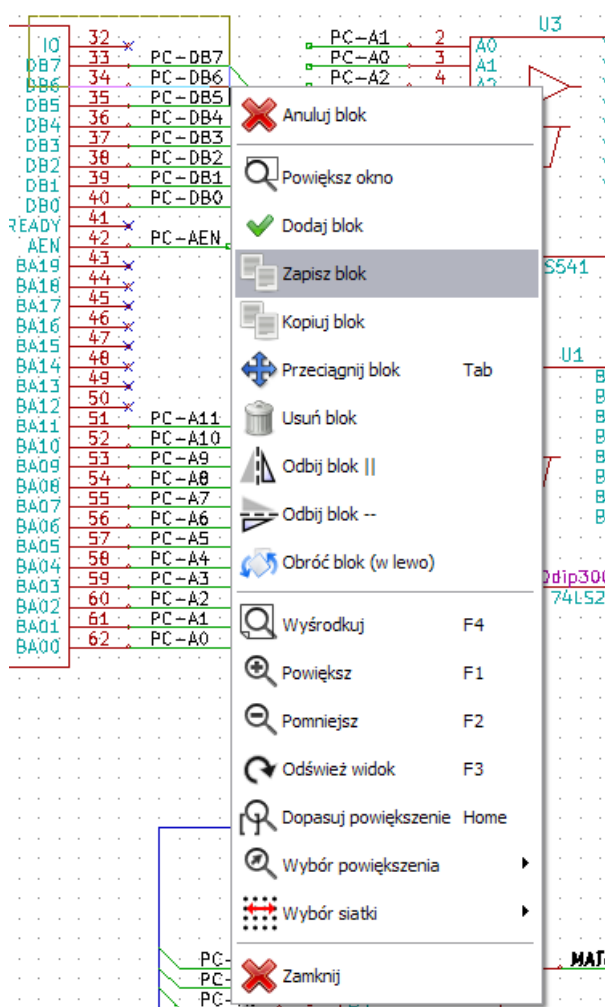
Lewy przycisk myszy	Przesunięcie zaznaczenia.
Shift + Lewy przycisk myszy	Skopiowanie zaznaczenia.
Ctrl + Lewy przycisk myszy	Przeciąganie zaznaczenia.
Control + Shift + Lewy przycisk myszy	Usunięcie zaznaczenia.

Polecenia są wykonywane po puszczeniu klawisza.

Podczas przesuwania zaznaczenia możesz:

- Kliknąć ponownie by przywrócić elementy na poprzednią pozycję.
- Kliknąć prawy klawisz by przerwać operację.

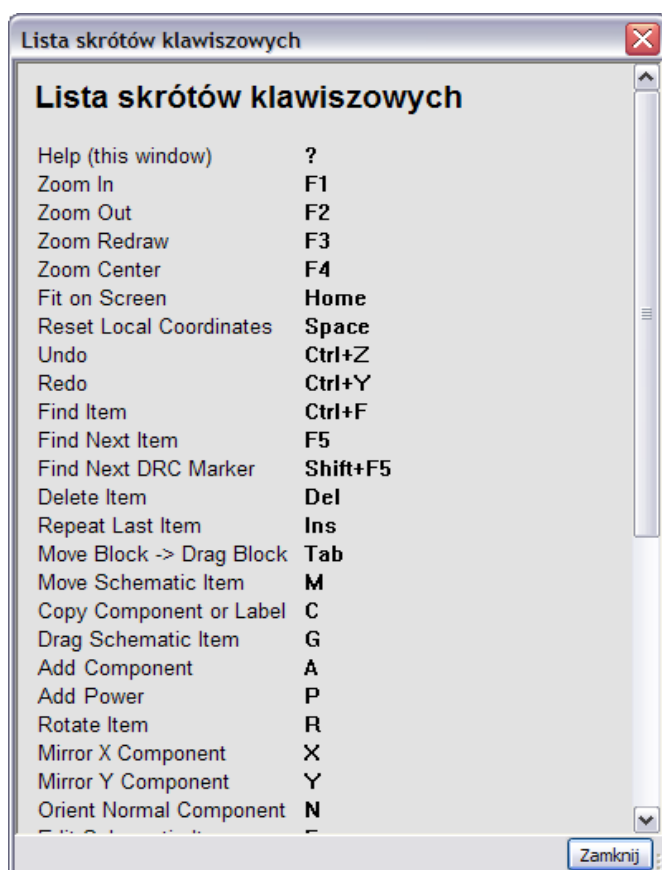
Jeśli operacja blokowa została rozpoczęta, można wybrać inną operację blokową z menu kontekstowego (myszą, prawym klawiszem):



2.3. Klawisze skrótów

Klawisze skrótów nie rozróżniają wielkości liter.

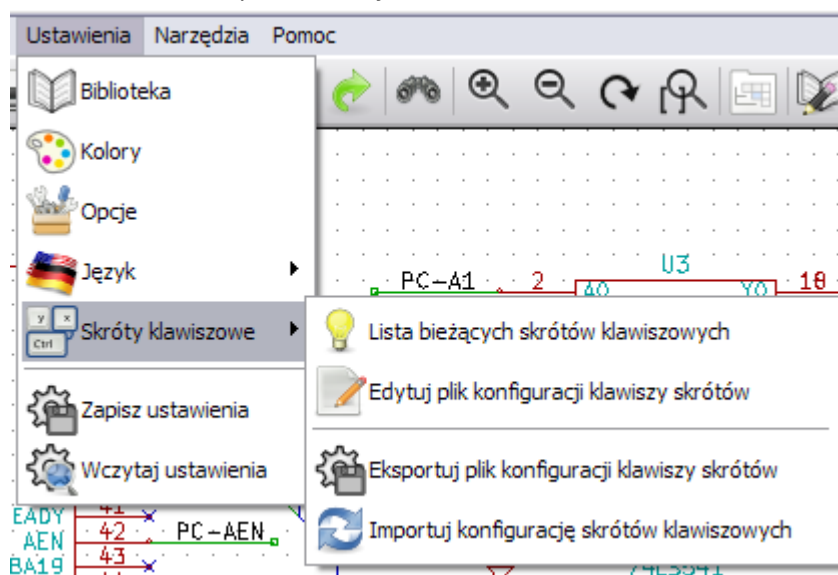
- Klawisz **?** Pokazuje obecną listę klawiszy skrótów.
- Menu Preferencje zarządza klawiszami skrótów.



Tutaj jest przykład

Klawisze skrótów mogą być programowane przez użytkowników
By to zrobić:

1. Stwórz lub odtwórz plik klawiszy skrótów:



2. Dokonaj modyfikacji pliku (jest skomentowany).
3. By użyć nowego układu klawiszy skrótów ponownie wczytaj plik konfiguracji klawiszy (lub zrestartuj **Eeschema**).

2.4. Wybór rozmiaru siatki

Kursor porusza się według siatki, która może być wyświetlana lub ukryta (siatka ta jest wyświetlana zawsze w edytorze elementów/bibliotek).

Możesz zmienić rozmiar siatki za pomocą menu kontekstowego lub w menu **Ustawienia / Opcje**.

Domyślnym rozmiarem siatki jest 50 milsów (0,050 cala) lub inaczej 1,27 milimetrów.

Można pracować z siatką przybliżoną (20 milsów) lub siatką dokładną (10 milsów).

Nie jest to jednak zalecane.

Siatka przybliżona lub dokładna jest specjalnie przeznaczona dla elementów z dużą ilością wyprowadzeń (liczoną w setkach).

2.5. Wybór powiększenia - Zoom

By zmienić powiększenie (Zoom):

- Kliknij prawym klawiszem by otworzyć menu kontekstowe i wybierz potrzebne powiększenie.
- Lub użyj klawiszy funkcyjnych:
 - **F1**: Przybliż.
 - **F2**: Oddal.
 - **F3**: Odśwież widok.
 - **F4**: Centruj wokół kursora
- Lub po prostu kliknij środkowym klawiszem (bez poruszania myszą)
 - **Powiększ zaznaczenie**: Przeciągnij myszą z wciśniętym środkowym klawiszem.
 - **Kółko myszy**: Przybliżanie/Oddalanie
 - **SHIFT+Kółko myszy**: Przesuwanie widoku góra/dół
 - **CTRL+Kółko myszy**: Przesuwanie widoku lewo/prawo

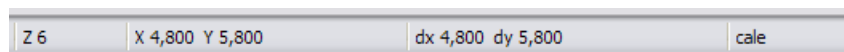
2.6. Wyświetlanie pozycji kursora

Jednostki w jakich pozycja jest wyświetlana to cale (cal lub ") albo milimetry.

Jednakże, **Eeschema** wewnątrz zawsze przelicza wszystko na tysięczne cali (1/1000").

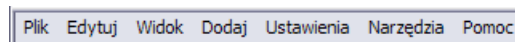
Informacje wyświetlane na dole okna od prawej strony są następujące:

- Poziom powiększenia.
- Pozycja absolutna kursora.
- Względna pozycja kursora.
- Punkt początkowy dla pozycji względnej (x, y) może być ustalony klawiszem spacji. Wyświetlana pozycja odnosi się do tego punktu.



2.7. Pasek menu

Menu pokazane poniżej pozwala na otwieranie i zapisywanie schematów, ustawianie konfiguracji programu, a także zawiera dostęp do pomocy.



2.8. Górny pasek narzędzi

Ten pasek umożliwia dostęp do głównych funkcji programu **Eeschema**.



Tworzenie nowego schematu.



Otwarcie schematu.



Zapis kompletnego schematu (razem z pełną hierarchią).



Wybór rozmiaru arkusza i edycja zawartości ramki.



Uruchomienie edytora symboli **LibEdit** (Sprawdzanie, modyfikacje oraz edycja symboli bibliotecznych).



Przeglądanie bibliotek (**ViewLib**).



Wywołanie okna „nawigatora”, pozwalającego na wyświetlanie struktury hierarchicznej schematów (jeśli posiadają schematy podrzędne) w postaci drzewa oraz błyskawiczne przełączanie się pomiędzy hierarchicznymi arkuszami.



Usuwanie wybranych elementów podczas przesuwania bloków.



Kopiowanie wybranych elementów do schowka podczas przesuwania bloków.



Kopiowanie ostatnio wybranego elementu lub bloku w bieżącym arkuszu.



Cofnij: Cofa ostatnią zmianę (do 10 wstecz).



Ponów: Przywraca ostatnią zmianą (do 10 wstecz).



Otwarcie menu wydruku.



Wywołanie **CvPcb**.



Wywołanie **Pcbnew**.



Przybliżanie i oddalanie wokół punktu centralnego ekranu.



Odświeżanie widoku i optymalizacja powiększenia.



Wywołanie menu wyszukiwania komponentów lub opisów na schemacie.



Tworzenie listy połączeń (różne formaty *Pcbnew*, *Spice* ...).



Numeracja komponentów.



ERC (*Electrical Rules Check*): automatyczna kontrola połączeń elektrycznych.



Tworzenie plików materiałowych **BOM** (*Bill Of Materials*) oraz/albo listę etykiet hierarchicznych.



Reimport pliku części z **CvPcb** (wypełnianie pól opisujących moduły w komponentach na schemacie).




















2.9. Prawy pasek narzędzi



Ten pasek umożliwia dostęp do **narzędzi** do:

- Stawiania komponentów, połączeń i magistral, węzłów, etykiet, opisów...
- Poruszania się po hierarchii schematów.
- Tworzenia hierarchicznych schematów podrzędnych oraz symboli połączeń.
- Usuwania komponentów.

Szczegóły dotyczące ich użycia zostały opisane w rozdziale **Tworzenie i edycja schematu**. Krótki opis ich użycia znajduje się poniżej.

	Zatrzymane polecenie lub narzędzie w trakcie.
	Nawigacja w strukturze hierarchicznej: to narzędzie pozwala na otwarcie arkuszy podrzędnych dla wyświetlanego schematu (kliknij w symbol arkusza podrzędnego), albo cofnięcie się wyżej w hierarchii (kliknij na pustym polu arkusza podrzędnego)
	Wywołuje menu stawiania komponentów.
	Wywołuje menu stawiania portów zasilania (specjalne elementy).
	Stawianie połączeń.
	Stawianie magistral.
	Stawianie wejść połączeń do magistrali. Elementy te spełniają rolę dekoracyjną i nie pozwalają na połączenia; dlatego nie powinny być używane do łączenia połączeń.
	Stawianie wejść magistral do magistral. Tylko one mogą połączyć dwie magistrale ze sobą.
	Symbol „Nie połączone”. Powinny być umieszczone na wyprowadzeniach pinów, które nie są do niczego podłączone. Jest to przydatne w funkcji ERC, aby zaznaczyć, że wyprowadzenia są celowo niepołączone i nie zostały pominięte.
	Stawianie etykiet lokalnych. Dwa połączenia mogą być złączone ze sobą jeśli oba posiadają te same etykiety w tym samym schemacie . Do łączenia pomiędzy dwoma odrębnymi arkuszami, powinieneś użyć etykiet globalnych.
	Stawiania etykiet globalnych. Stwarza to możliwość łączenia połączeń pomiędzy arkuszem podrzędnym a nadrzędnym który zawiera symbole arkuszy podrzędnych.
	Stawianie węzłów. By połączyć dwa przecinające się połączenia, lub dołączyć połączenie do wyprowadzenia, gdy jest to niejednoznaczne (np. gdy wyprowadzenie posiada już inne połączenie w innym kierunku).
	Stawianie etykiet hierarchicznych. Pozwala to na stworzenie połączenia pomiędzy arkuszem schematu i schematem nadrzędnym który zawiera symbol tego schematu.
	Stawianie symboli hierarchicznych schematów podrzędnych (prostokąt o zmiennym rozmiarze). Musisz określić nazwę pliku by zapisać dane tego schematu podrzędnego.
	Globalna etykieta importowa etykiety z arkusza podrzędnego, by utworzyć połączenie z symbolem na arkuszu podrzędnym. Dla tego symbolu hierarchicznego, stworzone punkty połączeń są odpowiednikami tradycyjnych wyprowadzeń elementów, i muszą zostać podłączone.
	Stawianie globalnych etykiet w arkuszach podrzędnych do stworzenia punktów połączeń. Funkcja ta jest podobna do poprzedniej, która nie wymaga zdefiniowanych symboli globalnych.
	Linie graficzne dla ramek... Tylko jako dekoracja. Nie tworzą one połączeń.
	Stawianie opisów. Tylko do dekoracji.
	Wstawianie map bitowych. Tylko do dekoracji.



Usuwanie wybranych elementów. Jeśli kilka elementów nakłada się na siebie w wybranym miejscu, priorytet mają wtedy najmniejsze elementy (priorytety ustalone są według: węzeł, symbol *Nie połączone*, połączenie, magistrala, tekst, element). Dotyczy to także arkuszy hierarchicznych. **Uwaga:** Funkcja **Cofnij** w głównym pasku narzędzi pozawala na wycofanie ostatnich usunięć.

2.10. Lewy pasek narzędzi



Ten pasek narzędzi zarządza opcjami wyświetlania:

- Siatki,
- Jednostek miary,
- Typu kursora,
- Widoczności ukrytych wyprowadzeń zasilania.

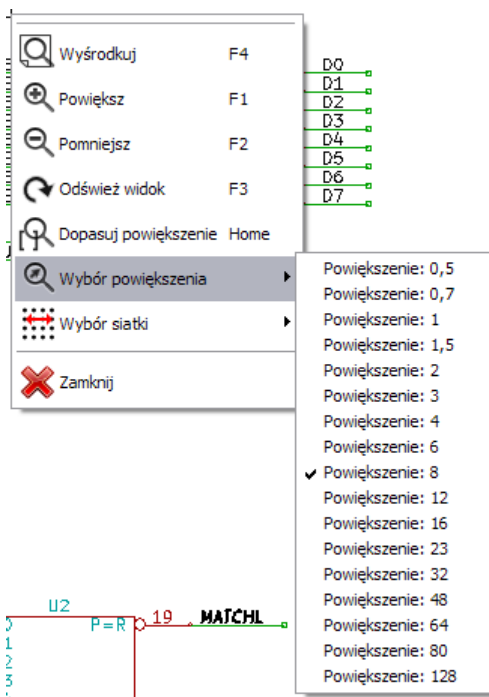
Oraz ustala dozwolone kierunki połączeń i magistral.

2.11. Menu kontekstowe i szybka edycja komponentów

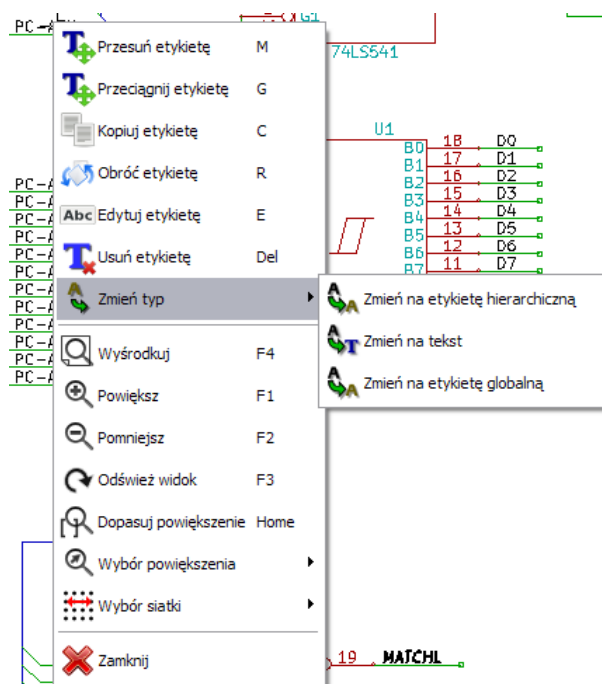
Kliknięcie prawym klawiszem otwiera menu kontekstowe, którego zawartość zależy od wybranego tym kliknięciem elementu (o został wybrany).

Masz możliwość szybkiego dostępu do:

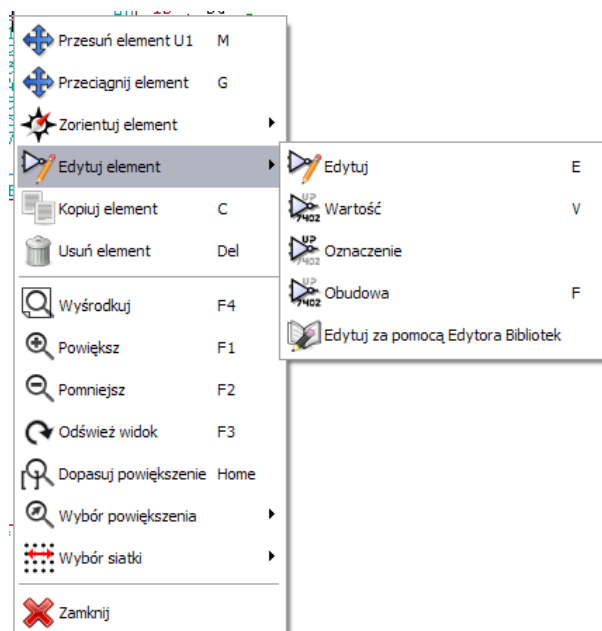
- Stopnia powiększenia,
- Ustawień siatki,
- W zależności od miejsca kliknięcia, do edycji najczęściej zmienianych parametrów.



Menu kontekstowe bez wybranego elementu.



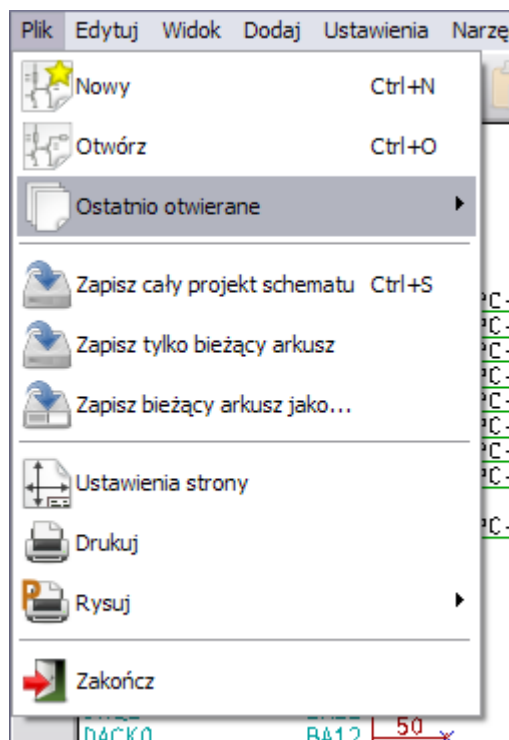
Edycja etykiety.



Edycja komponentów.

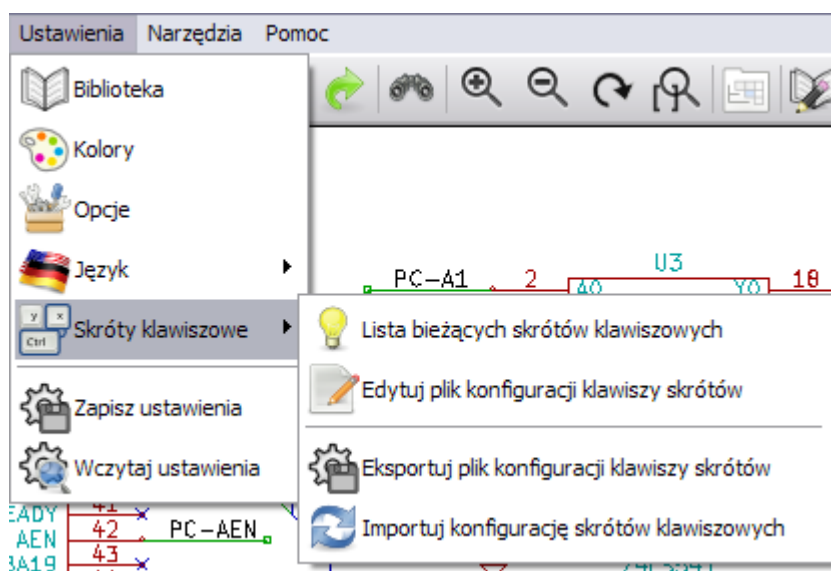
3. Menu główne

3.1. Menu Plik



<u>Nowy</u>	Czyści bieżący schemat i przygotowuje nowy.
<u>Otwórz</u>	Otwiera schemat lub hierarchię schematów.
<u>Otwórz ostatnio użyte</u>	Otwiera listę ostatnio otwartych plików, pozwalając wybrać plik w celi jego otwarcia.
<u>Zapisz cały projekt schematu</u>	Zapisuje bieżący schemat i całą hierarchię.
<u>Zapisz tylko bieżący schemat</u>	Zapisuje bieżący schemat, ale bez pozostałych w hierarchii.
<u>Zapisz bieżący arkusz jako...</u>	Zapisuje bieżący arkusz pod inną nazwą.
<u>Drukuj</u>	Dostęp do okna dialogowego z opcjami wydruku (Zobacz także rozdział Drukowanie i rysowanie).
<u>Rysuj</u>	Rysuje schemat w formacie Postscript, HPGL lub SVF (Zobacz rozdział Drukowanie i rysowanie).
<u>Wyjście</u>	Kończy pracę z programem.

3.2. Menu Ustawienia

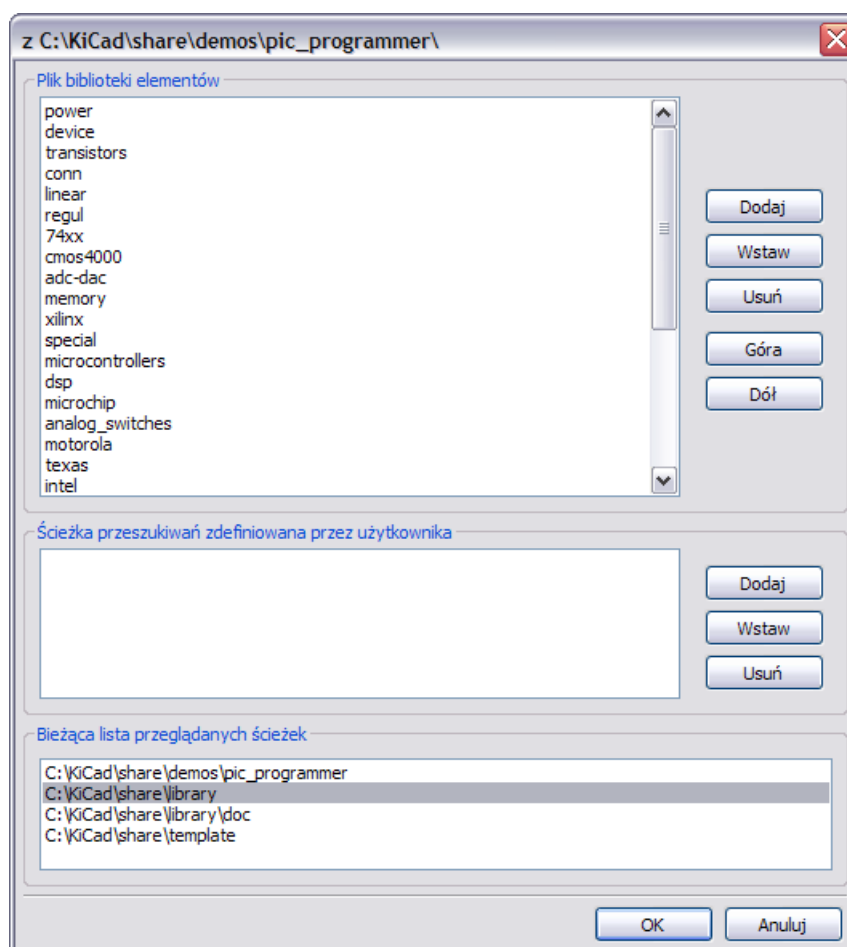


<u>Biblioteka</u>	Wybiera używane biblioteki oraz określa ścieżki do bibliotek.
<u>Kolory</u>	Otwiera okno dialogowe z wyborem kolorów.
<u>Opcje</u>	Otwiera okno dialogowe z opcjami programu (Jednostki miary, siatka...).
<u>Język</u>	Udostępnia bieżącą listę tłumaczeń interfejsu programu, pozwalając wybrać język komunikacji z użytkownikiem.
<u>Skróty klawiszowe</u>	Udostępnia menu z klawiszami skrótów.
<u>Zapisz ustawienia</u> <u>Wczytaj ustawienia</u>	Odczytuje i zapisuje pliki konfiguracji.

3.2.1. Menu podrzędne: Skróty klawiszowe

<u>Lista bieżących skrótów klawiszowych</u>	Pokazuje bieżącą listę skrótów klawiszowych. To samo powoduje naciśnięcie klawisza ?
<u>Edytuj plik konfiguracji skrótów klawiszowych</u>	Otwiera okno konfiguracji skrótów klawiszowych, który można zmodyfikować (zwykłym edytorem tekstu), by dostosować listę skrótów do własnych upodobań.
<u>Eksportuj plik konfiguracji klawiszy skrótów</u>	Eksportuje do pliku bieżące ustawienia skrótów klawiszowych.
<u>Importuj konfigurację skrótów klawiszowych</u>	Importuje z pliku ustawienia skrótów klawiszowych.

3.2.2. Menu Ustawienia: Biblioteka



Konfiguracja programu **Eeschema** to w zasadzie tylko:

- Ścieżki do bibliotek.
- Lista używanych bibliotek.

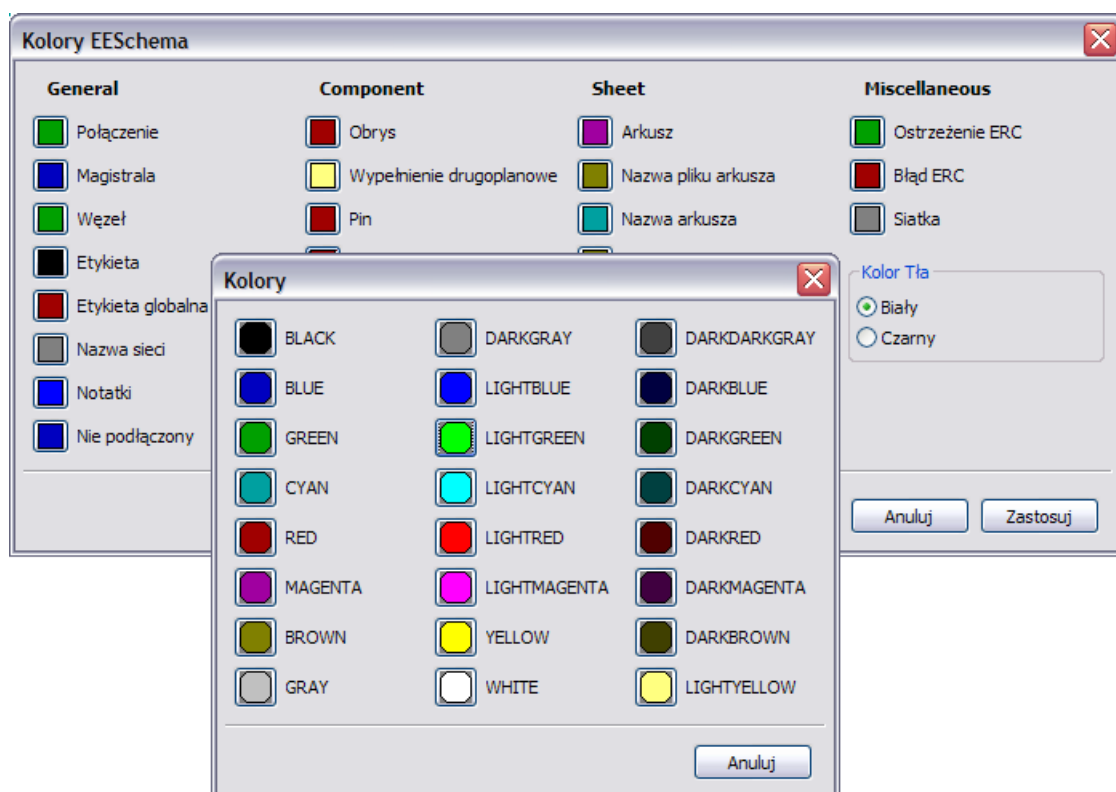
Parametry konfiguracyjne są zapisywane do pliku .pro.

Są możliwe różne konfiguracje dla różnych katalogów.

Eeschema wyszukuje i używa konfiguracji według określonej reguły:

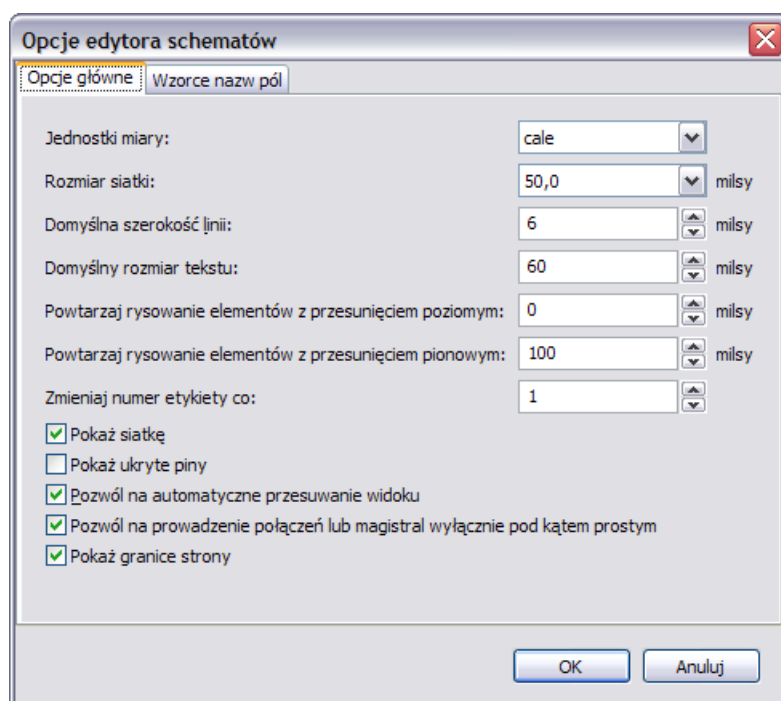
1. Najpierw plik konfiguracji (<nazwa_projektu>.pro) w bieżącym katalogu.
2. Potem plik konfiguracji kicad.pro w katalogu kicad. Ten plik można traktować jako domyślną konfigurację.
3. Jak żaden z plików nie został znaleziony to przywracane są wartości domyślne. Będzie wówczas wymagane przynajmniej wypełnienie listy bibliotek do załadowania oraz zapisanie nowej konfiguracji.

3.2.3. Menu Ustawienia: Kolory



Za pomocą tego okna dialogowego, można wybrać kolorystykę wyświetlania poszczególnych elementów oraz koloru tła z dwóch dostępnych: biały (domyślny) i czarny.

3.2.4. Menu Ustawienia: Opcje



<i>Jednostki miary:</i>	Wybiera jednostki miary (cale lub milimetry) jakie będą używane w całym programie.
<i>Rozmiar siatki:</i>	Wybiera rozmiar siatki. Podczas normalnej pracy rozmiarem siatki dla schematu musi być 0,050 cala lub 1,27 mm. Mniejsza siatka jest używana podczas tworzenia elementów.

<i>Domyślna szerokość linii:</i>	Rozmiar ten jest używany do rysowania elementów, które nie posiadają określonej szerokości linii.
<i>Domyślny rozmiar tekstu:</i>	Wartość ta jest używana przy tworzeniu nowych opisów lub etykiet.
<i>Powtarzaj rysowanie elementów z przesunięciem poziomym:</i>	Ustala przesunięcie w osi X podczas powielania elementów (zwykle 0 – nie przesuwaj). (Po wstawieniu elementu, którym może być symbol na schemacie, etykieta lub połączenie; można powielić element naciskając klawisz Insert)
<i>Powtarzaj rysowanie elementów z przesunięciem pionowym:</i>	Ustala przesunięcie w osi Y podczas powielania elementów. (zwykle przesuwaj o 0,100 cala lub 2,54 mm)
<i>Zmieniaj numer etykiety co:</i>	Zwiększanie lub zmniejszanie numeracji podczas powielania elementów (zwykle 1 lub -1).
<i>Pokaż siatkę</i>	Jeśli zaznaczone: pokazuje siatkę na arkuszu.
<i>Pokaż ukryte piny</i>	Pokazuje niewidoczne (lub ukryte) piny. Jeśli zaznaczone, pozwala wyświetlać piny zasilania.
<i>Pozwól na automatyczne przesuwanie widoku</i>	Jeśli zaznaczone, ekran będzie się automatycznie przesuwał gdy kursor będzie próbował wyjść poza okno podczas rysowania połączeń lub przesuwania elementów.
<i>Pozwól na prowadzenie połączeń lub magistral wyłącznie pod kątem prostym</i>	Jeśli zaznaczone, magistrale i połączenia można prowadzić tylko w pionie lub poziomie. W przeciwnym wypadku magistrale i połączenia można prowadzić w dowolnym kierunku.
<i>Pokaż granice strony</i>	Jeśli zaznaczone, pokazuje granice strony na ekranie.

3.2.5. Menu Ustawienia: Język

Można użyć trybu domyślnego. Wtedy program będzie się komunikował w języku zgodnym z językiem systemu – o ile taki język będzie dostępny w programie.

Można też wybrać inny język. **Eeschema** zostanie wtedy zrestartowany.

3.3. Menu Pomoc

Udostępnia pomoc *on-line* (ten dokument) a także dostarcza informacji o bieżącej wersji programu **Eeschema** (**O programie**).

Można również z poziomu tego menu otworzyć krótki samouczek, który w kilkunastu krokach wyjaśnia proces tworzenia kompletnego projektu.

4. Główny pasek narzędziowy

4.1. Ustawienia strony



Umożliwia dostęp do ustawień strony:

Pozwala na edycję tabliczki oraz zmianę rozmiaru strony.

Data jest uaktualniana automatycznie.

Tak samo automatycznie uaktualniana jest całkowita liczba arkuszy jak i numer arkusza.

4.2. Znajdź



Pozwala na wywołanie następującego okna narzędziowego.

Za jego pomocą można wyszukać na schemacie jeden z umieszczonych tam komponentów, jego wartość lub dowolny tekst jaki znajduje się na bieżącym schemacie czy w całej jego hierarchii.

Po odnalezieniu jednego z w/w elementów kursor przeskakuje do miejsca jego umieszczenia o ile ostatnia opcja nie została włączona.

4.3. Lista sieci



Pozwala na użycie narzędzia do tworzenia list sieci w kilku formatach.

Plik z listą sieci może być zastosowany do całej hierarchii (zwykle taka opcja jest wybrana), lub tylko do bieżącego arkusza (lista sieci jest tylko częściowa, ale ta opcja może być przydatna przy niektórych programach).

W hierarchii wieloarkuszowej, lokalna etykieta jest znana tylko w obrębie arkusza, do której należy. Dlatego etykieta T0T0 z arkusza 3 różni się od etykiety T0T0 z arkusza 5 (jeśli ich połączenie nie było celowo wprowadzone).

Wynika to z faktu, że liczba arkuszy (aktualizowana za pomocą polecenia **Numeracja elementów**) jest wiązana z lokalną etykietą. W poprzednim przykładzie, pierwsza etykieta T0T0 rzeczywiście jest nazwana T0T0_3, a postać drugiej etykiety to T0T0_5.

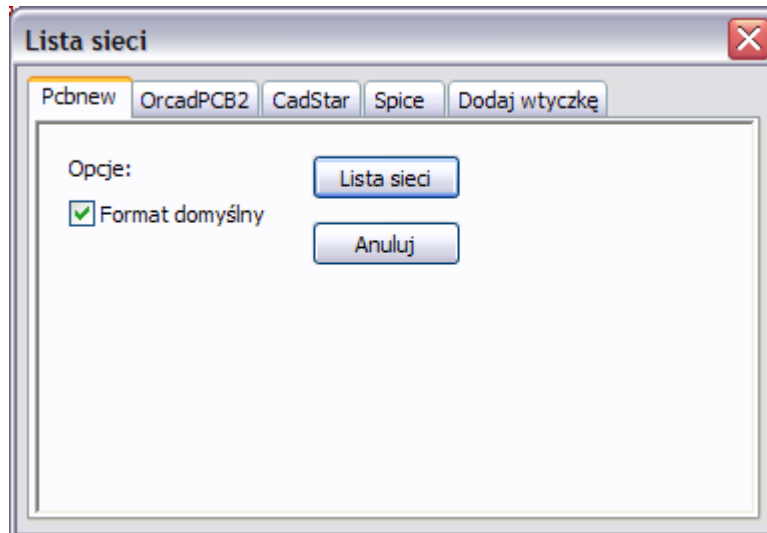
Taka możliwość wiązania może być wstrzymana na żądanie użytkownika, ale trzeba mieć świadomość możliwych w tym wypadku niepożądanych połączeń.

Uwaga 1:

Długość etykiet nie jest ograniczana przez **Eeschema**, lecz oprogramowanie eksportujące *netlisty* może tę długość ograniczać.

Uwaga 2:

Unikaj spacji w nazwach etykiet, bo zostaną one zinterpretowane jako dwa różne słowa. Nie jest to ograniczenie narzucane przez **Eeschema**, lecz wiele z formatów list sieci nie dopuszcza do ich stosowania.



Opcje:

Format domyślny - Zaznacz tę opcję jeśli chcesz skorzystać z domyślnego formatu programu **Pcbnew**.

Można też wygenerować *netlistę* w innych formatach:

- Orcad PCB2
- CadStar
- Spice, dla symulatora Spice.

Za pomocą zewnętrznych wtyczek można uruchamiać konwersje do innych formatów.

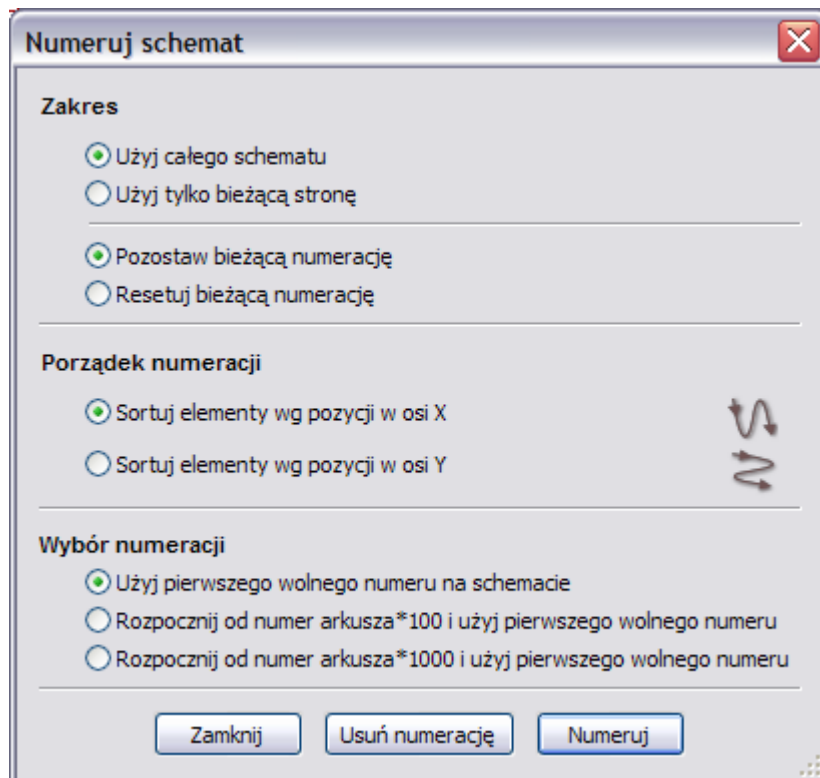
4.4. Numeracja komponentów



Pozwala na użycie narzędzia automatycznego numerowania komponentów.

Dla komponentów, które składają się z kilku części (jak na przykład 7400 TTL który posiada 4 takie same 4 bramki), przyrostek oznaczający poszczególne części także będzie zachowany (w przypadku 7400 TTL przypisana nazwa U3 będzie podzielona na U3A, U3B, U3C oraz U3D).

Można bezwarunkowo ponumerować wszystkie elementy, bądź tylko te, które są nowe na schemacie, tzn. takie, które dotychczas nie zostały jeszcze ponumerowane.



Zakres:

Użyj całego schematu:

Wszystkie arkusze zostaną poddane numeracji (opcja najczęściej używana).

Użyj tylko bieżącej strony:

Tylko bieżąca strona zostanie poddana numeracji (opcja ta jest używana tylko w szczególnych przypadkach, na przykład do oceny ilości rezystorów w bieżącym arkuszu).

Pozostaw bieżącą numerację:

Numeracja warunkowa, tylko nowe elementy na schemacie zostaną ponumerowane ponownie (opcja najczęściej używana).

Resetuj bieżącą numerację:

Numeracja bezwarunkowa, wszystkie elementy składowe schematu zostaną ponumerowane (opcja ta przydatna jest po wykonaniu operacji kopiowania bloku, po której to mogą pojawić się zdublowane elementy).

Porządek numeracji:

Określa w jakim kierunku ma poruszać się narzędzie numeracji przy sortowaniu listy elementów. Są tylko dwie opcje: pierwsza pozwala by numeracja była zwiększana najpierw kierując się w dół, druga pozwala by numeracja była zwiększana najpierw kierując się w prawo.

Wybór numeracji:

Pozwala określić dodatkowe opcje numeracji, by ją dodatkowo uporządkować i podzielić na poszczególne arkusze. Opcja podziału na arkusze zwykle jest używana przy dużych projektach.

4.5. Kontrola reguł projektowych - ERC



Pozwala na dostęp do narzędzia sprawdzania reguł projektowych schematu (zwanego ERC).

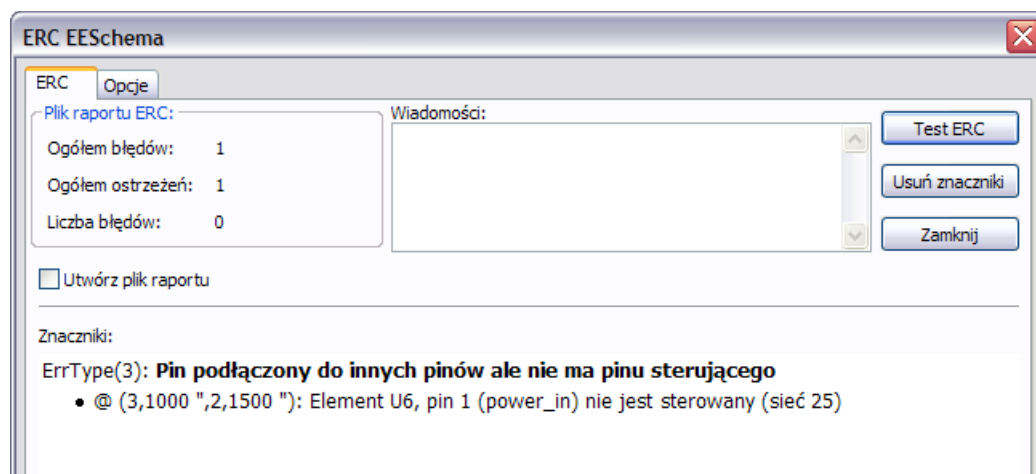
Funkcja ta generalnie służy do wykrywania złych lub nieistniejących połączeń lub innych niespójności schematu.

Eeschema po przeprowadzeniu testu ERC umieszcza na schemacie znaczniki w okolicy wyprowadzeń lub etykiet by wskazać miejsce wykrytego problemu.

Opis problemu może zostać wyświetlony klikając lewym klawiszem myszy na znaczniku ERC.

Można także wygenerować plik z listą błędów do dalszej analizy zaznaczając odpowiednią opcję.

4.5.1. Zakładka ERC



Podsumowanie ilości błędów lub ostrzeżeń jest wyświetlane w polu po lewej stronie na górze:

- *Ogółem błędów* – to całkowita liczba znalezionych nieprawidłowości.
- *Ogółem ostrzeżeń* – to liczba wykrytych ostrzeżeń.
- *Liczba błędów* – to liczba wykrytych błędów.

Opcje:

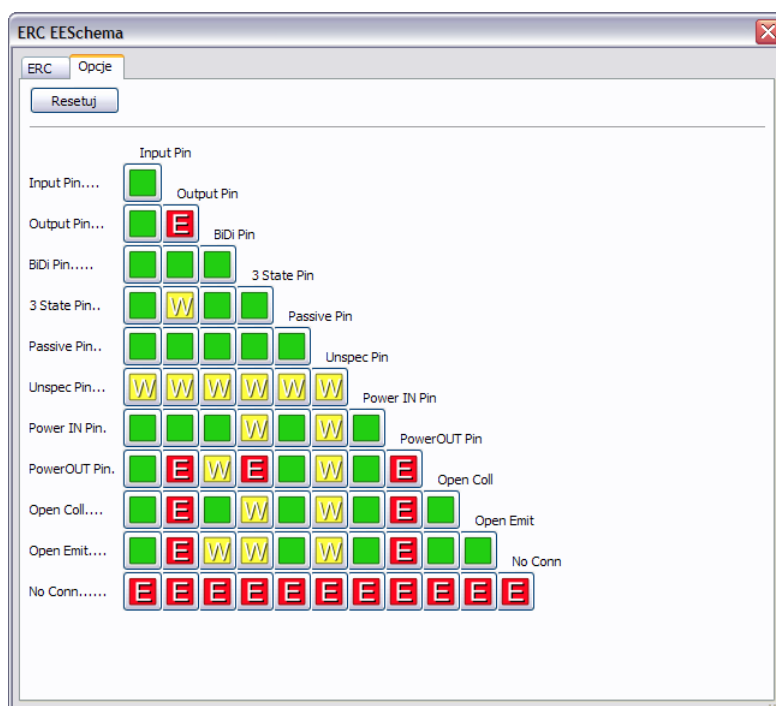
- *Utwórz plik raportu.* Jeśli chcemy dodatkowo wygenerować plik z listą błędów należy zaznaczyć tą opcję.

Polecenia:

- **Test ERC** : rozpoczyna proces sprawdzenia.
- **Usuń znaczniki** : usuwa znaczniki błędów po poprzednim testowaniu.
- **Zamknij** : zamyka okno.

Uwaga. Gdy kliknie się w komunikat o błędzie na liście, kursor przeskakuje do miejsca gdzie znajduje się znacznik tego błędu.

4.5.2. Zakładka Opcje



Ta zakładka pozwala na określenie, jaki błąd powinien zostać wygenerowany po zestawieniu ze sobą dwóch typów wyprowadzeń. Można wybrać jedną z 3 opcji zgłoszenia dla danego przypadku:

- **Brak błędu** - zielone pole
- **Ostrzeżenie** - żółte pole W
- **Błąd** - czerwone pole E

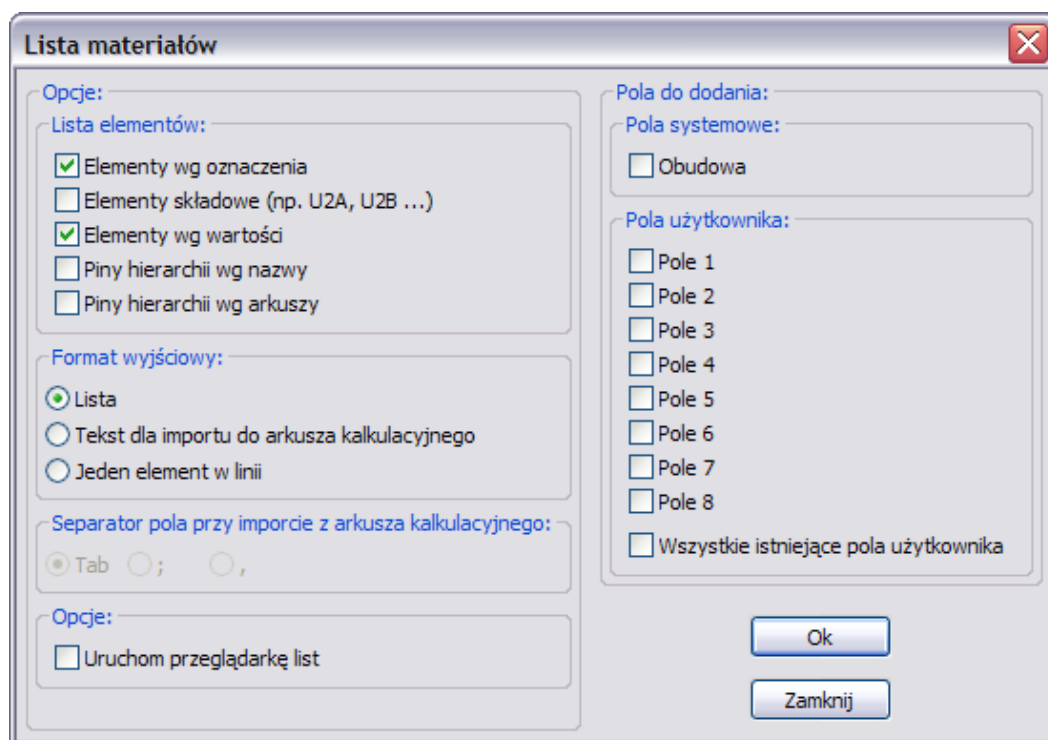
Każde pole na matrycy błędów i ostrzeżeń może być zmienione klikając w nie. Zmiany są wykonywane cykliczne.

W każdej chwili można przywrócić ustawienia domyślne poprzez przycisk Resetuj.

4.6. Lista materiałowa - BOM



Polecenie to pozwala na wygenerowanie pliku z listą elementów i/lub ich połączeniami w hierarchii (za pomocą etykiet globalnych).



Elementy na liście można sortować z pomocą:

- ich oznaczenia,
- wartości im przypisanych.

Elementy wieloczęściowe mogą być dodatkowo wyszczególnione.

Globalne etykiety mogą być sortowane według:

- klasyfikacji alfabetycznej,
- arkuszy podrzędnych.

Można te dwie metody również łączyć.

Dostępne opcje:

<i>Elementy według oznaczenia</i>	Lista materiałowa będzie posortowana według oznaczeń elementów.
<i>Elementy według wartości</i>	Lista materiałowa będzie posortowana według wartości elementów.
<i>Elementy składowe</i>	Lista materiałowa będzie zawierać informacje o każdym elemencie wchodzącym w skład elementu wieloczęściowego (np. U2A, U2B ...).
<i>Piny w hierarchii według nazw</i>	Połączenia hierarchiczne zostaną posortowane alfabetycznie.
<i>Piny w hierarchii według arkuszy</i>	Połączenia hierarchiczne zostaną posortowane wg arkuszy w których występują.
<i>Lista</i>	Lista będzie miała postać zwykłego tekstu, gotowego do wydrukowania.
<i>Tekst dla importu do arkusza kalkulacyjnego</i>	Lista będzie miała postać pliku ASCII który można będzie łatwo zaimportować do arkusza kalkulacyjnego.
<i>Jeden element w linii</i>	Lista będzie miała postać pliku CSV w którym w jednej linii połączone zostaną elementy z tą samą wartością. Dane zostaną oddzielone przecinkami.
<i>Uruchom przeglądarkę list</i>	Po stworzeniu pliku z listą materiałową zostanie otwarty domyślny edytor tekstowy, w którym będzie można podejrzeć jej zawartość.

Przydatny zestaw właściwości elementów używany zwykle do tworzenia list materiałowych to:

- ♦ *Wartość* – unikalna nazwa dla każdego z użytych elementów,
- ♦ *Obudowa* – zarówno wpisany ręcznie lub za pomocą numeracji wstecznej (zobacz następny punkt),
- ♦ *Pole1* – nazwa producenta
- ♦ *Pole2* – nazwa elementu według producenta
- ♦ *Pole3* – nazwa elementu według jego dystrybutora

Przykładowo:

Korzystanie z formatu *Jeden element w linii* wymaga tylko by właściwości jednego z elementów o takiej samej wartości zostały w pełni ustalone. Reszta będzie dzielić te informacje z tym jednym elementem.

Jednakże, jeśli na schemacie istnieją różne elementy o wartości 33K, z których jeden może być np. o mocy 0.1W, a inny ¼W; lub mogą mieć różne wymiary, będzie trzeba określić jeden jako 33K, a drugi np. jako 33KBig, co pozwoli by na liście materiałowej oba wystąpiły jako dwa różne elementy.

Formatem wyjściowym jest plik, który można zaimportować do arkusza kalkulacyjnego, gdzie koszty (lub opcjonalnie zawartość *Pola4*) łatwo będzie można podliczyć by otrzymać całościowe koszty elementów i uzyskać dane do ich zamówienia.

4.7. Narzędzie importu dla numeracji wstecznej



Ta funkcja pozwala by po narysowaniu schematu, utworzeniu listy sieci i przypisaniu elementom obudów za pomocą **CvPcb**, dokonać późniejszego importu danych o obudowach z powrotem do schematu.

Funkcja ta odczytuje pliki .stf tworzone przez **CvPcb** oraz inicjalizuje pole *Obudowa (Pole3)* dla elementów znajdujących się na liście.

Plik ten nie ma zastosowania w **Pcbnew**, lecz jest użyteczny przy wypełnianiu pola z nazwą obudowy podczas tworzenia plików materiałowych (*BOM*) i listy sieci.

Funkcja ta przechowuje informacje o obudowach/odnośnikach w jednym pliku źródłowym/schemacie, który jest źródłem danych dla listy sieci i zbędnym plik .cmp.

Przypisywanie obudów pojawi się w przyszłym eksporcie list sieci z **Eeschema**.

Jest to przydatne podczas korzystania z niektórych formatów *netlist*.

Uwaga dla Pcbnew:

Gdy **Pcbnew** nie potrafi znaleźć pliku .cmp powiązanego z plikiem listy sieci .net, używa wtedy domyślnie danych jakie znajdują się w pliku listy sieci.

Użycie pliku .cmp jest lepszym rozwiązaniem, ponieważ jeśli projektant zmieni przypisania obudów z **Pcbnew**, odpowiedni plik .cmp także zostanie zaktualizowany.

5. Tworzenie i edycja schematu

5.1. Definicje

Schemat może zostać przedstawiony na jednym arkuszu, ale głównie będzie to jednak wymagać kilku arkuszy.

Schemat prezentowany na kilku arkuszach jest zwany **schematem hierarchicznym**, a wszystkie te arkusze (każdy reprezentowany przez własny plik) stanowi dla **Eeschema projekt**.

Projekt składa się z głównego schematu, zwanego schematem głównym, i arkuszy podrzędnych stanowiących pewną hierarchię.

W celu znalezienia wszystkich plików projektu, trzeba będzie przestrzegać zasad przy jego rysowaniu, które zostaną opisane poniżej.

Dalej, kiedy będziemy mówić o projekcie jako takim, będziemy odnosić się do obu ich wersji: zarówno do schematu z pojedynczym arkuszem, jak i do hierarchicznej struktury projektu.

Zaś w dodatkowym rozdziale rozwiniemy stosowanie hierarchii i jej właściwości.

5.2. Uwagi ogólne

Schemat zaprojektowany za pomocą **Eeschema** jest czymś więcej niż tylko prostą reprezentacją graficzną urządzenia elektronicznego.

Zwykle jest punktem wyjścia dla łańcucha rozwoju, który umożliwia:

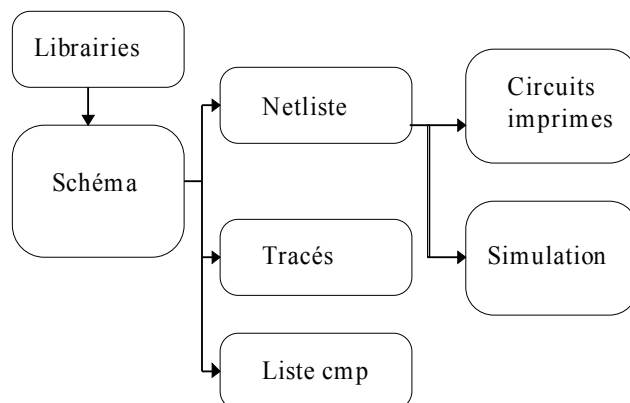
- ◆ Kontrolę poprawności (ERC), która pozwala na wykrycie błędów lub braków w schemacie.
- ◆ Automatyczne generowanie list materiałowych.
- ◆ Generowanie listy sieci dla oprogramowania do symulacji, takiego jak **PSpice**.
- ◆ Generacja listy sieci do projektowania obwodów drukowanych (**Pcbnew**). Sprawdzenie zgodności między schematem a obwodem drukowanym jest automatyczne i natychmiastowe.

W celu skorzystania z tych możliwości, trzeba będzie jednak przestrzegać pewnych ograniczeń i konwencji, aby uniknąć przykrych niespodzianek i błędów.

Schemat składa się głównie z elementów, połączeń między nimi, etykiet, węzłów, magistral i portów zasilania.

Dla zwiększenia czytelności schematu, można umieścić elementy graficzne takie jak: wejścia do magistral, komentarze i linie przerywane do budowy ramek.

5.3. Proces tworzenia



Oprogramowanie do tworzenia schematów wykorzystuje **biblioteki symboli**.

Oprócz pliku z projektem schematu, szczególnie ważny jest również **plik listy sieci**, ponieważ jest używany przez inne oprogramowanie projektowe.

Plik z listą sieci zawiera wykaz elementów i połączeń wynikających ze schematu.

Niestety (dla użytkownika) istnieje wiele formatów list sieci, z których niektóre są bardziej znane inne nie. Przykładem może być tu format Spice.

5.4. Wstawianie i edycja komponentów

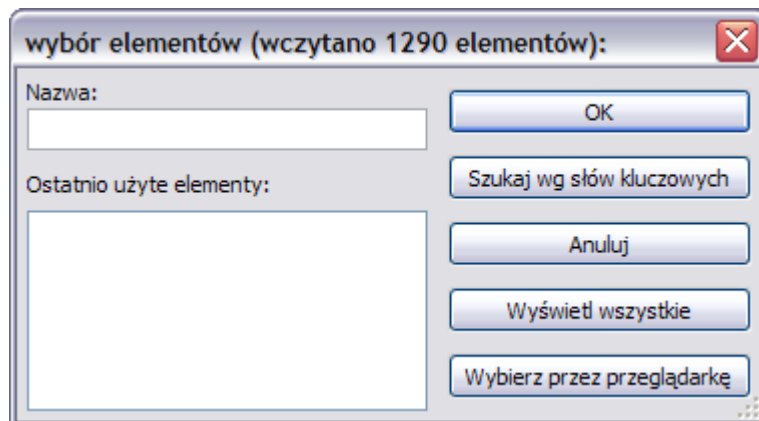
5.4.1. Wyszukiwanie i wstawianie komponentów

Aby umieścić komponent na schemacie należy użyć narzędzia wywołanego za pomocą



ikony

By umieścić nowy komponent, należy kliknąć w miejscu gdzie element ma być wstawiony. Pojawi się okienko dialogowe, które pozwoli na wpisanie nazwy elementu.



Okno dialogowe pokazuje listę ostatnio używanych elementów.

Jeśli w pole *Nazwa* zostanie wpisana „*”, lub po wybraniu przycisku *Wyświetl wszystkie*, **Eeschema** wyświetli listę bibliotek, a następnie listę dostępnych komponentów.

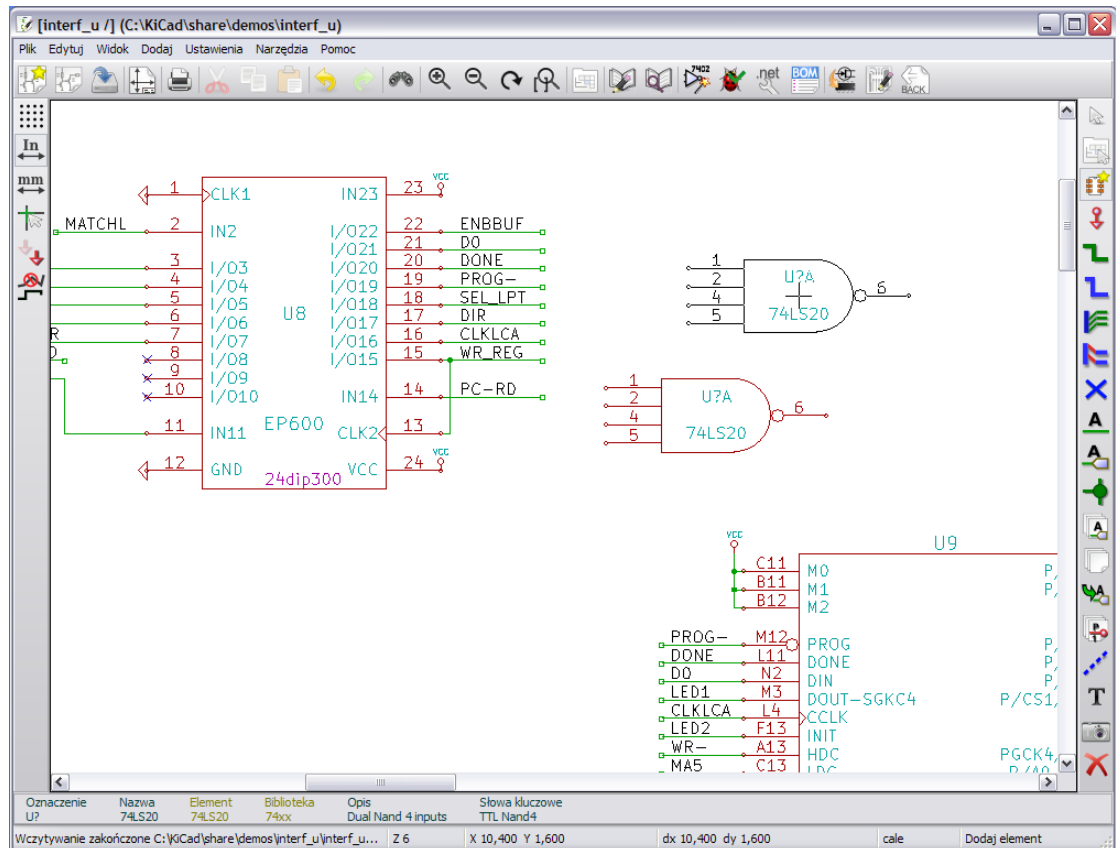
Jeśli zostanie wpisany symbol „=”, a następnie słowa kluczowe, **Eeschema** wyświetli listę elementów pasujących **do wszystkich** tych słów kluczowych.

Można również stosować filtry: na przykład jeśli jako nazwa wpisany zostanie ciąg **LM2***, wtedy wszystkie elementy, których nazwa zaczyna się od **LM2** zostaną wymienione na liście. Wybrany element pojawi się na ekranie, w miejscu gdzie zostało wywołane narzędzie.

Przed umieszczeniem elementu na wybranej pozycji (za pomocą lewego przycisku myszy), można go obrócić o 90 stopni, odbić w osi X lub Y, lub wybrać jego reprezentację poprzez wywołanie podręcznego menu. Można to również łatwo wykonać już po umieszczeniu elementu.

Należy pamiętać, że w przypadku gdy wymagany element nie istnieje, można wybrać podobny element i go zmodyfikować. Na przykład: gdy szukanym elementem jest 54LS00, można zamiast niego wybrać tożsamy element 74LS00 (wykonany jednak w innej technologii) i zmienić jego nazwę z 74LS00 na 54LS00.

Poniższy obrazek pokazuje symbol podczas operacji wstawiania go do schematu:



5.4.2. Porty zasilania

Port zasilania to także element (symbole te zostały zgrupowane w bibliotece power.lib). Dlatego też można go umieszczać tak samo jak inne elementy. Jednakże, ponieważ są to elementy używane dość często, dostępne jest specjalne narzędzie



wywoływane za pomocą ikony

Narzędzie to jest podobne do poprzednio przedstawionego, z tą różnicą, że odwołuje się bezpośrednio do biblioteki power.lib, skracając czas potrzebny na przeszukanie bibliotek.

5.4.3. Edycja / modyfikacja elementów (umieszczonych na schemacie)

Edycje / modyfikacje są dwójakiego rodzaju.

- Modyfikacja samego elementu (położenie, orientacja, wybór części dla elementu wieloczęściowego).
- Modyfikacja jednego z pól elementu (oznaczenie, wartość lub inne).

Gdy element został właśnie umieszczony na schemacie, może zaistnieć potrzeba zmiany jego wartości (szczególnie dla rezystorów, kondensatorów...), ale nie ma sensu natychmiastowego przypisania temu elementowi jego oznaczenia, lub wyboru części składowej dla elementów wieloczęściowych (np. bramka logiczna z układu 7400). To może zostać wykonane z automatu za pomocą funkcji **automatycznej numeracji**.

5.4.3.1. Modyfikacja elementów

W tym celu należy umieścić kursor myszy na elemencie (ale nie w miejscu gdzie znajdują się jego pola). Można wówczas:

- Kliknąć dwukrotnie na elemencie aby otworzyć okno dialogowe **Właściwości elementu** z opcjami jego pełnej edycji.
- Kliknąć prawym przyciskiem myszy, aby otworzyć menu podręczne, i użyć jednego z wyświetlonych poleceń (**Przesuń**, **Zorientuj**, **Edytuj**, **Usuń**).

5.4.3.2. Modyfikacja pól tekstowych elementów

Można dokonać modyfikacji oznaczeń, wartości, pozycji, orientacji, rozmiaru i widoczności pól. W celu prostych edycji można:

- Kliknąć dwukrotnie na pole tekstowe aby go zmienić.
- Kliknąć prawym przyciskiem myszy i użyć jednego z wyświetlonych poleceń (**Przesuń**, **Zorientuj**, **Edytuj**, **Usuń**) z wyskakującego menu podręcznego.

W celu edycji bardziej zaawansowanej, lub w celu stworzenia pola; należy kliknąć dwukrotnie na element, by otworzyć okno dialogowe **Właściwości elementu**:

Można ustawić orientację i inne opcje elementu, oraz dokonywać edycji, dodawać lub usuwać pola.

Każde pole może być widoczne lub nie oraz wyświetlane poziomo lub pionowo.

Wyświetlana (i zmienna) pozycja jest zawsze pokazywana dla normalnie wyświetlanego elementu (bez obrotu lub lustra) i odnosi się do punktu aktywnego elementu.

Przycisk **Resetuj do standardu biblioteki** przywraca element do zerowej orientacji, oraz przywraca domyślne opcje, wielkość i położenie każdego z pól.

Zawartość pól nie jest jednak modyfikowana, gdyż mogłoby to doprowadzić do przekłamań na schemacie.

5.5. Połączenia, Magistrale, Etykiety i Symbole zasilania

5.5.1. Podstawy

Wszystkie te elementy rysunkowe mogą zostać umieszczone na schemacie za pomocą narzędzi z prawego pionowego paska narzędziowego.

Te elementy to:

- ♦ **Połączenia** dla zwykłych połączeń.
- ♦ **Magistrale** (których stosowanie jest dozwolone wyłącznie do łączenia etykiet magistral, do poprawy estetyki rysunku)
- ♦ **Linie przerywane**, dla graficznych prezentacji.
- ♦ **Węzły**, by wymusić połączenia pomiędzy przecinającymi się połączeniami lub magistralami.
- ♦ **Wejścia do magistral** zwykłe lub do łączenia magistral, do poprawy estetyki rysunku.
- ♦ **Etykiety** dla zwykłych połączeń.
- ♦ **Etykiety globalne**, do połączeń pomiędzy arkuszami.
- ♦ **Teksty** jako komentarze.
- ♦ Symbol „**Nie połączone**” dla narzędzia ERC.
- ♦ **Arkusze hierarchiczne**, oraz ich wyprowadzenia.

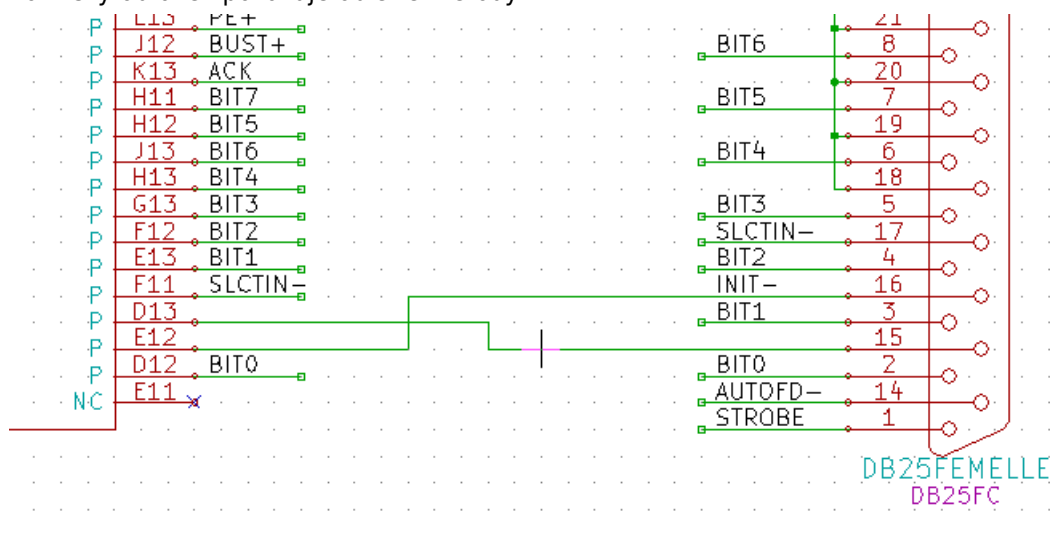
5.5.2. Połączenia (Łączy i etykiety)

Są dwie możliwości tworzenia połączeń:

- Połączenia bezpośrednio pomiędzy wyprowadzeniami.

- Połączenia z pomocą **etykiet**.

Poniższy obrazek pokazuje obie te metody:



Uwaga 1:

Punktem „kontaktu” (lub podpięcia) etykiet to dolny lewy narożnik pierwszej litery z lewej strony.

Aby etykieta była brana pod uwagę, punkt ten musi być w kontakcie z połączeniem lub nakładać się na styku połączenia z wyprowadzeniem.

Uwaga 2:

By nawiązać połączenie, jeden z segmentów połączenia musi być dołączony swoim końcem do innego zakończenia segmentu lub do punktu aktywnego u wyprowadzenia elementu.

Jeśli połączenie się nakłada na wyprowadzenie (gdy połączenie przechodzi przez wyprowadzenie ale nie trafia w jego punkt aktywny), **wtedy takie połączenie nie jest prawidłowym połączeniem.**

Jednakże, etykieta zostanie zaczepiona do połączenia niezależnie w którym miejscu jej punkt kontaktu znajdzie się na takim połączeniu.

Uwaga 3:

Jeśli połączenie musi być podłączone do innego połączenia, inaczej niż za pomocą ich zakończeń, należy umieścić węzeł w miejscu ich przecięcia.

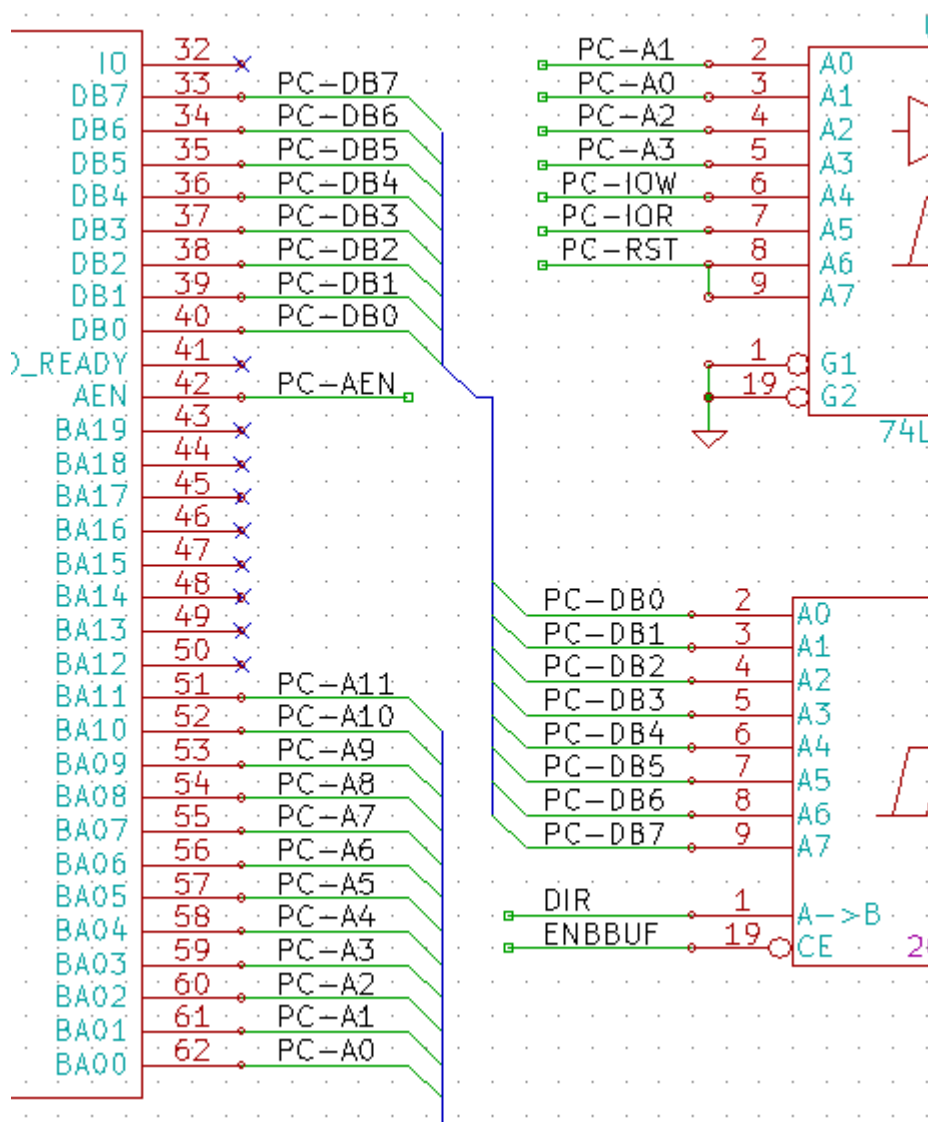
Poprzedni rysunek (połączenia doprowadzone do szpilek 22, 21, 20, 19 złącza DB25FEMALE) ukazuje taki przypadek połączeń za pomocą węzła.

Uwaga 4:

Jeśli dwie różne etykiety są umieszczone na tym samym połączeniu, zostają one połączone ze sobą i stają się równoważne: wszystkie inne elementy związane z jedną lub drugą etykietą zostają połączone razem.

5.5.3. Magistrale.

Na poniższym rysunku znajduje się następujący fragment schematu:



Jak widać, wiele wyprowadzeń (zwłaszcza pochodzących z układów scalonych) zostało połączonych magistralami.

5.5.3.1. Składniki magistral

Z punktu widzenia schematu, magistrala jest zbiorem sygnałów, począwszy od wspólnego prefiksu, a skończywszy na liczbie.

Taka koncepcja nie jest dokładnym odzwierciedleniem magistral znanych z techniki mikroprocesorowej. **Każdy sygnał jest tu członkiem magistrali.** PCA0, PCA1, PCA2 są zatem członkami magistrali PCA.

Cała magistrala jest zwana PCA[N...m], gdzie *N* i *m* to pierwszy i ostatni numer połączenia tej magistrali.

Zatem, jeśli PCA posiada 20-tu członków od 0 do 19, kompletna magistrala to PCA [0...19].

Ale zbiór sygnałów, takich jak PCA0, PCA1, PCA2, WRITE, READ nie może być zawarty w magistrali.

5.5.3.2. Połączenia pomiędzy składnikami magistral

Wyprowadzenia między tymi samymi członkami magistrali muszą być połączone poprzez **etykiety**.

Rzeczywiście, bezpośrednie połączenie wyprowadzeń do magistrali jest nonsensowne, ponieważ magistrala jest tylko zbiorem sygnałów, a takie połączenia będą nadal ignorowane przez **Eeschema**.

W powyższym przykładzie, połączenia wykonane są za pomocą etykiet umieszczonych na połączeniach podłączonych do wyprowadzeń.

Połączenia poprzez **wejścia do magistral** (odcinki połączeń pod kątem 45 stopni) mają wyłącznie wartość estetyczną, i nie są konieczne na poziomie czystego schematu.

W rzeczywistości, ze względu na **polecenie powtórzenia** (klawisz **Insert**), połączenia mogą być bardzo szybko wykonane tym sposobem, jeśli wyprowadzenia elementu są ustawione w porządku rosnącym (częsty przypadek w praktyce, na przykład w pamięciach, procesorach...):

- Najpierw należy umieścić pierwszą etykietę (np. PCA0)
- Użyć polecenia powtórzenia tyle razy ile potrzeba, aby umieścić kolejne etykiety. **Eeschema** automatycznie utworzy następne etykiety (PCA1, PCA2 ...) pionowo, teoretycznie w miejscu innych wyprowadzeń.
- Narysować połączenie pod pierwszą z etykiet. Następnie użyć polecenia powtarzania umieszczając dalsze połączenia pod etykietami.
- W razie potrzeby umieścić wejścia do magistrali w ten sam sposób (Umieścić pierwsze wejście, a następnie użyć polecenia powtarzania).

Uwaga:

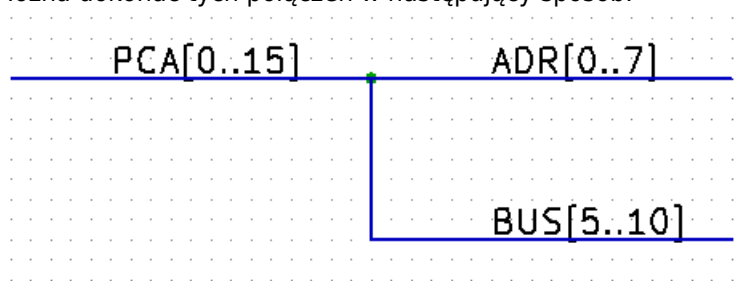
W menu **Ustawienia / Opcje**, można ustalić parametry powtarzania:

- ♦ Przeskok w pionie.
- ♦ Przeskok w poziomie.
- ♦ Przyrost etykiety (który może być dodatni np. 2, 3. lub ujemny -2, -3).

5.5.3.3. Globalne połączenia pomiędzy magistralami

Konieczne może być też połączenie pomiędzy magistralami, w celu połączenia dwóch magistral o różnych nazwach, lub w przypadku hierarchii, do tworzenia połączeń między różnymi arkuszami.

Można dokonać tych połączeń w następujący sposób:



Magistrale PCA [0..15], ADR [0..7] oraz BUS [5..10] są ze sobą połączone (Należy zwrócić uwagę na węzeł, ponieważ segment pionowej magistrali łączy się w środku poziomego segmenty magistrali).

Dokładniej, odpowiedni członkowie są ze sobą złączeni: PCA0, ADR0 są połączone (tak samo jak PCA1 i ADR1 ... PCA7 i ADR7). Ponadto PCA5, BUS5 i ADR5 są podłączone (tak jak PCA6, BUS6 i ADR6 jak PCA7, BUS7 i ADR7). A także PCA8 i BUS8 są podłączone (podobnie jak PCA9 i BUS9, PCA10 i BUS10).

Z drugiej strony nie jest możliwe połączenie członków różnych wag w ten sposób.

Jeśli chcemy połączyć członków o różnych wagach z różnych magistral, będzie trzeba to zrobić za pomocą dwóch zwykłych etykiet, umieszczając je na tym samym połączeniu.

5.5.4. Połączenia z symbolami zasilania

Gdy wyprowadzenia zasilania elementów są widoczne, muszą być podłączone tak, jak inne sygnały.

Problem pojawia się przy elementach (takich jak bramki i przerzutniki), w których wyprowadzenia zasilania są normalnie niewidoczne (**ukryte wyprowadzenia**).

Trudność jest wtedy podwójna, ponieważ:

- ♦ Nie można ich połączyć, ze względu na ich niewidzialność.
- ♦ Nie znamy ich nazw.

Poza tym, złym pomysłem będzie ich uwidocznienie i łączenie tak jak inne wyprowadzenia, bo schemat stanie się nieczytelny, i nie będzie zgody z przyjętą konwencją.

Uwaga:

Jeśli chcemy aby były one widoczne, należy zaznaczyć opcję *Pokaż ukryte piny* w oknie dialogowym **Ustawienia / Opcje** dostępnym z głównego menu aplikacji, lub za pomocą ikony



znajdującej się na lewym pasku narzędzi (pasek opcji).

Eeschema łączy automatycznie niewidzialne piny zasilania.

Wszystkie niewidzialne piny zasilania o tej samej nazwie są automatycznie łączone razem bez dodatkowych powiadomień.

Jednakże te automatyczne połączenia muszą być czasem uzupełnione:

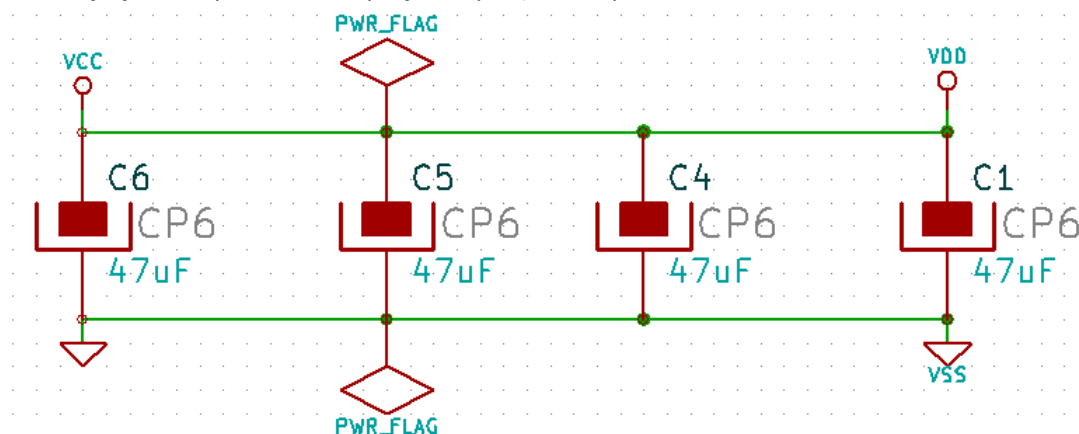
- ◆ Przez połączenie z innymi widocznymi wyprowadzeniami, podłączonymi do tego portu zasilania.
- ◆ Być może także przez połączenia pomiędzy grupami niewidocznych wyprowadzeń o różnych nazwach (np. wyprowadzenie masy nazywa się GND w elementach TTL i VSS w elementach CMOS, i muszą one być połączone ze sobą).

Dla takich połączeń, należy użyć symboli portów zasilania (komponenty zaprojektowane specjalnie w tym celu, które można tworzyć i modyfikować za pomocą edytora bibliotek).

Symbole takie składają się z niewidzialnych wyprowadzeń przypisanych do właściwych rysunków.

Nie należy w tym celu używać etykiet, które mają tylko "lokalne" możliwości łączeniowe i nie mogą łączyć niewidzialnych wyprowadzeń zasilania. (Zobacz koncepcję hierarchii by uzyskać więcej informacji).

Poniższy rysunek przedstawia przykład połączenia portów zasilania:



W tym wypadku, masa (GND) jest połączona z portem zasilania o nazwie VSS, a port zasilania VCC jest połączony do VDD.

Widoczne są też dwa symbole PWR_FLAG. Informują one, że dwa porty zasilania VCC i GND są rzeczywiście podłączone do źródła napięcia zasilania.

Bez tych symboli, narzędzie ERC wygenerowałoby ostrzeżenie: *Ostrzeżenie: Port zasilania nie jest zasilony.*

Wszystkie te symbole są elementami biblioteki power.lib.

5.5.5. Symbol „Nie połączone”

Te symbole są bardzo użyteczne podczas sprawdzania poprawności schematu (ERC) by narzędzie to, nie zwracało ostrzeżeń o niepodłączonych wyprowadzeniach. (ERC sprawdza czy żadne z połączeń zostało pozostawione niepodłączone).

Jeśli wyprowadzenia muszą pozostać niepodłączone, jest wymagane aby dołączyć do tych

wyprowadzeń specjalny symbol **Nie połączone** (narzędziem .


Symbole te nie mają żadnego wpływu na tworzone listy połączeń.

5.6. Elementy uzupełniające

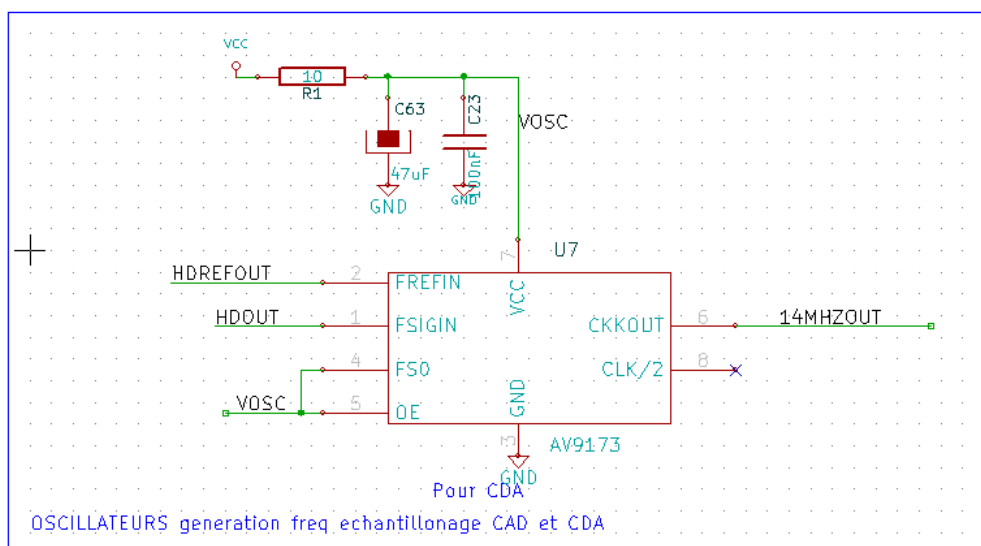
5.6.1. Komentarze

Mogą być bardzo użyteczne (na przykład do pełnego zrozumienia zawartości schematu) w celu umieszczenia na schemacie pól tekstowych i ramek.

Do tego celu przeznaczone są **Pola tekstowe** (narzędzie ) oraz **linie przerywane**

(narzędzie ) , w przeciwieństwie do etykiet i połączeń, które są elektrycznymi elementami łączącymi.

Przykład ramki:



5.6.2. Tabelka

Zawartość tabelki jest modyfikowana za pomocą narzędzia wywoływanego ikoną . Po wybraniu tego narzędzia otworzy się okno dialogowe z szeregiem parametrów i opcji:

Kompletna tabelka w wypadku takiego ustawienia parametrów wyglądać będzie następująco:

Comment 4		
Comment 3		
Comment 2		
Comment 1		
KICAD		
File: interf_u.sch		
Sheet: /		
Title: INTERFACE UNIVERSEL		
Size: A3	Date: 25 sep 2011	Rev: 2B
KiCad E.D.A. eeschema (2011-09-19 BZR 3136)-testing		Id: 1/1

Data oraz numer arkusza (*Sheet X/Y*) są automatycznie aktualizowane:

- *Data*: w momencie modyfikacji schematu.
- *Numer arkusza* (zwykle w hierarchii): poprzez narzędzie do numeracji elementów.

6. Schematy o strukturze hierarchicznej

6.1. Wprowadzenie

Hierarchiczna reprezentacja jest szczególnie dobrym rozwiązaniem dla projektów większych niż kilka arkuszy. Jeśli chcielibyśmy zarządzać tego rodzaju projektem, będzie niezbędne:


- ♦ Użycie dużych arkuszy, co spowoduje problem z drukowaniem i obsługą.
- ♦ Wykorzystanie kilku arkuszy, która tym samym doprowadzi do stworzenia hierarchii.

Cały schemat następnie składa się w głównym arkuszu schematu zwanym arkuszem nadrzędnym (głównym) i arkuszy podrzędnych stanowiących hierarchię.

Co więcej, umiejętny podział projektu na oddzielne arkusze często poprawia jego czytelność.

Z arkusza głównego, należy być w stanie znaleźć wszystkie arkusze podrzędne.

Zarządzanie schematem hierarchicznym jest bardzo proste w **Eeschema** dzięki

zintegrowanemu narzędziu jakim jest Nawigator (klawisz  na górnym i prawym pasku narzędzi, opisanym dalej).

W rzeczywistości istnieją dwa typy hierarchii (które mogą występować jednocześnie):

Pierwszy z nich został właśnie opisany i jest ogólnie używany.

Drugi polega na stworzeniu elementów w bibliotece, które pojawiają się jak tradycyjne elementy na schemacie, ale które faktycznie odpowiadają schematom, które opisują ich wewnętrzną strukturę.

Ten drugi typ jest raczej wykorzystany do opracowania układów scalonych, ponieważ w tym przypadku należy skorzystać z bibliotek funkcji w schemacie który rysujemy.

Eeschema obecnie nie obsługuje tego drugiego przypadku.

Hierarchia może być:

- ♦ **prosta**: dany arkusz jest używany tylko raz,
- ♦ **złożona**: dany arkusz jest używany więcej niż raz (przypadek zwielokrotnienia),
- ♦ **płaska**, która jest prostą hierarchią, ale połączenia między arkuszami nie są rysowane.


Eeschema uznaje wszystkie te hierarchie.

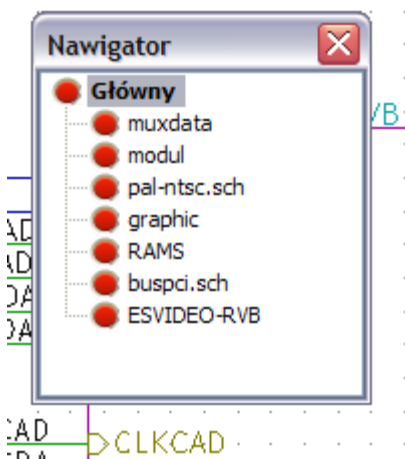
Stworzenie struktury hierarchicznej schematu jest łatwe, gdyż całość hierarchii jest obsługiwana z poziomu schematu głównego, tak jak gdyby był to tylko jeden schemat.

By opanować tworzenie hierarchii należy poznać dwie rzeczy:

- ♦ Jak stworzyć arkusz podrzędny.
- ♦ Jak zbudować połączenia elektryczne między arkuszami podrzędnymi.

6.2. Nawigacja wewnątrz hierarchii

Nawigacja jest prosta dzięki narzędziu nawigacji (ikona  na **głównym** pasku narzędzi) pokazanym poniżej:



Każdy z arkuszy jest dostępny poprzez kliknięcie w jego nazwę.

Szybka nawigacja:

Należy nacisnąć prawym klawiszem na symbolu arkusza i wybrać z menu podręcznego polecenie **Wejdź w arkusz**.

Można także szybko dostać się do arkusza głównego lub podrzędnego dzięki narzędziu



nawigacji znajdujący się na prawym pasku narzędzi.

Po wybraniu narzędzia:

- Kliknąć na nazwę arkusza by wybrać ten arkusz.
- Kliknąć gdziekolwiek indziej by wybrać główny arkusz.

6.3. Etykiety lokalne, hierarchiczne i globalne



Lokalne etykiety (narzędzie) łączą sygnały tylko w bieżącym arkuszu.



Hierarchiczne etykiety (narzędzie) łączą sygnały **tylko wewnątrz arkusza z hierarchicznym pinem** umieszczonym w arkuszu nadrzędnym.



Etykiety globalne (narzędzie) łączą sygnały **we wszystkich** elementach hierarchii.

Niewidoczne wyprowadzenia zasilania (typu **wejście zasilania** i **wyjście zasilania**) są traktowane jak globalne etykiety, ponieważ są one postrzegane jako związane między sobą we wszystkich elementach hierarchii.

Uwagi:

- Wewnątrz hierarchii (prostej lub złożonej) można wykorzystać obie: hierarchiczne jak i globalne etykiety.

6.4. Hierarchie proste

6.4.1. Wprowadzenie

Aby stworzyć **hierarchię prostą** należy:

- Umieścić w arkuszu głównym symbol zwany **arkuszem hierarchicznym**, który stanowił będzie **arkusz podrzędny**.
- Wejść do nowego schematu (arkusza podrzędnego) za pomocą narzędzi nawigacji i narysować schemat tak jak zwykle.
- Narysować połączenia elektryczne pomiędzy tymi dwoma schematami poprzez umieszczenie **etykiel hierarchicznych** w nowym schemacie (arkuszu podrzędnym), oraz etykiet (**pinów hierarchicznych**) o tej samej nazwie na arkuszu hierarchicznym z arkusza głównego. Te etykiety zostaną dołączone do symbolu w arkuszu głównym jak standardowe wyprowadzenia elementów, tak by móc połączyć je z innymi elementami schematu.

6.4.2. Arkusze podrzędne

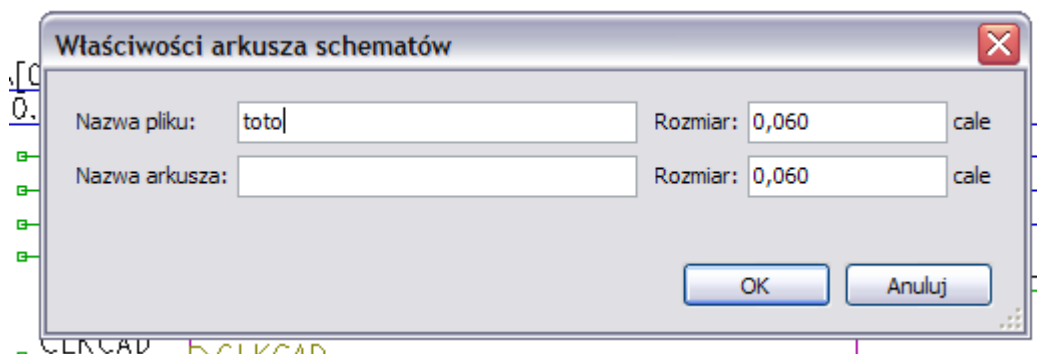
Aby narysować arkusz podrzędny, należy narysować prostokąt symbolizujący arkusz podrzędny.



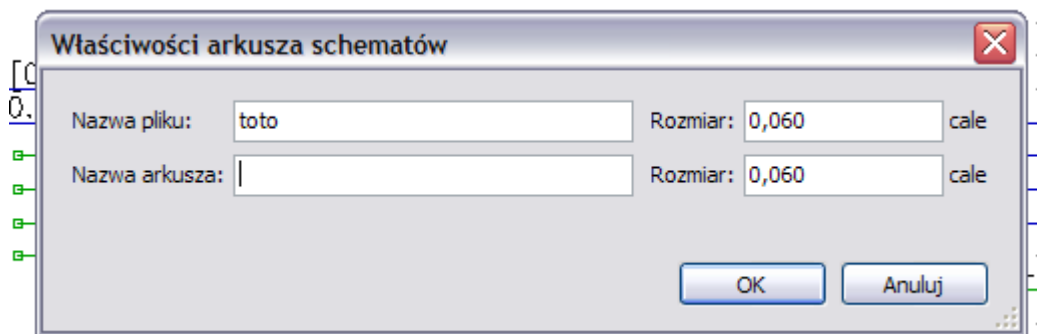
W tym celu trzeba posłużyć się narzędziem ukrytym pod ikoną .

Kliknąć następnie w miejscu górnego lewego narożnika prostokąta. Kliknąć ponownie w miejscu dolnego prawego narożnika, pozostawiając prostokąt o dostatecznym rozmiarze.

Przykładowo:



Następnie program poprosi o wpisanie **nazwy pliku** oraz **nazwy arkusza** dla tego arkusza podrzędnego (w celu dotarcia do odpowiedniego schematu, z pomocą nawigatora).



Ostatecznie można podać tylko samą nazwę pliku. Jeśli nie ma nazwy arkusza, nazwa pliku będzie pełniła rolę nazwy arkusza (zwykle tak się robi).

Wielkość tego prostokąta musi pozwolić umieścić później poszczególne piny hierarchiczne odpowiadające etykiatom hierarchicznym (*HLabels*) w arkuszach podrzędnych.

Etykiety te będą spełniać podobną rolę jak zwykłe wyprowadzenia elementów.

6.4.3. Piny hierarchiczne

W tym punkcie stworzone zostaną punkty połączeń (piny hierarchiczne) dla symbolu, który został właśnie utworzony.

Te punkty połączeń są podobne do zwykłych wyprowadzeń elementów, jednak z możliwością połączenia kompletnej magistrali za pomocą tylko jednego punktu łączącego.


Są dwie możliwości by wykonać takie połączenia:

- ♦ Umieścić różne piny hierarchiczne w symbolu arkusza **przed** narysowaniem zawartości arkusza podrzędnego (wstawianie ręczne).
- ♦ Umieścić różne piny hierarchiczne **po** narysowaniu zawartości arkusza podrzędnego oraz etykiet hierarchicznych (wstawianie półautomatyczne).

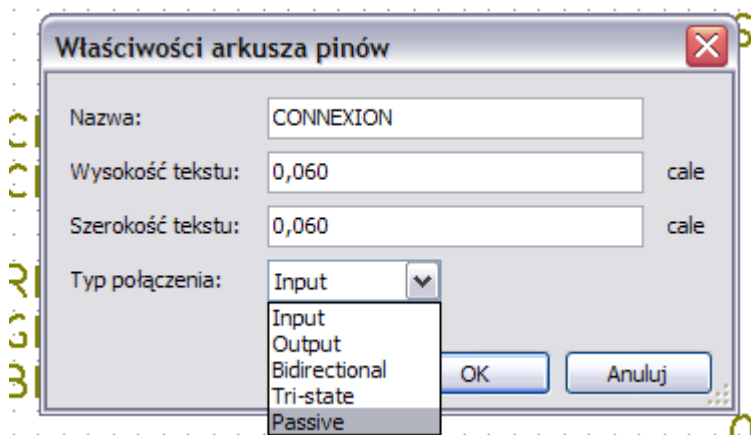
Drugie rozwiązanie wydaje się bardziej korzystne.

6.4.3.1. Wstawianie ręczne pinów hierarchicznych

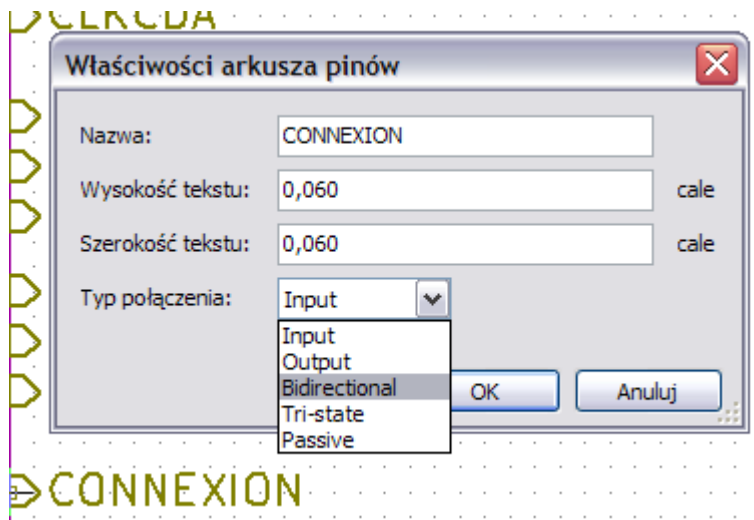
By wstawić pin hierarchiczny do symbolu tego arkusza należy:

- ♦ Wybrać narzędzie .
- ♦ Kliknąć na symbolu hierarchicznym gdzie chcielibyśmy utworzyć taki pin.

Poniżej znajduje się przykład tworzenia pinu hierarchicznego nazwanego CONNEXION.



Można później zdefiniować dodatkowe atrybuty graficzne oraz rozmiar opisu edytując ten pin hierarchiczny (Klikając prawym klawiszem, wybierając polecenie **Edycja** w podręcznym menu):




Dostępne są różne postacie symboli:

- *Input* - Wejście
- *Output* - Wyjście
- *Bidirectional* - Dwukierunkowy
- *Tri-state* - Trój-stanowy
- *Passive* - Nieokreślony

Te symbole to tylko rozszerzenie graficzne, nie mające żadnej innej roli.

6.4.3.2. Wstawianie półautomatyczne pinów hierarchicznych

Przy wstawianiu półautomatycznym postępujemy podobnie:

- ◆ Należy wybrać narzędzie ukryte pod ikoną .
- ◆ Kliknąć na symbol hierarchiczny, z którego chcemy zaimportować piny odpowiadające etykiatom hierarchicznym umieszczonym w odpowiednim schemacie. Piny hierarchiczne pojawią się, jeśli istnieją nowe etykiety, tzn. pominięte zostaną już dodane wcześniej piny.
- ◆ Kliknąć w miejscu gdzie taki pin ma się pojawić.

Wszystkie niezbędne piny mogą być więc umieszczone szybko i bez błędów.


Przewagą w stosunku do wstawiania ręcznego jest to, że są one zgodne z odpowiednimi etykietami hierarchicznymi jakie istnieją na schemacie.

6.4.4. Etykiety hierarchiczne

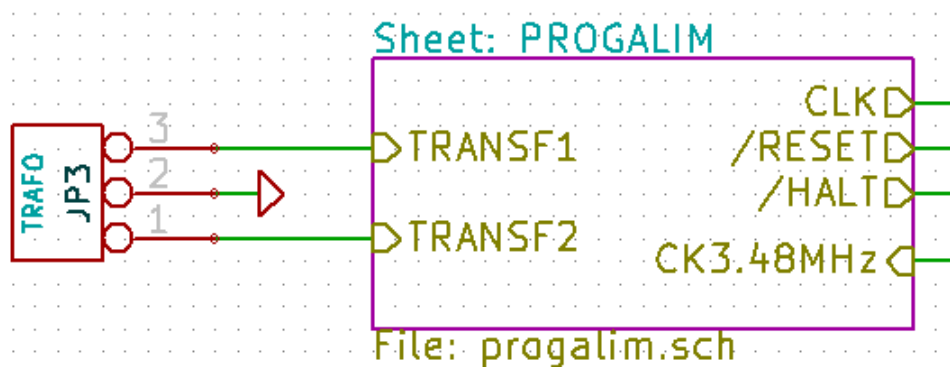
Każdy stworzony pin w symbolu arkusza musi odpowiadać **etykiecie hierarchicznej** w arkuszu podrzędnym.

Etykiety hierarchiczne są podobne do zwykłych etykiet, ale w przeciwieństwie do nich dostarczają one połączeń pomiędzy arkuszami podrzędnymi a arkuszem głównym (nadrzędnym).

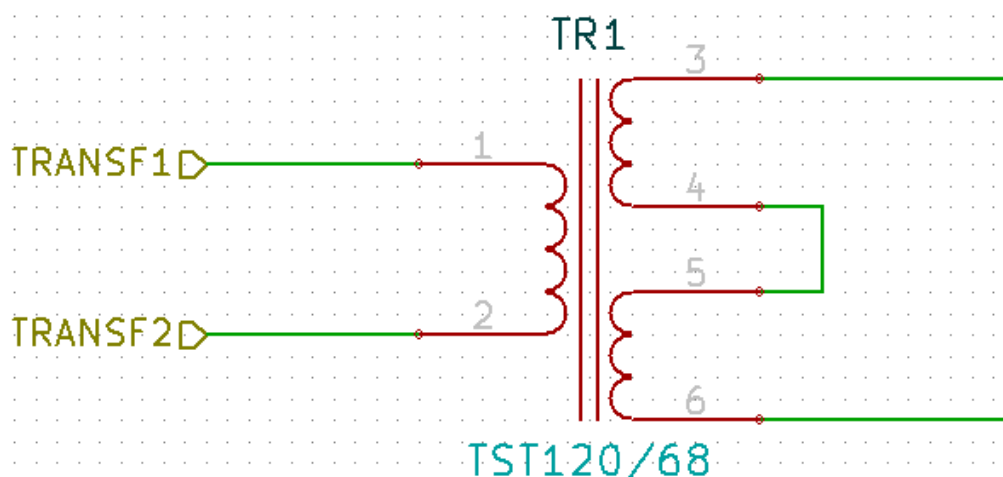
Graficzna reprezentacja dwóch etykiet uzupełniających (pinu i etykiety hierarchicznej) jest podobna.

Tworzeniem etykiet hierarchicznych (*HLabel*) zajmuje się narzędzie ukryte pod ikoną .

Poniżej znajduje się przykład arkusza głównego:



Należy zwrócić baczną uwagę na piny TRANSF1 and TRANSF2, połączone ze złączem JP3. Na następnym rysunku znajdują się odpowiednie połączenia w arkuszu podrzędnym:



Należy zwrócić też uwagę na dwie odpowiadające pinom hierarchicznym **etykiety hierarchiczne**, pozwalające na zbudowanie połączenia pomiędzy arkuszami hierarchicznymi.

Uwaga:

Za pomocą etykiet hierarchicznych oraz pinów hierarchicznych, istnieje możliwość łączenia dwóch magistral, zgodnie z tym co opisano wcześniej przy opisie ich składników (Magistrala [N. .m]).

6.4.5. Etykiety, etykiety hierarchiczne, etykiety globalne oraz piny ukryte

Oto kilka uwag na temat różnych sposobów zapewnienia połączeń innych niż połączenia bezpośrednie.

6.4.5.1. Etykiety

Etykiety (zwykle) mają lokalne możliwości łączeniowe, czyli ograniczone do arkusza schematu w którym się znajdują. Wynika to z faktu, że:

- ♦ Każdy arkusz posiada numer arkusza.
- ♦ Numer arkusza jest związany z nazwą etykiety.

Dlatego, jeśli umieścimy etykietę T0T0 na arkuszu nr 3, w rzeczywistości prawdziwa jej nazwa będzie brzmieć T0T0_3.

Jeśli w arkuszu numer 1 (arkusz główny) również umieścimy etykietę T0T0 to w rzeczywistości zostanie tam umieszczona etykieta T0T0_1, różna od T0T0_3.

Jest to prawdziwe, nawet jeśli istnieje tylko jeden arkusz.

6.4.5.2. Etykiety hierarchiczne

To, co zostało napisane w związku z etykietami prostymi również jest prawdą w stosunku do etykiet hierarchicznych.

Zatem w tym samym arkuszu, etykieta hierarchiczna T0T0 jest uważana za podłączoną do lokalnej etykiety T0T0, ale nie jest podłączona do etykiety hierarchicznej lub etykiety hierarchicznej o nazwie T0T0 w innym arkuszu.

Jednak etykieta hierarchiczna jest uważana za podłączoną do odpowiedniego pinu hierarchicznego w symbolu hierarchicznym umieszczonym w arkuszu głównym.

6.4.5.3. Ukryte piny

Jak zostało wcześniej napisane, są one połączone ze sobą gdy posiadają tę samą nazwę. Tak więc wszystkie piny zadeklarowane jako *Ukryte piny zasilania* o nazwie VCC są połączone i tworzą szynę VCC, bez względu na to na którym arkuszu są one umieszczone. Gdyby etykieta o nazwie VCC została umieszczona na arkuszu podrzędnym, to **nie byłaby** ona połączona z wyprowadzeniem VCC, ponieważ etykieta byłaby faktycznie VCC_n, gdzie *n* to numer arkusza.

Jeśli chcemy, by etykieta VCC była naprawdę podłączona do szyny VCC, będzie trzeba ją **jednoznacznie** połączyć do ukrytych wyprowadzeń zasilania, dzięki portowi zasilania VCC.

6.4.5.4. Etykiety globalne

Globalne etykiety, które mają identyczne nazwy połączone są w całej hierarchii (porty zasilania jak np. VCC ... są globalnymi etykietami).

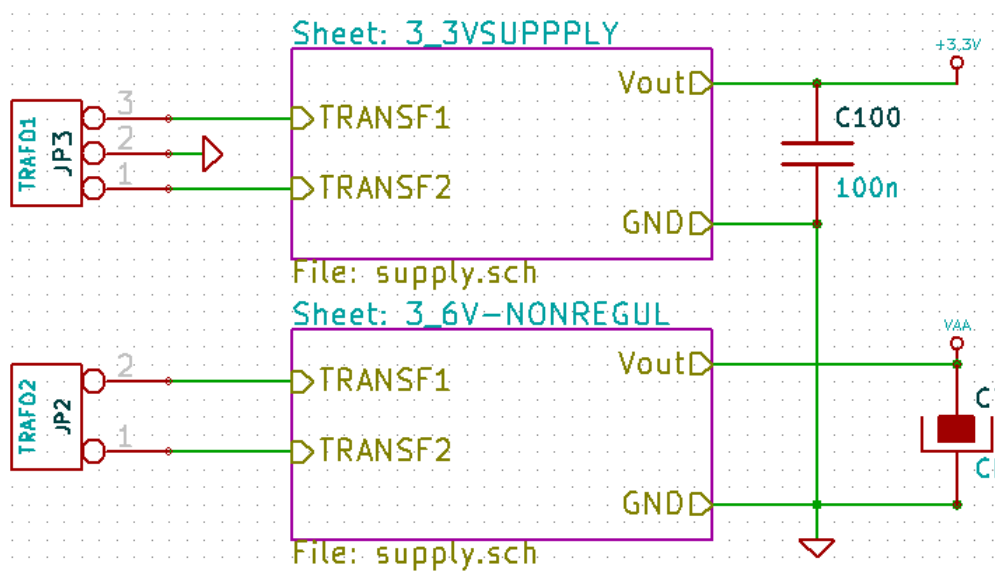
6.5. Hierarchia złożona

Hierarchia złożona występuje tam gdzie jeden z arkuszy hierarchicznych został użyty np. dwukrotnie (dwie jego postaci). Przykład takiej hierarchii został zaprezentowany poniżej.

Dwa arkusze posiadają taki sam schemat, ponieważ nazwa pliku jest taka sama dla dwóch arkuszy (supply.sch). Ich nazwy jednak **muszą pozostać różne**.

Uwaga:

Po utworzeniu drugiego arkusza (jego następnej postaci) w hierarchii złożonej należy **ponownie** zaimportować jego piny hierarchiczne.



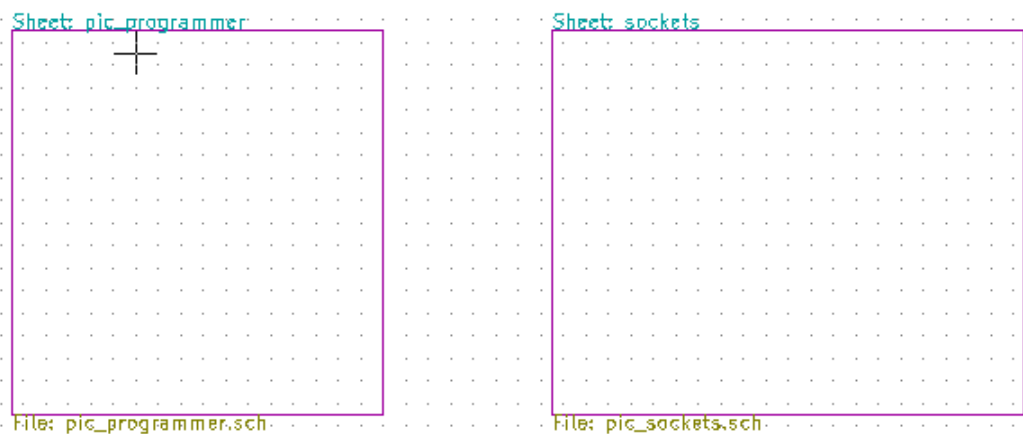
6.6. Hierarchia płaska

Można utworzyć projekt z wykorzystaniem wielu arkuszy bez tworzenia połączeń między tymi arkuszami (**płaski model hierarchii**), jeśli zastosowano następujące reguły:

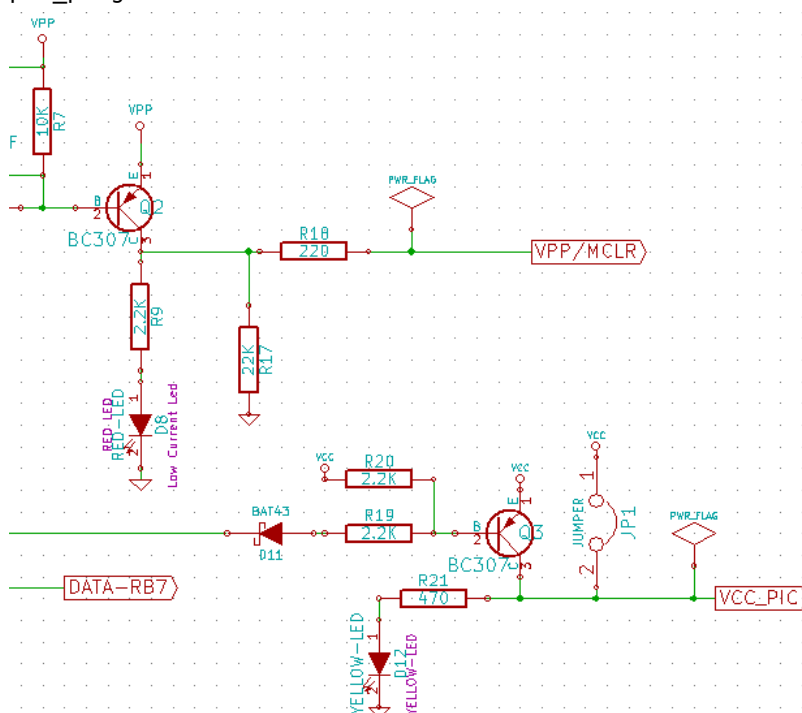
- Należy stworzyć **arkusz główny** zawierający inne arkusze, który działa jako łącznik między innymi arkuszami.
- Nie są potrzebne wyraźne połączenia między nimi.
- Wszystkie połączenia między arkuszami zostają wykonane z użyciem **etykiet globalnych** zamiast etykiet hierarchicznych.

Przykład:

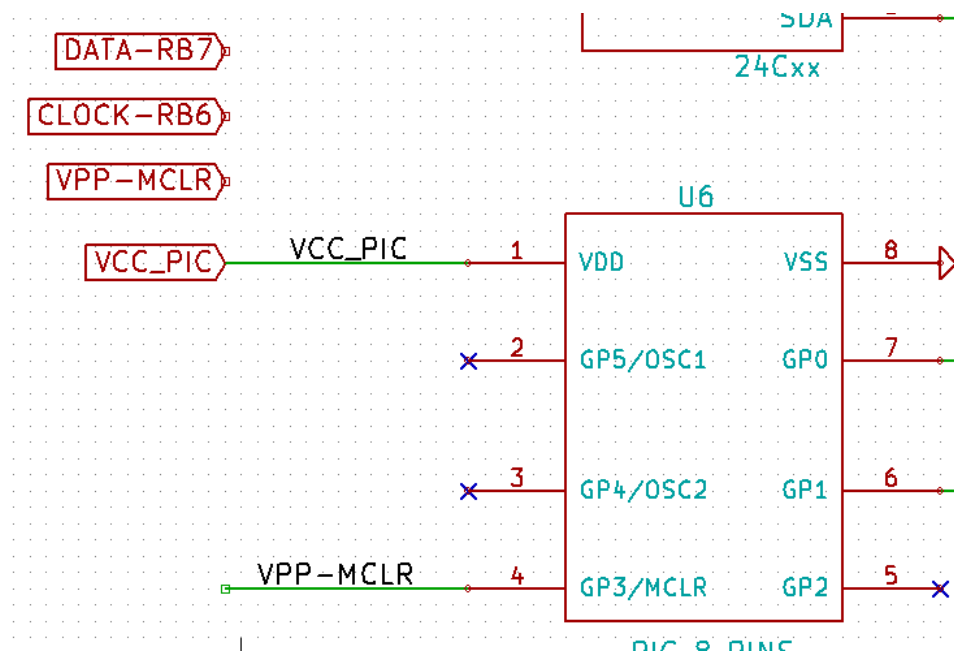
Poniżej zawartość głównego schematu:



Poniżej znajdują się dwie strony połączone za pomocą etykiet globalnych. Arkusz pic_programmer:



Arkusz sockets:




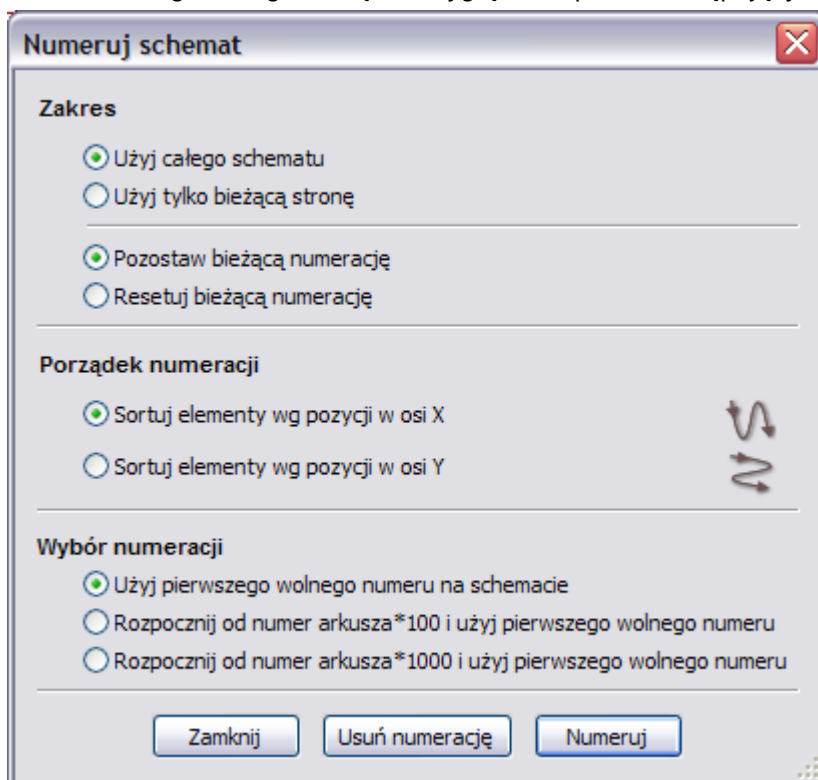
Same etykiety globalne
wyglądają w ten sposób:



7. Automatyczna numeracja elementów schematu

7.1. Przeznaczenie

Narzędzie **Numeruj schemat** (ikona ) pozwala na przeprowadzenie procesu automatycznego przypisania oznaczeń elementom, a dla elementów wieloczęściowych także na przypisanie sufiksów, tak by zminimalizować liczbę takich elementów. Okienko dialogowe tego narzędzia wygląda w sposób następujący:



7.2. Dostępne opcje

Dostępne są różne możliwości przeprowadzenia numeracji automatycznej:

- Opisywanie wszystkich składników (opcja *Resetuj bieżącą numerację*)
- Opisywanie wyłącznie nowych części (np. tylko tych, których odniesienia zakończone są przez ? Jak np IC?) (Opcja *Pozostaw bieżącą numerację*).
- Opisywanie całej hierarchii (opcja *Użyj całego schematu*).
- Opisywanie tylko bieżącego arkusza (opcja *Użyj tylko bieżącą stronę*).

Opcje zawarte w grupie *Wybór numeracji* pozwalają wybrać metodę jaka zostanie wykorzystana podczas przypisywania numerów referencyjnych wewnątrz każdego arkusza w hierarchii.

Za wyjątkiem szczególnych przypadków, automatyczna numeracja ma zastosowanie do całego projektu (wszystkie arkusze) oraz tylko do nowych elementów, jeśli nie chcemy modyfikować poprzedniej numeracji.

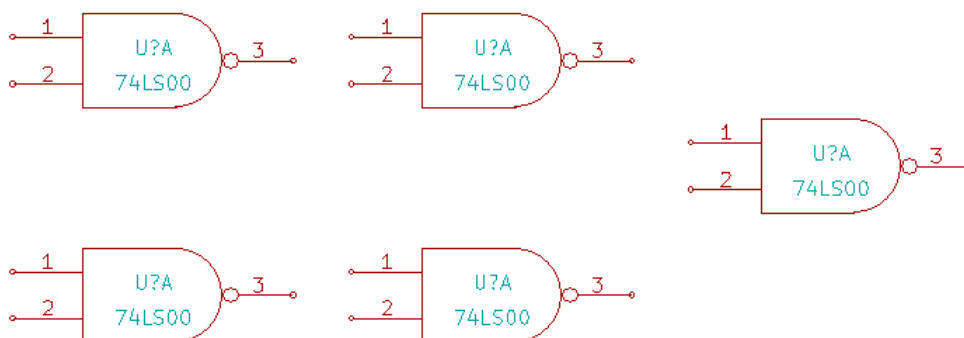
Wybór numeracji daje wybór metody użytej podczas obliczania numerów referencyjnych:

- **Użyj pierwszego wolnego numeru na schemacie:** elementy są notowane od 1 (dla każdego prefiksu odniesienia). Jeżeli istnieje poprzednia numeracja, wybrane zostaną liczby jeszcze nie wykorzystywane.
- **Rozpocznij od numer arkusza*100 i użyj pierwszego wolnego numeru:** Numeracja zostanie rozpoczęta od liczby 101 dla arkusza 1, 201 dla arkusza 2, itd... Jeśli istnieje więcej niż 99 pozycji z tym samym prefiksem w nazwie odniesienia (np. U czy R) wewnątrz arkusza 1, numeracja będzie kontynuowana od liczby 200 i dalej, a numeracja w arkuszu 2 rozpocznie się od następnego wolnej liczby.
- **Rozpocznij od numer arkusza*1000 i użyj pierwszego wolnego numeru:** Numeracja rozpocznie się od liczby 1001 dla arkusza 1, 2001 dla arkusza 2, itd...

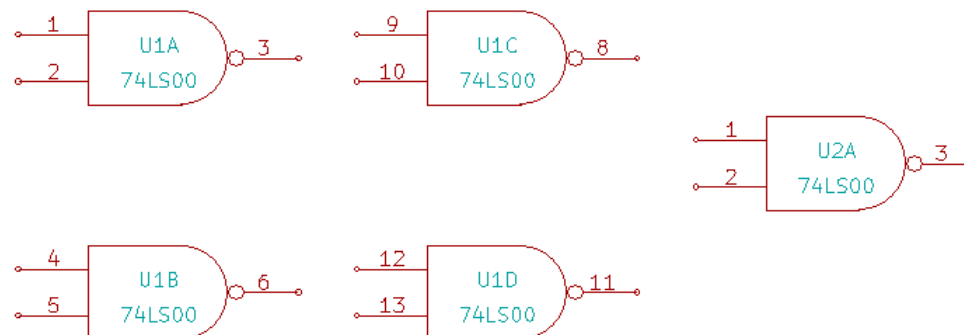
7.3. Przykłady

7.3.1. Zmiany porządku numeracji

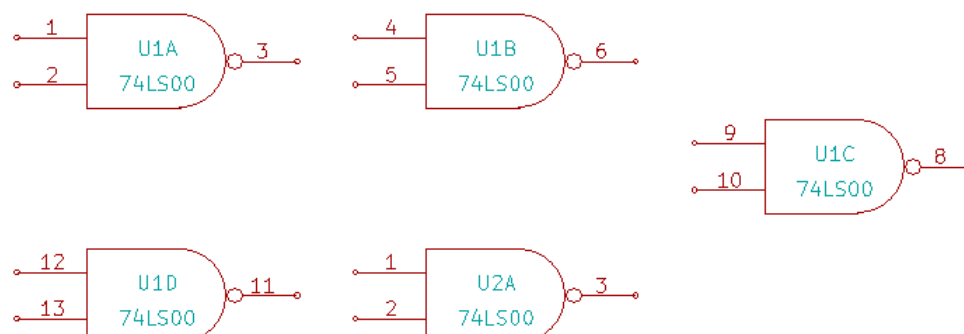
Poniższy obrazek ukazuje 5 umieszczonych elementów, lecz jeszcze nie ponumerowanych.



Po przeprowadzeniu automatycznej numeracji:



Z sortowaniem elementów według pozycji w osi X



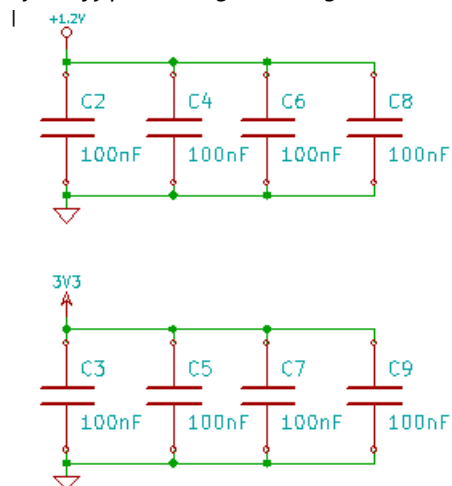
Z sortowaniem elementów według pozycji w osi Y

Można zauważyć, że cztery bramki układu 74LS00 zostały zawarte w układzie U1, a piąta bramka została przypisana do następnego układu U2.

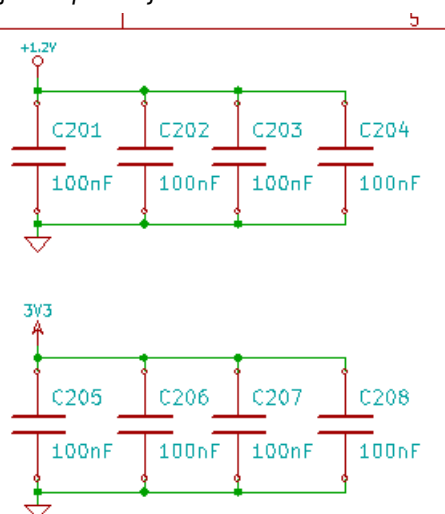
7.3.2. Wybór numeracji

Poniżej znajdują się wyniki procesu numeracji dla arkusza numer 2, w zależności od wybranej opcji:

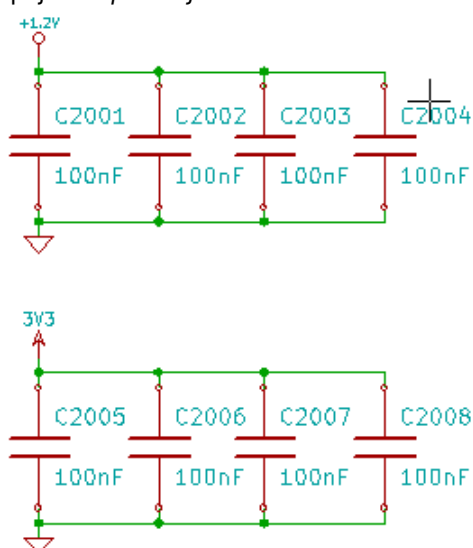
Opcja *Użyj pierwszego wolnego numeru na schemacie:*



Opcja *Rozpocznij od numer arkusza*100 i użyj pierwszego wolnego numeru*



Opcja *Rozpocznij od numer arkusza*1000 i użyj pierwszego wolnego numeru*



8. Kontrola reguł projektowych - ERC

8.1. Wprowadzenie

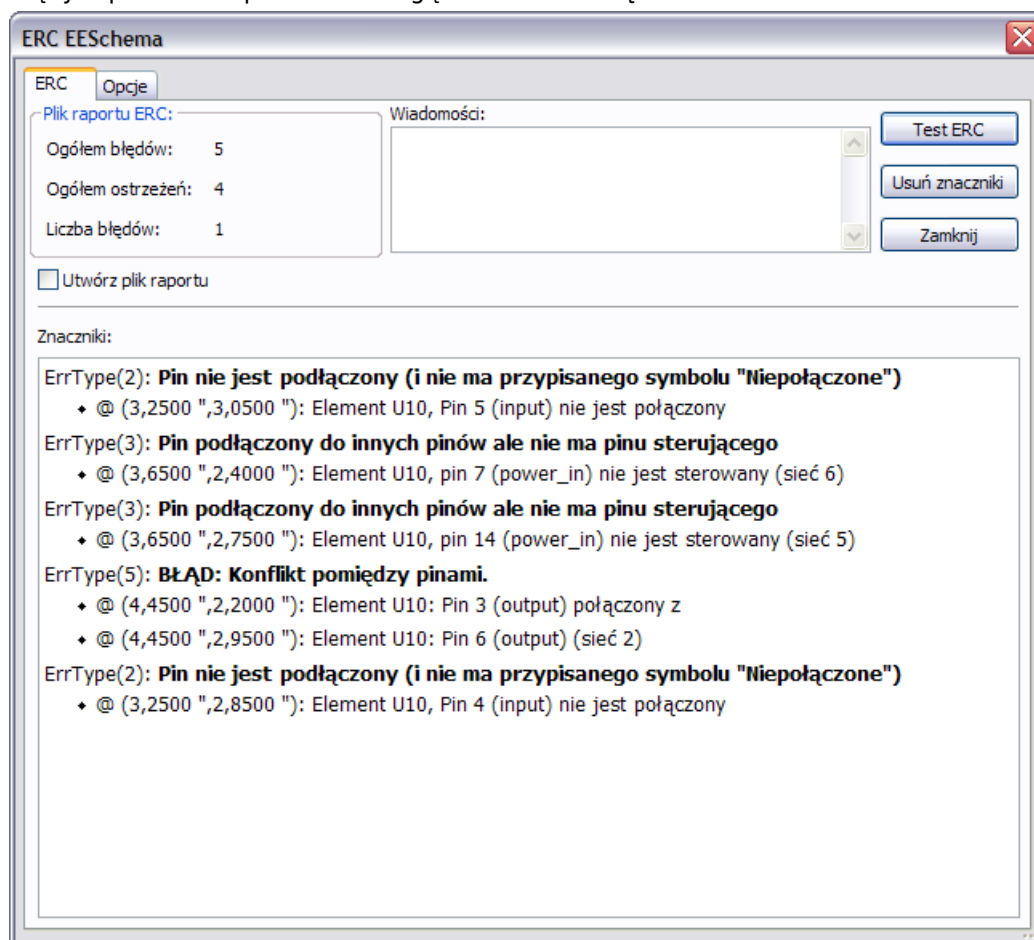
Funkcja **Kontrola reguł projektowych ERC** przeprowadza automatyczne sprawdzenie poprawności elektrycznej schematu.

Wskazuje ona błędy na arkuszu schematu, takie jak: niepodłączone wyprowadzenia, niepodłączone wyprowadzenia w symbolach hierarchicznych, zwarcia pomiędzy wyjściami, itp.

Naturalnie, sprawdzanie automatyczne nie jest bezbłędne, a oprogramowanie które mogłoby wykryć wszelkie błędy nie zostało jeszcze napisane. Aczkolwiek zwykła kontrola jest bardzo użyteczna, ponieważ pozwala na wykrycie wielu niedopatrzeń oraz małych błędów.

W zasadzie wszystkie wykryte błędy muszą zostać sprawdzone i poprawione przed kontynuacją dalszej pracy nad projektem. Jakość procesu sprawdzenia jest bezpośrednio zależna od staranności jaka została podjęta podczas tworzenia elementów bibliotecznych, zwłaszcza przy definiowaniu typu wyprowadzeń.

Błędy raportowane przez ERC mogą mieć status **błędów** lub **ostrzeżeń**.



8.2. Sposób użycia

Uruchomienie procesu kontroli reguł projektowych jest uruchamiane z pomocą narzędzia



ukrytego pod ikoną .

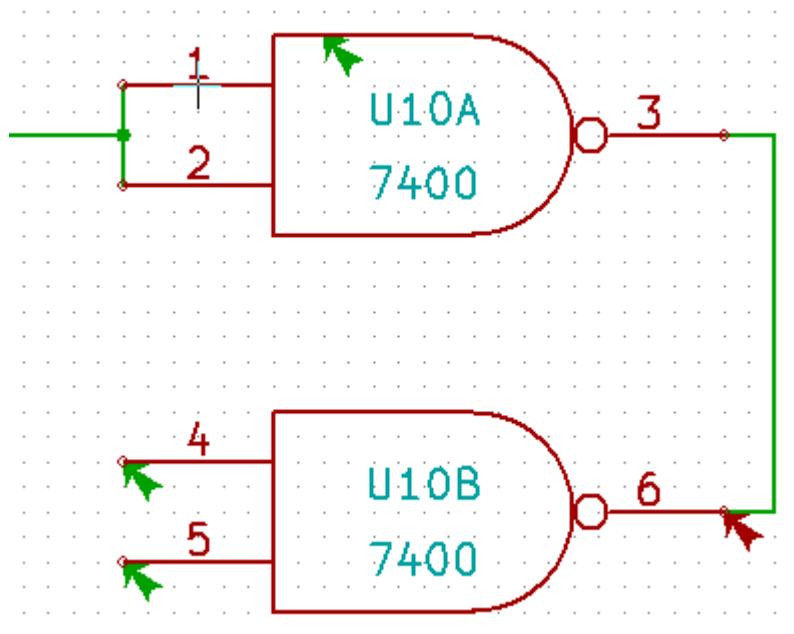
Znaczniki błędów są umieszczane na elementach lub w miejscach, które były powodem błędów (zwykle wyprowadzenia lub etykiety).

Uwaga:

- Kliknięcie w informację o błędzie w tym oknie dialogowym spowoduje, że kursor zostanie przeniesiony do miejsca w którym znajduje się znacznik tego błędu na schemacie.
- Na schemacie, kliknięcie prawym klawiszem na znacznik błędu umożliwia dostęp do informacji o powodzie błędu.

Z pomocą tego okna dialogowego można również usunąć znaczniki błędów.

8.3. Przykładowy rezultat działania ERC

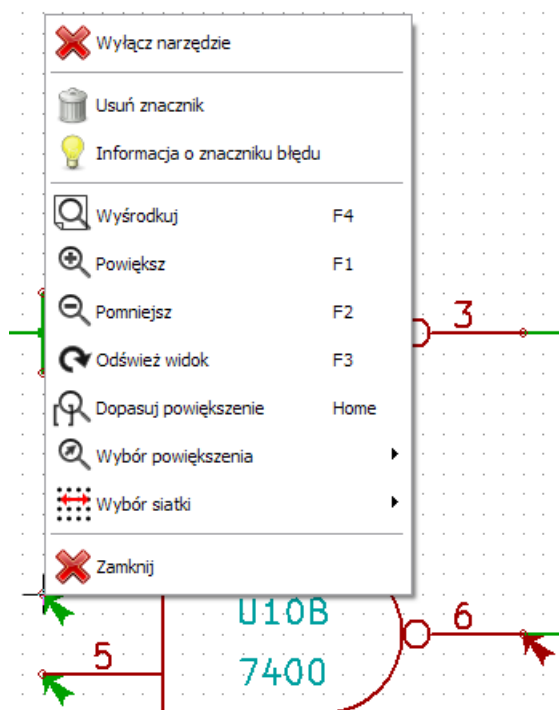


Widać tu cztery błędy:

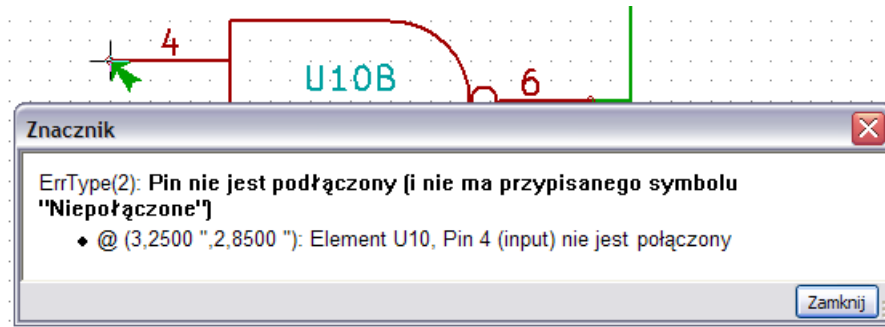
- ◆ Dwa wyjścia zostały połączone razem co z pewnością doprowadzi do superpozycji.
- ◆ Dwa wejścia nie zostały podłączone i „wiszą w powietrzu”.
- ◆ Jeden z błędów dotyczy **ukrytego** pinu zasilania (tu: linia zasilania nie posiada ustawionej flagi PWR_FLAG).

8.4. Wyświetlanie informacji o błędzie

W przypadku kliknięcia prawym klawiszem myszy na **znaczniku błędu**, zostanie wyświetlone menu podręczne pozwalające na wyświetlenie informacji o przyczynie tego błędu.



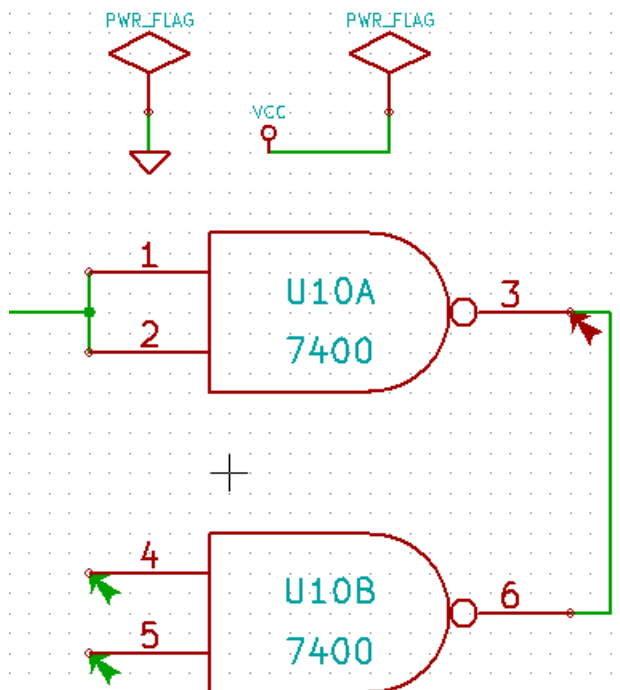
Po wybraniu polecenia **Informacja o znaczniku o błędu** wyświetlone zostanie okno z informacją o typie i powodzie wystąpienia tego błędu:



8.5. Porty zasilania a flagi zasilania

Dosyć często występuje błąd (ostrzeżenie) na wyprowadzeniach zasilania, podczas gdy na pierwszy rzut oka wszystko wydaje się poprawne (patrz przykład powyżej). To dlatego, że w większości projektów zasilanie jest dostarczane przez złączki, które **nie są źródłami zasilania** gdyż ich wyprowadzenia mają funkcję **pasywną** (nie tak jak na przykład wyjście regulatora napięcia, którego piny są zwykle zadeklarowane jako **źródło zasilania**).

ERC wobec tego nie znajduje żadnego źródła zasilania do wysterowania takiej sieci i uzna ją za nie wysterowaną (nie połączoną ze źródłem zasilania). W takich przypadkach należy do takiej sieci przypiąć specjalny element: **flagę** PWR_FLAG z biblioteki power.lib, które sygnalizuje, że ta sieć jest w istocie źródłem zasilania.

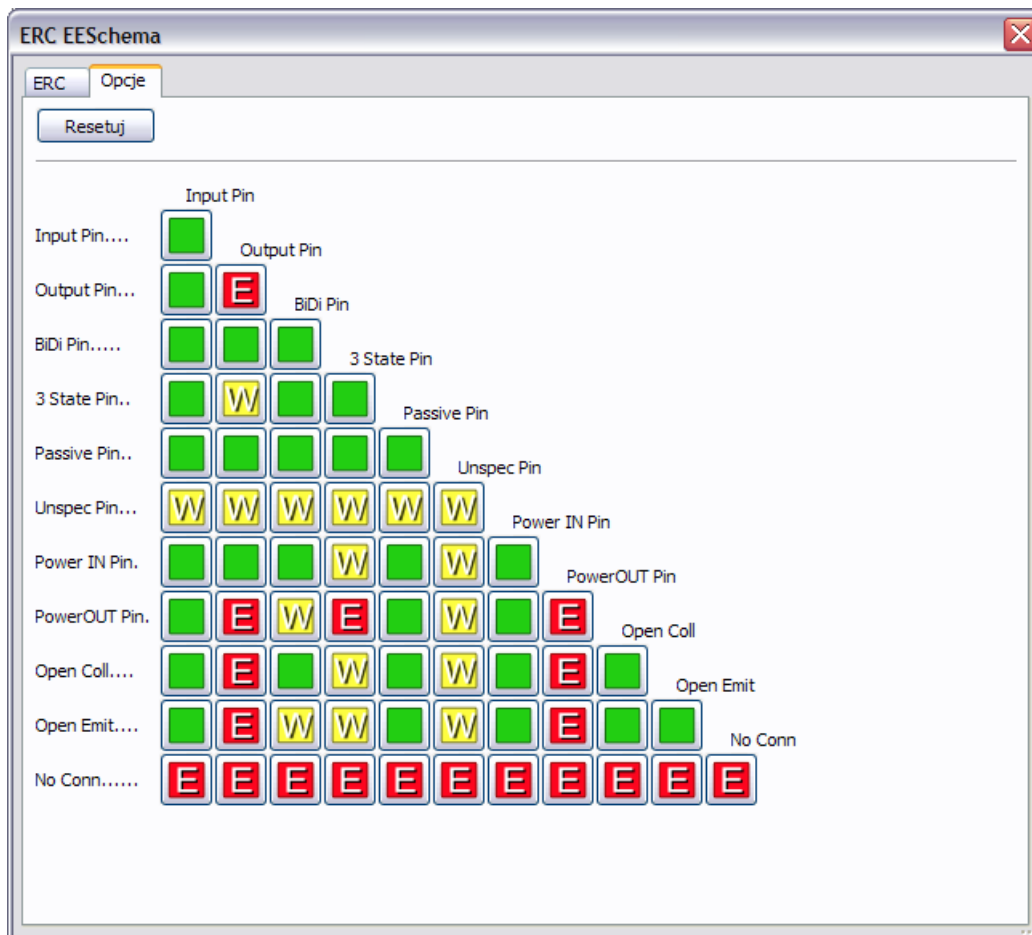


Po umieszczeniu na szynach zasilania tej flagi, błędy związane z brakiem sterowania powinny zniknąć podczas ponownej kontroli ERC.

W większości przypadków, PWR_FLAG musi zostać podpięta do sieci GND, ponieważ regulatory napięć posiadają co prawda wyjścia zadeklarowane jako źródło zasilania, ale ich wyprowadzenia masy (GND) zwykle nigdy nie są źródłami zasilania (normalny atrybut to **wejście zasilania**). Tak więc, masy nigdy nie występują jako źródła zasilania bez podpiętej PWR_FLAG.

8.6. Konfiguracja

Panel opcji pozwala na skonfigurowanie reguł jakimi się ma kierować ERC w określonych przypadkach zestawienia połączenia, i czy w danym przypadku ma zostać wygenerowany **błąd** czy tylko **ostrzeżenie**, albo też takie zestawienie nie powinno generować błędu.



Poszczególne reguły mogą zostać zmienione poprzez kliknięcie na wybranym polu na powyższej matrycy. Kolejne kliknięcia pozwalają na wybranie: **brak błędu** (zielony), **ostrzeżenie** (żółty, W) i **błąd** (czerwony, E). Zmiany odbywają się w zamkniętym cyklu, zatem aby wrócić do poprzedniego stanu należy ponownie kliknąć (jedno- lub dwukrotnie). W każdej chwili można przywrócić ustawienia domyślne poprzez przycisk **Resetuj**.

8.7. Pliki raportów generowane przez ERC

Plik raportu ERC może zostać wygenerowany i zapisany poprzez zaznaczenie opcji *Utwórz plik raportu*. Pliki takie są zapisywane z rozszerzeniem .erc.

Poniżej znajduje się przykład zawartości takiego pliku:

```
Raport ERC (2011-09-24 08:46:02)

***** Arkusz / (Root)
ErrType(2): Pin nie jest podłączony (i nie ma przypisanego symbolu "Niepołączony")
@ (3,2500 ",3,0500 "): Element U10, Pin 5 (input) nie jest połączony
ErrType(3): Pin podłączony do innych pinów ale nie ma pinu sterującego
@ (3,6500 ",2,4000 "): Element U10, pin 7 (power_in) nie jest sterowany (sieć 6)
ErrType(3): Pin podłączony do innych pinów ale nie ma pinu sterującego
@ (3,6500 ",2,7500 "): Element U10, pin 14 (power_in) nie jest sterowany (sieć 5)
ErrType(5): BŁĄD: Konflikt pomiędzy pinami.
@ (4,4500 ",2,2000 "): Element U10: Pin 3 (output) połączony z
@ (4,4500 ",2,9500 "): Element U10: Pin 6 (output) (sieć 2)
ErrType(2): Pin nie jest podłączony (i nie ma przypisanego symbolu "Niepołączony")
@ (3,2500 ",2,8500 "): Element U10, Pin 4 (input) nie jest połączony

>> Błędy ERC: 5
```

9. Generowanie list sieci

9.1. Przegląd zagadnień

Lista sieci to plik, który opisuje połączenia pomiędzy elementami na schemacie. Znajduje się w nim:

- Lista elementów,
- Lista połączeń pomiędzy nimi, zwana siecią ekwipotencjalną.

Istnieje wiele formatów list sieci. Czasem listę elementów i listę ekwipotencjałów tworzą dwa oddzielne pliki.


Lista sieci jest elementem fundamentalnym w przypadku oprogramowania do tworzenia schematów, ponieważ lista sieci to łączy do innego elektronicznego oprogramowania CAD, takiego jak:

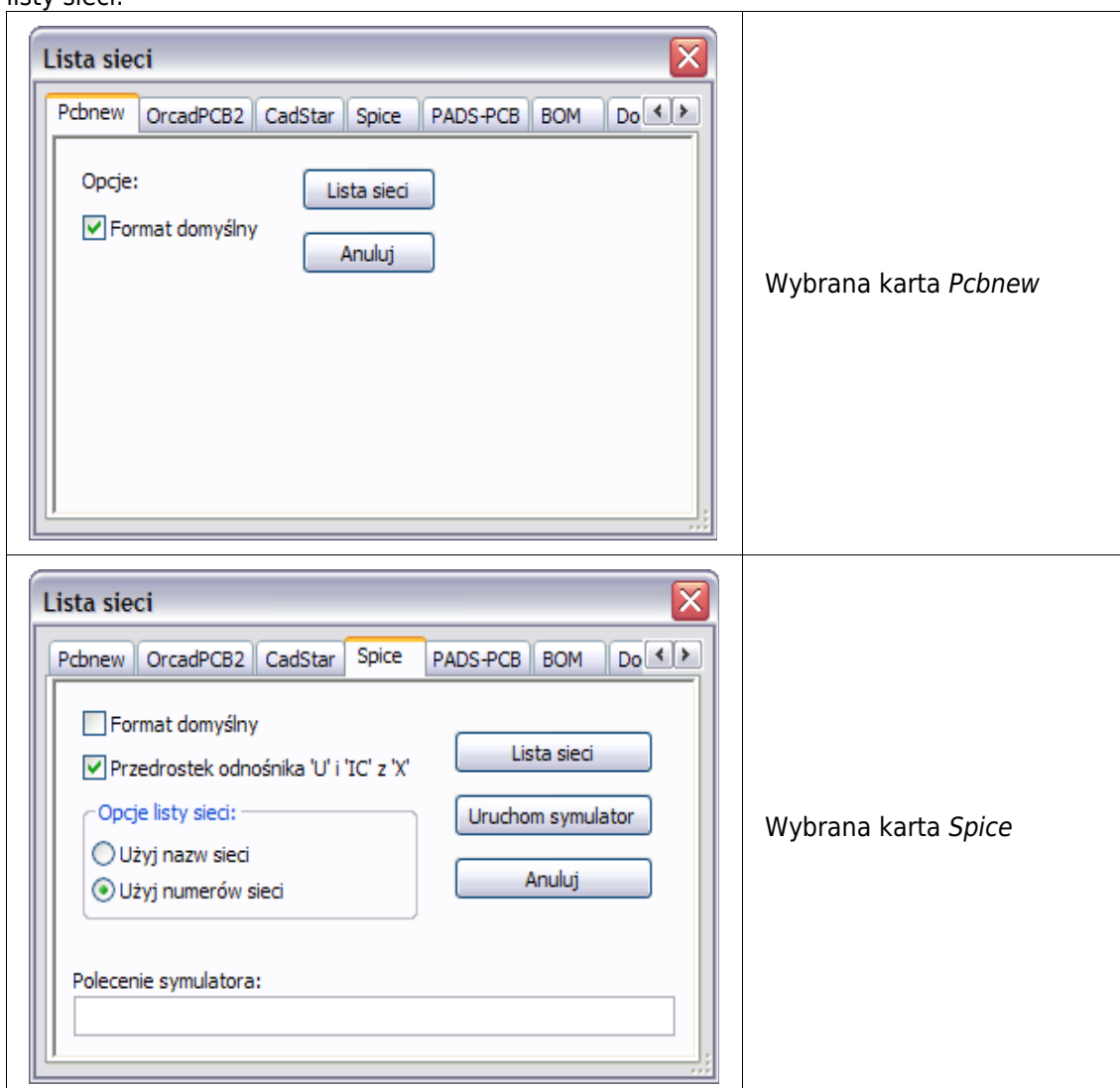
- ◆ Oprogramowanie do trasowania obwodów drukowanych (PCB).
- ◆ Symulatory układów.
- ◆ Syntetyzery układów PAL/PLA (oraz innych układów programowalnych).

Eeschema wspiera kilka formatów list sieci :

- ◆ Format **Pcbnew** (obwody drukowane).
- ◆ Format **ORCAD PCB2** (obwody drukowane).
- ◆ Format **CADSTAR** (obwody drukowane).
- ◆ Format **Spice**, używany przez sporą grupę symulatorów nie tylko przez PSpice.

9.2. Format listy sieci

Należy wybrać narzędzie dostępne spod ikony  by otworzyć okno dialogowe tworzenia listy sieci:



Korzystając z różnych kart, można wybrać żądany format jako format domyślny.

W formacie Spice, można wygenerować listę sieci z nazwami ekwipotencjałów (jest to bardziej czytelne) lub w postaci liczbowej (starsze wersje Spice akceptują tylko liczby).

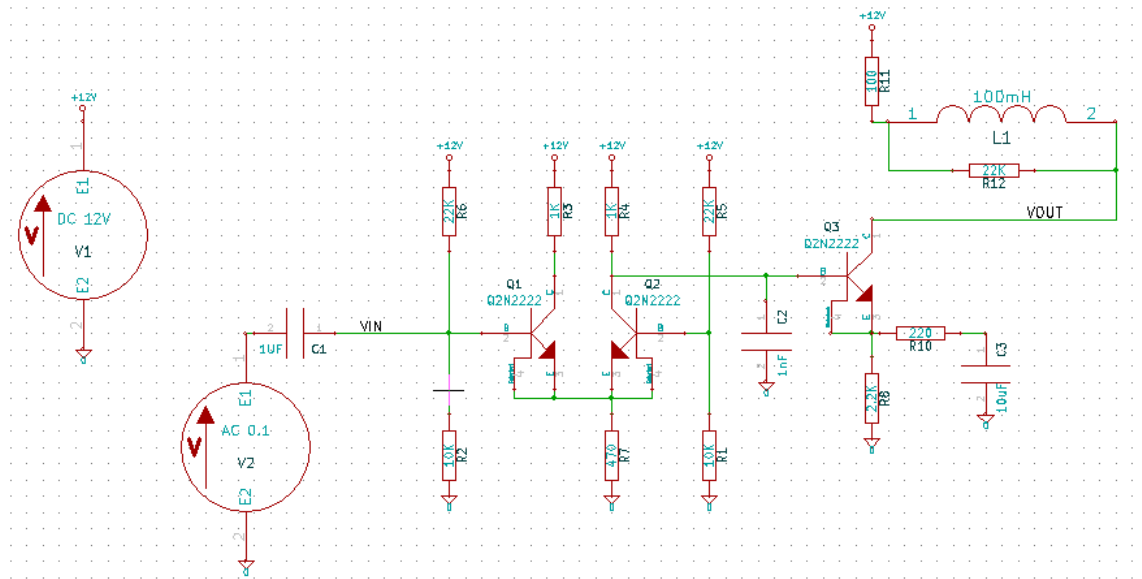
Klikając w przycisk **Lista sieci**, będziemy poproszeni o podanie nazwy pliku z listą sieci.

Uwaga:

W przypadku dużych projektów, generowanie listy sieci może zająć więcej czasu.

9.3. Przykłady

Na poniższym rysunku znajduje się schemat używający biblioteki PSPICE.



Struktura listy sieci programu **Pcbnew**:

```
# Eeschema Netlist Version 1.0 generee le 21/1/1997-16:51:15
(
(32E35B76 $noname C2 1NF {Lib=C}
(1 0)
(2 VOUT_1)
)
(32CFC454 $noname V2 AC_0.1 {Lib=VSOURCE}
(1 N-000003)
(2 0)
)
(32CFC413 $noname C1 1UF {Lib=C}
(1 INPUT_1)
(2 N-000003)
)
(32CFC337 $noname V1 DC_12V {Lib=VSOURCE}
(1 +12V)
(2 0)
)
(32CFC293 $noname R2 10K {Lib=R}
(1 INPUT_1)
(2 0)
)
(32CFC288 $noname R6 22K {Lib=R}
(1 +12V)
(2 INPUT_1)
)
(32CFC27F $noname R5 22K {Lib=R}
(1 +12V)
(2 N-000008)
)
(32CFC277 $noname R1 10K {Lib=R}
(1 N-000008)
(2 0)
)
(32CFC25A $noname R7 470 {Lib=R}
(1 EMET_1)
(2 0)
)
(32CFC254 $noname R4 1K {Lib=R}
(1 +12V)
(2 VOUT_1)
)
(32CFC24C $noname R3 1K {Lib=R}
(1 +12V)
(2 N-000006)
)
(32CFC230 $noname Q2 Q2N2222 {Lib=NPN}
(1 VOUT_1)
(2 N-000008)
```

```
(3 EMET_1)
)
(32CFC227 $noname Q1 Q2N2222 {Lib=NPN}
(1 N-000006)
(2 INPUT_1)
(3 EMET_1)
)
)
# End
```

W formacie **PSpice**, lista sieci byłaby następująca:

```
* Eeschema Netlist Version 1.1 (Spice format) creation date: 18/6/2008-08:38:03

.model Q2N2222 npn (bf=200)
.AC 10 1Meg *1.2
.DC V1 10 12 0.5

R12 /VOUT N-000003 22K
R11 +12V N-000003 100
L1 N-000003 /VOUT 100mH
R10 N-000005 N-000004 220
C3 N-000005 0 10uF
C2 N-000009 0 1nF
R8 N-000004 0 2.2K
Q3 /VOUT N-000009 N-000004 N-000004 Q2N2222
V2 N-000008 0 AC 0.1
C1 /VIN N-000008 1uF
V1 +12V 0 DC 12V
R2 /VIN 0 10K
R6 +12V /VIN 22K
R5 +12V N-000012 22K
R1 N-000012 0 10K
R7 N-000007 0 470
R4 +12V N-000009 1K
R3 +12V N-000010 1K
Q2 N-000009 N-000012 N-000007 N-000007 Q2N2222
Q1 N-000010 /VIN N-000007 N-000007 Q2N2222

.print ac v(vout)
.plot ac v(nodes) (-1,5)

.end
```

9.4. Uwagi

9.4.1. Zalecane środki ostrożności

Wiele wersji oprogramowania, które wykorzystują listy sieci nie akceptują spacji w nazwach elementów, wyprowadzeń, ekwipotencjałach lub innych elementów. Należy zatem unikać spacji w nazwach etykiet lub w nazwach i wartościach elementów, lub też w nazwach ich wyprowadzeń.

W ten sam sposób, niektóre znaki inne niż litery i cyfry mogą również powodować problemy. Należy pamiętać, że ograniczenie to nie jest związane z **Eeschema**, ale z formatami list sieci, które mogą następnie stać się nieprzetłumaczalne dla oprogramowania, które korzysta z tych list sieci.

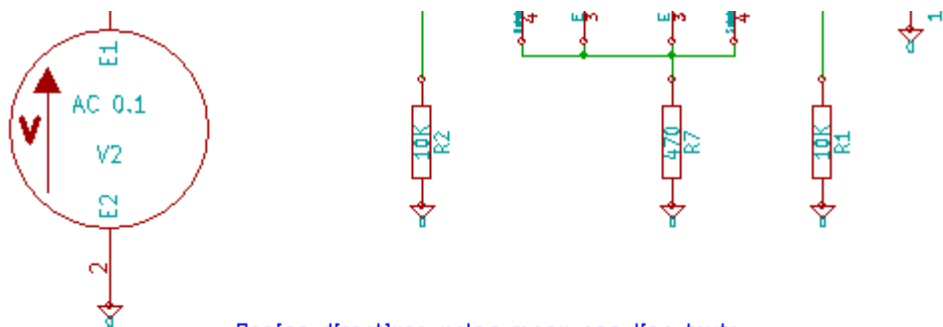
9.4.2. Listy sieci PSpice

Dla symulatora PSpice trzeba do listy sieci dodać kilka linii z poleceniami dla symulatora (.PROBE, .AC ...). Można je umieścić bezpośrednio na schemacie.

Każdy wiersz tekstu umieszczonego na schemacie, rozpoczynający się od słów kluczowych: -pspice lub -gnuap zostanie wstawiony (bez słów kluczowych) na początku listy sieci.

Każdy wiersz tekstu umieszczonego na schemacie rozpoczynający się od słów kluczowych: +gnuap lub +pspice zostanie dopisany (bez słów kluczowych) na koniec listy sieci.

Poniżej znajduje się przykład, na którym użyto wielu jednoliniowych tekstów poleceń, a także jeden wieloliniowy tekst polecenia:



Pspice directives using many one line texts

```
-PSPICE .model Q2N2222 npn (bf=200)
-gnucap .AC dec 10 1Meg *1.2
-PSPICE .DC V1 10 12 0.5
+PSPICE .print ac v(vout)
+gnucap .plot ac v(nodes) (-1.5)
```

Pspice directives using one multiline text:

```
+PSPICE .model NPN NPN
.model PNP PNP
.lib C:\Program Files\LTC\LTspiceIV\lib\cmp\standard.bjt
.backanno
```

Przykładowo: Jeśli zostanie wpisany następujący tekst (nie jako etykieta!!):

```
-PSPICE .PROBE
```

linia .PROBE zostanie wstawiona do listy sieci.

W poprzednim przykładzie dzięki tej technice, trzy linie poleceń zostaną wstawione na początek listy sieci, oraz dwie linie poleceń na końcu.

Jeśli użyty został format wieloliniowy poleceń, +pspice lub +gnucap są wymagane tylko na początku:

```
+PSPICE .model NPN NPN
.model PNP PNP
.lib C:\Program Files\LTC\LTspiceIV\lib\cmp\standard.bjt
.backanno
```

Taki zapis utworzy następujący tekst:

```
.model NPN NPN
.model PNP PNP
.lib C:\Program Files\LTC\LTspiceIV\lib\cmp\standard.bjt
.backanno
```

Poza tym, należy również pamiętać, że ekwipotencjał GND musi dla PSpice być nazwany **0** (zero).

9.5. Inne formaty, użycie „wtyczek”

Dla innych formatów listy sieci można dodać specjalne **konwertery** w formie wtyczek. Konwertery te są automatycznie uruchamiane przez **Eeschema**.

Rodział 14 zawiera odpowiednie wskazówki jak i przykłady takich konwerterów.

9.5.1. Podstawy

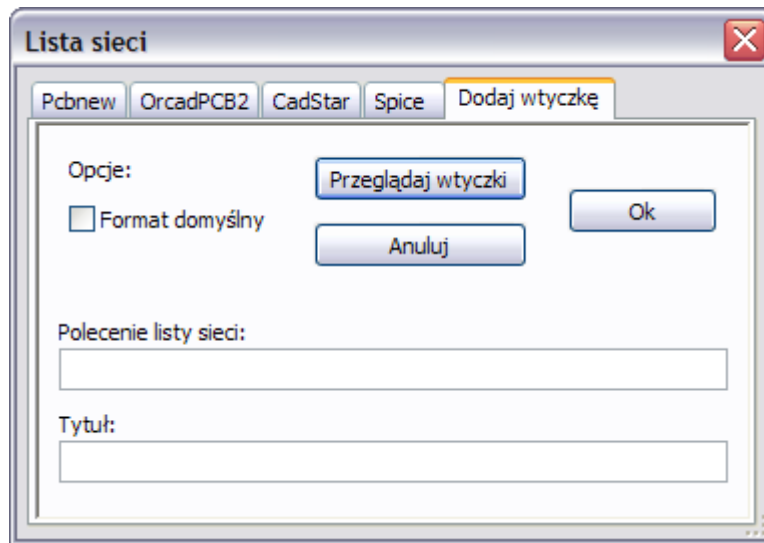
Konwerter to plik tekstowy (format **xml**, ale można użyć innego języka, np. PYTHON).

Użycie formatu **xml**, narzędzie (xsltproc.exe or xsltproc) odczytuje plik pośredni tworzony przez **Eeschema**, a następnie konwerter tworzy odpowiedni plik wyjściowy.

W takim przypadku, plik konwertera (arkusz stylu) jest bardzo mały i bardzo łatwo go napisać.

9.5.2. Inicjalizacja okna dialogowego wtyczki

Aby dodać i zainicjować nową wtyczkę należy przejść do zakładki **Dodaj wtyczkę**:

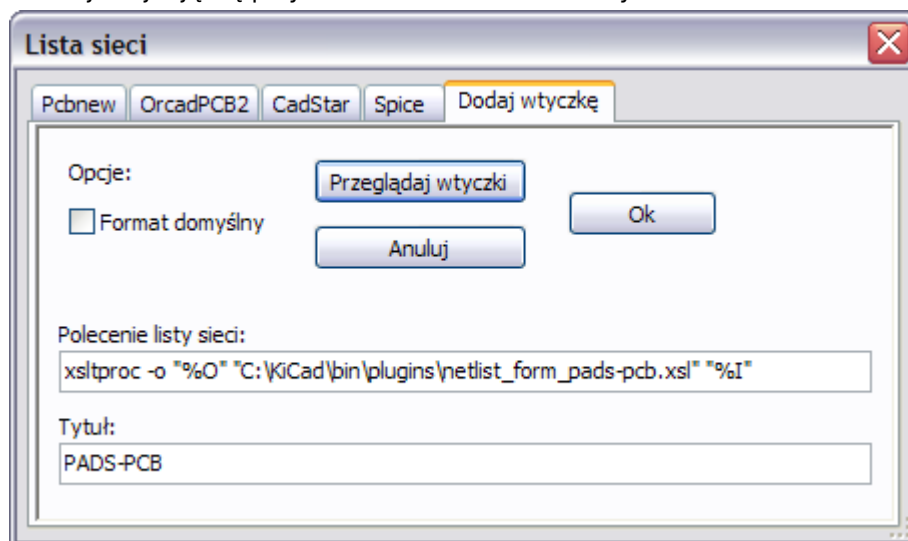


Tam klikając przycisk **Przełączaj wtyczki** można wybrać jeden z dostępnych plików konwertera (np. z \kicad\bin\plugins).

Ustawienia które będą niezbędne to:

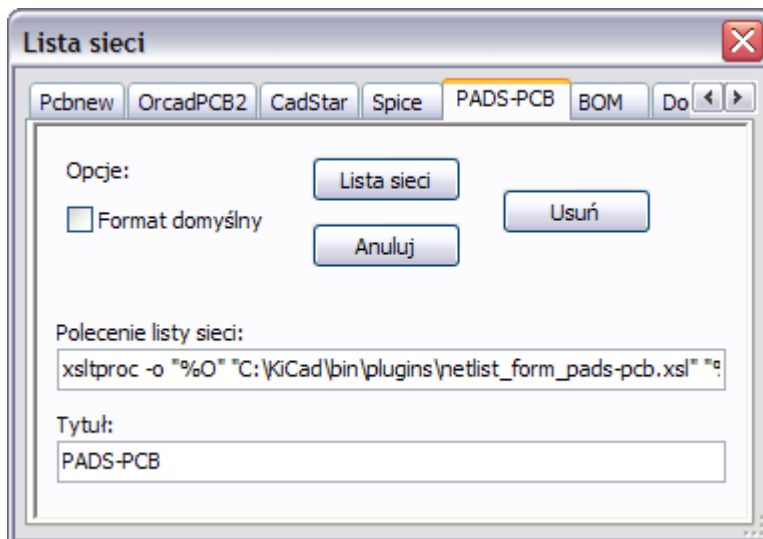
- Tytuł (na przykład: nazwa formatu listy sieci), który będzie stanowił nazwę zakładki.
- Polecenie uruchamiające tą wtyczkę.

Poniżej znajdują się przykładowe ustawienia dla wtyczki PADS-PCB:



W tym wypadku **Eeschema** sam dopisał potrzebne polecenie użycia konwertera i jedyną czynnością pozostało wpisanie nazwy tytułu.

Po kliknięciu przycisku **OK** zostanie utworzona nowa zakładka **PADS-PCB**, za pomocą której będzie można uruchomić polecenie utworzenia listy sieci w tym formacie przyciskiem **Lista sieci**. Zaznaczając opcję *Format domyślny* możemy ustawić zakładkę tego formatu jako domyślnie otwieraną przy każdym wywołaniu polecenia utworzenia listy sieci. Za pomocą klawisza **Usuń** można też usunąć tą wtyczkę by nie była dostępna.



Zasada działania programu Eeschema w przypadku takich wtyczek jest prosta. Przy wywołaniu klawiszem **Lista sieci**:

1. **Eeschema** tworzy **plik pośredni** *.tmp, dla przykładu test.tmp
2. **Eeschema** uruchamia wtyczkę, która odczytuje plik test.tmp i tworzy test.net

9.5.3. Format linii poleceń

Poniżej mamy przykład użycia xsltproc.exe jako narzędzia do konwersji plików pośrednich, oraz jako arkusz stylów, plik /netlist_form_pads-pcb.xml:

```
f:/kicad/bin/xsltproc.exe -o %0.net f:/kicad/bin/plugins/netlist_form_pads-pcb.xml %I
```

Gdzie:

f:/kicad/bin/xsltproc.exe	Narzędzie odczytujące i konwertujące pliki na podstawie plików xsl
-o %0.net	Plik wyjściowy: %0 zostanie zastąpione przez nazwę listy sieci (nazwę schematu root)
f:/kicad/bin/plugins/netlist_form_pads-pcb.xml	Nazwa pliku konwertera (arkusz stylów, format xsl).
%I	Zostanie zastąpione przez plik pośredni stworzony przez Eeschema (*.tmp).

W przypadku schematu zapisanego w pliku test.sch, kompletne polecenie będzie miało postać:

```
f:/kicad/bin/xsltproc.exe -o test.net f:/kicad/bin/plugins/netlist_form_pads-pcb.xml test.tmp
```

9.5.4. Konwerter i arkusze stylów - Program „xsltproc”

Jest to bardzo prosty program, ponieważ jego celem jest tylko konwersja wejściowych plików tekstowych (przez **pośredni plik tekstowy**) do innego pliku tekstowego.

Ponadto, z pliku pośredniego, może także tworzyć **listę materiałową BOM**.

Podczas korzystania z xsltproc jako konwertera, należy wyłącznie napisać odpowiedni arkusz stylów.

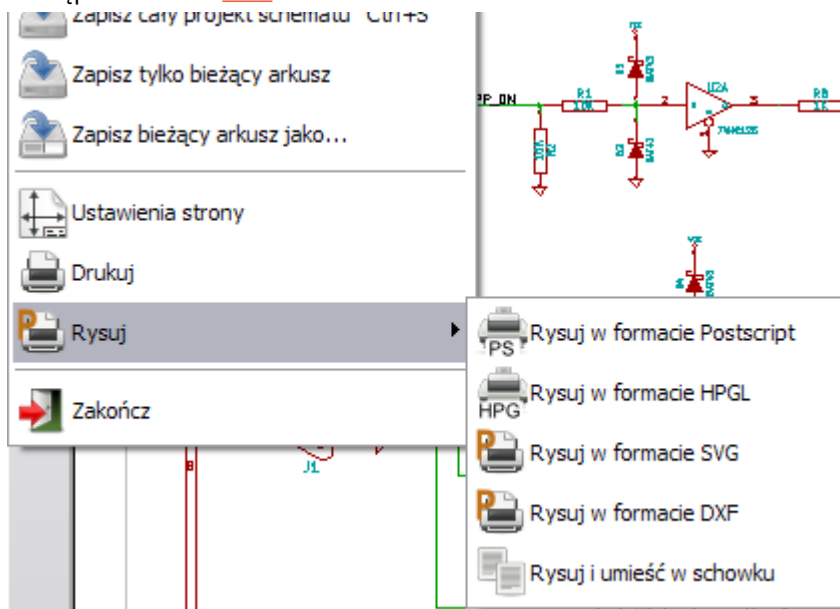
9.5.5. Format pliku pośredniego listy sieci

Zobacz zawartość **rozdziału 14** aby uzyskać więcej wyjaśnień na temat xsltproc, opisu formatu pliku pośredniego oraz przykłady arkuszy stylów dla konwerterów.

10. Drukowanie i rysowanie schematów na drukarkach lub ploterach

10.1. Wprowadzenie

Obie możliwości przenoszenia schematów na papier (bądź inny materiał drukarski) są dostępne z menu **Plik**:




Formatami wyjściowymi mogą być **POSTSCRIPT**, **HPGL**, **SVG** lub **DXF**. Można także drukować bezpośrednio na zwykłej drukarce.

10.2. Polecenia wspólne

W każdym oknie dialogowym dotyczącym operacji rysowania schematu znajdują się dwie opcje:

- ♦ **Rysuj wszystko** pozwala na narysowanie całej hierarchii (pliki są generowane dla każdego z arkusza).
- ♦ **Rysuj strone** generuje plik wyjściowy wyłącznie dla bieżącego arkusza.

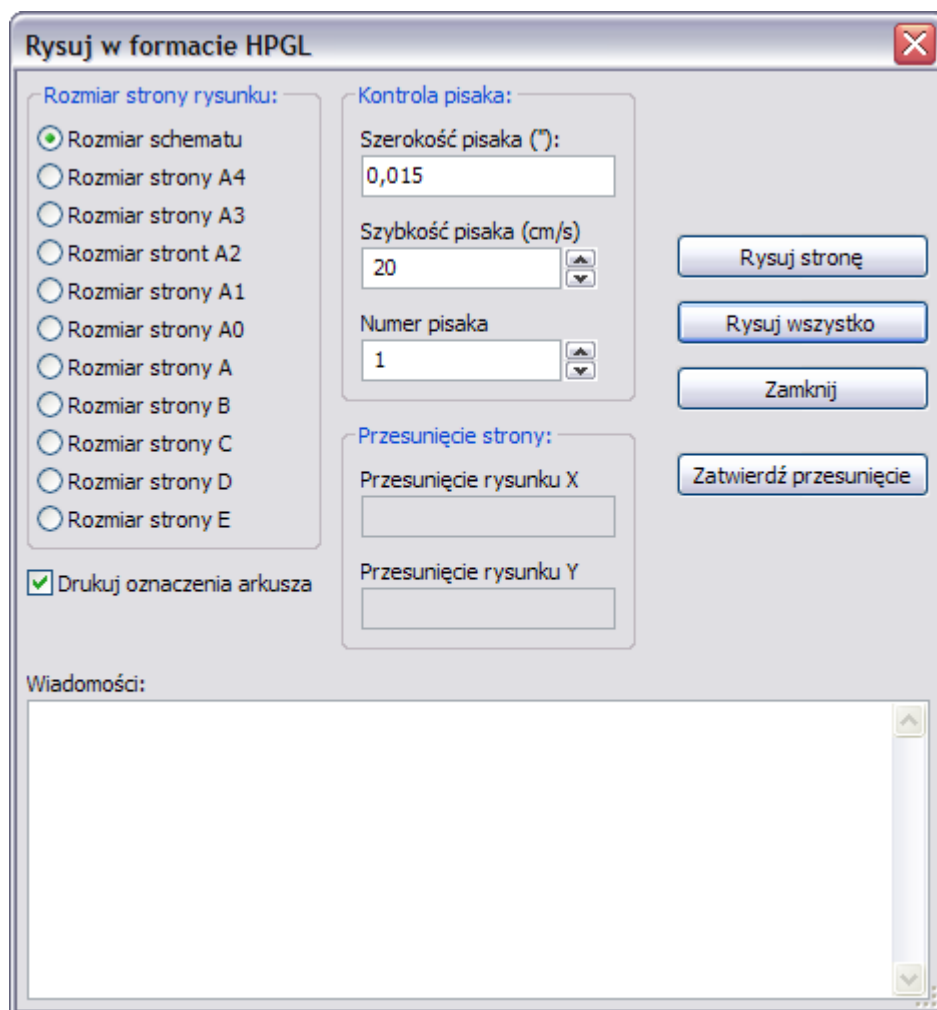
10.3. Rysuj / Rysuj w formacie HPGL

Polecenie ukryte pod ikoną  pozwala na stworzenie pliku dla plotera obsługującego **format HPGL**.

W tym formacie można zdefiniować kilka parametrów dla plotera:

- Numer pisaka
- Rozmiar pisaka (w jednostkach 0,001cala).
- Prędkość rysowania (w cm/s).
- Rozmiar pola arkusza.
- Przesunięcie strony - offset.

Po wybraniu tego polecenia otworzy się następujące okno:



Nazwa pliku wyjściowego składała się będzie z nazwy arkusza i rozszerzenia .plt.

10.3.1. Wybór rozmiaru arkusza

Normalnie jest zaznaczona opcja *Rozmiar schematu*. W takim przypadku, rozmiar arkusza plotera będzie taki sam jak rozmiar arkusza określony w **Eeschema**, a skala będzie wynosić 1. Jeśli wybrano inny rozmiar arkusza docelowego (od **A4** do **A0**, lub **A** do **E**), to skala zostanie automatycznie dobrana, tak aby rysunek wypełnił w pełni stronę plotera.

10.3.2. Ustawienie przesunięcia strony

Dla wszystkich standardowych rozmiarów, można ustawić **przesunięcie strony** by wyrównać rysunek na stronie jak najdokładniej.

Ponieważ plotery mogą posiadać punkt początkowy w centrum pola roboczego lub w dolnym lewym rogu, wymagana jest możliwość wprowadzenia offset-u (przesunięcia) by rysunek został narysowany poprawnie.

Ogólnie rzecz biorąc:

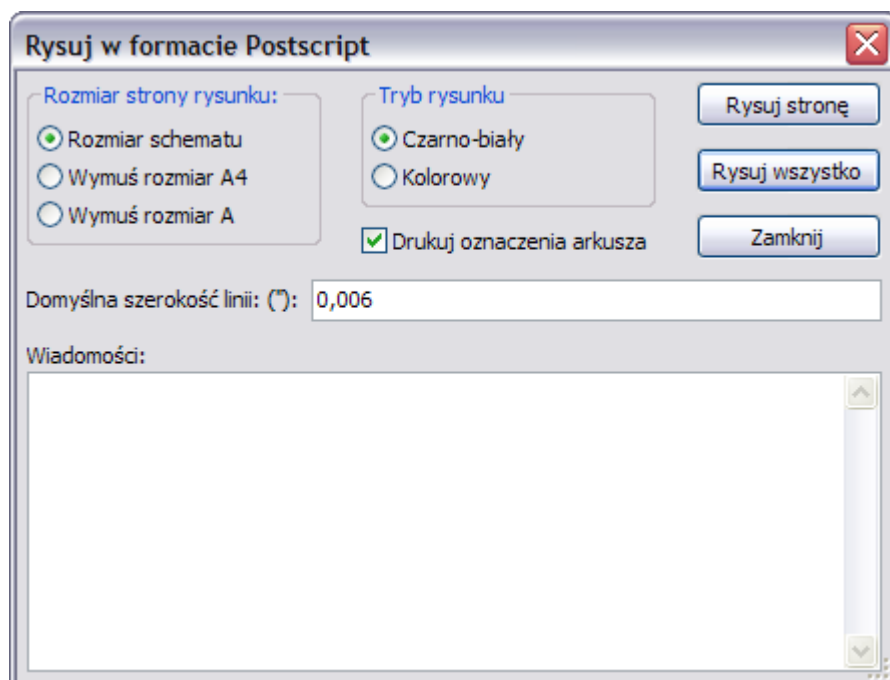
- ♦ Dla ploterów mających **punkt wyjściowy na środku arkusza**, offset musi być **ujemny** i ustawiony **w połowie** rozmiaru arkusza.
- ♦ Dla ploterów posiadających **punkt początkowy w lewym dolnym rogu** arkusza, offset **musi być ustawiony blisko zera**.

Ustawienie offsetu jest dość proste:

- Wybrać rozmiar arkusza.
- Ustawić *Przesunięcie rysunku X i Y*
- Kliknąć na *Zatwierdź przesunięcie*.

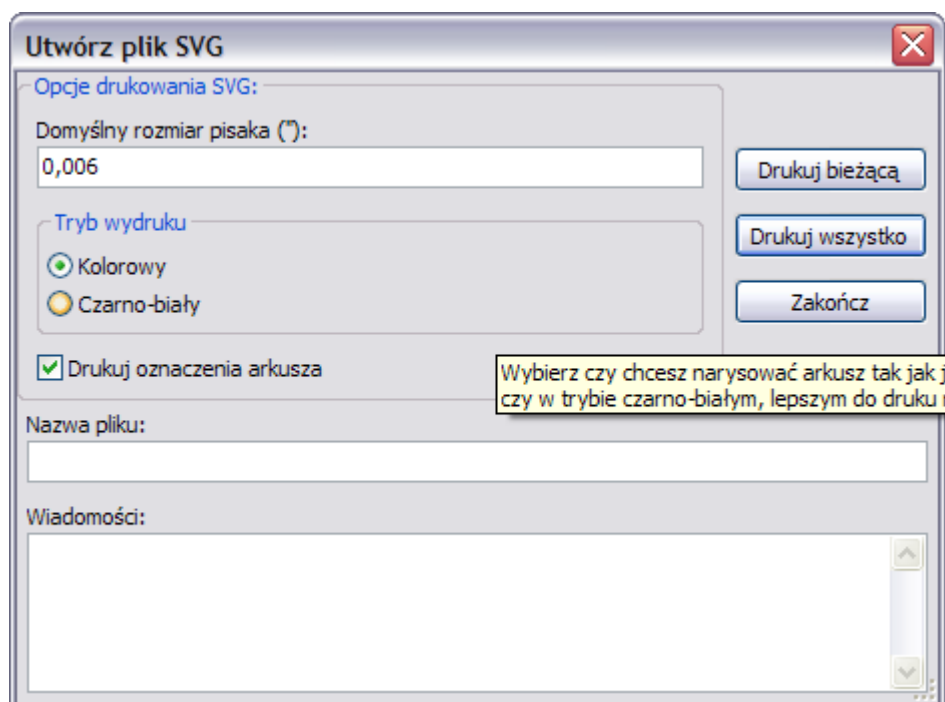
10.4. Rysuj / Rysuj w formacie PostScript

Polecenie to pozwala na stworzenie pliku w formacie *PostScript*.



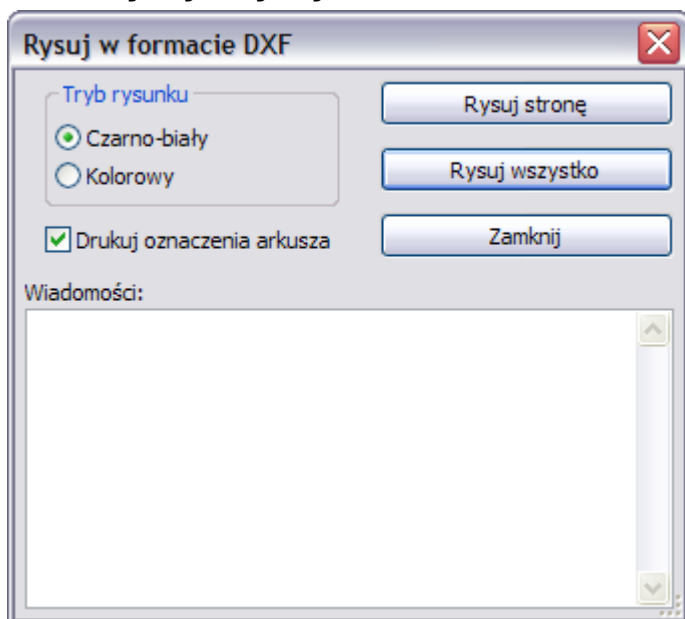
Nazwa pliku wyjściowego składała się będzie z nazwy arkusza i rozszerzenia .ps. Można dodatkowo odznaczyć opcję: *Drukuj oznaczenia arkusza*. Jest to użyteczne w przypadku tworzenia pliku postscriptowego do późniejszej obróbki (format .eps), aby umożliwić wstawianie rysunków do procesora tekstu. Okno wiadomości zawierać będzie nazwy plików jakie zostały utworzone.

10.5. Rysuj / Rysuj w formacie SVG



To polecenie pozwala na utworzenie plików, które zawierać będą **skalowane rysunki wektorowe - SVG**. Nazwa pliku wyjściowego składała się będzie z nazwy arkusza i rozszerzenia .svg.


10.6. Rysuj / Rysuj w formacie DXF

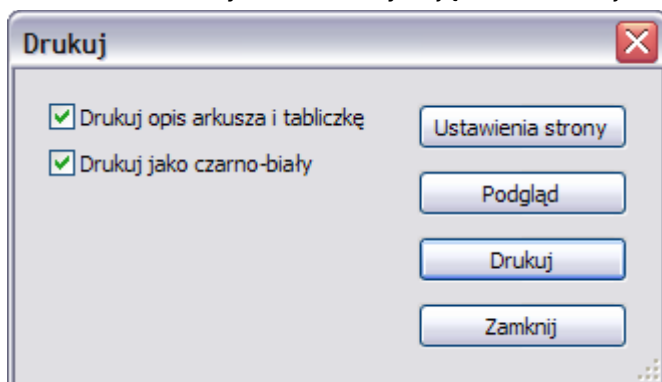


Pozwala na utworzenie plików z rysunkami CAD używając popularnego formatu DXF. Nazwa pliku wyjściowego składała się będzie z nazwy arkusza i rozszerzenia .dxf.

10.7. Drukuj



To polecenie, podobne do polecenia  na głównym pasku narzędzi, pozwala na podgląd oraz utworzenie wydruków korzystając z normalnej drukarki.



Pierwsza opcja *Drukuj opis arkusza i tabliczkę* pozwala na wydrukowanie także odnośników arkuszy oraz tabliczki która znajduje się w prawym dolnym rogu. Opcja *Drukuj jako czarno-biały* wymusza zaś wydruk monochromatyczny.

Opcja ta zwykle jest stosowana, gdy do wydruków jest używana laserowa drukarka monochromatyczna, ponieważ większość drukarek dla jasnych kolorów korzysta z dość nieczytelnej symulacji pół-tonalnej. Stąd też połączenia, rysowane kolorem zielonym, mogłyby stać się mało widoczne.

11. Edytor bibliotek LibEdit - Podstawy

11.1. Podstawowe informacje na temat bibliotek


11.1.1. Czym jest biblioteka symboli ?

Wszystkie symbole używane na schematach są opisane w bibliotekach symboli. Aby umożliwić stosunkowo proste zarządzanie tymi składnikami, używa się kilku bibliotek, z symbolami pogrupowanymi według jednego klucza (wg funkcji lub producentów). Polecenia służące do zarządzania bibliotekami symboli umożliwiają tworzenie bibliotek, dodawanie, usuwanie lub też przenoszenie symboli. Oczywiście pozwala również na szybką wizualizację elementów z biblioteki.

11.1.2. Polecenia związane z bibliotekami

Istnieją dwa polecenia uruchamiające narzędzia związane z bibliotekami:

- ♦ **ViewLib** który pozwala wyłącznie na przegląd symboli, ale umożliwia za to szybki

dostęp do nich. Uruchamiany jest poleceniem . Jego opis zostanie przedstawiony w jednym z następnych rozdziałów.

- ♦ **LibEdit** który pozwala na pełne zarządzanie zawartością bibliotek. Uruchamiane

poprzez .

11.2. Elementy składowe symboli

Symbol znajdujący się w bibliotece **jest złożony z:**

- ♦ Jego postaci graficznej (**linie, okręgi, pola tekstowe**).
- ♦ **Wyprowadzeń**, inaczej *pinów* (które muszą być narysowane w ściśle określony sposób używając powszechnie przyjętych standardów (zwykły pin, lub wejście zegarowe, lub aktywny niskim poziomem, albo aktywny zboczem...) opisując ich właściwości elektryczne, używane przez ERC).
- ♦ Pola (tekstowe) takie jak **oznaczenie, wartość, nazwa modułu** potrzebna do wstawienia go na płytkę...

Symbol może posiadać również **aliasy**, czyli nazwy alternatywne (na przykład 7400 może także występować jako 74LS00, 74HC00, 7437, ponieważ wszystkie te symbole są identyczne z punktu widzenia schematu).

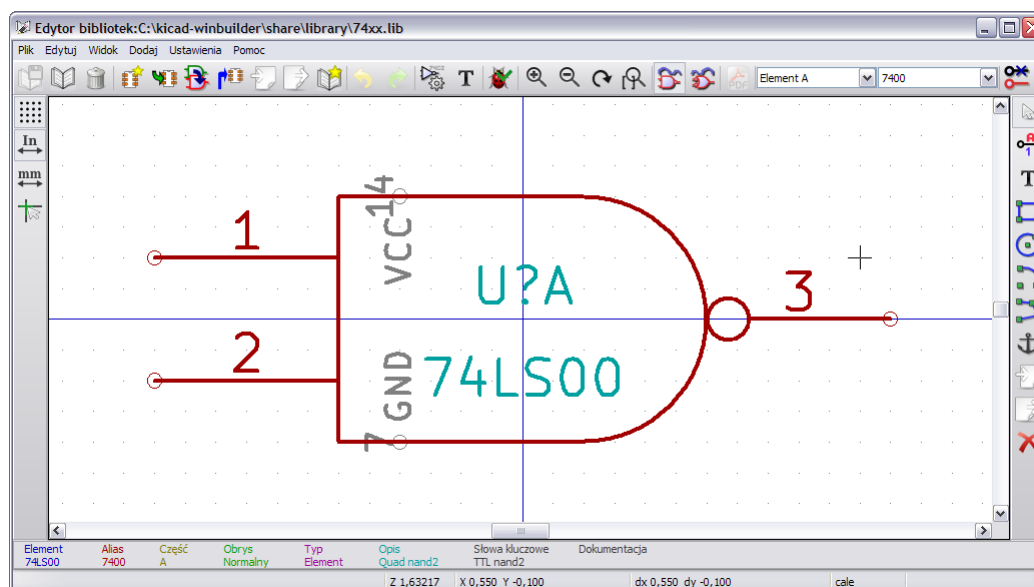
Korzystanie z aliasów jest bardzo ciekawą metodą tworzenia kompletnych, ale zwartych i (względnie) szybko budowanych bibliotek.

Tworzenie symboli to:

1. Definiowanie ogólnych właściwości: czy posiada wiele części składowych i ile ich jest, czy posiada podwójną reprezentację (znany jako DeMorgan, a w **Eeschema** reprezentowany normalnie i jako skonwertowany).
2. Projektowanie wyglądu (z wyjątkiem pinów) z użyciem linii, prostokątów, okręgów, wielokątów i tekstów.
3. Dodawanie wyprowadzeń, dokładnie określając ich projekt graficzny, nazwę oraz numer pinu, a także ich właściwości elektryczne (wejście, wyjście, trzy-stanowe, port zasilania...).
4. Dodawanie aliasów, jeśli inne komponenty są tej samej konstrukcji i mają ten sam zestaw wyprowadzeń (lub też usunięcie jednego w przypadku, gdy symbol został stworzony przez skopiowanie innego symbolu).
5. Dodawanie pól w razie potrzeby (jest to opcjonalne, nazwa modułu jest wykorzystywany przez oprogramowanie PCB) i/lub określenie ich widoczności.
6. Dokumentowanie składnika np. poprzez dodanie słów kluczowych i adresu dokumentacji w sieci lub na lokalnym nośniku.
7. Zapisując go do wybranej biblioteki.

11.3. Odczyt symbolu w celu edycji

Klikając w ikonę  otworzy się okno **LibEdit**, w którym będzie można dokonać edycji symboli. Okno **LibEdit** ma postać:



11.3.1. Główny pasek menu



Zapisuje bibliotekę na dysku.



Wybór biblioteki aktywnej w danej chwili



Usunięcie symbolu z aktywnej biblioteki



Stworzenie nowego symbolu od podstaw



Odczyt symbolu z aktywnej biblioteki w celu edycji



Tworzenie nowego symbolu na podstawie symbolu aktualnie załadowanego.



Zapisanie bieżącego symbolu w aktywnej bibliotece w buforze pamięci RAM. Plik biblioteki na dysku nie jest modyfikowany.



Importowanie symboli.



Eksportowanie symboli.



Tworzenie nowej biblioteki z bieżącym symbolem w polu edycyjnym.



Przywracanie / wykonywanie ponowne poleceń



Edycja właściwości symboli.



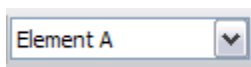
Edycja pól symbolu: nazwa na schemacie, wartość/nazwa pod jaką symbol będzie zapisany w bibliotece, pola użytkownika



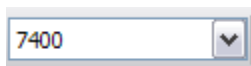
Przełączanie reprezentacji: Normalny lub skonwertowany (De Morgana)



Pokazuje dokumentację elementu (jeśli taka została określona)



Wybór elementu składowego (dla symboli wieloczęściowych)



Wybór aliasu (jeśli symbol je posiada)



Edycja pinów: niezależna edycja kształtów i pozycji pinów (dla symboli wieloczęściowych i reprezentacji De Morgana)

11.3.2. Wybór biblioteki



Wybranie bieżącej biblioteki jest możliwe za pomocą ikony , która otwiera okno z listą dostępnych bibliotek.

Gdy element jest ładowany lub zapisywany, to wszystkie te operacje będą przeprowadzane w tej bibliotece.

Uwaga:

Należy wcześniej określić nazwy bibliotek dostępnych dla **Eeschema**, by móc je wybierać w ten sposób.



Bieżąca biblioteka może zostać zapisana po modyfikacjach klikając w ikonę .



Symbol może zostać usunięty z biblioteki poprzez kliknięcie w ikonę .

11.3.3. Wybór symbolu i sposoby jego zapisu

Podczas edycji symbolu, w rzeczywistości wszelkie zmiany nie są dokonywane bezpośrednio w bibliotece, ale w jej kopii w pamięci RAM. W ten sposób można z łatwością cofnąć wszelkie zmiany od ostatniego zapisu.

Symbol może pochodzić z biblioteki lub z poprzedniego symbolu ze schematu. Po załadowaniu będzie on wyświetlony.

11.3.3.1. Wybór symbolu

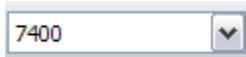


Ikona pozwala wyświetlić listę dostępnych symboli, by wybrać jeden i załadować potrzebny element.

Uwaga 1:

Jeśli wybrano **alias** symbolu, zostanie załadowany **symbol główny** do którego się on odnosi (**Eeschema** zawsze wyświetla nazwę symbolu, który w rzeczywistości został załadowany).

- Lista aliasów zawsze zostaje załadowana dla każdego symbolu, wobec czego można ją edytować.
- Gdy chcemy edytować jeden alias, musi on zostać wybrany z listy w oknie



narzędziowym:

Pierwszy element listy jest symbolem głównym.

Uwaga 2:




Alternatywnie, można użyć polecenia **Import** (), które pozwala na załadowanie symbolu



jaki został wcześniej zapisany poleceniem **Eksport** ().

11.3.3.2. Zapis symbolu

Po modyfikacji, symbol może zostać zapisany w bieżącej bibliotece, nowej bibliotece, lub wyeksportowany do pliku kopii zapasowej.

By umieścić symbol w bieżącej bibliotece, należy użyć polecenia **Zaktualizuj element** (). Jednakże aktualizacja symbolu zostanie przeprowadzona **tylko do pamięci RAM** (W ten sposób można zdecydować się czy symbol pasuje do schematu).


Jeśli zajdzie potrzeba pełnego zapisu symbolu, należy użyć polecenia ukrytego pod ikoną



, które zmodyfikuje zawartość biblioteki na dysku twardym.


Jeśli chcemy by aktualny komponent trafił do nowej biblioteki, należy użyć polecenia *Utwórz*



nową bibliotekę (). W takim przypadku program poprosi o nazwę nowej biblioteki.

Uwaga:






Nowe elementy w nowych bibliotekach będą widoczne **dopiero po** zmianie konfiguracji bibliotek w programie **Eeschema**.

Kończąc, można użyć polecenia **Eksport** () by utworzyć plik zawierający tylko ten symbol (jest to standardowy plik biblioteki, lecz zawierający tylko jeden element).

W praktyce polecenia **Utwórz nową bibliotekę** i **Eksport** są identyczne. Pierwsze z nich proponuje jednak utworzenie biblioteki w domyślnym folderze z bibliotekami, drugie zaś w katalogu użytkownika.

11.3.3.3. Przenoszenie symboli do innych bibliotek

Można bardzo łatwo przenosić symbole z jednej biblioteki do drugiej, używając tego schematu postępowania :

- ◆ Wybrać bibliotekę źródłową jako bieżącą .
- ◆ Załadować symbol, który chcemy przenieść .
- ◆ Wybrać bibliotekę docelową jako bieżącą .
- ◆ Zapisać symbol do jej kopii w pamięci RAM .
- ◆ Zapisać zmodyfikowaną bibliotekę .

11.3.3.4. Zaniechanie edycji symbolu

Symbol poddawany edycji jest tylko kopią roboczą symbolu, jaki rzeczywiście znajduje się w bibliotece. Tak długo jak nie zostanie on zapisany do pamięci RAM, można go przywrócić z biblioteki (lub przeładować z innej biblioteki) by porzucić zmiany dokonane w tym symbolu.

Jeśli symbol jest już zapisany w pamięci RAM, a nie został zapisany w pliku biblioteki na dysku, można zamknąć i ponownie uruchomić **Eeschema**, a następnie odczytać go z biblioteki ponownie.

11.4. Tworzenie symboli

11.4.1. Tworzenie nowego symbolu

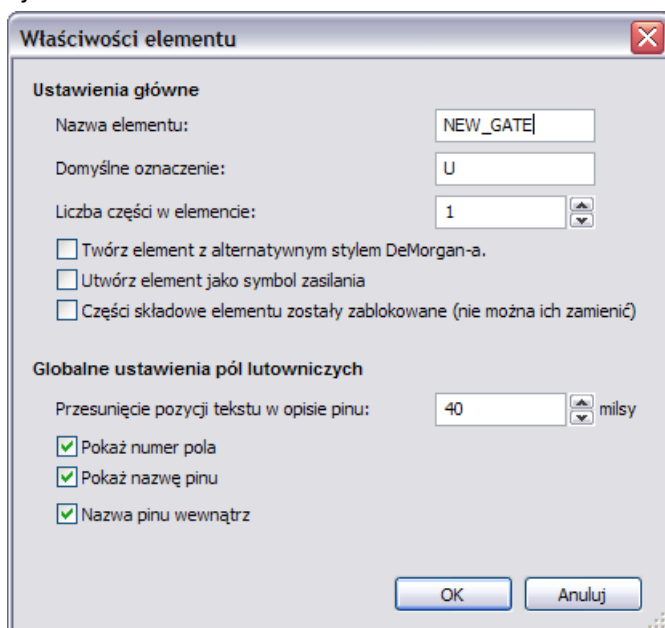
Nowe symbole mogą być tworzone za pomocą polecenia **Utwórz nowy element** ().

Program poprosi o podanie : nazwy symbolu, aby móc potem załadować go z biblioteki (nazwa ta jest także zawartością pola Wartość dla **LibEdit** i używana jako wartość domyślna dla pola **Wartość** w edytorze schematów), nazwy domyślnego oznaczenia na schemacie (U, IC, R...), liczby elementów w pakiecie (np. standardowy komponent 7400 posiada 4 części w

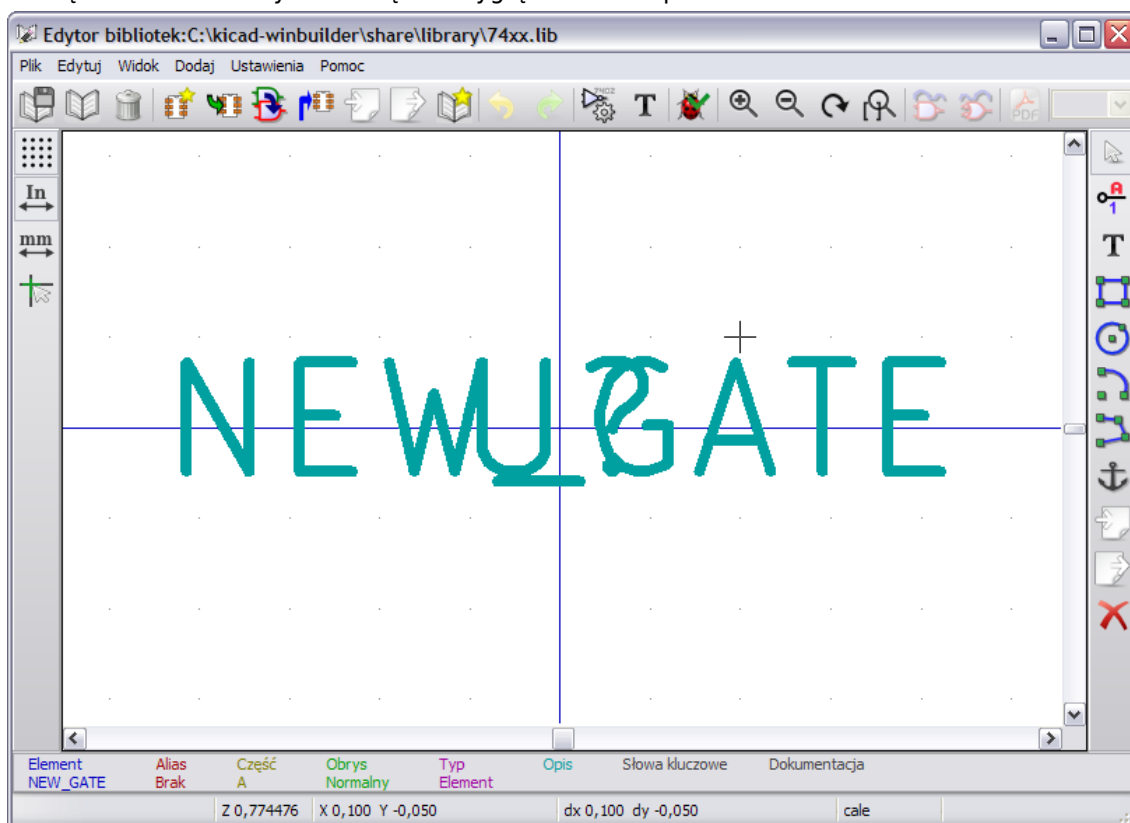
jednej obudowie) i czy istnieje przekształcona reprezentacja tego symbolu (standardowo De Morgan).

Jeśli nazwa odnośnika będzie pusta, domyślnie zostanie wpisane U.

Wszystkie te dane mogą być ustalone później, ale lepiej jest ustawić je na początku tworzenia symbolu.



Początkowe stadium symbolu będzie wyglądać w ten sposób :



11.4.2. Tworzenie nowego symbolu na podstawie innego



Jeśli nowy symbol bardzo przypomina inny symbol, częściej korzystniej jest załadować ten inny symbol i go zmodyfikować. Aby to zrobić należy :

- ◆ Załadować symbol który będzie użyty jako wzorcowy.



- ◆ Kliknąć polecenie lub zmodyfikować jego nazwę (klikając prawym klawiszem na nazwie i poddać edycji tekst pola *Wartość*).
Użytkownik zostanie poproszony o nazwę nowego symbolu.

- ◆ Jeśli symbol wzorcowy posiadał **aliasy**, użytkownik zostanie poproszony o ich usunięcie z nowego symbolu, gdyż pozostawienie ich spowoduje konflikt. Jeśli podczas tego pytania odpowiemy „Nie”, tworzenie nowego symbolu zostanie zaniechane.
- ◆ Zmodyfikować nazwę symbolu.
- ◆ Wykonać niezbędne edycje.

- ◆ Zapisać symbol w załadowanej bibliotece poprzez  lub zapisać go do nowej biblioteki z pomocą polecenia  lub (jeśli chcemy zapisać nowy element do innej istniejącej biblioteki) wybrać inną bibliotekę i zapisać nowy symbol.

- ◆ Zapisać bibliotekę na dysk poleceniem Zapisz bieżącą bibliotekę na dysk. 

11.4.3. Edycja głównych właściwości symboli

Główne właściwości symboli to przede wszystkim :

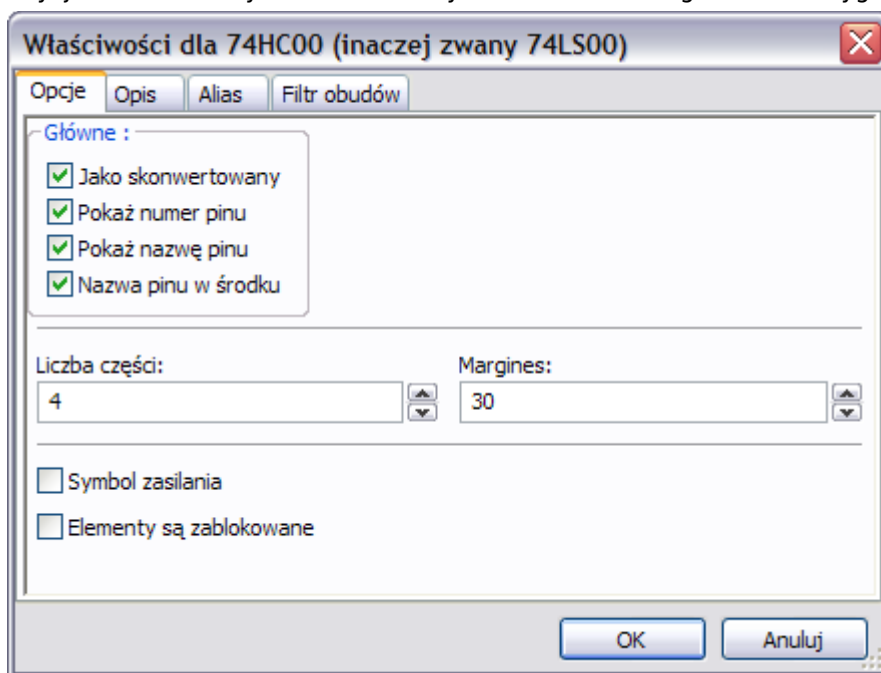
- ◆ Liczba elementów w pakiecie.
- ◆ Obecność reprezentacji alternatywnej.
- ◆ Powiązania z dokumentacją.
- ◆ Aktualizacja różnych pól.

Te cechy powinny być ustalone poprawnie, ponieważ są one wymagane podczas tworzenia symboli lub mogą pochodzić z symbolu wzorcowego (przy tworzeniu symboli na podstawie innych symboli) i nie muszą być z nim zgodne.

Zatem, jeśli w jakikolwiek sposób są one modyfikowane, konieczne jest wywołanie polecenia



edycji właściwości symbolu . Wywoła to okno dialogowe które wygląda w ten sposób :



Najważniejsze opcje, które określają ogólne właściwości to:

- *Liczba części* określająca liczbę elementów w pakiecie.
- *Jako skonwertowany*: zaznaczone, jeśli składnik ma podwójną reprezentację.

Ważne jest, aby te dwa parametry były ustawione prawidłowo, ponieważ gdy są tworzone i edytowane wyprowadzenia, odpowiednie piny wszystkich części są publikowane lub tworzone razem.

Jeśli liczba części po utworzeniu pinu/edycji zostanie zwiększona, trzeba będzie wykonać kilka dodatkowych modyfikacji wynikających z tej zmiany. Niemniej jednak istnieje możliwość modyfikacji tych parametrów w każdej chwili.

Opcje grafiki:

- *Pokaż numer pinu*

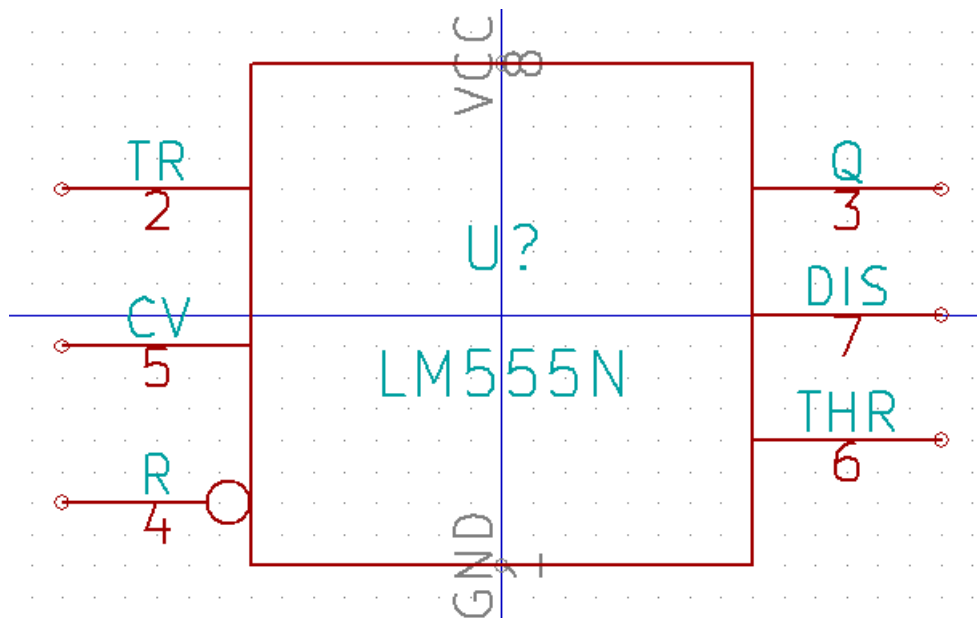
- *Pokaż nazwę pinu*

definiują widoczność numeru pinu i jego nazwy (teksty te będą dostępne, jeśli odpowiednie opcje są aktywne).

Opcja *Nazwa pinu w środku* określa położenie nazwy pinu: tekst ten będzie wyświetlany wewnątrz konturu elementu, jeśli opcja ta jest aktywna.

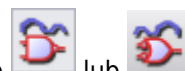
W tym przypadku parametr *Margines* określa przesunięcie tekstu w kierunku wnętrza konturu. Wartości od 30 do 40 (w jednostkach 1/1000 cala) są rozsądnymi wartościami.

Poniższy przykład pokazuje symbol, w którym opcja *Nazwa pinu w środku* została odznaczona (należy zwrócić uwagę na położenie nazw i numerów pinów):

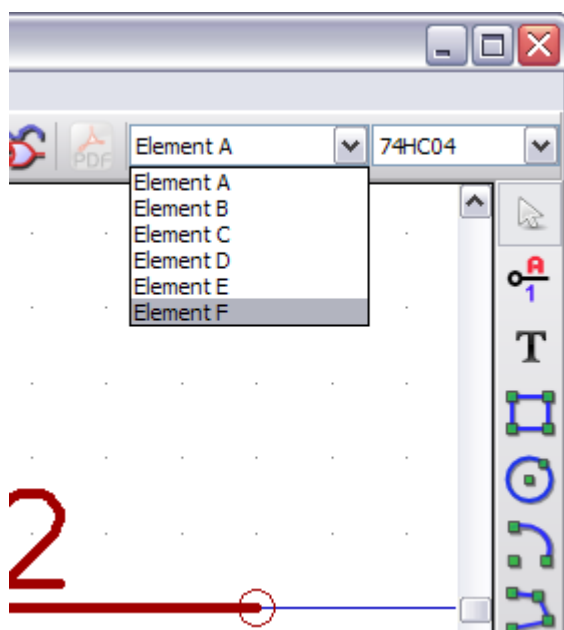


11.4.4. Edycja symboli wieloczęściowych

Podczas edycji składników symboli w przypadku gdy symbol posiada więcej części składowych lub podwójną reprezentację, zachodzi potrzeba wyboru części składowych lub innej reprezentacji takiego symbolu.



Aby wybrać jedną z alternatywnych reprezentacji należy kliknąć ikonę lub .
Aby wybrać jeden z elementów składowych należy użyć rozwijanej listy na pasku narzędzi :



11.5. Tworzenie postaci graficznej symbolu

Pasek narzędzi znajdujący się po prawej stronie pozwala na stworzenie graficznej postaci symbolu :



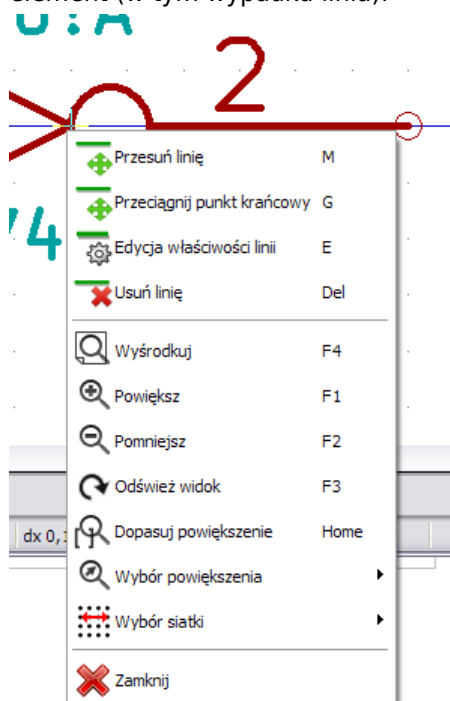
Aby narysować graficzną postać symbolu można użyć następujących elementów graficznych :

- Linie (oraz wielokąty, proste lub wypełnione).
- Prostokąty
- Okręgi
- Łuki.
- Teksty (inne niż zawartość pól elementu lub opisy pinów).

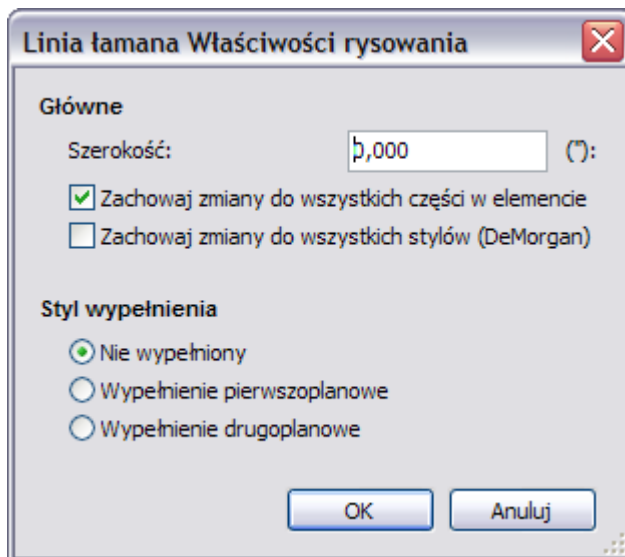
Piny jak i pola (wartość, oznaczenie) są traktowane w inny sposób, ponieważ nie są one elementami czysto graficznymi.

11.5.1.Elementy graficzne symbolu

Każdy element graficzny może być zdefiniowany jako zwyczajny lub specyficzny, w zależności od typu reprezentacji (normalna lub skonwertowana) czy ilości części składowych. Podręczne menu z opcjami jest dostępne przez kliknięcie prawym przyciskiem na dany element (w tym wypadku linia):



Można także kliknąć dwukrotnie na dany element aby wywołać okno z jego właściwościami :



Normalnie opcje głównych elementów graficznych mają postać :

- ♦ *Zachowaj zmiany do wszystkich części w elemencie* jest **zaznaczone**, ponieważ na ogół różne części komponentu mają takie same reprezentacje graficzne i wystarczające jest narysowanie jednej części składowej.
- ♦ *Zachowaj zmiany do wszystkich stylów (DeMorgan)* jest **odznaczone**, ponieważ podwójna reprezentacja została wprowadzona, aby móc wybrać inną reprezentację dla każdego rodzaju reprezentacji.

Wtedy będzie konieczne sporządzenie każdej reprezentacji graficznej.

Można sprawę wielu części składowych potraktować w ten sposób (na szczęście rzadko), projektując ich różne typy graficzne, poprzez zaznaczenie opcji *Części szczegółowe*. Każda część będzie musiała być wtedy narysowana osobno, a jeśli opcja *Specyficzne dla reprezentacji* będzie zaznaczona, dla każdej części trzeba będzie narysować również dwie reprezentacje. Wreszcie, interesujące może być zaznaczenie opcji *Wspólna dla reprezentacji* dla elementów narysowanych z uwzględnieniem standardów IEEE, ponieważ właściwe grafiki są identyczne w reprezentacji normalnej i skonwertowanej.

Dla elementów typu **wielokąt** (linie łamane) opcje *Styl wypełnienia* pozwalają także na wygenerowanie wypełnionego wielokąta.

11.5.2.Elementy geometryczne w symbolach

Tworzenie elementów geometrycznych jest możliwe dzięki narzędziom do rysowania :

- ♦ **Linie i wielokąty**, proste lub wypełnione jeśli odpowiednie opcje są zaznaczone.
- ♦ **Prostokąty** definiowane za pomocą dwóch punktów po przekątnej.
- ♦ **Okręgi** definiowane poprzez punkt środkowy i promień.
- ♦ **Łuki** definiowane za pomocą punktów początkowych i końcowych oraz punktu centralnego. Łuki zawsze składają się z półokręgu (od 0 do 180st.).

11.5.3.Tekst jako grafika w symbolu

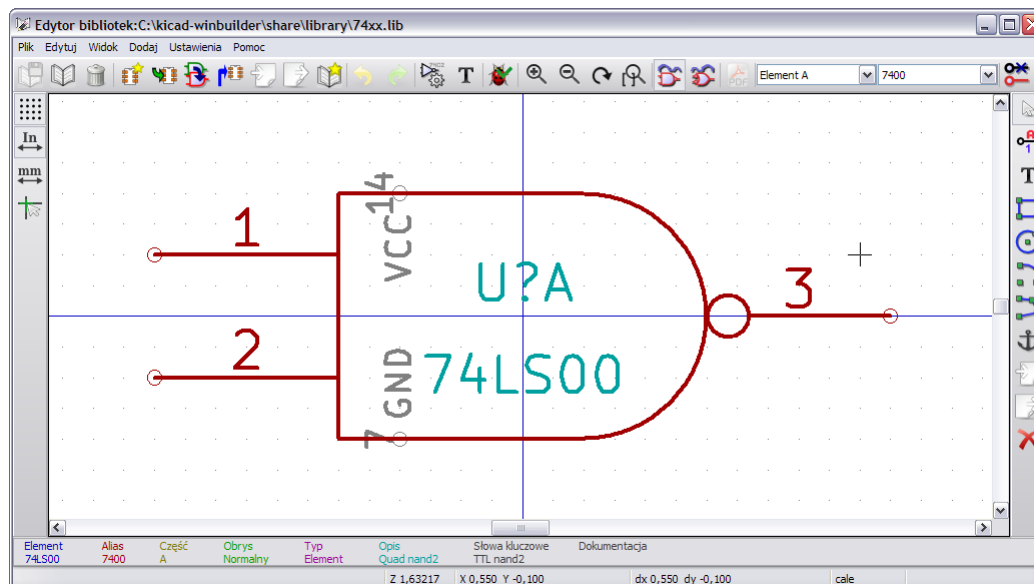



Pozwala na tworzenie tekstów (tekst swobodny). Taki tekst jest zawsze wyświetlany poprawnie, nawet jeśli symbol zostanie odwrócony.

11.6. Ustalanie punktu zaczepienia symbolu

Punkt zaczepienia elementu wyznacza punkt zerowy układu współrzędnych w edytorze bibliotek oraz punkt za który element będzie przeciągany podczas wstawiania go do schematu. Obrót lub odbicie lustrzane jest również wykonywane w stosunku do tego punktu zaczepienia, a podczas wstawiania, punkt ten jest używany jako punkt odniesienia. Zalecane jest zatem ustalenie jego pozycji dość dokładnie.

Punkt zaczepienia jest wyświetlany jako dwie skrzyżowane niebieskie linie:

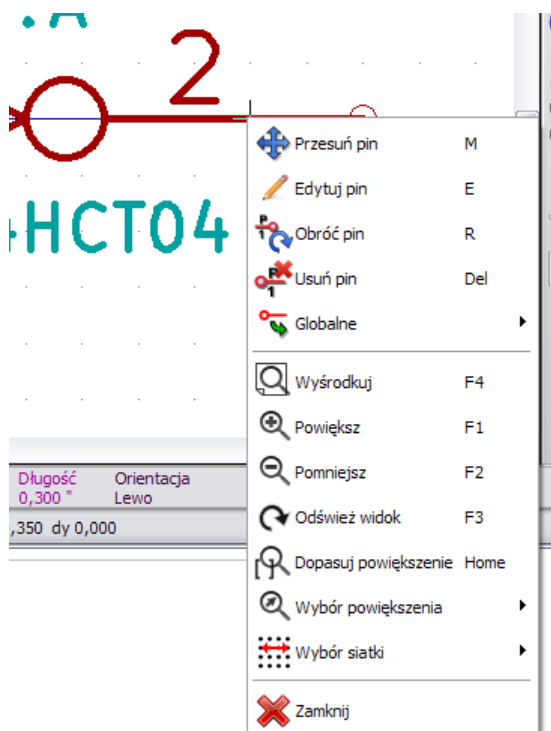


Punkt zaczepienia może zostać przemieszczony poprzez wybranie narzędzia , po czym należy kliknąć w miejscu nowego punktu zaczepienia. Rysunek zostanie automatycznie wycentrowany ponownie wokół nowego punktu zaczepienia.

11.7. Tworzenie i edycja wyprowadzeń (pinów)

Aby utworzyć pin należy kliknąć w polecenie **Dodaj wyprowadzenie** . Otworzy się okno dialogowe z opcjami, których dokładniejszy opis będzie przedstawiony później.

Aby dokonać prostych edycji wyprowadzeń najlepiej jest kliknąć dwukrotnie na wybranym wyprowadzeniu. Można także skorzystać z menu podręcznego wywoływanego przez kliknięcie prawym klawiszem :



Piny muszą być tworzone starannie, ponieważ każdy błąd będzie mieć wpływ na projekt PCB, lub funkcja sprawdzania poprawności elektrycznej schematu ERC będzie działać nieefektywnie.

Każdy dodany już pin można ponownie edytować, usunąć, obrócić lub przenieść. Najłatwiej jest to robić w wykorzystaniem skrótów klawiszowych.

11.7.1. Wyprowadzenia - Informacje podstawowe

Pin jest zdefiniowany przez swoją formę (długość, postać graficzną), jego nazwę i "liczbę", która nie zawsze jest liczbą (piny w obudowach PGA/BGA są definiowane za pomocą liter i liczb, jak na przykład : A12 czy AB45).

W **Eeschema**, "numer pinu" jest definiowany za pomocą zestawu 4 liter lub cyfr, także mieszanych.

Dla narzędzia kontroli ERC, musi zostać definiowany **typ "elektryczny"** każdego wyprowadzenia (wejście, wyjście, trój-stanowy...). Jeśli typ nie będzie dobrze zdefiniowany, kontrola ERC będzie nieefektywna.

Uwaga:

- **Nie należy wstawiać znaków spacji** w nazwach pinów i ich numeracji. Wstawienie spacji w nazwie spowoduje, że zostanie ona automatycznie zastąpiona przez znak dolnej kreski „_”.
- Nazwa pinu z sygnałem zanegowanym rozpoczyna się od znaku tyldy „~”. Wtedy nad tekstem zostanie dorysowana pozioma kreska. Znak ten działa jak przełącznik, zatem możliwe jest również stosowanie go tylko dla wybranego fragmentu nazwy, np. PA0/~INT0~/PCI0 (zanegowany będzie tylko INT0).
- Jeśli nie ma nazwy pinu to w pole nazwa należy wstawić jeden znak tyldy.
- Nazwy pinów rozpoczynające się od znaku *hash* „#”, są zarezerwowane dla symboli zasilania.
- Numery pinów mogą składać się z maksymalnie czterech liter lub liczb (również mieszanie) 1, 2, ... 9999 to prawidłowe nazwy, ale prawidłowymi nazwami będą A1, B3... (standardowa notacja w układach PGA/BGA), lub Anod, Gnd, Wine...

11.7.2. Symbole wieloczęściowe, podwójna reprezentacja symboli

Przypomnijmy, że szczególnie w przypadku bramek logicznych, symbol może mieć dwie reprezentacje (znane jako "De Morgan", a i taki układ scalony może zawierać kilka części, np. kilka bramek NOR). Dla niektórych układów scalonych, może być zatem potrzebne kilka różnych elementów graficznych i wyprowadzeń.

Na przykład przełącznik może być reprezentowane przez trzy różne elementy:

1. Cewka
2. Pierwszy zestyk
3. Drugi zestyk

Zarządzanie wieloczęściowymi układami scalonymi i komponentami z podwójną reprezentacją jest elastyczne.

Rzeczywiście, pin może być:

- ♦ Wspólny lub specyficzny dla różnych części.
- ♦ Wspólny dla obu reprezentacji lub specyficzny dla każdej reprezentacji z osobna.

Domyślnie, piny są specyficzne dla każdej reprezentacji każdej części, ponieważ ich liczba różni się dla każdej części, a ich konstrukcja jest inna dla każdej reprezentacji.

Gdy pin jest wspólny wystarczy wyprowadzić go tylko raz (np. w przypadku pinów zasilania).

Również w przypadku projektu, który jest prawie zawsze identyczny dla każdej części (ale różni się pomiędzy normalną a skonwertowaną reprezentacją).

11.7.3. Wyprowadzenia - Opcje podstawowe

Symbole z wieloma częściami i/lub reprezentacjami stanowią szczególny problem dla tworzenia pinów i ich edycji.

O ile większość z pinów jest specyficzna dla każdej części (bo ich numer pinu jest specyficzny dla każdej części) i do każdej reprezentacji (bo ich forma jest specyficzna dla każdej z reprezentacji), tworzenie i edycja pinów byłaby prawdopodobnie długa i męcząca.

W rzeczywistości, **Eeschema** umożliwia jednocześnie obsługiwanie pinów:

- ♦ Domyślnie, dla symboli z wieloma częściami i/lub reprezentacją podwójną, zmiany te są wykonywane dla wszystkich pinów odpowiadających częściom i reprezentacjom podczas tworzenia, edycji (z wyjątkiem formy i numeracji), usuwania lub przenoszenia pinu, (tj. **dla wszystkich pinów umieszczonych w tej samej lokacji**).
- ♦ W przypadku projektu, dokonane zmiany w bieżącej reprezentacji, są wykonywane dla wszystkich części.
- ♦ Numery pinów są modyfikowane dla bieżącego elementu, dla dwóch reprezentacji.

♦ Nazwy są modyfikowane niezależnie od siebie.
Zależność ta powstała by umożliwić szybkie zmiany dla większości przypadków.
Zależność tą podczas modyfikacji można wyłączyć z poziomu menu **Opcje**, pozwalając na tworzenie cech symboli z wieloma częściami i podwójną reprezentacją całkowicie niezależnie.

Opcja ta jest zarządzana również z poziomu paska narzędzi :



- ♦ Jeśli przełącznik **nie jest aktywny (nie podświetlony)**: edycja jest stosowana **do wszystkich** części składowych i wszystkich reprezentacji. **Jest to opcja domyślna.**



- ♦ Jeśli przełącznik **jest aktywny (podświetlony)**: edycja jest stosowana **tylko dla bieżącej** części i w bieżącej reprezentacji (czyli tylko w tej części widocznej na ekranie). Ta opcja jest rzadziej używana.

11.7.4. Wyprowadzenia - Określanie właściwości

Okno z właściwościami pinu pozwala na zmiany charakterystycznych cech wyprowadzeń.

Właściwości pinu

Nazwa pinu: ~ Rozmiar nazwy: 0,060 cale

Numer pinu: 1 Rozmiar numeru pinu: 0,060 cale

Orientacja: Prawo Długość: 0,300 cale

Typ elektryczny: Wejście

Styl grafiki: Linia

Piny Współdzielone

☐ Dodaj do wszystkich części w paczce

☐ Dodaj do wszystkich alternatywnych stylów (DeMorgan)

Właściwości schematu

☒ Widoczny

OK Anuluj

Okno to ukazuje się zawsze podczas tworzenia nowego pinu, albo gdy w pin zostanie kliknięty dwukrotnie myszą. Jego zawartość pozwala na zdefiniowanie lub modyfikację parametrów takich jak :

- ♦ **Nazwa** oraz rozmiar nazwy pinu.
- ♦ **Numer** oraz rozmiar numeru pinu.
- ♦ **Długość pinu** (jego postać graficzna).
- ♦ **Typ elektryczny** i formę jego prezentacji.
- ♦ Jego przynależność (wspólny lub normalny, w reprezentacji DeMorgan czy też nie)
- ♦ Widoczność pinu (używana przy pinach zasilania).

Pamiętajmy:

- Nazwa pinu rozpoczynająca się od znaku tyldy „~” oznacza negację nazwy (kreska nad nazwą).
- Przy nazwie składającej się wyłącznie ze znaku tyldy, oznacza to, że pin nie posiada nazwy i nie jest ona wyświetlana. Przydatne w elementach pasywnych.
- Numer pinu składa się z 1 do 4 znaków (cyfry lub litery, albo łączone). Zatem 1, 2, ... 9999 to nazwy poprawne, ale też A1, B3... (standardowa notacja w PGA/BGA), Anod, Gnd, V.in ...

11.7.4.1. Kształty wyprowadzeń

Na poniższym rysunku w jednym elemencie zastosowano wszystkie rodzaje kształtów wyprowadzeń :



Wybór formy ma wyłącznie znaczenie graficzne i nie ma żadnego znaczenia dla narzędzia ERC jak i dla funkcji związanych z tworzeniem list sieci.

11.7.4.2. Typy elektryczne

Wybór elektrycznego typu pinu jest wysoce istotny z punktu widzenia kontroli poprawności schematu ERC.

Zazwyczaj będzie to wybór pomiędzy **Wejściem** a **Wyjściem** w przypadku układów scalonych. Jednakże dostępne są także inne typy :

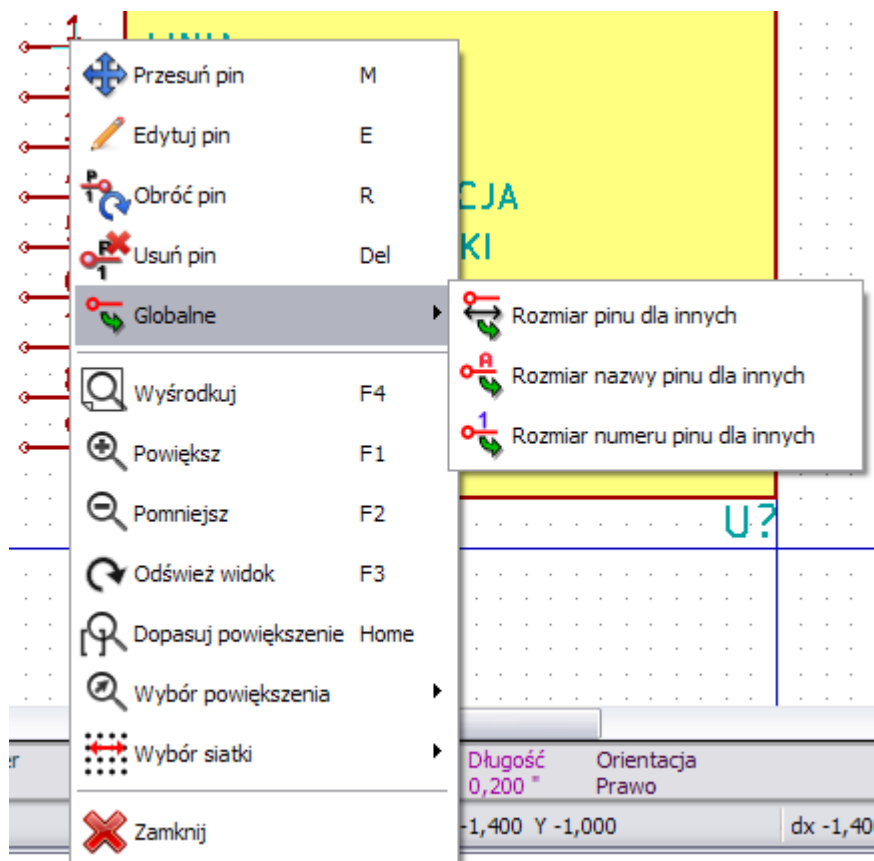
- **Typ dwukierunkowy** oznacza, że pin może być zarówno wejściem jak i wyjściem (jak w współczesnych mikroprocesorach czy układach FPGA).
- **Typ trój-stanowy** oznacza, że pin może przechodzić jako wyjście w trzeci stan logiczny Z (wysoka impedancja).
- **Typ Pasywny** używany jest dla elementów pasywnych takich jak wyprowadzenia rezystorów, złącz itp.
- **Typ Niezdefiniowany** może być używany gdy testowanie poprawności ERC nie ma zwracać uwagi na ten pin.
- **Typ Wejście zasilania** jest używany dla pinów zasilania znajdujących się w symbolu.
- **Typ Wyjście zasilania** jest przeznaczony dla wyjść regulatorów napięć jako źródło napięcia.

W szczególności, **jeśli pin jest portem zasilania (Wejście zasilania lub Wyjście zasilania, i jest zadeklarowany jako "Niewidoczny", nie jest on wyświetlany na schemacie, i jest automatycznie podłączony do pozostałych pinów tego samego rodzaju i tej samej nazwie (tzw. Niewidoczne piny zasilania)**

- Dodatkowo można używać typów **Otwarty Emiter** lub **Otwarty kolektor**. Rzadko spotykane opcje.

11.7.5. Wyprowadzenia - Zmiany globalne


W przypadku wielokrotnych zmian jednego typu z parametrów : rozmiar pinu, rozmiar jego nazwy, czy numeru pinu, można użyć menu podręcznego, a w nim opcji zgrupowanych pod poleceniem **Globalne**. Za pomocą tych opcji można jedną z tych cech skopiować do innych pinów na podstawie pinu wskazywanego w danej chwili.



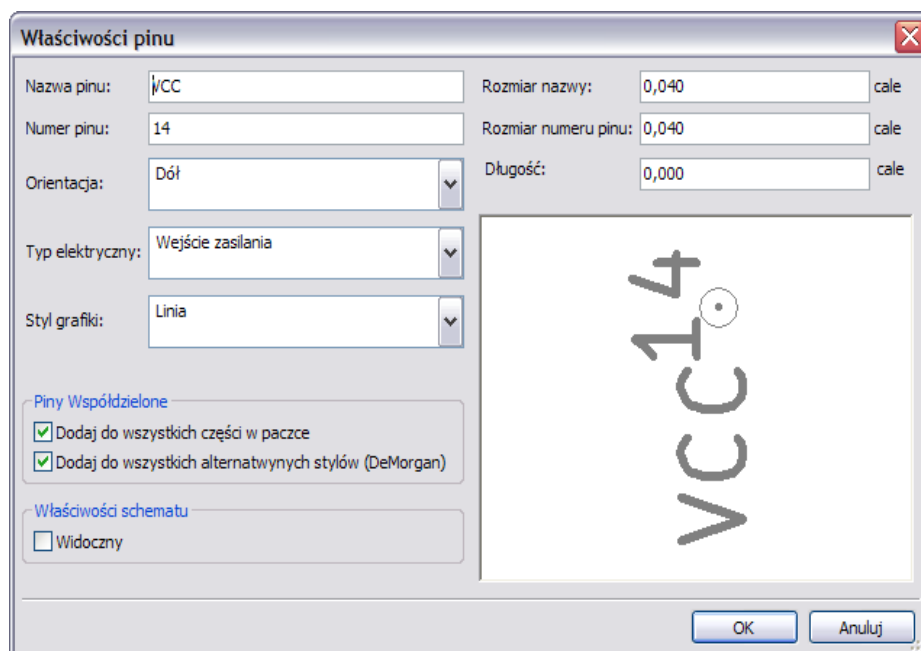
Jeśli zatem w elemencie istnieją piny o różnych długościach, a chcielibyśmy to ujednolicić, to zamiast kolejno zmieniać właściwości pinów, można wybrać jeden z pinów wzorcowych i wybrać **Rozmiar pinu dla innych**, co spowoduje, że reszta pinów też będzie miała taką samą długość.

11.7.6. Wyprowadzenia - Symbole wieloczęściowe I podwójna reprezentacja

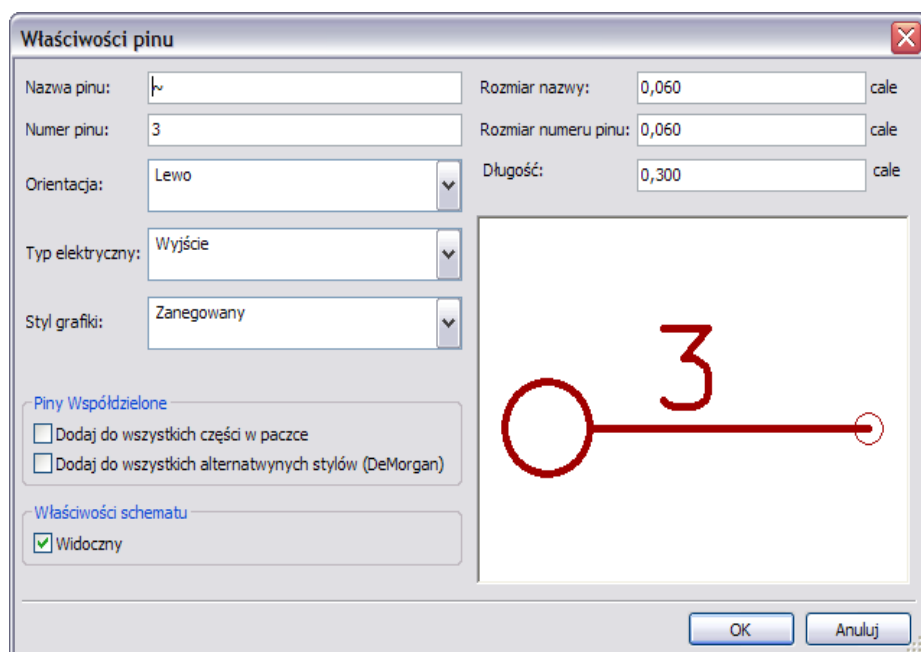
Różne części lub reprezentacje (takie jak spotyka się w układach 7400, 7402 ...) mogą potrzebować pewnych uzupełnień. Takie uzupełnianie będzie znacznie ograniczone w przypadku zachowania następujących środków ostrożności:

- Główna opcja *Edytuj piny element po elemencie*  musi **pozostać odznaczona**.
- Piny zasilania zostaną utworzone z aktywnymi atrybutami *Dodaj do wszystkich części w paczce* i *Dodaj do wszystkich alternatywnych stylów* (mogą być również niewidoczne (odznaczone pole *Widoczne*)).

Poprawne ustawienia wyglądają tak jak na poniższym obrazku :



Gdy piny zostaną utworzone, będą utworzone dla każdego elementu i każdej reprezentacji.



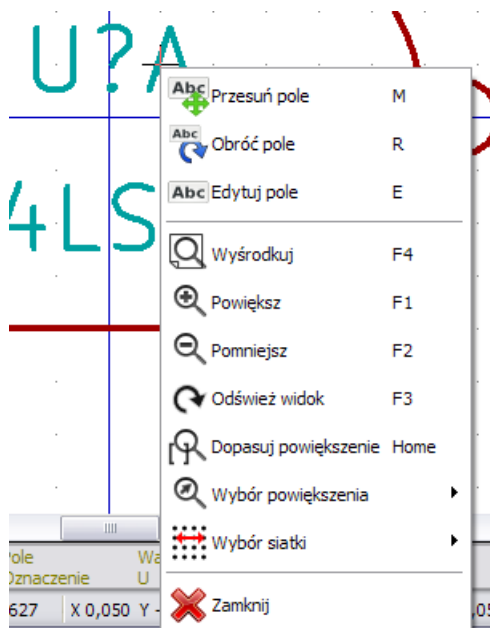
Na przykład wyjście bramki A w układzie 7400 zostanie utworzone przez **Eeschema** w 8 próbkach: 2 na część (są 4 części A, B, C, D i dla każdej części, reprezentacja normalna i De Morgan).


Jednakże będzie trzeba na początku, prawdopodobnie poprawnie stworzyć część A w normalnej reprezentacji. Z tego wynika, że dla każdej części będzie konieczne:

- ◆ Wybranie reprezentacji skonwertowanej i dokonanie edycji formy i długości każdego z pinów.
- ◆ W przypadku pozostałych części, dokonanie edycji numerów pinów.

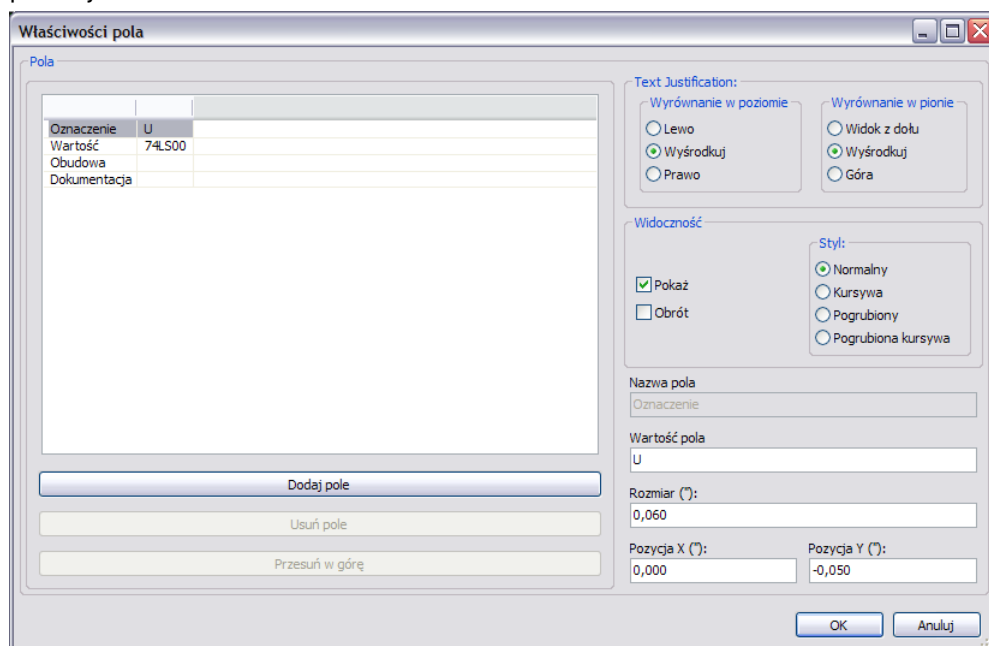
11.8. Edycja pól

Dla pól już istniejących można użyć poleceń szybkiej edycji dostępnej z menu podręcznego wywoływanego prawym klawiszem myszy :



Dla pełnej edycji pól będzie niezbędne użycie narzędzia do edycji tych pól, ukrywającego się pod ikoną: .

Okno jakie zostanie wtedy otwarte będzie wyglądać podobnie do tego znajdującego się poniżej:



Na przedstawionym rysunku aktywne jest pole *Oznaczenie* i to jego dotyczą parametry znajdujące się po prawej stronie.

Pola to teksty przypisane do symbolu, ale nie należy ich mylić z tekstem graficznym jaki może zostać umieszczony w graficznej reprezentacji symbolu.

Dostępne są następujące pola :

- ◆ **Wartość**
- ◆ **Oznaczenie**
- ◆ Nazwa przypisanego modułu (**Obudowa** dla PCB)
- ◆ **Łączy do pliku z dokumentacją** (przeznaczone głównie dla schematów).
- ◆ Pola użytkownika **Pole1** do **Pole8** lub dalsze (jako komentarze)

Pola *Wartość* i *Oznaczenie* są definiowane podczas tworzenia symbolu i mogą być tutaj modyfikowane.

Jest możliwość użycia pola *Obudowa* jako nazwy zawierającej nazwę modułu przypisanego do symbolu by móc bezpośrednio generować kompletne pliki list sieci (z wstępnie określonymi

modułami dla PCB). Pole Obudowa powiązane ze schematem jest szczególnie użyteczne w przypadku innych programów ECAD.

Pola 1 do 8 i dalsze nie są naprawdę przeznaczone do wykorzystania w bibliotece, ponieważ są one raczej stosowane przy tworzeniu projektu schematu. Można im tutaj zmienić nazwę. Dla biblioteki, edycja pól *Wartość* i *Oznaczenie* umożliwia określenie wielkości i położenia.

Ważne :

- ♦ **Modyfikacja tekstu** w polu *Wartość*, równa się **utworzeniu nowego symbolu** na podstawie starego, służącego wtedy za wzór, gdyż symbol po zapisaniu go w bibliotece ma nazwę taką jak zawarta w polu *Wartość*.
- ♦ Aby edytować **niewidoczne pole** (tzn. puste, bo nawet jeśli pole ma atrybut *Niewidoczny*, jest ono wyświetlane w LibEdit) trzeba będzie skorzystać z okna edycji przedstawionego powyżej.

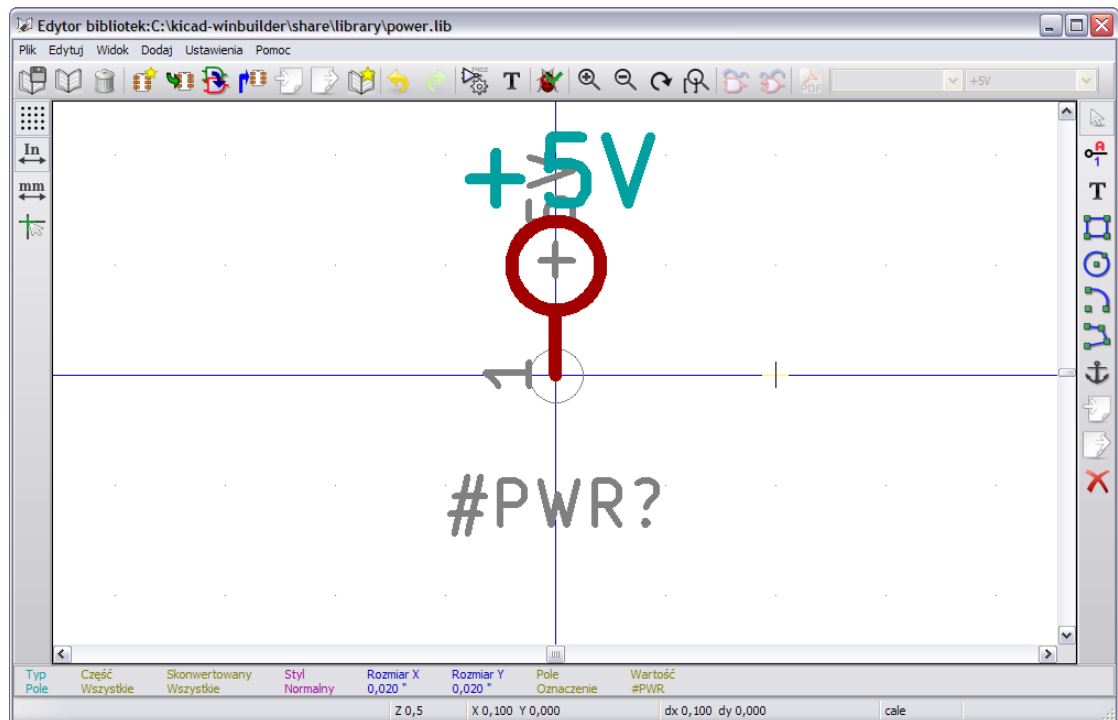
11.9. Tworzenie symboli zasilania

Symbole portów zasilania są tworzone tak jak zwykłe symbole.

Może być to przydatne w celu zebrania ich w specjalnej bibliotece takiej jak power.lib.

Zawierają się one w swoim symbolu graficznym (żądanej postaci) i w pinie o typie *Niewidoczne źródło zasilania*. Będą one zatem traktowane jak każdy inny symbol przeznaczony dla oprogramowania do tworzenia schematów elektronicznych. Jednakże, pewne środki ostrożności są dalej niezbędne.

Poniżej mamy symbol zasilania (zasilanie +5V) :



Na symbol taki składają się:

- ♦ Wyprowadzenie typu **Niewidoczne źródło zasilania** nazwane +5V (ważne, gdyż za pomocą tej nazwy będzie możliwe nawiązanie połączenia do sieci +5V), z numerem pinu 1 (numer nie jest jednak ważny) oraz z długością pinu równą zero.
- ♦ Kształt to **Linia**, oczywiście z zadeklarowanym typem **Zasilanie** oraz z atrybutem **Niewidoczny**.
- ♦ Postać graficzna : tu jest to mały okrąg i krótki segment linii prowadzący od pinu do okręgu.
- ♦ **Punkt aktywny** symbolu jest w miejscu pinu.
- ♦ Wartość to +5V tak samo jak nazwa pinu, i służy do wyświetlania wartości tego symbolu (pin nie będzie widoczny, zatem jego nazwa by się nie pojawiła).
- ♦ Oznaczenie to #PWR?. Oznaczenie to nie ma większego znaczenia poza pierwszym znakiem, który musi być znakiem kratki "#". **Zgodnie z przyjętą konwencją, każdy element, którego odniesienia rozpoczynają się tym znakiem nie pojawią się ani na liście składników, ani w liście sieci.** Co więcej, opcjach symbolu, odniesienie jest zadeklarowane jako niewidoczne.

Tworzenie nowych symboli zasilania jest znacznie prostsze gdy zostanie ono wykonane na podstawie innego symbolu tego typu. Zatem najprościej jest :

- ◆ Załadować wzorzec (inny symbol zasilania).
- ◆ Dokonać zmiany **nazwy pinu** (która zostanie nazwą nowego portu zasilania).
- ◆ Dokonać zmiany pola *Wartość* (taka sama jak nazwa pinu, jeśli chcemy by wartość odpowiadająca temu portowi była wyświetlana...).
- ◆ Zapisanie nowego symbolu.

12. Edytor bibliotek LibEdit - Dodatkowe możliwości

12.1. Wprowadzenie

Symbol składa się z kilku elementów:

- ◆ Jego postaci graficznej (kształty geometryczne, teksty).
- ◆ Wyprowadzenia (piny).
- ◆ Pola lub teksty powiązane, wykorzystywane przez post-procesory: tworzące listy sieci, listy materiałowe...

W tym rozdziale skupimy się na ostatniej pozycji, gdyż umiejętne zarządzanie narzędziami dodatkowymi związanymi z symbolami ułatwia przeszukiwanie, obsługę i konserwację bibliotek.

Dwa pola są inicjowane zawsze: *Oznaczenie* i *Wartość*. Nazwa projektu powiązana ze składnikiem, nazwa powiązanego z nim modułu, albo inne pola pozostałe są dowolne i mogą pozostać ogólnie puste, albo mogą być wypełnione podczas rysowania schematu.

Powiązana dokumentacja zawiera m.in.:

- Liniję komentarza.
- Liniję ze słowami kluczowymi takimi jak np.: TTL CMOS NAND2..., oddzielonymi spacjami
- Liniję z nazwą pliku zewnętrznej dokumentacji (np. notę aplikacyjną, notę katalogową...). Domyślnie pliki te są wyszukiwane w następujących katalogach:
kicad/share/library/doc lub
kicad/library/doc.
w systemie Linux także w:
/usr/local/kicad/share/library/doc
/usr/share/kicad/library/doc
/usr/local/share/kicad/library/doc

Słowa kluczowe pozwalają na selektywne wyszukiwanie symboli według różnych kryteriów wyboru.

Komentarze i słowa kluczowe są wyświetlane w różnych miejscach, szczególnie jednak po wybraniu elementu w bibliotece.


Element posiada również **aliasy**, czyli nazwy alternatywne. Pozwala to na znaczne zmniejszenie liczby elementów składowych bibliotek, które muszą być tworzone (np. 74LS00 może posiadać aliasy takie jak 7400, 74HC00, 74HCT00 ...).

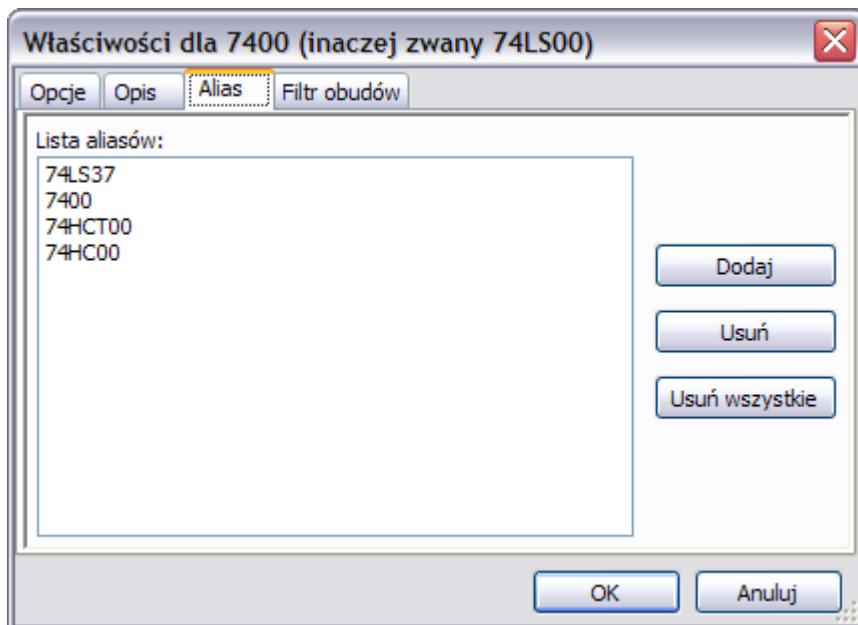
Wreszcie, symbole są zwykle umieszczane w kilku bibliotekach (klasyfikowanych według tematów, lub producentów...), w celu ułatwienia zarządzania nimi.

12.2. Aliasy

Alias jest nazwą alternatywną odpowiadającą temu samego symbolowi w bibliotece.

Symbole o podobnym rozkładzie pinów i podobnej reprezentacji mogą być reprezentowane tylko przez jeden symbol o kilku aliasach (np.: 7400 może posiadać aliasy 74LS00, 74HC00, 74LS37.). Korzystanie z aliasów pozwala na tworzenie kompletnych bibliotek symboli znacznie szybciej. Oprócz tego biblioteki takie są znacznie bardziej zwarte i są ładowane szybciej.

Aby zmodyfikować listę aliasów, należy wybrać w głównym oknie edycji narzędzie  i wybrać zakładkę **Alias**:



Za pomocą klawiszy obok listy można dodawać lub usuwać aliasy. Bieżący alias nie może być oczywiście zmieniony ponieważ jest edytowany.

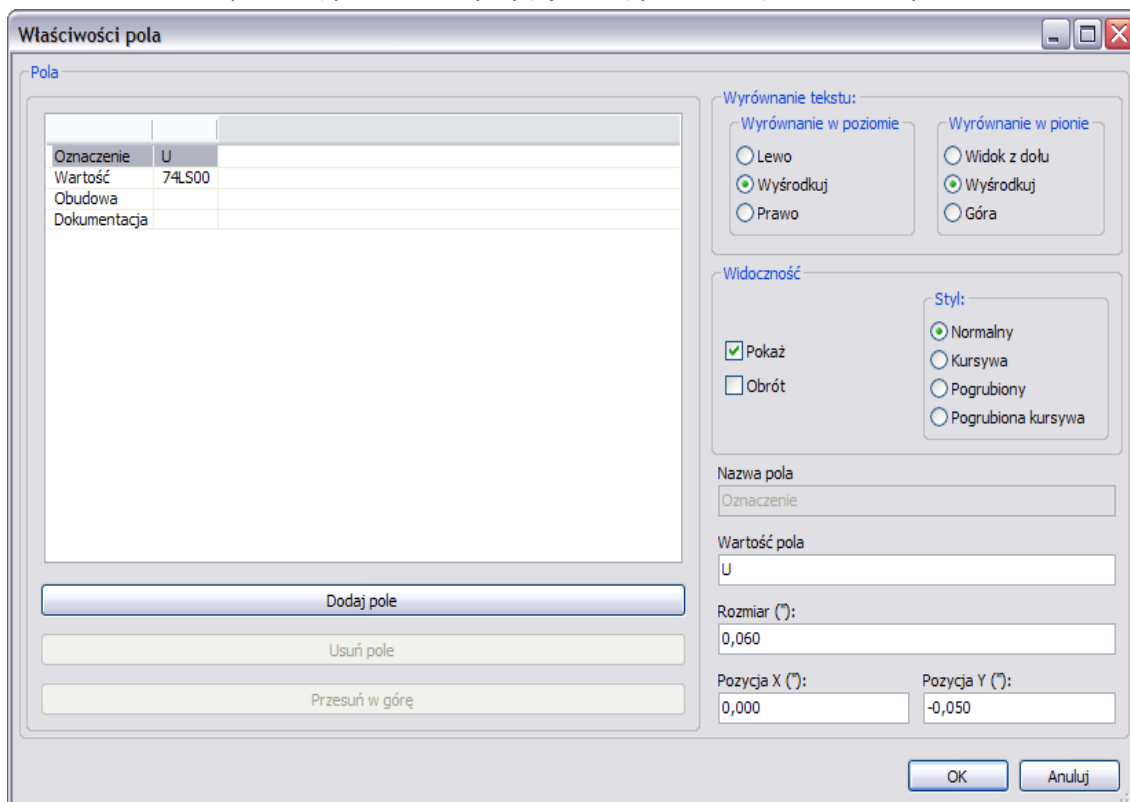
By usunąć wszystkie aliasy **należy najpierw wybrać symbol główny** (pierwszy z listy aliasów na górnym pasku narzędzi edytora bibliotek).

12.3. Pola specjalne



Edytor pól jest wywoływany za pomocą ikony

Istnieją cztery specjalne pola (tekst przypisany do symbolu) oraz pola użytkownika, które może dodawać za pomocą poleceń znajdujących się pod tabelą zawartości pól.




Pola specjalne to:

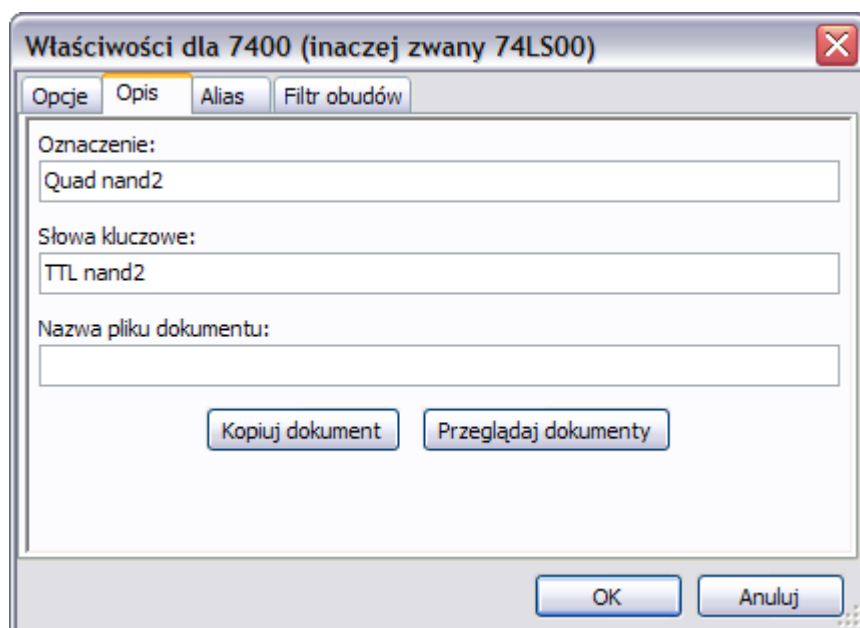
- **Oznaczenie.**
- **Wartość:** Jest to nazwa symbolu w bibliotece, oraz domyślna wartość jaka zostanie umieszczona na schemacie.
- **Obudowa:** nazwa modułu używanego na PCB. Nie jest zbyt potrzebna jeśli używamy **CvPcb** do ustalania listy modułów.

- **Dokumentacja:** zarezerwowane (tutaj nie używane).

12.4. Dokumentowanie symboli

Aby dokonać edycji opcji związanych z dokumentacją symboli, należy wywołać główne okno

właściwości symbolu (narzędzie ) oraz przełączyć się na zakładkę **Opis**.



Ostrożnie:

Należy pamiętać, aby **wybrać odpowiedni alias lub składnik główny**, ponieważ opcje dokumentacyjne są jedyną cechą, którą różnią się poszczególne aliasy. Polecenie **Kopiuje dokument** pozwala na skopiowanie tych informacji z dokumentacji symbolu głównego do aktualnie edytowanego aliasu.

12.4.1. Słowa kluczowe

Słowa kluczowe pozwalają na wyszukiwanie symboli w sposób selektywny według określonych kryteriów wyboru (funkcja, rodzina, technologia, itp.). Wielkość liter nie jest istotna, bo narzędzie zawarte w **Eeschema** nie rozróżnia wielkość liter.

Najbardziej popularne słowa kluczowe użyte w bibliotekach to:

- CMOS TTL dla rodzin układów logicznych
- AND2 NOR3 XOR2 INV... dla bramek (AND2 = 2 wejściowa bramka AND, NOR3 = 3 wejściowa bramka NOR).
- JKFF, DFF... dla przerzutników typu JK lub D.
- ADC, DAC, MUX... dla przetworników.
- OpenCo1 dla bramek posiadających wyjścia typu otwarty kolektor.

Tak więc, jeśli w oprogramowaniu do edycji schematów będziemy szukać symboli za pomocą słów kluczowych NAND2 OpenCo1, **Eeschema** wyświetli listę symboli pasujących do tych 2 słów kluczowych.

12.4.2. Dokumentacja symbolu

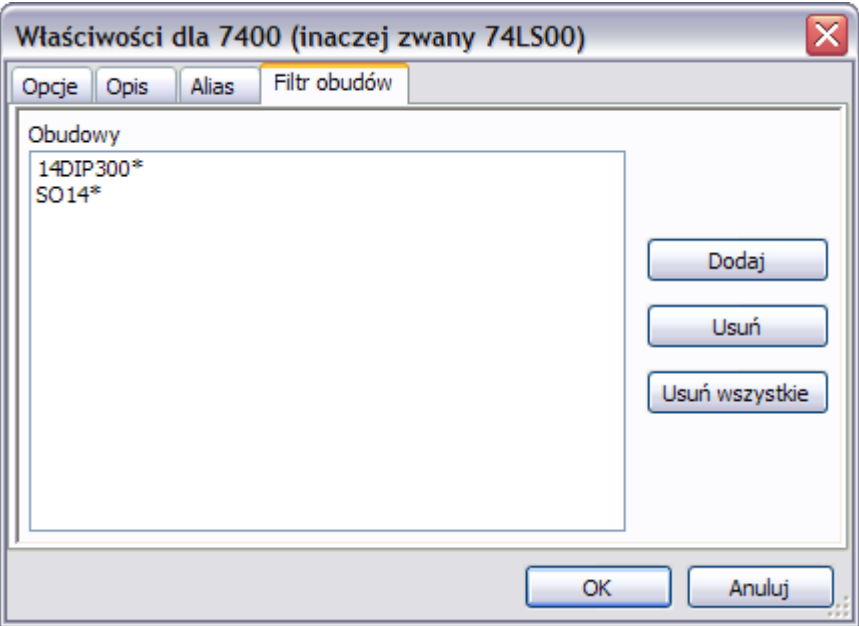
Linia oznaczenie (i **słowa kluczowe**) jest wyświetlana w różnych menu, w szczególności po wybraniu elementu na wyświetlonej liście komponentów biblioteki i w menu **ViewLib**.

Linia *Nazwa pliku dokumentu* wskazuje na dostępny plik z dokumentacją symbolu lub jego schematem aplikacyjnym. Może to być plik w formacie PDF (zwykle stosowany w takim przypadku) dostępny lokalnie na dysku twardym, ale można stosować również ścieżki URL by umożliwić dostęp do zasobów zdanych (np. na stronie WWW producenta).

Jeśli plik dokumentacji istnieje (jest wpisany w to pole), jest on również dostępny z poziomu oprogramowania do tworzenia schematów w menu wyświetlanym przy kliknięciu prawym przyciskiem myszy na komponentcie.

12.4.3.Filtrowanie modułów dla CvPcb

Za pomocą listy znajdującej się na zakładce **Filtr obudów** można określić jakie moduły byłby odpowiednie dla obecnie projektowanego symbolu. Listę tą wykorzystuje **CvPcb** by podczas przypisywania obudów można było odfiltrować listę dostępnych modułów tylko do tych najbardziej odpowiednich. Jeśli nie chcemy korzystać z tych możliwości można albo opcję filtracji w **CvPcb** wyłączyć, albo pozostawić tą listę pustą.



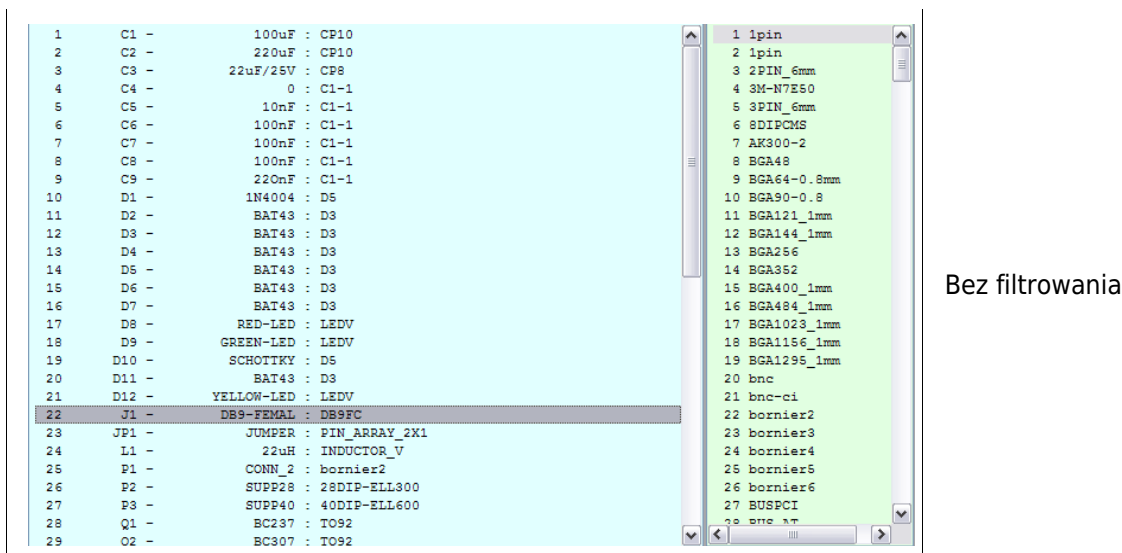
Na liście można stosować szeroko stosowane znaki maskujące: * oraz ?. Przykładowo: S014* pozwala **CvPcb** na wyświetlenie wszystkich modułów których nazwa rozpoczyna się znakami SO14. Zaś R? Pozwala na wyświetlenie tylko tych modułów, których nazwa będzie składać się z dwóch znaków, przy czym nazwa taka będzie musiała się rozpoczynać znakiem R.

Poniżej przedstawiono widok okna **CvPcb** z włączoną jak i wyłączoną filtracją:

1	C1 -	100uF :	CP10
2	C2 -	220uF :	CP10
3	C3 -	22uF/25V :	CP8
4	C4 -	0 :	C1-1
5	C5 -	10nF :	C1-1
6	C6 -	100nF :	C1-1
7	C7 -	100nF :	C1-1
8	C8 -	100nF :	C1-1
9	C9 -	220nF :	C1-1
10	D1 -	1N4004 :	D5
11	D2 -	BAT43 :	D3
12	D3 -	BAT43 :	D3
13	D4 -	BAT43 :	D3
14	D5 -	BAT43 :	D3
15	D6 -	BAT43 :	D3
16	D7 -	BAT43 :	D3
17	D8 -	RED-LED :	LEDV
18	D9 -	GREEN-LED :	LEDV
19	D10 -	SCHOTTKY :	D5
20	D11 -	BAT43 :	D3
21	D12 -	YELLOW-LED :	LEDV
22	J1 -	DB9-FEMAL :	DB9FC
23	JP1 -	JUMPER :	PIN_ARRAY_2X1
24	L1 -	22uH :	INDUCTOR_V
25	P1 -	CONN_2 :	bornier2
26	P2 -	SUPP28 :	28DIP-ELL300
27	P3 -	SUPP40 :	40DIP-ELL600
28	Q1 -	BC237 :	TO92
29	Q2 -	BC307 :	TO92

1	DB9FC
2	DB9FD
3	DB9F_CI
4	DB9F_CI_INVERT
5	DB9MC
6	DB9MD
7	DB9M_CI
8	DB9M_CI_INVERT

Z filtrowaniem
(DB9*)



12.5. Biblioteka wzorców

W łatwy sposób można skompilować podręczną biblioteczkę z rysunkami, zawierającą często używane grafiki. Można to wykorzystywać do tworzenia podstawowych komponentów składowych symboli (np. trójkąty, prostokąty, kształty bramek AND, OR, XOR...) w celu ich późniejszego ponownego użycia.

Pliki te są przechowywane domyślnie w katalogu biblioteki jako poszczególne pliki z rozszerzeniem .sym. Wzorce te nie są zbierane w jednym pliku bibliotecznym jak symbole, ponieważ zazwyczaj nie są zbyt liczne.

12.5.1. Eksport/Tworzenie wzorca

Symbol może być łatwo wyeksportowany jako wzorec graficzny za pomocą narzędzia



eksportu znajdującego się na prawym pasku narzędzi. Generalnie przy tworzeniu wzorców głównie chodzi o samą grafikę, zatem dobrym pomysłem jest by przed eksportem, usunąć istniejące piny.

12.5.2. Importowanie wzorca

Podczas edycji symbolu można za pomocą narzędzia **Importuj istniejące rysunki** dodać zapisany wcześniej wzorec. Zaimportowana grafika zostanie dodana tak jakby została ona właśnie narysowana.



13. Przeglądarka bibliotek ViewLib

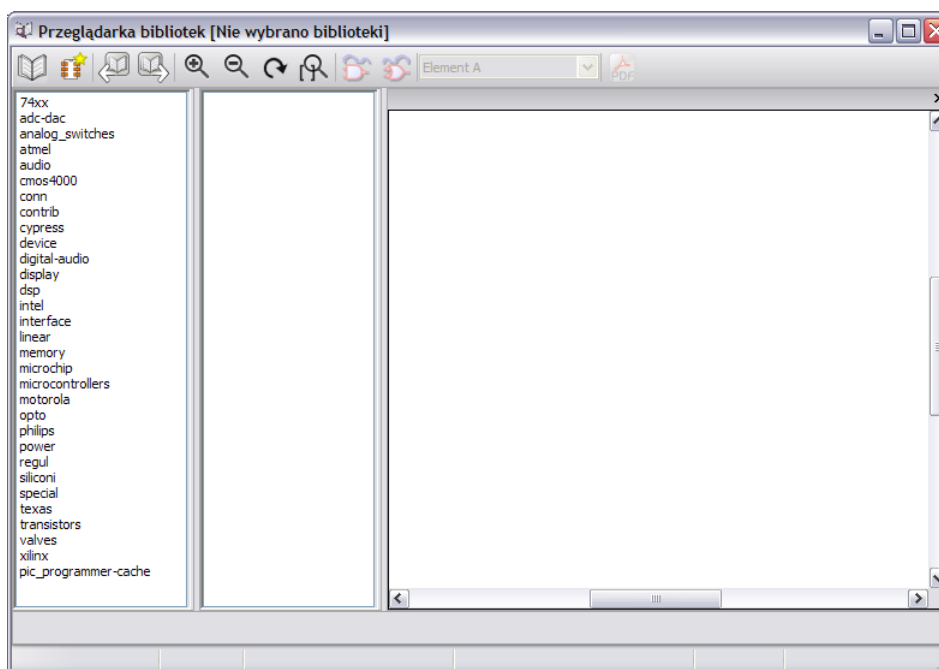
13.1. Przeznaczenie

Przeglądarka bibliotek pozwala na szybkie sprawdzenie zawartości aktywnych bibliotek.



Przeglądarkę można wywołać klikając w ikonę

13.2. Ekran główny



By sprawdzić zawartość biblioteki należy ją wybrać z listy znajdującej się po lewej stronie okna.

Jej zawartość zostanie pokazana na drugiej liście, z której można wybrać jeden z elementów, którego podgląd pojawi się w panelu po prawej stronie.

13.3. Pasek narzędzi przeglądarki bibliotek

Podstawowy pasek narzędzi wygląda w ten sposób:



lub (jeśli przeglądarka została wywołana z okna dialogowego **Eeschema** podczas operacji wstawiania elementów):



Poszczególne polecenia to:



Wybór przeglądanej biblioteki (który może być również zrealizowany poprzez wybór z listy).



Wybór elementu z biblioteki (który może być również zrealizowany przez wybór z listy).



Przejdzie do poprzedniego elementu na liście.



Przejdzie do następnego elementu na liście.



Narzędzia do dostosowywania powiększenia.



Wybór reprezentacji elementu (normalny lub skonwertowany jeśli takowy istnieje).



Wybór części składowej elementu (w przypadku elementów wieloczęściowych).



Uruchamia zewnętrzną przeglądarkę z dokumentacją elementu (jeśli została określona).



Wybór przeglądanej biblioteki (który może być również zrealizowany poprzez wybór z listy).



Zamyka przeglądarkę bibliotek i wstawia wybrany element do aktywnego schematu.

14. Dostosowywanie list sieci oraz zestawień materiałowych (BOM)

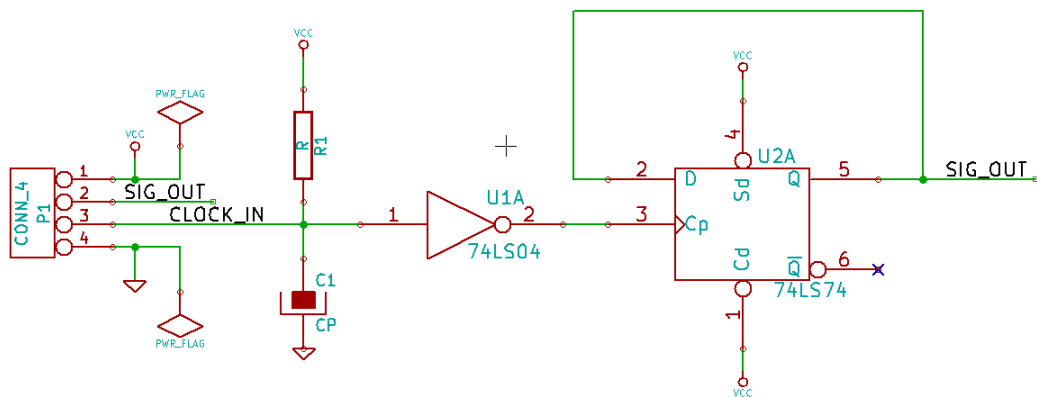
14.1. Pośrednia lista sieci

Lista materiałowa BOM i lista sieci może być skonwertowana z formatu pośredniego tworzonego przez **Eeschema**. Plik ten jest zbudowany z tagów XML i nosi nazwę **pliku pośredniego listy sieci**.

Ponieważ zawiera on jednak wiele danych na temat komponentów schematu, może być też wykorzystywany do tworzenia list materiałowych BOM lub innych raportów – nie tylko list sieci.

W zależności od formatu wyjściowego (BOM, nowe listy sieci), tylko niektóre sekcje tego pliku będą wykorzystywane.

14.1.1. Przykładowy schemat



14.1.2. Przykładowa pośrednia lista sieci

Odpowiadająca temu schematowi pośrednia lista sieci (używając składni XML) wygląda tak:

```
<?xml version="1.0" encoding="utf-8"?>
<export version="D">
  <design>
    <source>F:\kicad_aux\netlist_test\netlist_test.sch</source>
    <date>29/08/2010 20:35:21</date>
    <tool>eeschema (2010-08-28 BZR 2458) -unstable</tool>
  </design>
  <components>
    <comp ref="P1">
      <value>CONN_4</value>
      <libsource lib="conn" part="CONN_4"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E2141</tstamp>
    </comp>
    <comp ref="U2">
      <value>74LS74</value>
      <libsource lib="74xx" part="74LS74"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E20BA</tstamp>
    </comp>
```

```

<comp ref="U1">
  <value>74LS04</value>
  <libsource lib="74xx" part="74LS04"/>
  <sheetpath names="/" tstamps="/" />
  <tstamp>4C6E20A6</tstamp>
</comp>
<comp ref="C1">
  <value>CP</value>
  <libsource lib="device" part="CP"/>
  <sheetpath names="/" tstamps="/" />
  <tstamp>4C6E2094</tstamp>
</comp>
<comp ref="R1">
  <value>R</value>
  <libsource lib="device" part="R"/>
  <sheetpath names="/" tstamps="/" />
  <tstamp>4C6E208A</tstamp>
</comp>
</components>
<libparts>
  <libpart lib="device" part="C">
    <description>Condensateur non polarise</description>
    <footprints>
      <fp>SM*</fp>
      <fp>C?</fp>
      <fp>C1-1</fp>
    </footprints>
    <fields>
      <field name="Reference">C</field>
      <field name="Value">C</field>
    </fields>
    <pins>
      <pin num="1" name="~" type="passive"/>
      <pin num="2" name="~" type="passive"/>
    </pins>
  </libpart>
  <libpart lib="device" part="R">
    <description>Resistance</description>
    <footprints>
      <fp>R?</fp>
      <fp>SM0603</fp>
      <fp>SM0805</fp>
      <fp>R?-*</fp>
      <fp>SM1206</fp>
    </footprints>
    <fields>
      <field name="Reference">R</field>
      <field name="Value">R</field>
    </fields>
    <pins>
      <pin num="1" name="~" type="passive"/>
      <pin num="2" name="~" type="passive"/>
    </pins>
  </libpart>
  <libpart lib="conn" part="CONN_4">
    <description>Symbole general de connecteur</description>
    <fields>
      <field name="Reference">P</field>
      <field name="Value">CONN_4</field>
    </fields>
    <pins>
      <pin num="1" name="P1" type="passive"/>
      <pin num="2" name="P2" type="passive"/>
      <pin num="3" name="P3" type="passive"/>
      <pin num="4" name="P4" type="passive"/>
    </pins>
  </libpart>
  <libpart lib="74xx" part="74LS04">
    <description>Hex Inverseur</description>
    <fields>
      <field name="Reference">U</field>
      <field name="Value">74LS04</field>
    </fields>
    <pins>
      <pin num="1" name="~" type="input"/>
      <pin num="2" name="~" type="output"/>
      <pin num="3" name="~" type="input"/>
      <pin num="4" name="~" type="output"/>
      <pin num="5" name="~" type="input"/>
    </pins>
  </libpart>

```

```

        <pin num="6" name="~" type="output"/>
        <pin num="7" name="GND" type="power_in"/>
        <pin num="8" name="~" type="output"/>
        <pin num="9" name="~" type="input"/>
        <pin num="10" name="~" type="output"/>
        <pin num="11" name="~" type="input"/>
        <pin num="12" name="~" type="output"/>
        <pin num="13" name="~" type="input"/>
        <pin num="14" name="VCC" type="power_in"/>
    </pins>
</libpart>
<libpart lib="74xx" part="74LS74">
    <description>Dual D FlipFlop, Set & Reset</description>
    <docs>74xx/74hc_hct74.pdf</docs>
    <fields>
        <field name="Reference">U</field>
        <field name="Value">74LS74</field>
    </fields>
    <pins>
        <pin num="1" name="Cd" type="input"/>
        <pin num="2" name="D" type="input"/>
        <pin num="3" name="Cp" type="input"/>
        <pin num="4" name="Sd" type="input"/>
        <pin num="5" name="Q" type="output"/>
        <pin num="6" name="~Q" type="output"/>
        <pin num="7" name="GND" type="power_in"/>
        <pin num="8" name="~Q" type="output"/>
        <pin num="9" name="Q" type="output"/>
        <pin num="10" name="Sd" type="input"/>
        <pin num="11" name="Cp" type="input"/>
        <pin num="12" name="D" type="input"/>
        <pin num="13" name="Cd" type="input"/>
        <pin num="14" name="VCC" type="power_in"/>
    </pins>
</libpart>
</libparts>
<libraries>
    <library logical="device">
        <uri>F:\kicad\share\library\device.lib</uri>
    </library>
    <library logical="conn">
        <uri>F:\kicad\share\library\conn.lib</uri>
    </library>
    <library logical="74xx">
        <uri>F:\kicad\share\library\74xx.lib</uri>
    </library>
</libraries>
<nets>
    <net code="1" name="GND">
        <node ref="U1" pin="7"/>
        <node ref="C1" pin="2"/>
        <node ref="U2" pin="7"/>
        <node ref="P1" pin="4"/>
    </net>
    <net code="2" name="VCC">
        <node ref="R1" pin="1"/>
        <node ref="U1" pin="14"/>
        <node ref="U2" pin="4"/>
        <node ref="U2" pin="1"/>
        <node ref="U2" pin="14"/>
        <node ref="P1" pin="1"/>
    </net>
    <net code="3" name="">
        <node ref="U2" pin="6"/>
    </net>
    <net code="4" name="">
        <node ref="U1" pin="2"/>
        <node ref="U2" pin="3"/>
    </net>
    <net code="5" name="/SIG_OUT">
        <node ref="P1" pin="2"/>
        <node ref="U2" pin="5"/>
        <node ref="U2" pin="2"/>
    </net>
    <net code="6" name="/CLOCK_IN">
        <node ref="R1" pin="2"/>
        <node ref="C1" pin="1"/>
        <node ref="U1" pin="1"/>
        <node ref="P1" pin="3"/>
    </net>

```

```

    </net>
  </nets>
</export>

```

14.2. Konwersja na nowy format

Poprzez zastosowanie filtra dla tego pliku pośredniego z listą sieci, można wygenerować listę sieci w wielu innych formatach oraz listę materiałową BOM.

Ponieważ jest to tylko transformacja jednej postaci tekstu na drugą postać, filtr ten można łatwo napisać w języku PYTHON lub XSLT.

XSLT sam w sobie jest językiem XML przeznaczonym do definiowania transformacji dla XML. W przypadku użycia XSLT, można użyć programu `xsltproc.exe` do odczytu pliku XML z zastosowaniem przy tym **arkusza stylu** (tzw. "filtr") i zapisaniu danych wynikowych. Dlatego użytkownik musi wcześniej utworzyć odpowiedni **arkusz stylu** pliku wyjściowego używając konwencji przyjętych w XSLT.

Cały proces konwersji jest przejrzysty i obsługiwany przez **Eeschema**.

14.3. Język skryptowy XSLT

Dokumentacja XSL Transformations (XSLT) znajduje się pod adresem <http://www.w3.org/TR/xslt/>

14.3.1.Przykład tworzenia pliku z listą sieci PADS-PCB

Format ten posiada dwie sekcje pliku listy sieci :

- ◆ Listę footprint-ów
- ◆ Listę sieci (zgrupowane wyprowadzenia według sieci)

Poniżej znajduje się przykład arkusza stylów, na podstawie którego można skonwertować plik pośredni listy do jej odpowiednika w formacie akceptowanym przez PADS-PCB:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--XSL style sheet to EESCHEMA Generic Netlist Format to PADS netlist format
Copyright (C) 2010, SoftPLC Corporation.
GPL v2.

How to use:
https://lists.launchpad.net/kicad-developers/msg05157.html
-->

<!DOCTYPE xsl:stylesheet [
  <!ENTITY nl "&#xd;&#xa;"> <!--new line CR, LF -->
]>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="text" omit-xml-declaration="yes" indent="no"/>

<xsl:template match="/export">
  <xsl:text>*PADS-PCB*&nl;*PART*&nl;</xsl:text>
  <xsl:apply-templates select="components/comp"/>
  <xsl:text>&nl;*NET*&nl;</xsl:text>
  <xsl:apply-templates select="nets/net"/>
  <xsl:text>*END*&nl;</xsl:text>
</xsl:template>

<!-- for each component -->
<xsl:template match="comp">
  <xsl:text> </xsl:text>
  <xsl:value-of select="@ref"/>
  <xsl:text> </xsl:text>
  <xsl:choose>
    <xsl:when test = "footprint != ''">
      <xsl:apply-templates select="footprint"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>unknown</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:text>&nl;</xsl:text>
</xsl:template>

<!-- for each net -->
<xsl:template match="net">
  <!-- nets are output only if there is more than one pin in net -->
```

```

<xsl:if test="count(node)>1">
  <xsl:text>*SIGNAL* </xsl:text>
  <xsl:choose>
    <xsl:when test = "@name != '' ">
      <xsl:value-of select="@name"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>N-</xsl:text>
      <xsl:value-of select="@code"/>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:text>&nl;</xsl:text>
  <xsl:apply-templates select="node"/>
</xsl:if>
</xsl:template>

<!-- for each node -->
<xsl:template match="node">
  <xsl:text> </xsl:text>
  <xsl:value-of select="@ref"/>
  <xsl:text>.</xsl:text>
  <xsl:value-of select="@pin"/>
  <xsl:text>&nl;</xsl:text>
</xsl:template>

</xsl:stylesheet>

```

Finalny plik wyjściowy po zastosowaniu tego arkusza jako filtra dla *xsltproc*:

```

*PADS-PCB*
*PART*
P1 unknown
U2 unknown
U1 unknown
C1 unknown
R1 unknown

*NET*
*SIGNAL* GND
U1.7
C1.2
U2.7
P1.4
*SIGNAL* VCC
R1.1
U1.14
U2.4
U2.1
U2.14
P1.1
*SIGNAL* N-4
U1.2
U2.3
*SIGNAL* /SIG_OUT
P1.2
U2.5
U2.2
*SIGNAL* /CLOCK_IN
R1.2
C1.1
U1.1
P1.3
*END*

```

Polecenie które dokonało takiej konwersji wygląda następująco:

```
f:/kicad/bin/xsltproc.exe -o test.net f:/kicad/bin/plugins/netlist_form_pads-
pcb.xml test.tmp
```

14.3.2. Przykład tworzenia listy sieci programu Cadstar

Ten format składa się z dwóch sekcji:

- ◆ Listy footprintów
- ◆ Listy sieci (pogrupowanych według sieci odnośników do wyprowadzeń)

14.3.2.1. Plik z arkuszem stylu

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!--XSL style sheet to EESCHEMA Generic Netlist Format to CADSTAR netlist format
Copyright (C) 2010, Jean-Pierre Charras.
Copyright (C) 2010, SoftPLC Corporation.
GPL v2.

<!DOCTYPE xsl:stylesheet [
  <!ENTITY nl "&#xd;&#xa;"> <!--new line CR, LF -->
]>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="text" omit-xml-declaration="yes" indent="no"/>

<!-- Netlist header -->
<xsl:template match="/export">
  <xsl:text>.HEA&nl;</xsl:text>
  <xsl:apply-templates select="design/date"/> <!-- Generate line .TIM <time> -->
  <xsl:apply-templates select="design/tool"/> <!-- Generate line .APP <eeschema
version> -->
  <xsl:apply-templates select="components/comp"/> <!-- Generate list of components
-->
  <xsl:text>&nl;&nl;</xsl:text>
  <xsl:apply-templates select="nets/net"/> <!-- Generate list of nets and
connections -->
  <xsl:text>&nl;.END&nl;</xsl:text>
</xsl:template>

  <!-- Generate line .TIM 20/08/2010 10:45:33 -->
<xsl:template match="tool">
  <xsl:text>.APP "</xsl:text>
  <xsl:apply-templates/>
  <xsl:text>"&nl;</xsl:text>
</xsl:template>

  <!-- Generate line .APP "eeschema (2010-08-17 BZR 2450)-unstable" -->
<xsl:template match="date">
  <xsl:text>.TIM </xsl:text>
  <xsl:apply-templates/>
  <xsl:text>&nl;</xsl:text>
</xsl:template>

  <!-- for each component -->
<xsl:template match="comp">
  <xsl:text>.ADD_COM </xsl:text>
  <xsl:value-of select="@ref"/>
  <xsl:text> </xsl:text>
  <xsl:choose>
    <xsl:when test = "value != '' ">
      <xsl:text>"</xsl:text> <xsl:apply-templates select="value"/>
<xsl:text>"</xsl:text>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>"</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:text>&nl;</xsl:text>
</xsl:template>

  <!-- for each net -->
<xsl:template match="net">
  <!-- nets are output only if there is more than one pin in net -->
  <xsl:if test="count(node)>1">
    <xsl:variable name="netname">
      <xsl:text>"</xsl:text>
      <xsl:choose>
        <xsl:when test = "@name != '' ">
          <xsl:value-of select="@name"/>
        </xsl:when>
        <xsl:otherwise>
          <xsl:text>N-</xsl:text>
          <xsl:value-of select="@code"/>
        </xsl:otherwise>
      </xsl:choose>
      <xsl:text>"&nl;</xsl:text>
    </xsl:variable>
    <xsl:apply-templates select="node" mode="first"/>

```

```

        <xsl:value-of select="$netname"/>
        <xsl:apply-templates select="node" mode="others"/>
    </xsl:if>
</xsl:template>

<!-- for each node -->
<xsl:template match="node" mode="first">
    <xsl:if test="position()=1">
        <xsl:text>.ADD_TER </xsl:text>
        <xsl:value-of select="@ref"/>
        <xsl:text>.</xsl:text>
        <xsl:value-of select="@pin"/>
        <xsl:text> </xsl:text>
    </xsl:if>
</xsl:template>

<xsl:template match="node" mode="others">
    <xsl:choose>
        <xsl:when test='position()=1'>
            </xsl:when>
        <xsl:when test='position()=2'>
            <xsl:text>.TER </xsl:text>
        </xsl:when>
        <xsl:otherwise>
            <xsl:text> </xsl:text>
        </xsl:otherwise>
    </xsl:choose>
    <xsl:if test="position()>1">
        <xsl:value-of select="@ref"/>
        <xsl:text>.</xsl:text>
        <xsl:value-of select="@pin"/>
        <xsl:text>&nl;</xsl:text>
    </xsl:if>
</xsl:template>

</xsl:stylesheet>

```

14.3.2.2. Plik wyjściowy

```

.HEA
.TIM 21/08/2010 08:12:08
.APP "eeschema (2010-08-09 BZR 2439)-unstable"
.ADD_COM P1 "CONN_4"
.ADD_COM U2 "74LS74"
.ADD_COM U1 "74LS04"
.ADD_COM C1 "CP"
.ADD_COM R1 "R"

.ADD_TER U1.7 "GND"
.TER C1.2
    U2.7
    P1.4
.ADD_TER R1.1 "VCC"
.TER U1.14
    U2.4
    U2.1
    U2.14
    P1.1
.ADD_TER U1.2 "N-4"
.TER U2.3
.ADD_TER P1.2 "/SIG_OUT"
.TER U2.5
    U2.2
.ADD_TER R1.2 "/CLOCK_IN"
.TER C1.1
    U1.1
    P1.3

.END

```

14.3.3. Przykład tworzenia listy sieci programu OrcadPCB2

Ten format posiada tylko jedną sekcję – listę footprintów. Każdy z footprintów zawiera swoją listę wyprowadzeń z odnośnikami do właściwych sieci.

14.3.3.1. Arkusz stylów

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!--XSL style sheet to EESchema Generic Netlist Format to CADSTAR netlist format
  Copyright (C) 2010, SoftPLC Corporation.
  GPL v2.

  How to use:
    https://lists.launchpad.net/kicad-developers/msg05157.html
-->

<!DOCTYPE xsl:stylesheet [
  <!ENTITY nl "&#xd;&#xa;"> <!--new line CR, LF -->
]>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="text" omit-xml-declaration="yes" indent="no"/>

<!--
  Netlist header
  Creates the entire netlist
  (can be seen as equivalent to main function in C
-->
<xsl:template match="/export">
  <xsl:text>({ EESchema Netlist Version 1.1 </xsl:text>
  <!-- Generate line .TIM <time> -->
  <xsl:apply-templates select="design/date"/>
  <!-- Generate line eeschema version ... -->
  <xsl:apply-templates select="design/tool"/>
  <xsl:text>}&nl;</xsl:text>

  <!-- Generate the list of components -->
  <xsl:apply-templates select="components/comp"/> <!-- Generate list of components
-->

  <!-- end of file -->
  <xsl:text>)&nl;*&nl;</xsl:text>
</xsl:template>

<!--
  Generate id in header like "eeschema (2010-08-17 BZR 2450)-unstable"
-->
<xsl:template match="tool">
  <xsl:apply-templates/>
</xsl:template>

<!--
  Generate date in header like "20/08/2010 10:45:33"
-->
<xsl:template match="date">
  <xsl:apply-templates/>
  <xsl:text>&nl;</xsl:text>
</xsl:template>

<!--
  This template read each component
  (path = /export/components/comp)
  creates lines:
    ( 3EBF7DBD $noname U1 74LS125
      ... pin list ...
    )
  and calls "create_pin_list" template to build the pin list
-->
<xsl:template match="comp">
  <xsl:text>(</xsl:text>
  <xsl:choose>
    <xsl:when test = "tstamp != '' ">
      <xsl:apply-templates select="tstamp"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>00000000</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:text></xsl:text>
  <xsl:choose>
    <xsl:when test = "footprint != '' ">
      <xsl:apply-templates select="footprint"/>
    </xsl:when>

```

```

        <xsl:otherwise>
            <xsl:text>$noname</xsl:text>
        </xsl:otherwise>
    </xsl:choose>
    <xsl:text> </xsl:text>
    <xsl:value-of select="@ref"/>
    <xsl:text> </xsl:text>
    <xsl:choose>
        <xsl:when test = "value != '' ">
            <xsl:apply-templates select="value"/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:text>"~"</xsl:text>
        </xsl:otherwise>
    </xsl:choose>
    <xsl:text>&nl;</xsl:text>
    <xsl:call-template name="Search_pin_list" >
        <xsl:with-param name="cmplib_id" select="libsource/@part"/>
        <xsl:with-param name="cmp_ref" select="@ref"/>
    </xsl:call-template>
    <xsl:text> )&nl;</xsl:text>
</xsl:template>

<!--
    This template search for a given lib component description in list
    lib component descriptions are in /export/libparts,
    and each description start at ./libpart
    We search here for the list of pins of the given component
    This template has 2 parameters:
        "cmplib_id" (reference in libparts)
        "cmp_ref" (schematic reference of the given component)
-->
<xsl:template name="Search_pin_list" >
    <xsl:param name="cmplib_id" select="0" />
    <xsl:param name="cmp_ref" select="0" />
    <xsl:for-each select="/export/libparts/libpart">
        <xsl:if test = "@part = $cmplib_id ">
            <xsl:apply-templates name="build_pin_list" select="pins/pin">
                <xsl:with-param name="cmp_ref" select="$cmp_ref"/>
            </xsl:apply-templates>
        </xsl:if>
    </xsl:for-each>
</xsl:template>

<!--
    This template writes the pin list of a component
    from the pin list of the library description
    The pin list from library description is something like
        <pins>
            <pin num="1" type="passive"/>
            <pin num="2" type="passive"/>
        </pins>
    Output pin list is ( <pin num> <net name> )
    something like
        ( 1 VCC )
        ( 2 GND )
-->
<xsl:template name="build_pin_list" match="pin">
    <xsl:param name="cmp_ref" select="0" />

    <!-- write pin nummer and separator -->
    <xsl:text> ( </xsl:text>
    <xsl:value-of select="@num"/>
    <xsl:text> </xsl:text>

    <!-- search net name in nets section and write it: -->
    <xsl:variable name="pinNum" select="@num" />
    <xsl:for-each select="/export/nets/net">
        <!-- net name is output only if there is more than one pin in net
            else use "?" as net name, so count items in this net
        -->
        <xsl:variable name="pinCnt" select="count(node)" />
        <xsl:apply-templates name="Search_pin_netname" select="node">
            <xsl:with-param name="cmp_ref" select="$cmp_ref"/>
            <xsl:with-param name="pin_cnt_in_net" select="$pinCnt"/>
            <xsl:with-param name="pin_num"> <xsl:value-of select="$pinNum"/>
        </xsl:with-param>
    </xsl:apply-templates>

```

```

</xsl:for-each>

<!-- close line -->
<xsl:text> )&nl;</xsl:text>
</xsl:template>

<!--
This template writes the pin netname of a given pin of a given component
from the nets list
The nets list description is something like
<nets>
  <net code="1" name="GND">
    <node ref="J1" pin="20"/>
    <node ref="C2" pin="2"/>
  </net>
  <net code="2" name="">
    <node ref="U2" pin="11"/>
  </net>
</nets>
This template has 2 parameters:
"cmp_ref"    (schematic reference of the given component)
"pin_num"    (pin number)
-->

<xsl:template name="Search_pin_netname" match="node">
  <xsl:param name="cmp_ref" select="0" />
  <xsl:param name="pin_num" select="0" />
  <xsl:param name="pin_cnt_in_net" select="0" />

  <xsl:if test = "@ref = $cmp_ref">
    <xsl:if test = "@pin = $pin_num">
      <!-- net name is output only if there is more than one pin in net
      else use "?" as net name
      -->
      <xsl:if test = "$pin_cnt_in_net>1">
        <xsl:choose>
          <!-- if a net has a name, use it,
          else build a name from its net code
          -->
          <xsl:when test = "../@name != ''">
            <xsl:value-of select="../@name"/>
          </xsl:when>
          <xsl:otherwise>
            <xsl:text>$N-0</xsl:text><xsl:value-of select="../@code"/>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:if>
      <xsl:if test = "$pin_cnt_in_net < 2">
        <xsl:text>?</xsl:text>
      </xsl:if>
    </xsl:if>
  </xsl:if>
</xsl:template>

</xsl:stylesheet>

```

14.3.3.2. Plik wyjściowy

```

( { EESchema Netlist Version 1.1 29/08/2010 21:07:51
eeschema (2010-08-28 BZR 2458)-unstable}
( 4C6E2141 $noname P1 CONN_4
( 1 VCC )
( 2 /SIG_OUT )
( 3 /CLOCK_IN )
( 4 GND )
)
( 4C6E20BA $noname U2 74LS74
( 1 VCC )
( 2 /SIG_OUT )
( 3 N-04 )
( 4 VCC )
( 5 /SIG_OUT )
( 6 ? )
( 7 GND )
( 14 VCC )
)

```

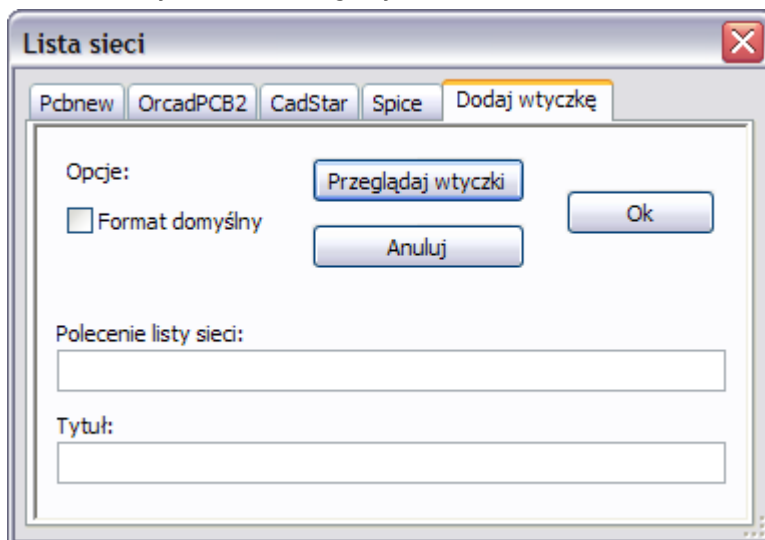
```
( 4C6E20A6 $nname U1 74LS04
( 1 /CLOCK_IN )
( 2 N-04 )
( 7 GND )
( 14 VCC )
)
( 4C6E2094 $nname C1 CP
( 1 /CLOCK_IN )
( 2 GND )
)
( 4C6E208A $nname R1 R
( 1 VCC )
( 2 /CLOCK_IN )
)
)
*
```

14.3.4. Używanie systemu wtyczek Eeschema

Konwertery pośrednich list sieci mogą być wywoływane bezpośrednio z **Eeschema**. Wystarczy tylko odpowiednio skonfigurować narzędzie do tworzenia list sieci.

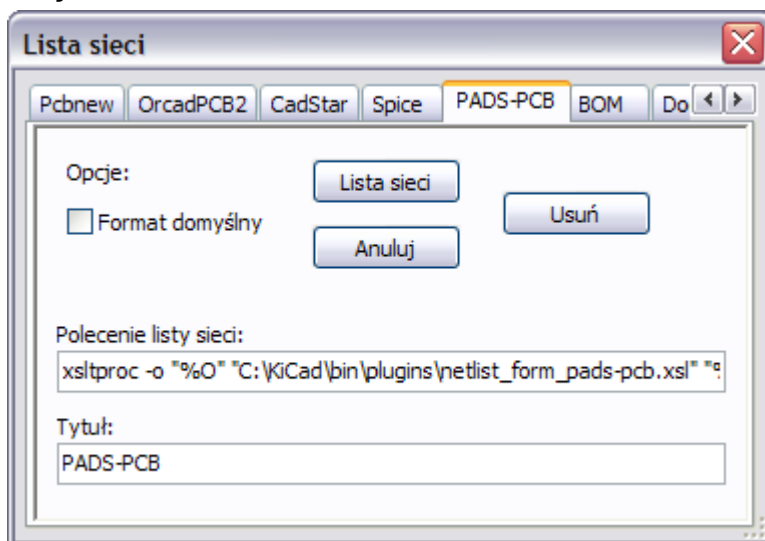
14.3.4.1. Inicjowanie okna dialogowego

Jeśli uruchomimy narzędzie do tworzenia list sieci, to w ostatniej zakładce znajdziemy specjalne narzędzie **Dodaj wtyczkę**, służące do konfigurowania i inicjowania nowych zakładek w tym oknie dialogowym.

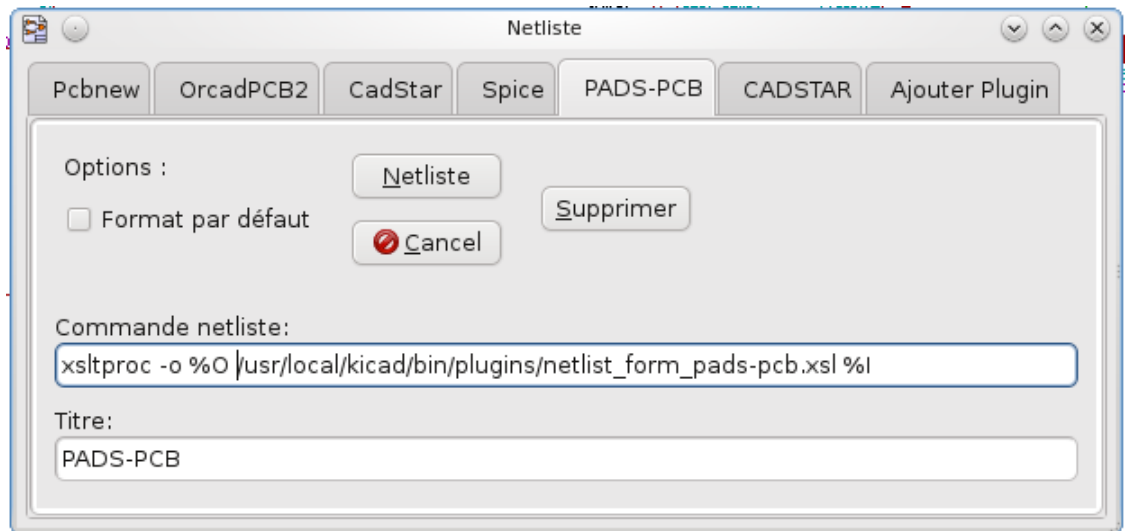


Poniższy obrazek ukazuje skonfigurowaną wtyczkę **PadsPcb**

W systemie Windows:

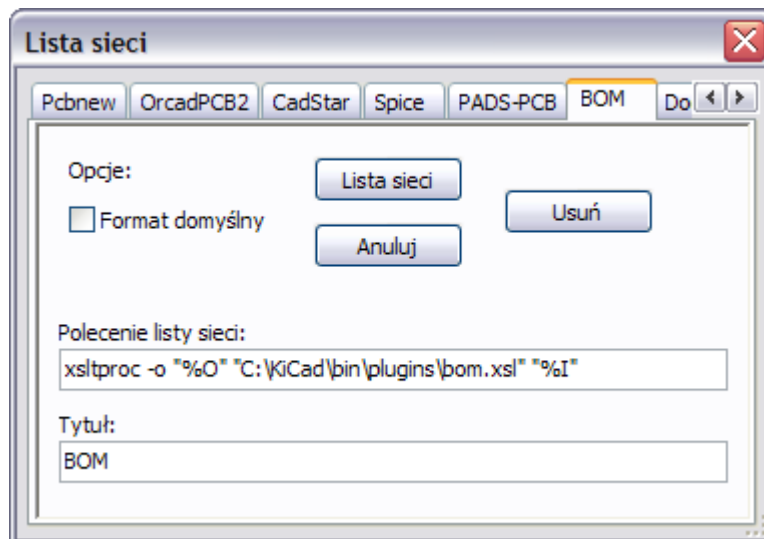


W systemie Linux:



Ponieważ pośredni plik zawiera odpowiednią ilość informacji o komponentach schematu, równie dobrze można z tego pliku stworzyć plik materiałowy BOM (*Bill Of Materials*). Konfiguracja takiego filtra została pokazana na następnym obrazku.

W systemie Windows:



14.3.4.2. Ustawienia nowych wtyczek

Zakładka z ustawieniami wymaga następujących informacji:

- ♦ **Tytułu zakładki** (określająca również nazwę formatu wyjściowego listy sieci)
- ♦ **Lini poleceń**, której przekazanie do systemu operacyjnego uruchomi konwersję.

Przy aktywacji klawisza **Lista sieci** na takiej zakładce:

1. **Eeschema** tworzy pośredni plik z listą sieci *.xml, na przykład test.xml
2. **Eeschema** uruchamia wtyczkę, która czyta plik test.xml i tworzy plik test.net

14.3.4.3. Linia poleceń

Zakładając, że zostanie użyty program xsltproc.exe by zastosować filtr z arkusza stylów na pliku pośrednim listy sieci, xsltproc.exe będzie potrzebował odpowiednio skonstruowaną listę parametrów, zgodnie ze wzorcem:

```
xsltproc.exe -o <output filename> <style-sheet filename> <input XML file to convert>
```

Tak więc, używając systemu Windows linia poleceń przekazana do systemu będzie miała postać :

```
"f:/kicad/bin/xsltproc.exe" -o "%O" "f:/kicad/bin/plugins/netlist_form_pads-pcb.xml" "%I"
```

lub używając systemu Linux :

```
xsltproc -o "%0" "/usr/local/kicad/bin/plugins/netlist_form_pads-pcb.xml" "%I"
```

Plik netlist_form_pads-pcb.xml jest w tym przypadku arkuszem stylu.

Należy pamiętać by nazwy plików zawierające (lub mogące zawierać) znaki spacji zamknąć w cudzysłowach.

14.3.4.4. Format linii poleceń

```
<ścieżka do xsltproc >xsltproc <parametry xsltproc>
```

Jako parametry obsługiwane są sekwencje :

- ◆ %B - nazwa bazowa pliku oraz ścieżka do wybranego pliku wyjściowego, pozbawiona ścieżki oraz rozszerzenia.
- ◆ %I - kompletna nazwa pliku oraz ścieżka do tymczasowego pliku wejściowego.
- ◆ %0 - kompletna nazwa pliku oraz ścieżka do wybranego przez użytkownika pliku wyjściowego.

%I zostanie zastąpione przez nazwę aktywnego w danej chwili pliku pośredniego. %0 zostanie zastąpiony przez nazwę aktywnego w danej chwili pliku wyjściowego (końcowy plik z listą sieci).

Linia poleceń może mieć postać :

- w systemie Windows:
f:/kicad/bin/xsltproc.exe -o %0 f:/kicad/bin/plugins/netlist_form_pads-pcb.xml %I
- w systemie Linux :
xsltproc -o %0 /usr/local/kicad/bin/plugins/netlist_form_pads-pcb.xml %I

Zakładając, że xsltproc został zainstalowany na komputerze (w systemie Windows, wszystkie pliki są w kicad/bin).

14.4. Plik pośredni listy sieci

Poniższy przykład ukazuje ideę samego pliku pośredniego.

```
<?xml version="1.0" encoding="utf-8"?>
<export version="D">
  <design>
    <source>F:\kicad_aux\netlist_test\netlist_test.sch</source>
    <date>29/08/2010 21:07:51</date>
    <tool>eeschema (2010-08-28 BZR 2458)-unstable</tool>
  </design>
  <components>
    <comp ref="P1">
      <value>CONN_4</value>
      <libsource lib="conn" part="CONN_4"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E2141</tstamp>
    </comp>
    <comp ref="U2">
      <value>74LS74</value>
      <libsource lib="74xx" part="74LS74"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E20BA</tstamp>
    </comp>
    <comp ref="U1">
      <value>74LS04</value>
      <libsource lib="74xx" part="74LS04"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E20A6</tstamp>
    </comp>
    <comp ref="C1">
      <value>CP</value>
      <libsource lib="device" part="CP"/>
      <sheetpath names="/" tstamps="/" />
      <tstamp>4C6E2094</tstamp>
    </comp>
    <comp ref="R1">
      <value>R</value>
```

```

<libsource lib="device" part="R"/>
<sheetpath names="/" tstamps="/" />
<tstamp>4C6E208A</tstamp>
</comp>
</components>
<libparts/>
<libraries/>
<nets>
  <net code="1" name="GND">
    <node ref="U1" pin="7"/>
    <node ref="C1" pin="2"/>
    <node ref="U2" pin="7"/>
    <node ref="P1" pin="4"/>
  </net>
  <net code="2" name="VCC">
    <node ref="R1" pin="1"/>
    <node ref="U1" pin="14"/>
    <node ref="U2" pin="4"/>
    <node ref="U2" pin="1"/>
    <node ref="U2" pin="14"/>
    <node ref="P1" pin="1"/>
  </net>
  <net code="3" name="">
    <node ref="U2" pin="6"/>
  </net>
  <net code="4" name="">
    <node ref="U1" pin="2"/>
    <node ref="U2" pin="3"/>
  </net>
  <net code="5" name="/SIG_OUT">
    <node ref="P1" pin="2"/>
    <node ref="U2" pin="5"/>
    <node ref="U2" pin="2"/>
  </net>
  <net code="6" name="/CLOCK_IN">
    <node ref="R1" pin="2"/>
    <node ref="C1" pin="1"/>
    <node ref="U1" pin="1"/>
    <node ref="P1" pin="3"/>
  </net>
</nets>
</export>

```

14.4.1.Struktura ogólna

Plik pośredni listy sieci posiada 5 sekcji :

- ◆ sekcja nagłówka
- ◆ sekcja komponentów
- ◆ sekcja elementów bibliotecznych
- ◆ sekcja bibliotek
- ◆ sekcja sieci połączeń.

Cały plik został objęty w tag <export> :

```

<export version="D">
...
</export>

```

14.4.2.Sekcja nagłówka

Nagłówek znajduje się w tagu <design>

```

<design>
  <source>F:\kicad_aux\netlist_test\netlist_test.sch</source>
  <date>21/08/2010 08:12:08</date>
  <tool>eeschema (2010-08-09 BZR 2439) -unstable</tool>
</design>

```

Sekcja ta może być widoczna jako komentarze.

14.4.3.Sekcja komponentów

Sekcja komponentów zawiera się w tagu <components>

```
<components>
  <comp ref="P1">
    <value>CONN_4</value>
    <libsource lib="conn" part="CONN_4"/>
    <sheetpath names="/" tstamps="/" />
    <tstamp>4C6E2141</tstamp>
  </comp>
</components>
```

Jest to lista na której znajdują się poszczególne komponenty schematu. Każdy komponent jest opisany w następujący sposób :

```
<comp ref="P1">
  <value>CONN_4</value>
  <libsource lib="conn" part="CONN_4"/>
  <sheetpath names="/" tstamps="/" />
  <tstamp>4C6E2141</tstamp>
</comp>
```

libsource	Nazwa biblioteki gdzie można dany komponent odnaleźć.
part	Nazwa komponentu wewnątrz tej biblioteki.
sheetpath	Ścieżka do arkusza wewnątrz hierarchii. Identyfikuje arkusz w całej hierarchii.
tstamps (time stamps)	Odcisk czasowy dla pliku schematu.
tstamp (time stamp)	Odcisk czasowy dla komponentu.

14.4.3.1. Uwagi na temat odcisków czasowych dla komponentów

Aby zidentyfikować składnik listy sieci (a także na płytce), jest używane jego oznaczenie i jest ono unikalne dla każdego z komponentów. Jednak **KiCad** udostępnia pomocniczą informację by jednoznacznie zidentyfikować komponent i odpowiadający mu moduł na płytce. Pozwala to na ponowną numerację symboli na schemacie w taki sposób by nie utracić powiązań pomiędzy komponentem i jego modulem.

Znacznik czasowy jest unikalnym identyfikatorem dla każdego składnika lub arkusza schematu w projekcie. Jednak w złożonych hierarchiach, w tym samym arkuszu składnik może być używany więcej niż raz, a zatem arkusz ten zawiera elementy o tym samym znaczniku czasowym.

Dany arkusz (wewnątrz złożonej hierarchii) ma jednak unikalny identyfikator: jego ścieżka wewnętrzna. Dany składnik zaś (wewnątrz złożonej hierarchii) ma unikalny identyfikator: ścieżka wewnętrzna + jego odcisk czasowy.

14.4.4. Sekcja elementów bibliotecznych

Sekcja elementów bibliotecznych znajduje się w tagu <libparts>, a dane w tej sekcji są zdefiniowane w bibliotekach schematu. Dla każdego komponentu sekcja ta zawiera dane :

- ◆ Nazwy dozwolonych footprintów (nazwy używają masek), zawarte w tagu <fp>.
- ◆ Pola zdefiniowane w bibliotece, zawarte w tagu <fields>.
- ◆ Lista pinów, zawarte w tagu <pins>.

```
<libparts>
  <libpart lib="device" part="CP">
    <description>Condensateur polarise</description>
    <footprints>
      <fp>CP*</fp>
      <fp>SM*</fp>
    </footprints>
    <fields>
      <field name="Reference">C</field>
      <field name="Valeur">CP</field>
    </fields>
    <pins>
      <pin num="1" name="1" type="passive"/>
      <pin num="2" name="2" type="passive"/>
    </pins>
```



```
</libpart>
</libparts>
```

Linie jak `<pin num="1" type="passive"/>` określają również typ elektryczny pinów. Dostępne są typy :

Input	Zwykły pin wejściowy
Output	Zwykły pin wyjściowy
Bidirectional	Wejście lub wyjście
Tri-state	Wejście lub wyjście trójstanowe
Passive	Pasywny, zwykle w komponentach biernych.
Unspecified	Nieznany
Power input	Wejście zasilania dla komponentu
Power output	Wyjście zasilania z komponentu
Open collector	Otwarty kolektor
Open emitter	Otwarty emiter
Not connected	Musi być pozostawiony niepodłączony

14.4.5.Sekcja bibliotek

Sekcja bibliotek znajduje się w tagu `<libraries>`. Dostarcza ona listę bibliotek używanych w danym projekcie schematu.

```
<libraries>
  <library logical="device">
    <uri>F:\kicad\share\library\device.lib</uri>
  </library>
  <library logical="conn">
    <uri>F:\kicad\share\library\conn.lib</uri>
  </library>
</libraries>
```

14.4.6.Sekcja sieci

Sekcja sieci znajduje się w tagu `<nets>`. Zawiera ona listę wszystkich połączeń na schemacie.

```
<nets>
  <net code="1" name="GND">
    <node ref="U1" pin="7"/>
    <node ref="C1" pin="2"/>
    <node ref="U2" pin="7"/>
    <node ref="P1" pin="4"/>
  </net>
  <net code="2" name="VCC">
    <node ref="R1" pin="1"/>
    <node ref="U1" pin="14"/>
    <node ref="U2" pin="4"/>
    <node ref="U2" pin="1"/>
    <node ref="U2" pin="14"/>
    <node ref="P1" pin="1"/>
  </net>
</nets>
```

Poszczególne sieci są pogrupowane wewnątrz tagu `<net>` :

```
<net code="1" name="GND">
  <node ref="U1" pin="7"/>
  <node ref="C1" pin="2"/>
  <node ref="U2" pin="7"/>
  <node ref="P1" pin="4"/>
</net>
```

net code	Jest to wewnętrzny identyfikator danej sieci
----------	--

name	Jest to nazwa danej sieci
node	Zawiera odnośniki do poszczególnych pinów w danej sieci

14.5. Więcej informacji na temat xsltproc

Aby uzyskać więcej informacji na temat programu, polecamy zajrzeć na stronę [www: http://xmlsoft.org/XSLT/xsltproc.html](http://xmlsoft.org/XSLT/xsltproc.html)

14.5.1. Wprowadzenie

xsltproc jest narzędziem uruchamianym z linii poleceń do filtrowania za pomocą arkuszy stylów XSLT dokumentów XML. Jest on częścią libxslt, biblioteki *XSLT C Library* przeznaczonej dla GNOME. Chociaż powstała ona jako część projektu GNOME, może również działać niezależnie od GNOME.

xsltproc jest wywoływany z linii poleceń z podaną nazwą arkusza stylów do wykorzystania, a następnie z nazwą pliku lub plików, do którego arkusz stylów ma być zastosowany. Jeśli nazwa pliku wejściowego nie będzie podana, czyli parametr `-i` nie zostanie użyty, będzie wykorzystane standardowe wejście.

Jeśli arkusz stylów jest wbudowany w dokument XML z instrukcjami *Stylesheet Processing Instruction*, nie będzie trzeba dodatkowo podawać nazwy arkusza stylów w linii poleceń. xsltproc automatycznie wykryje i użyje zawartych stylów.

Domyślnie dane wyjściowe zostaną skierowane na standardowe wyjście. Można jednak określić plik wyjściowy przy użyciu opcji `-o`.

14.5.2. Synopsis

```
xsltproc [[-V] | [-v] | [-o file] | [--timing] | [--repeat] | [--debug] | [--novalid] | [--noout] | [--maxdepth val] | [--html] | [--param name value] | [--stringparam name value] | [--nonet] | [--path paths] | [--load-trace] | [--catalogs] | [--xinclude] | [--profile] | [--dumpextensions] | [--nowrite] | [--nomkdir] | [--writesubtree] | [--nodtdattr]] [stylesheet] [file1] [file2] [...]
```

14.5.3. Opcje linii poleceń

`-V` lub `--version`

Pokazuje używaną wersję libxml i libxslt.

`-v` lub `--verbose`

Pokazuje każdy krok wykonany przez xsltproc podczas przetwarzania arkusza stylów i dokumentów.

`-o` lub `--output file`

Przekierowuje wyjście do pliku o nazwie file. Dla wyjść wielokrotnych, zwanych także jako "chunking", `-o directory/` przekierowuje pliki wyjściowe do określonego katalogu. Katalog ten musi być wcześniej utworzony.

`--timing`

Pokazuje czas zużyty na przetworzenie arkusza stylów, przetworzenia dokumentu oraz zastosowania arkusza stylów, a także czas zapisu danych wynikowych. Wartości pokazywane są milisekundach.

`--repeat`

Uruchamia transformację 20 razy. Używane przy testach czasowych.

`--debug`

Pokazuje drzewo XML transformowanego dokumentu w celu usuwania usterek w oprogramowaniu.

`--novalid`

Opuszcza ładowanie dokumentów DTD.

`--noout`

Nie generuje danych wyjściowych.

`--maxdepth value`

Określa maksymalną głębokość stosu wzorców, przed stwierdzeniem o wejściu libxslt do nieskończonej pętli. Domyślnie jest to 500.

`--html`

Dokument wejściowy jest plikiem HTML.

- -param name value
Przekazuje parametr name i wartość value do arkusza stylów. Można przekazać wiele par nazwa/wartość, jednak nie więcej niż 32. Jeśli wartość przekazywana jest łańcuchem a nie identyfikatorem węzła, należy użyć --stringparam zamiast tej opcji.
- -stringparam name value
Przekazuje parametr name i wartość value gdzie value jest łańcuchem znaków a nie identyfikatorem węzła. (Uwaga : Ciąg musi posiadać znaki kodowane w UTF-8.)
- -nonet
Zabrania użycia sieci Internet w celu pobrania DTD, podmiotów lub dokumentów.
- -path paths
Używa listy (separowanej za pomocą spacji lub przecinków) ścieżek systemu plików określonych przez paths w celu załadowania DTD, podmiotów lub dokumentów.
- -load-trace
Wysyła na stderr wszystkie dokumenty ładowane podczas przetwarzania.
- -catalogs
Używa katalogu SGML określonego w SGML_CATALOG_FILES by określić lokalizację zewnętrznych podmiotów. Domyślnie, xsltproc zagląda do katalogu określonego w XML_CATALOG_FILES. Jeśli nie jest to określone, używa /etc/xml/catalog.
- -xinclude
Przetwarza dokumenty wejściowe używając specyfikacji Xinclude. Więcej szczegółów na ten temat można znaleźć na stronie Web specyfikacji Xinclude: <http://www.w3.org/TR/xinclude/>
- -profile lub - -norman
Zwraca sprofilowane informacje na temat czasu spędzonego w każdej części arkusza stylów. Jest to przydatne w optymalizacji wydajności arkuszy stylów.
- -dumpextensions
Zwraca listę wszystkich zarejestrowanych rozszerzeń na stdout.
- -nowrite
Odrzuca polecenia tworzenia plików lub zasobów.
- -nomkdir
Odrzuca polecenia utworzenia katalogów.
- -writesubtree path
Pozwala na zapis tylko do wybranej podgałęzi path.
- -nodtdattr
Nie stosuje domyślnych atrybutów pochodzących z dokumentów DTD.

14.5.4.Zwracane wartości

xsltproc zwraca także kody błędów, których można użyć w przypadku wywołań programu wewnątrz skryptów :

- 0 : normalne zakończenie
- 1 : brak argumentu
- 2 : za dużo parametrów
- 3 : opcja nieznana
- 4 : niepowodzenie przy parsowaniu arkusza stylów
- 5 : błąd arkusza stylu
- 6 : błąd w jednym z dokumentów
- 7 : nieobsługiwana metoda xsl:output
- 8 : parametry w postaci ciągów zawierają zarówno znaki apostrofów jak i cudzysłowów
- 9 : błąd wewnętrzny
- 10 : przetwarzanie zostało zatrzymane przez komunikat o przerwaniu
- 11 : nie można zapisać danych wyjściowych do pliku wyjściowego

14.5.5. Więcej informacji

Strona Web libxml <http://www.xmlsoft.org/>

Strona Web W3C XSLT <http://www.w3.org/TR/xslt>