

Autorouter Specctra

część 2

W drugim artykule na temat autoroutera Specctra postaramy się przybliżyć jej środowisko GUI, opisać zadania plików wykorzystywanych przez autorouter, powiemy też kilka słów na temat sposobów sterowania pracą autoroutera.

Zanim rozpoczniemy prezentację interfejsu użytkownika Specctry, kilka słów na temat uruchamiania tego programu. Specctra może stanowić rozszerzenie każdego systemu do projektowania PCB, zapewniając funkcje związane z interaktywnym i automatycznym rozmieszczaniem elementów oraz wytyczaniem ścieżek. Ponieważ zazwyczaj każdy system PCB zapisuje informacje w swoim własnym, unikalnym formacie, przed przejściem do Specctry konieczne jest dokonanie odpowiedniej konwersji (rys. 1). Podobnie rzecz się ma podczas przekazywania danych w kierunku odwrotnym, czyli z autoroutera do edytora PCB. Ze względu na silną pozycję Specctry na rynku, właściwie wszyscy producenci systemów do projektowania PCB starają się wyposażać swoje produkty w odpowiednie konwertery służące do wymiany danych z zewnętrznym autorouterem. Niektóre z nich posiadają nawet wbudowane funkcje służące do wymiany danych do i ze Specctry, a nawet do automatycznego wywoływania tego programu z poziomu własnego GUI. Pliki czytane i zapisywane przez Specctrę są plikami „czysto” tekstowymi, więc jest możliwe ich przeglądanie za pomocą zwykłego windowsowego Notatnika.

Najważniejsze z nich to:

Design file (.dsn) - tworzony przez translator podczas konwersji pliku z edytora PCB do Specctry. Zawiera

wszystkie niezbędne informacje służące do prawidłowej interpretacji projektu przez autorouter, takie jak: kształt i wymiary płytki, definicje warstw, definicje padów, dane elementów, lista połączeń oraz reguły projektowe narzucone przez użytkownika w edytorze PCB.

Session file (.ses) - tworzony przez autorouter. Zawiera wskaźnik do oryginalnego pliku projektu (.dsn), historię poprzednich sesji, w tym informacje dotyczące rozmieszczenia elementów na płycie i wytyczonych ścieżkach.

Routes file (.rte) - tworzony przez autorouter zawiera informacje dotyczące połączeń wykonanych przez Specctrę. Zbiór może być konwertowany i przekazywany z powrotem do systemu PCB użytkownika, może również być czytany przez autorouter.

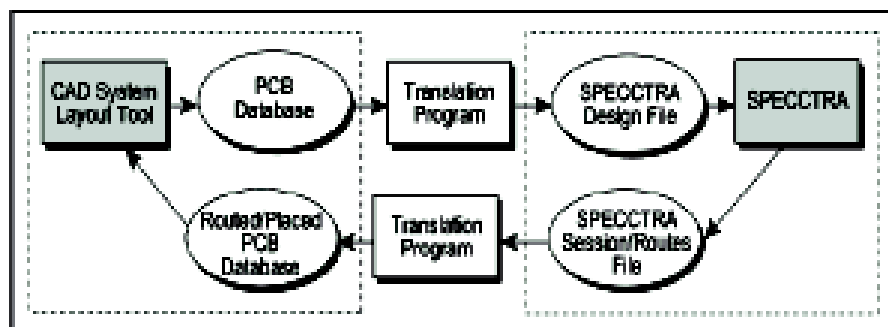
Wires file (.w) - tworzony przez autorouter, zawiera informacje o wytyczonych połączeniach. Plik może być czytany jedynie przez autorouter.

W przypadku systemów PCB ze zintegrowanym odwołaniem do Specctry konwersja i przekazanie projektu do Specctry odbywa się automatycznie po użyciu odpowiedniego polecenia bezpośrednio w module PCB.

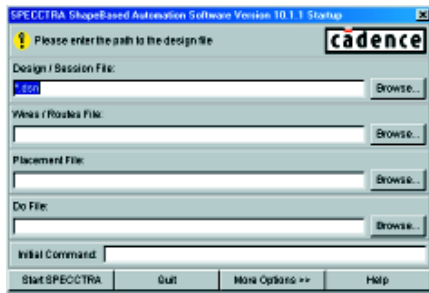
W przypadku uruchamiania Specctry jako niezależnej aplikacji konieczne jest wskazanie pliku projektu, z którym chcemy aktualnie pracować. Może to być plik typu

*.dsn, *.ses lub *.cct. Odpowiednie okno pozwalające na wskazanie ścieżki dostępu tego pliku pokazano na rys. 2. Uruchamiając projekt możemy jednocześnie wskazać pliki zawierające informacje o operacjach, które zostały już wykonane w poprzednich sesjach dotyczące np. rozmieszczeniu elementów czy połączeniach wytyczonych na określonym etapie projektowania, oraz do tzw. pliku DO, którego rolę opisujemy w dalszej części artykułu.

Główne okno programu (rys. 3), zawiera typowy dla Windows pasek menu pozwalający na dostęp do poszczególnych funkcji autoroutera oraz pasek narzędzi w postaci ikon umożliwiający szybki dostęp do podstawowych poleceń programu. Centralną część okna stanowi obszar roboczy, w ramach którego projekt jest wizualizowany i gdzie dokonywana jest jego „obróbka”. Poniżej obszaru roboczego znajduje się pasek poleceń zawierający pole służące do wprowadzania komend sterujących bezpośrednio z klawiatury (więcej o tym za chwilę) oraz pełniący rolę informacyjną. Dolną część głównego okna programu stanowi tzw. *output window*. W postaci tekstowej prezentowane są tu szczegółowe dane dotyczące wszystkich czynności (wydanych poleceń, wykonanych operacji) jakie miały miejsce w danej sesji (łącznie z ich wynikiem), informacje odnośnie ewentualnych błędów jakie pojawiły się podczas pracy oraz raporty końcowe dotyczące np. rezultatów *routingu* - rys. 4. Dzia-



Rys. 1



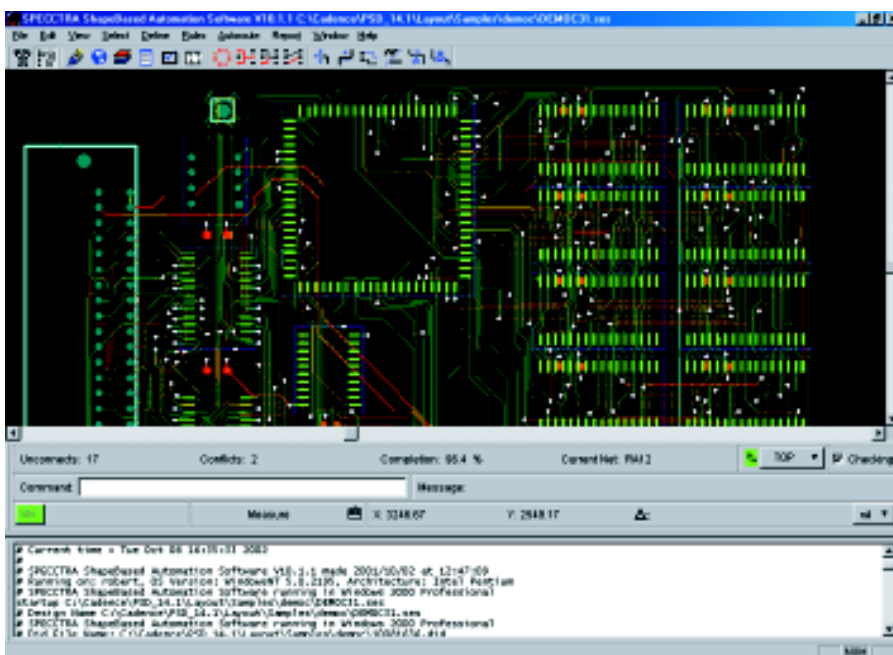
Rys. 2

łanie Specctry może być kontrolowane na trzy sposoby:

- bezpośrednio z poziomu interfejsu graficznego GUI, czyli w sposób typowy dla większości programów windowsowych,
- poprzez podawanie odpowiednich komend z klawiatury,
- za pomocą pliku *DO*.

W artykule skupimy się na pierwszym z wyżej wspomnianych sposobów kontrolowania autoroutera, który sprowadza się do projektowania przy wykorzystaniu interfejsu graficznego Specctry. Należy tutaj dodać, że działanie autoroutera można również bardzo precyzyjnie kontrolować za pomocą poleceń wydawanych w wewnętrznym języku Specctry, które mogą być wydawane pojedynczo z wiersza poleceń programu lub sekwencyjnie za pomocą pliku *DO*. Wszystkie trzy metody mogą być dowolnie przeplatanie (łączone) podczas pojedynczej sesji. W przypadku sterowania pracą programu z klawiatury, polecenia wpisywane są po prostu

w wierszu poleceń, który znajduje się w środkowej części okna głównego - **rys. 5**. Oczywiście wykorzystywanie tej metody sterowania autorouterem uwarunkowane jest znajomością znaczenia poszczególnych komend. W przypadku sterowania Specctrą za pomocą pliku *DO* konieczna jest również znajomość składni języka Specctry. *DO file* jest zwykłym plikiem tekstowym, który zawiera ciąg określonych przez użytkownika poleceń sterujących. Przykład jego zawartości pokazano na **rys. 6**. Każda linia pliku zawiera pojedynczą komendę. Polecenia wykonywane są sekwencyjnie - kolejno od początku do końca pliku. Niesie to za sobą konieczność przemyślenia kolejności wykonywania poszczególnych operacji. W przypadku natrafienia na błędy składni, które z takich czy innych względów zaistnieją w pliku *DO*, działanie routera zostaje wstrzymane. Dodajmy jeszcze, że wykonanie poleceń zawartych w omawianym pliku może być wyegzekwowane na dowolnym etapie projektowania (w dowolnym momencie sesji) lub już w trakcie uruchamiania Specctry. W tym drugim przypadku w omawianym wcześniej oknie Startup (rys. 2) należy po prostu wskazać odpowiednią ścieżkę dostępu do *DO file*. Korzyści płynące ze stosowania pliku *DO* to możliwość pełnej automatyzacji działania autoroutera, bardzo precyzyjna kontrola pracy programu oraz



Rys. 3

```
#PDSING STATISTICS
PCB Area=14621.487  EPC=41  Area/PCB=218.090
Component=0  DRCount=1
Trace Length=1798.822  Horizontal=428.772  Vertical=1798.822
Wire Junctions=144  et vias=0  Total Vias=990
Coefficients=0  Crossings=0  Clearances=0  Crests=0  Length=0
MaxDelta Length=1151.131  Horizontal=5747.181  Vertical=4830.047
Ratio Actual / Maximum=0.057
Unconnected Length= 0.000  Horizontal= 0.000  Vertical= 0.000
Layer Direct File Vias 720  Coefficients Length Horizontal Vertical
TOP Area 428 180 41 0 4481.678 3430.729 4070.714
NETSIN Vias 232 390 16 0 4481.678 1314.451 3952.731
```

Rys. 4

możliwość ponownego wykorzystania procedury (lub jej części) zawartej w pliku podczas pracy nad kolejnym projektem.

Najprostszym i najczęściej stosowanym sterowania pracą autoroutera jest wykorzystanie jego interfejsu graficznego. Specctra nie odbiega tutaj zasadniczo od innych aplikacji przeznaczonych dla Windows. Sprawne posługiwanie się programem wymaga oczywiście zaznajomienia się ze znaczeniem poszczególnych funkcji i okien, których ze względu na rozbudowane możliwości Specctry nie jest wcale mało. Aplikacja może pracować w dwóch trybach głównych: *Place Mode* oraz *Route Mode*. Jak łatwo się domyśleć *Place Mode* służy do projektowania płytki pod kątem rozmieszczenia elementów, natomiast w trybie *Route Mode* mamy możliwość wytyczania połączeń elektrycznych. W zależności od tego w jakim trybie aktualnie znajduje się Specctra, zmienia się nieco wygląd głównego okna programu, w którym wizualizowany jest projekt. Dotyczy to przede wszystkim funkcji jakie dostępne są w menu głównym oraz ikon znajdujących się powyżej obszaru roboczego. Wynika to oczywiście z faktu, że zestaw funkcji dostępnych podczas rozmieszczania elementów różni się od tego jaki mamy do dyspozycji w trakcie wytyczania ścieżek.

Procedura układania elementów na projektowanej płytce jest identyczna dla wszystkich programów wspomagających projektowanie i składa się z czterech podstawowych etapów: ustalenia reguł projektowych, rozmieszczenia elementów krytycznych, rozmieszczenia elementów dużych (o dużej liczbie wyprowadzeń) oraz rozmieszczenia elementów małych (o małej liczbie wyprowadzeń). Ustalenie zasad odnośnie rozmieszczania elementów jest pierwszym krokiem, który należy wykonać zanim przystąpimy do właściwych prac projektowych. Specctra kontroluje bowiem wyniki naszych „posunięć” zarówno podczas interaktywnego jak i automatycznego układania komponentów



Rys. 5

i sygnalizuje ewentualne przekroczenia zadanych wcześniej reguł. Autorouter zapewnia obszerny zestaw zasad projektowych związanych z rozmieszczaniem elementów. Podstawowe z nich umożliwiają kontrolę odległości między komponentami, kontrolę rozkładu elementów na poszczególne warstwy płytki czy kontrolę orientacji komponentów. Możemy jednak określać również bardziej zaawansowane reguły pozwalające na kontrolę rozmieszczenia elementów na przykład ze względu na konieczność podłączenia do konkretnej ścieżki zasilającej czy też od innych uwarunkowań takich jak np. wysokość komponentu. Zestaw reguł może być ponadto specyfikowany na różnych poziomach zgodnie z hierarchią narzuconą przez twórców Specctry. Reguły z wyższego poziomu „nadpisują” te, które plasują się niżej w hierarchii nawet jeśli fizycznie dotyczą one tego samego obiektu. To znaczy, że po ustawieniu na przykład *global spacing* na 0,25 cala dla wszystkich elementów i 0,8 cala dla wybranej grupy z nich, autoruter zastosuje odstępy 0,8 cala ale tylko na obszarze na którym znajduje się wybrana grupa elementów. Hierarchia poziomów na których stosowane mogą być reguły projektowe dotyczące rozmieszczenia elementów wygląda w Specctrze w następujący sposób:

- *image_image* - zestaw reguł pomiędzy określonymi obrazami (*footprintami*) elementów (najwyższy poziom w hierarchii),
- *family_family* - zestaw reguł pomiędzy rodzinami (grupami) elementów,
- *room_image_set* - zestaw reguł dla grupy *footprints* o podobnych właściwościach (np. z dużą liczbą wyprowadzeń, z małą liczbą wyprowadzeń), które będą stosowane na określonym obszarze płytki,

```
# Lines beginning with '#' are comments
# General purpose do file
# Initial Commands
bestsave on bestsave.vre
status_file route.stx
unit all
grid smart (vire 1) (via 1)
# Standard Routing Commands
smart_route
```

Rys. 6

- *room* - zestaw reguł, które będą stosowane na określonym przez użytkownika obszarze płytki niezależnie od rodzaju elementów jakie się na nim znajdują,
- *super_cluster* - zestaw reguł dla grupy komponentów, które połączone są w tzw. super cluster i traktowane przez autorouter jako jeden odrębny element,
- *component* - zestaw reguł dla określonego elementu,
- *image* - zestaw reguł dla określonego *footprintu*. Pojęcia *image* i *component* są pojęciami odrębnymi. *Component* określa konkretnie ten a nie inny element projektu. *Image* oznacza widok typu obudowy (*footprint*), który może być wspólny dla kilku różnych elementów,
- *image_set* - zestaw reguł dla grupy *footprints* o podobnych właściwościach,
- *pcb* - globalny zestaw reguł stosowany dla projektu (najniższy poziom w hierarchii).

RK

Dodatkowe informacje

Więcej informacji można uzyskać w firmie RK-System, www.rk-system.com.pl.