
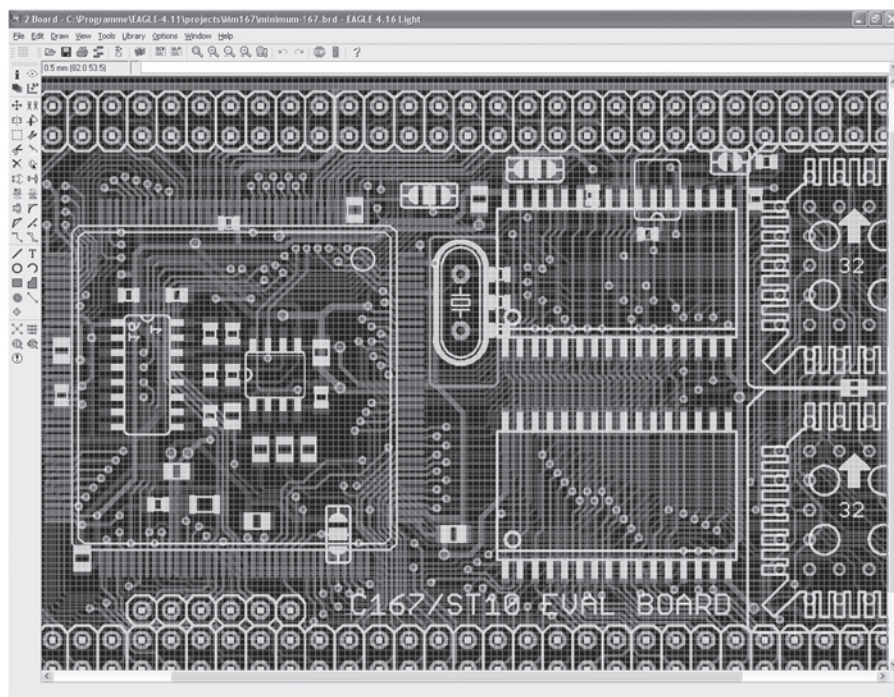


# Kurs obsługi EAGLE, część 5

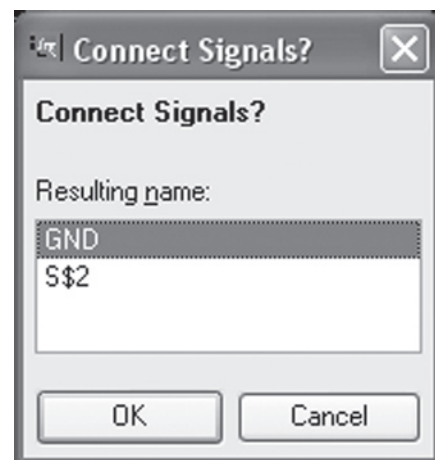
Ten odcinek kursu poświęcamy praktycznie w całości opisowi modułu autoroutera. Jest to druga i zarazem ostatnia część opisująca proces projektowania płytki. Obok autoroutera przedstawiamy także możliwości funkcji Polygon, dzięki której możemy kreślić figury o dowolnym kształcie.

Projektując płytkę często zachodzi potrzeba umieszczenia na jednej z warstw obszarów w pełni pokrytych miedzią. Jeżeli obszary te zostaną podłączone do istniejących już węzłów elektrycznych (przykładowo: GND, VCC, +5 V), to mogą one pełnić rolę ekranu lub zasilania. Obszary te można zastosować również w celach czysto ekonomicznych lub ekologicznych, aby zaoszczędzić środek trawiący, który zostanie użyty w procesie produkcji płytki.

Aby umieścić na płycie potrzebne nam obszary możemy użyć zwykłych narzędzia graficzne (*Wire*, *Circle*, *Arc*, *Rect*). Figurom powstałym w ten sposób nie możemy jednak nadać nazw (nie podłączymy ich do istniejących już sygnałów), nie podlegają one również zasadom określonym w DRC. Lepszą metodą jest określenie niezbędnych nam obszarów przy pomocy wielokąta – **POLYGON**. Polecenie to jest dostępne po naciśnięciu ikony . Po jej przyciśnięciu na pasku *parameter-toolbar* pojawiają się narzędzia, dzięki którym możemy zmienić poszczególne parametry kreślonej przez nas figury (**rys. 28**). Na początku musimy określić płaszczyznę, na której będziemy kreślić. Dokonujemy tego w polu *Select Layer* lub klikając środkowym klawiszem myszy. Potrzebny nam kształt rysujemy podobnie jak linię (polecenie **WIRE**). Na zagięciach klikamy lewym klawiszem, aby zamknąć figurę klikamy dwukrotnie. Sposób zagięcia możemy zmienić w polu *Wire Bend* lub klikając pra-



wym klawiszem myszy. W czasie rysowania *polygonu* musimy uważać, aby kontury na siebie nie nachodziły, oraz aby się nie przecinały. W przeciwnym wypadku program nie będzie mógł go prawidłowo obliczyć, o czym poinformuje nas komunikatem o błędzie. *Polygony* znajdujące się na innej niż górna lub dolna strona płytki są obliczane natychmiast po narysowaniu. *Polygony* umieszczone w warstwach miedzi (w naszym przypadku *Layer 1- Top* oraz *Layer 16- Bottom*) są symbolizowane jedynie przez obwódkę, aby je wyliczyć musimy użyć komendy **RATSNEST** (w ustawieniach *Options/Set.../Misc* musi być wcześniej zaznaczona opcja *Ratsnest processes polygons*). Jeżeli chcemy z wyliczonej już figury uzyskać tylko obwódkę (*polygon* może nam przysłać pewne elementy projektu, poza tym spowalnia odświeżanie ekranu) używamy komendy **RIPUP**, po czym klikamy na jej brzeg. Aby ograniczyć wielkość pliku zawierającego projekt płytki, wypełnienie *polygonu* nie jest w nim zapisywane. Tak więc po zamknięciu programu i jego po-



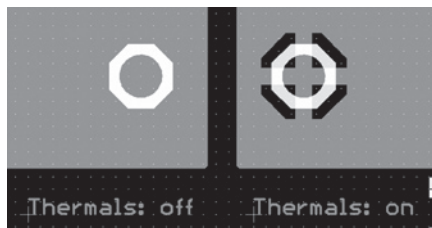
Rys. 29.

nownym uruchomieniu widzimy tylko kontury, a nie całą figurę.

Po narysowaniu *polygonu* należy nadać mu nazwę identyczną z sygnałem, do którego chcemy go podłączyć (przykładowo GND). Do tego użyjemy komendy **NAME**, po czym klikniemy jego krawędź. Program zapyta nas o nazwę, po jej wpisaniu zostaniemy poproszeni o potwierdzenie połączenia sygnałów (**rys. 29**) klikamy **OK**. W ten sposób wszystkie elementy znaj-



Rys. 28.



Rys. 30.

dujące się w obrębie *polygonu*, do których przyłączone są sygnały mające identyczną nazwę jak on, są do niego automatycznie podłączone. Oznacza to, iż nie musimy rysować odchodzących od nich ścieżek, program podłączy je automatycznie podczas obliczania *polygonu*.

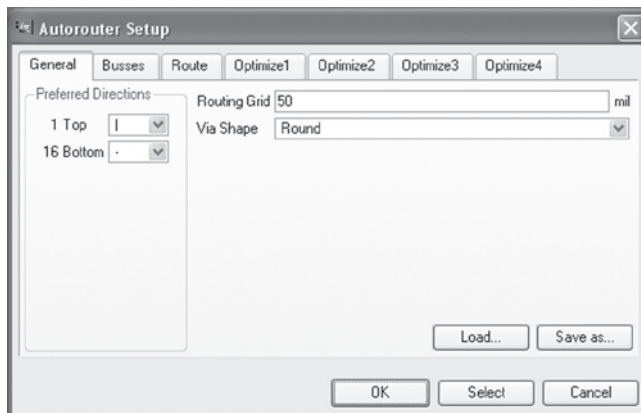
Na pasku *parameter-toolbar* oprócz wcześniej wymienionych elementów znajdziemy kolejne, symbolizują one następujące funkcje:

*Width* – szerokość linii, którą *polygon* jest kreślony. Należy sobie uzmysłowić, iż zaznaczony przez nas kontur nie jest zalewany, lecz kreskowany pisakiem, którego szerokość możemy w szerokim zakresie zmienić. *Polygony* kreślone cienkim pisakiem są dokładniejsze, jednakże pliki wyjściowe do produkcji (Gerber) mają większą objętość. Nie należy ustawiać grubości mniejszej niż minimalna szerokość ścieżek, ponieważ określa ona również minimalną szerokość miedzi na poligonie (przykładowo pomiędzy dwoma przelotkami) i jeżeli była by

za mała, to nie mamy gwarancji iż połączenie będzie na gotowej płytce rzeczywiście istniało (gdy jest za wąskie to może zostać przetrawione w czasie procesu produkcji). Test DRC sprawdza szerokość linii jaką jest rysowany *polygon*, gdy jest za wąska poinformuje nas o tym błędnie.

*Pour* – opcja ta pozwala na określenie rodzaju wypełnienia. Do wyboru mamy dwie możliwości: *Solid* – pełna powierzchnia, lub *Hatch* – wypełnienie w kratkę.

*Thermals* – umożliwia wybranie sposobu podłączenia padów. Aby parametr ten miał wpływ również na przelotki, w ustawieniach DRC należy zaznaczyć okienko *Generate thermals for vias*. Możemy wybrać całkowite połączenie do powierzchni miedzi, lub przy pomocy wąskich łączników (rys. 30). W przypadku pierwszej metody, w czasie lutowania, od pola lutowniczego jest odprowadzana duża ilość ciepła. Może to doprowadzić do nieprawidłowego lub zimnego lutu. Poza tym utrudnione jest późniejsze wylutowanie elementu (przy-



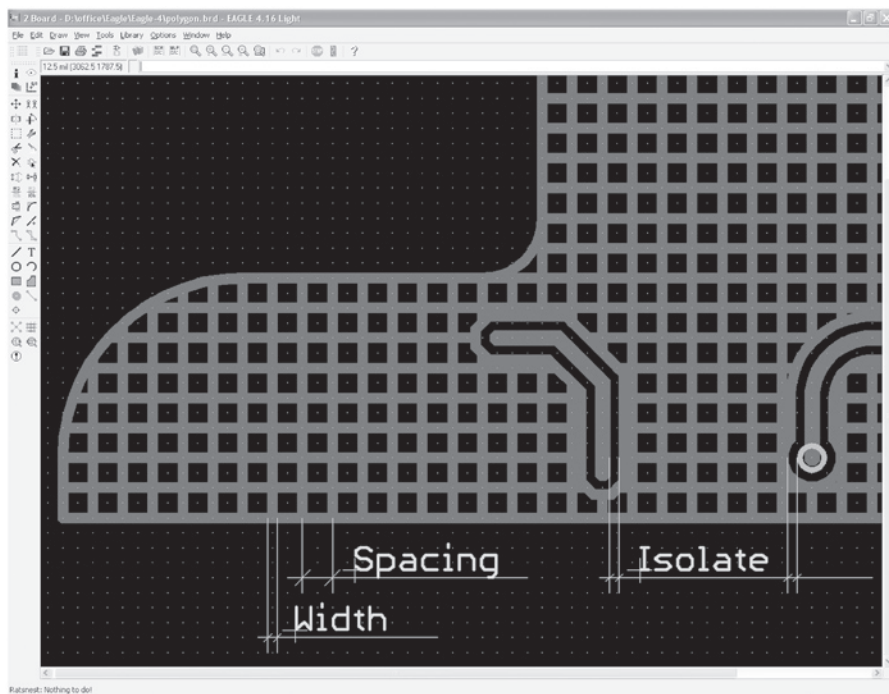
Rys. 32.

kładowo w czasie naprawy). Z tych powodów zalecana jest metoda druga. Szerokość łącznika jest obliczana automatycznie i ma wartość połowy średnicy otworu w danym padzie lub przelotce. Przy czym nie jest mniejsza od szerokości, oraz nie jest większa od podwójnej szerokości linii, którą *polygon* jest kreślony.

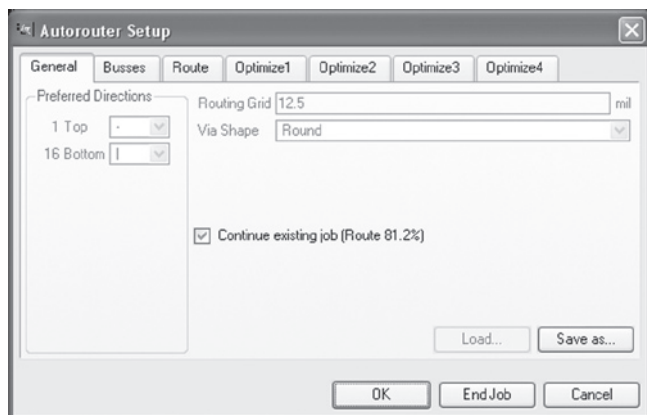
*Orphans* – w czasie prowadzenia ścieżek może się zdarzyć, iż *polygon* zostanie pocięty na mniejsze wysepki. Jeżeli w którejś z wysepek nie znajduje się pin lub przelotka, których sygnał jest podpięty do danego *polygonu* to program jej nie obliczy, pozostawiając puste pole. Aby zostały obliczone wszystkie wysepki należy parametr ten zmienić na *ON*. W niektórych przypadkach, gdy w obrębie *polygonu* nie ma żadnego do niego podpiętego elementu, to zostanie on również nieobliczony. Pozostanie tylko kontur, który widoczny jest jedynie na ekranie, na wydruku lub w plikach gerber wykonanych za pomocą procesora CAM go nie będzie. Ustawienie tej opcji na *ON* (obliczanie wszystkich wysepek) jest używane głównie do oszczędzania środka trawiącego, używanego podczas produkcji płytki.

*Isolate* – określa szerokość izolacji pomiędzy *polygonem* a innymi elementami posiadającymi inne potencjały niż on. Pod uwagę brane są również wartości ustawione w opcjach DRC oraz *Net Classes*. Szerokość izolacji jest równa najwyższej, z tych trzech wartości (rys. 31).

*Spacing* – jeżeli wypełnienie *polygonu* jest ustawione na *Hatch* (kratka), to parametr ten określa odstęp pomiędzy środkami linii kratki (rys. 32).



Rys. 31.



Rys. 33.

*Rank* – w przypadku gdy kilka *polygonów* o różnych potencjałach nachodzi na siebie, możemy przy pomocy tego parametru określić kolejność ich obliczania. *Polygon*, dla którego wartość *Rank* jest ustawiona na 1 ma najwyższy priorytet, zostanie on obliczony jako pierwszy i nie zostanie od niego nic odejęte. Ten, którego wartość wynosi 6 zostanie obliczony na końcu, jego kształt zostanie pomniejszony o obszar zajęty już przez inne *polygony*.

Za pomocą opisanych powyżej elementów można zmienić parametry tworzonego *polygonu*. Istniejący możemy zmienić używając komendy *Change*, po czym kolejno *Layer...*, *Width*, *Pour*, *Rank*, *Isolate*, *Spacing*, *Thermals* oraz *Orphans*

## Autorouter

Kolejnym elementem Eagle'a, który dzisiaj omówimy jest wbudowany w niego autorouter. Jego możliwości zostały opisane w pierwszej części artykułu, nie będziemy więc do nich teraz wracać. Autorouter wbudowany w Eagle'a należy do grupy routerów 100-procentowych. Określenie to oznacza, iż na płytce, na której teoretycznie można rozmieścić wszystkie ścieżki, wszystkie ścieżki zostaną przez autorouter rozmieszczone. Jedynym ograniczeniem jest to (i tu jest „haczyk”), że program ma nieskończenie wiele czasu. Ponieważ w rzeczywistości nie mamy nieskończonej ilości czasu, dlatego w niektórych projektach autorouter nie będzie w stanie poprowadzić wszystkich ścieżek. Autorouter działa na zasadzie *Ripup/Retry*, czyli w momencie gdy już nie ma miejsca na położenie kolejnej ścieżki, rozłącza (usuwa) ścieżki już położone, po-

czym próbuje je rozmieścić ponownie. Liczba ścieżek, które może w tym procesie rozłączyć (*RipupLevel*), jest określona przez użytkownika i ma bardzo duży wpływ na czas potrzebny programowi do zaprojektowania płytki. Od autoroutera nie należy oczekiwać, że wykona idealną

płytkę, na której użytkownik nie będzie musiał czegokolwiek poprawiać. Aby się ona zgadzała z wizją projektanta, musi on niestety zmodyfikować niektóre ścieżki poprowadzone przez program. Należy zaznaczyć, iż autorouter jest narzędziem jedynie wspomagającym człowieka i nie wykona za niego całej pracy.

Przed uruchomieniem autoroutera musimy zwrócić uwagę na parę ważnych elementów, od których będzie zależał wygląd automatycznie wygenerowanej płytki. A co się z tym wiąże – także ilość pracy, jaką będziemy musieli później poświęcić na jej poprawienie. Przede wszystkim należy skontrolować następujące elementy:

*Design Rules* – reguły projektowania (DRC), powinny zostać dostosowane do możliwości technologicznych producenta tworzonej płytki. W przeciwnym wypadku może się okazać, że naszej płytki, mimo ładnego wyglądu, nie uda się wyprodukować. Połączenia, których minimalna szerokość nie została sprecyzowana przy pomocy komendy *Net Classes*, są łączone możliwie największymi ścieżkami o szerokości określonej w DRC. Poszczególne jego elementy zostały opisane w poprzedniej części artykułu.

*Net Classes* – jeżeli w czasie tworzenia schematu, połączenia nie zostały przyporządkowane do odpowiednich klas, to jeżeli jest to konieczne, należy tego dokonać przed uruchomieniem autoroutera. Najlepszym przykładem są linie zasilania, które powinny być szersze niż ścieżki sygnałowe oraz sygnały wysokonapięciowe, których odstępstwa od wszystkich innych elementów powinny być w miarę duże. Autorouter sprawdza te wartości i prowa-


dzi ścieżki dla najmniejszych z możliwych parametrów. Przykładowo, jeżeli sygnał ma minimalną szerokość ścieżki 10 mil, to autorouter użyje takiej właśnie szerokości mimo, że w pobliżu może znajdować się dużo wolnego miejsca i można by ją poszerzyć, przez co ułatwiłoby się produkcję płytki.

*Grid* – czyli raster, w którym będzie pracował autorouter. Nie ma on nic wspólnego z rastrem ustawionym globalnie dla całej płytki. Należy również zwrócić uwagę na raster, w którym rozmieszczone zostały elementy (*Placement Grid*). Pomimo że program pozwala na rozmieszczenie elementów w dowolnym rastrze, nie jest to jednak najlepsze rozwiązanie. Należy przestrzegać dwóch zasad: raster ten nie powinien być mniejszy od używanego przez autorouter oraz powinien on być jego całkowitą wielokrotnością. Najlepsze rezultaty uzyskamy, gdy dobierzemy oba rastry tak, aby końcówki elementów (*Pins*) znajdowały się w rastrze autoroutera. Wybranie zbyt rzadkiego rastra nie gwarantuje poprowadzenia wszystkich ścieżek, natomiast zbyt dokładnego powoduje drastyczne zwiększenie czasu potrzebnego do zaprojektowania płytki, jak również wzrost zapotrzebowania na pamięć (RAM) niezbędną autorouterowi. Jeżeli jej zabraknie, to dane zostają odkładane na dysk twardy, co dodatkowo spowalnia pracę.

*Restricted Areas* – czyli obszary zabronione, w których autorouter nie powinien prowadzić ścieżek i przelotek. Opis umieszczania ich na płytce przedstawiliśmy w poprzedniej części artykułu. Warto zwrócić uwagę, że do kreślenia obszarów zabronionych możemy użyć również płaszczyzny nr 20 (*Layer 20 Dimension*). Autorouter nie umieści tam wtedy ani ścieżek ani przelotek. Nie należy zapomnieć o późniejszym ich usunięciu, ponieważ *Layer 20* służy do zaznaczania krawędzi płytki i jest używana w czasie wykonywania plików do produkcji.

Działanie autoroutera jest sterowane przez wiele parametrów, z których część została opisana wcześniej (*DRC* oraz *Net Classes*), pozostałe wprowadzamy w jego menu głównym. Są to *Cost Factors* oraz *Control parameters*. Mają one wpływ na przebieg ścieżek. Wstę-

nie są one ustawione na wartości domyślne, ustalone przez producenta. Użytkownik może je zmodyfikować. W czasie zapisu projektu, aktualne ustawienia są zapisywane do pliku z płytką \*.brd. Można je również zapisać do osobnego pliku (\*.ctl – autorouter control file), dzięki czemu będzie można wykorzystać daną strategię również w innym projekcie.

Po podaniu komendy **AUTO** lub przyciśnięciu ikony  zostaje wyświetlone okno *Autorouter Setup* (rys. 33), w którym określamy parametry sterujące pracą autoroutera. Działanie jego składa się ogólnie z trzech części, z których każdą można osobno włączyć lub wyłączyć:

**Busses** – jako pierwsze zostają poprowadzone ścieżki magistral. Dla autoroutera magistralami nie są sygnały określone na schemacie jako magistrale, lecz połączenia, które można poprowadzić na płycie w kierunku pionowym lub poziomym. Jeżeli dla żadnej warstwy nie zostanie określony preferowany kierunek, to autorouter tę część pominie.

**Route** – główna część, w której program łączy wszystkie elementy ścieżkami. W tym zostaje umieszczona duża liczba przelotek, co jest spowodowane próbą utworzenia 100% połączeń występujących w projekcie.

**Optimize** – na zakończenie można uruchomić dowolną liczbę procesów optymalizujących, których parametry są tak dobrane, aby zminimalizować liczbę przelotek oraz wygładzić poprowadzone już ścieżki. Optymalizacja usuwa jednorazowo tylko jedną ścieżkę, po czym próbuje ją ułożyć na nowo ze zmienionym kształtem. Liczba procesów optymalizacji musi zostać ustalona na początku, przed startem autoroutera. Późniejsza, dodatkowa optymalizacja nie jest możliwa, ponieważ po wykonaniu pracy autorouter nie może zmienić kształtu istniejących już ścieżek. Optymalizacja nie dotyczy więc ścieżek umieszczonych wcześniej przez projektanta lub przez poprzednią, zakończoną sesję autoroutera.

Przyjrzyjmy się teraz dokładniej oknu sterującemu pracą autoroutera. Składa się ono z kilku zakładek. W pierwszej – *General* – mamy możliwość modyfikacji ra-

stra używanego przez autorouter, określenia kształtu stosowanych przelotek, możemy również wczytać lub zapisać plik z ustawieniami autoroutera. W lewej części okna możemy zmienić preferowany kierunek rozmieszczenia ścieżek osobno dla każdej warstwy. Ponieważ Eagle w wersji *light*, której używamy z powodu łatwej dostępności (i braku konieczności płacenia za licencję), pozwala na projektowanie płytek maksymalnie dwustronnych, w polu tym znajdują się tylko warstwy *Top* oraz *Bottom*. Mamy możliwość wyboru jednego z następujących kierunków:

- poziomy
- | pionowy
- / pod kątem 45 stopni
- \ pod kątem 135 stopni
- \* brak preferowanego kierunku

N/A warstwa wyłączona (program nie poprowadzi na niej żadnych ścieżek)

W przypadku płytek jednostronnych składających się jedynie z elementów przewlekanych, uaktywniamy jedynie dolną warstwę płytki – *Bottom*. Jeżeli używamy elementów SMD to, w zależności od ich umieszczenia na płycie, należy ustalić preferowany kierunek na pionowy lub poziomy.

Następna zakładka: *Busses* pozwala na zdefiniowanie parametrów prowadzenia magistral. W jej lewej części – *Layer Costs* – mamy możliwość zdefiniowania warstwy, na której preferujemy prowadzenie połączeń. Jeżeli dla dolnej warstwy wpisujemy zero, a dla górnej inną większą wartość, program będzie próbował poprowadzić ścieżki tylko na warstwie dolnej. Możliwości tej możemy użyć, gdy przykładowo ze względów ekonomicznych lub technologicznych (będziemy sami trawili płytkę) chcemy wykonać płytkę jednostronną, jednakże stopień jej skomplikowania może na to nie pozwolić. Program będzie próbował wtedy wszystkie połączenia poprowadzić po jednej stronie, na drugiej znajdzie się jedynie mała ich liczba. Później, już na płycie jednowarstwowej, możemy je zastąpić mostkami.

W środkowej części okna mamy dostęp do określenia kolejnych parametrów. Mimo, iż wartości wpisywane mogą zawierać się od 0 do 99, to używanie całego przedziału nie zawsze jest sensowne. Dlatego

przy opisie poszczególnych parametrów zostaje podany zakres wartości zalecane przez producenta:

**Via** (0...99) – ma wpływ na liczbę przelotek. Wpisanie niewielkiej wartości powoduje wstawienie dużej liczby przelotek. Jeżeli wpisana zostanie duża wartość, autorouter będzie próbował unikać przelotek, jednakże ścieżki mogą zostać umieszczone pod innym niż preferowanym kierunkiem. Generalnie należy użyć małych wartości przy początkowym prowadzeniu ścieżek (*Buses*, *Route*), a dużych podczas optymalizacji.

**NonPref** (0...10) – prowadzenie ścieżek w innym niż preferowany kierunku. Jeżeli wpisujemy wartość "99", to, na danej płaszczyźnie ścieżki będą prowadzone tylko i wyłącznie w zadanym kierunku. Możliwości tej można użyć tylko wtedy, gdy jesteśmy w pełni pewni że potrzebujemy takiego właśnie zachowania.

**ChangeDir** (0...25) – określa częstotliwość zmian kierunku prowadzonej ścieżki. Przy małej wartości ścieżka na wiele zagięć, dla dużej jest w miarę prosta.

**OrthStep**, **DiagStep** – wprowadza zasadę, że przeciwprostokątna w trójkącie prostokątnym jest krótsza niż suma obu przyprostokątnych. Wartości domyślne wynoszą odpowiednio 2 oraz 3, co powoduje, że program wybiera kreślenie w kierunku przeciwprostokątnej. Zaleca się zachowanie szczególnej ostrożności podczas zmiany tych parametrów.

**ExtdStep** (0...30) – powoduje unikanie kawałków ścieżek, przebiegających pod kątem 45 stopni względem kierunku preferowanego. Wpisanie małej wartości pozwala autorouterowi na ich kreślenie, duża wartość ich zabrania. Współpracuje z parametrem *mnExtdStep* (opisanym dalej). Praktycznie jego działanie objawia się przebiegiem zagięć ścieżki. Jeżeli parametrem *mnExtdStep* jest ustawiony na 0 oraz *ExtdStep* ma wartość 99, to zagięcia ścieżek będą zawsze przebiegać pod kątem 90 stopni. Jeżeli zostaną nadane im inne wartości, to w zagięciu zostanie wprowadzony krótki kawałek ścieżki, przebiegającej pod kątem 45 stopni.

**BonusStep** **MalusStep** (1...3) – wzmacnia różnicę pomiędzy dobrymi (*Bonus*) oraz złymi (*Malus*)

obszarami płytki. Duże wartości prowadzą do silnego rozróżniania tych obszarów. Małe wartości zamazują różnice pomiędzy nimi. Parametry te powiązane są z parametrami z następnego punktu.

*PadImpact SmdImpact* (0...10) – wspomniane w poprzednim punkcie, „dobre” oraz „złe” obszary płytki znajdują się w okolicach padów oraz pół SMD. Są to miejsca w których program chętnie prowadzi ścieżki, lub raczej od tego się wstrzymuje. „Dobre” obszary przebiegają wzdłuż preferowanych kierunków, „złe” prostopadłe do nich. Powoduje to, iż odprowadzenia od pół lutowniczych (padów) oraz pół SMD przebiegają zgodnie z kierunkiem preferowanym. Zmieniając wartość opisywanych parametrów mamy wpływ na długość pierwszego segmentu ścieżki, od pola lutowniczego (lub SMD) do pierwszego zagięcia.

*BusImpact* (0...10) – powoduje rysowanie możliwie prostych, pozbawionych zagięć ścieżek. Ma wpływ na działanie autoroutera jedynie w czasie kreślenia *Busses*. Wpisanie małej wartości powoduje kreślenie prostych połączeń pomiędzy punktem startowym, a końcowym. Małe wartości pozwalają na stosowanie dużej liczby zagięć.

*Hugging* (0...5) – określa szerokość odstępów pomiędzy ścieżkami równoległymi. Wpisanie dużej wartości powoduje, że przebiegają one w miarę blisko siebie, małą wartość powoduje wprowadzenie pomiędzy nimi większych odstępów.

*Polygon* (0...30) – mała wartość powoduje że autorouter prowadzi ścieżki również w obszarach *poligonów*. Występuje wtedy duże prawdopodobieństwo, iż *polygony* te zostaną pocięte na mniejsze wysępki. Po wpisaniu dużej wartości, praktyka ta jest w miarę możliwości unikana.

*Avoid* (0...10) – ma wpływ na unikanie regionów, w których w wyniku operacji *Ripup*, zostały już usunięte ścieżki. Wysoka wartość powoduje, że autorouter unika ich, niska – mniej. Wartość ta używana jest jedynie na zakładce *Route*.

Po prawej stronie zakładki *Busses* znajdziemy jeszcze kontrolkę *Active*. Dzięki niej określić możemy czy autorouter dany krok wykona, czy też nie. Poza tym znajdują się tam jeszcze dodatkowe parametry konfiguracyjne pracę autoroutera:

*Via* (0...30) – ogranicza maksymalną liczbę przelotek, jaką można umieścić na jednej ścieżce.

*Segments* (0...9999) – określa maksymalną liczbę segmentów z jakiej może się składać jedna ścieżka.

*MnExtdStep* (0...9999) – określa liczbę kroków, które bez wpływu parametru *ExtdStep* mogą przebiegać pod kątem 45 stopni do kierunku preferowanego.

Następne zakładki w oknie *Autorouter Setup* kierują krokiem *Route* oraz dowolną liczbą procesów optymalizacyjnych, z nazwami rozszerzonymi o kolejne ich numery. Do wprowadzenia nowej lub usunięcia istniejącej już optymalizacji służy przyciski *Add* oraz *Del*. Każdą optymalizację możemy naturalnie włączyć lub wyłączyć przy pomocy kontrolki *Active*. Na zakładce *Route* znajdują się jeszcze dodatkowe parametry mające wpływ na mechanizm *Ripup/Retry*. Wpisane w te pola wartości domyślne są wynikiem kompromisu pomiędzy efektywnością autoroutera, a czasem jaki będzie mu potrzebny na wykonanie połączeń.

*RipupLevel* – parametr ten określa maksymalną liczbę ścieżek, które można w jednym kroku *Ripup* rozłączyć.

*RipupSteps* – maksymalna liczba zagnieżdżeń procesu *Ripup*.

*Ripuptotal* – określa jak wiele ścieżek może być w danej chwili rozłączonych.

Jeżeli któryś z tych parametrów zostanie przez autorouter przekroczony, przerywa on dla danej ścieżki proces *Ripup*, po czym przywraca stan początkowy. Ścieżka zostaje uznana jako niemożliwa do pociągnięcia i program zabiera się za rozrysowywanie następnego sygnału.

Czasami drobna zmiana któregoś z wyżej opisanych parametrów ma olbrzymi wpływ na wygląd gotowej płytki. Dlatego producent zaleca stosowanie wartości domyślnych, zostały one określone na podstawie wieloletnich doświadczeń pracowników firmy CadSoft i powinny zapewnić najlepsze wyniki. Warto jednak z ustawieniami tymi poeksperymentować. Można dzięki temu dostosować płytkę do naszych wymagań, oraz dokładniej poznać wpływ poszczególnych parametrów na działanie programu.

Gdy określone zostały już wszystkie potrzebne parametry, mo-

żemy przystąpić do uruchomienia autoroutera. Mamy dwie możliwości. Pierwsza pozwala na kreślenie tylko wybranych połączeń, druga – wszystkich. W przypadku pierwszej klikamy na przycisk *Select* po czym zaznaczamy interesujące nas sygnały, lub wpisujemy ich nazwy z klawiatury. Ostatecznie przyciskamy ikonkę **GO** . Jeżeli chcemy aby zostały połączone wszystkie sygnały, wystarczy przycisnąć *OK*. Postęp pracy jest na bieżąco wskazywany w liście statusu u dołu ekranu. W każdej chwili możemy przerwać działanie autoroutera przyciskając ikonkę **STOP** . Jeżeli chcielibyśmy kontynuować przerwana pracę, wystarczy uruchomić autorouter ponownie. W otwartym wtedy okienku (rys. 33) nie można zmienić żadnych parametrów. Kontrolka *Continue existing job* decyduje czy będziemy kontynuować przerwana pracę, czy chcemy rozpocząć nową, z nowymi parametrami.

Ponieważ proces automatycznego kreślenia ścieżek czasami może trwać wiele godzin, co 10 minut są zapisywane postępy pracy. Tak więc w przypadku zawieszenia komputera lub zaniku napięcia, zostanie utracone maksimum 10 minut pracy. Po ponownym załadowaniu projektu wystarczy zaznaczyć kontynuację pracy autoroutera. Jeżeli program nie zdoła poprowadzić wszystkich ścieżek, pomocnym może okazać się skontrolowanie i ewentualne zmniejszenie jeżeli jest to możliwe następujących wielkości:

- raster autoroutera oraz rozmieszczenia elementów,
- minimalne szerokości ścieżek oraz odstępów pomiędzy nimi,
- średnice przelotek.

Po zakończeniu pracy autoroutera należy dokładnie przyjrzeć się płytce i poprawić ewentualne niedociągnięcia oraz poprowadzić ścieżki, których nie udało się położyć automatycznie. Po tych zabiegach płytka jest już praktycznie gotowa, musimy ją jeszcze skontrolować testem DRC. Po poprawieniu wszystkich wskazanych błędów możemy uznać proces projektowy za zakończony i zająć się wykonaniem dokumentacji produkcyjnej, zostanie to opisane w jednej z najbliższych części artykułu.

**inż. Henryk Wieczorek**  
[henrykwieczorek@gmx.net](mailto:henrykwieczorek@gmx.net)