

Józef KULISZ  
Instytut Elektroniki  
Politechnika Śląska

## WSPÓŁCZESNE METODY PROJEKTOWANIA UKŁADÓW CYFROWYCH

**Streszczenie.** W artykule dokonano przeglądu metodologii i narzędzi wspomagających projektowanie charakterystycznych dla współczesnych układów cyfrowych. Skoncentrowano się na dwóch ścieżkach projektowania i wykonywania tych układów: ścieżce układów z elementów gotowych i ścieżce układów scalonych specjalizowanych (ASIC). Główny nacisk położony został na metody komputerowego wspomagania procesu projektowania.

### Modern methods of digital circuits design

**Summary.** In the paper a review of methodology and design tools characteristic for modern digital circuits is presented. The author's interests concentrate on two paths of their design and implementation: so called "catalog logic" path, and application specific integrated circuits (ASIC) path. Special attention was paid to computer aided design methods.

### Moderne Methoden für Digitalschaltkreisentwurf

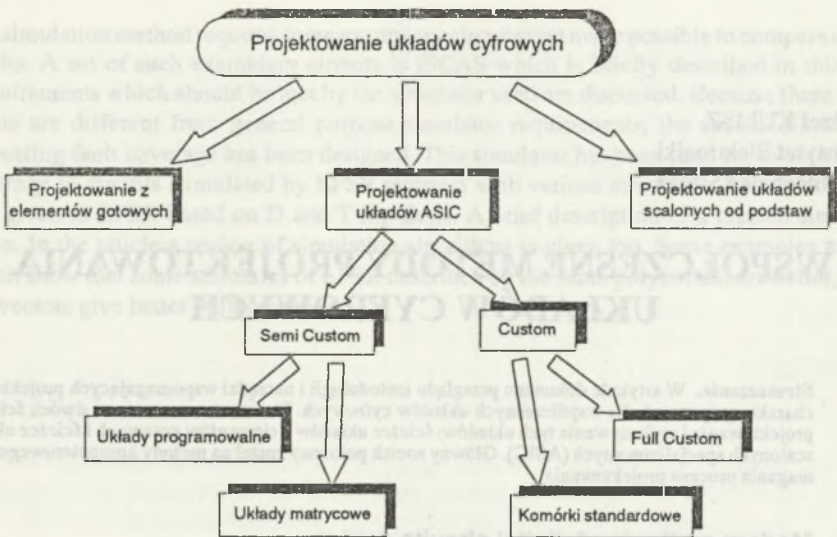
**Zusammenfassung.** Im Vortrag wurde ein Überblick der Methodologie und der CAD Werkzeuge für Entwurfsunterstützung moderner digitaler Schaltungen vorgestellt. Der Verfasser beschäftigt sich mit zwei Pfaden zur Planung und Erzeugung von solcher Schaltungen: dem Pfad von fertiggestellten Schaltkreisen und dem Pfad von kundenspezifischen Schaltkreisen (ASIC). Spezieller Druck wurde auf die Methoden des rechnergestützten Schaltungsentwurfs gelegt.

Obecnie wyróżnić można trzy podstawowe sposoby wykonywania układów cyfrowych (rys. 1). Są to:

- wykonywanie układów cyfrowych opierając się na gotowych elementach katalogowych,
- wykonywanie układów cyfrowych jako układów scalonych projektowanych od podstaw,
- wykonywanie układów specjalizowanych (ASIC).

Każdy z nich ma specyficzny dla siebie obszar zastosowań. Ponieważ sposób wykonania układu cyfrowego bardzo mocno określa metody jego projektowania, będziemy mówić o "ścieżkach" projektowania i wykonywania układów cyfrowych.

W praktyce bardzo rzadko spotyka się ostre i jednoznaczne podziały. Tak jest też i w tym przypadku. Między przedstawionymi powyżej ścieżkami występują różnego rodzaju zależności



Rys. 1. Trzy podstawowe ścieżki projektowania układów cyfrowych  
 Fig. 1. Three basic paths of digital circuits design

i związki, których nie oddaje rys. 1. Tradycyjnie powiązane ze sobą są ścieżki projektowania z elementów gotowych i układów scalonych projektowanych od podstaw. Układy scalone produkuje się przecież po to, by można je było później wykorzystywać jako elementy gotowe. Często też wykorzystuje się układy ASIC w większych systemach złożonych z elementów gotowych.

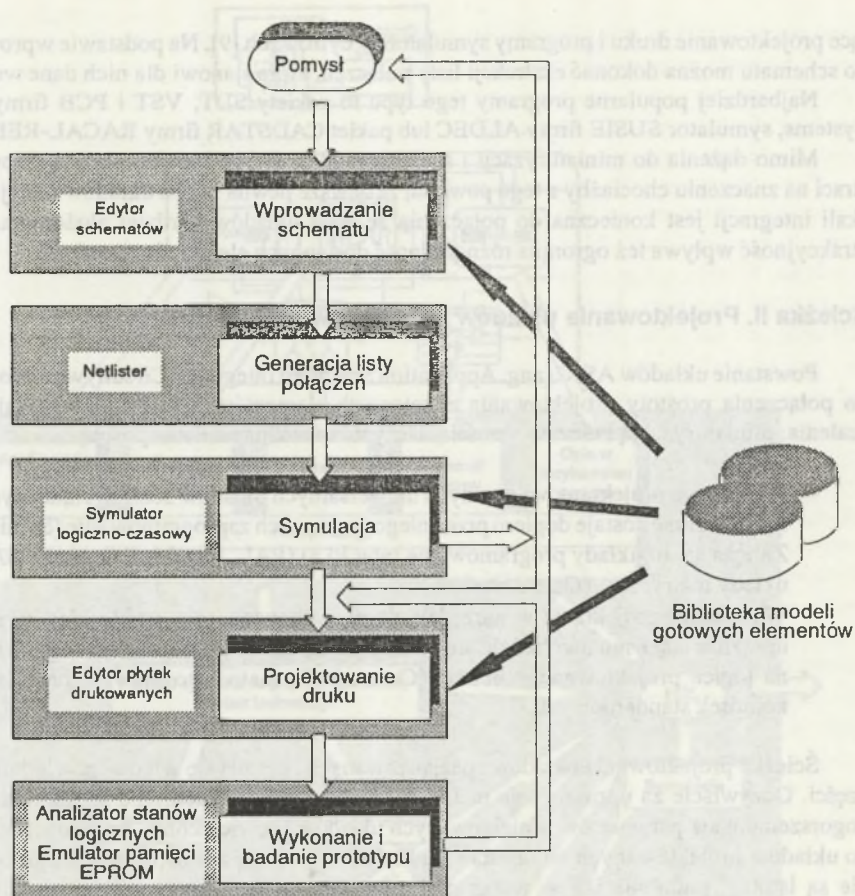
Niniejszy artykuł jest próbą przeglądu i porównania metodologii projektowania charakterystycznych dla dwóch z wymienionych ścieżek projektowania układów cyfrowych: projektowania z elementów gotowych i projektowania układów ASIC, które ze względu na koszty są bardziej dostępne od układów scalonych projektowanych od podstaw na krzemie.

## Ścieżka I. Projektowanie z gotowych elementów

Jest to tradycyjny, najstarszy i najtańszy sposób projektowania i prototypowania układów cyfrowych, dostępny także dla hobbystów-amatorów dysponujących bardzo ubogimi środkami (ołówki, kartka papieru, lutownica).

Prócz bardzo prostych układów projektuje się w ten sposób głównie systemy mikroprocesorowe oparte np. na mikrokontrolerach jednocukładowych. Postęp w technologii półprzewodników sprawił, że elementy katalogowe stają się coraz bardziej skomplikowane. Dostępność złożonych układów VLSI sprawia, że obecnie większość układów elektronicznych wykonuje się w postaci jednego lub kilku układów VLSI obudowanych pewną ilością układów małej i średniej skali integracji pełniących funkcje pomocnicze (np. buforów zwiększające obciążalność). Punkt ciężkości wysiłku projektanta projektującego z gotowych elementów coraz bardziej przenosi się z twórczej pracy koncepcyjnej na poznawanie działania coraz to nowszych i bardziej złożonych układów VLSI. Zastosowanie konkretnego układu VLSI zwykle wymusza sposób konstrukcji





Rys. 2. Projektowanie z gotowych elementów

Fig. 2. The "catalog logic" design flow

całego jego otoczenia. Twórcza praca projektanta elektronika przenosi się także na tworzenie oprogramowania na poziomie bliskim sprzętu, np. w językach assemblerowych mikrokontrolerów jednoukładowych.

Dzięki rozpowszechnieniu się w ostatnich latach stosunkowo tanich komputerów osobistych możliwe stało się pewne zautomatyzowanie procesu projektowania z elementów gotowych. Przebieg projektowania dla tej ścieżki przedstawiono na rys. 2. Rozpoczyna się ono od wprowadzenia schematu projektowanego układu za pomocą edytora schematów, opracowanego wcześniej przez projektanta. Wynika z tego, że właściwej syntezy logicznej dokonuje sam człowiek. Dostępne obecnie narzędzia programistyczne wspomagają projektowanie układów z elementów katalogowych na najniższym poziomie syntezy fizycznej. Edytory schematów ułatwiają tworzenie rysunków do celów dokumentacyjnych, a także są wygodnym sposobem wprowadzenia do komputera informacji wykorzystywanej później przez programy wspomagające.

jące projektowanie druku i programy symulatorów cyfrowych [9]. Na podstawie wprowadzonego schematu można dokonać ekstrakcji listy połączeń, która stanowi dla nich dane wejściowe.

Najbardziej popularne programy tego typu to pakiety SDT, VST i PCB firmy OrCAD Systems, symulator SUSIE firmy ALDEC lub pakiet CADSTAR firmy RACAL-REDAC.

Mimo dążenia do miniaturyzacji i scalenia wydaje się, że ten sposób projektowania nie straci na znaczeniu chociażby z tego powodu, że zawsze pewna liczba układów małej i średniej skali integracji jest konieczna do połączenia ze sobą układów bardziej złożonych. Na jego atrakcyjność wpływa też ogromna różnorodność dostępnych elementów i taniść.

## Ścieżka II. Projektowanie układów specjalizowanych (ASIC)

Powstanie układów ASIC (ang. Application Specified Integrated Circuit) wynikało z dążenia do połączenia prostoty projektowania z gotowych elementów z zaletami wynikającymi ze scalenia: miniaturyzacją i niezawodnością. Uczynić to można na dwa sposoby:

- Dostarczyć projektantowi pewnych uniwersalnych układów scalonych, których działanie określone zostaje dopiero przez niego poprzez ich zaprogramowanie (Semi-custom). Zalicza się tu układy programowalne typu PLD (PAL, PROM, FPLA), FPGA, a także układy matrycowe (Gate Array);
- Wyposażyć projektanta w narzędzia do projektowania topografii układów scalonych umożliwiające mu uwolnienie się od konkretnej technologii, tak aby mógł on skupić się na logice projektowanego układu (Custom). Ta metoda to głównie projektowanie z komórek standardowych.

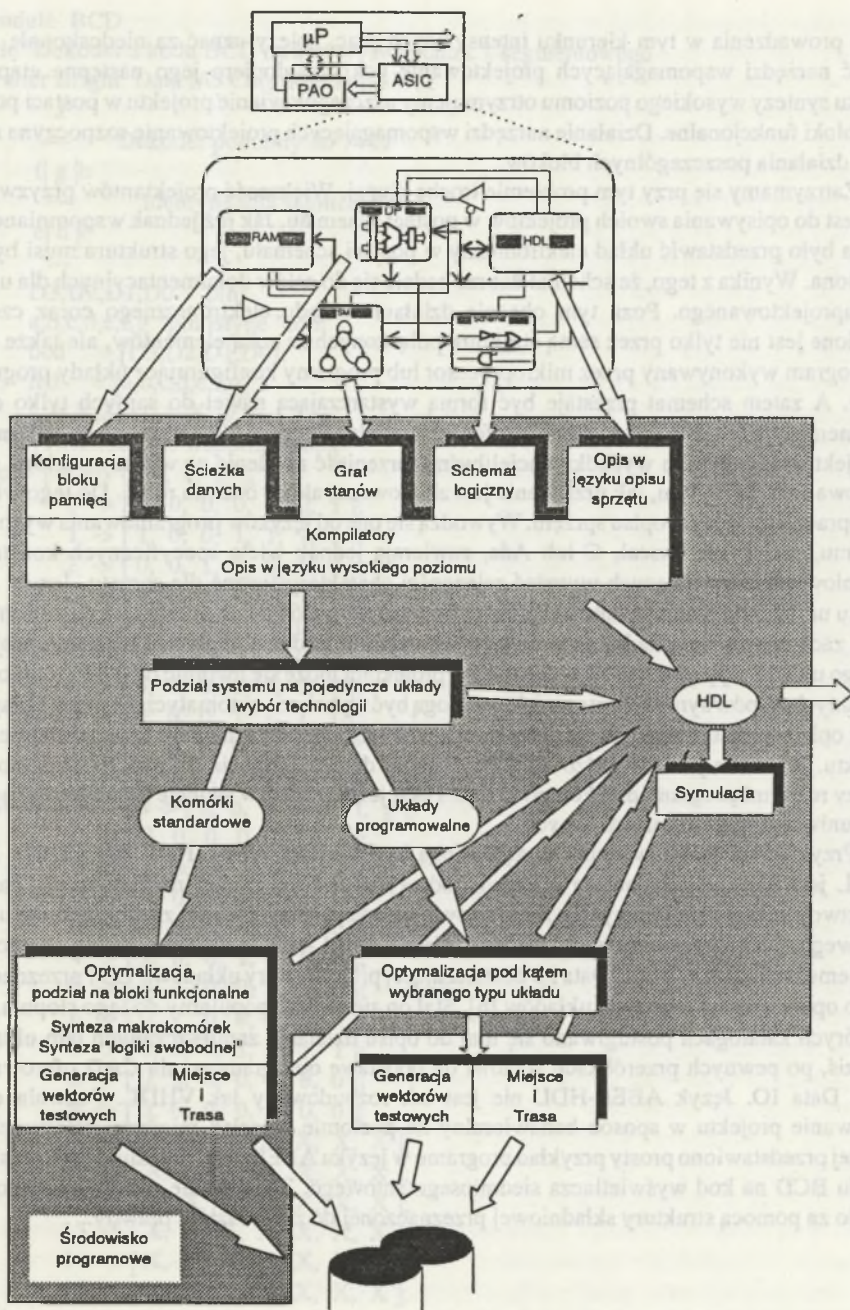
Ścieżka projektowania układów specjalizowanych rozpadła się więc odpowiednio na dwie części. Oczywiście za uproszczenie metod projektowania i oddalenie od technologii płacimy pogorszeniem się parametrów projektowanych układów (powierzchnia, szybkość) w stosunku do układów projektowanych od podstaw na krzemie. W wielu zastosowaniach ograniczenia te nie są istotne, podobnie jak w większości wypadków wystarcza programowanie w języku wysokiego poziomu, a nie w języku maszynowym.

Takie podejście wymaga bardzo wyrafinowanych narzędzi programistycznych oraz komputerów o większych możliwościach (przeważnie muszą to być stacje projektowe). Powstanie i rozwój układów ASIC przypada na ostatnie dziesięć lat. Obecnie dla tej ścieżki projektowania dostępne są najnowocześniejsze i najbardziej rozbudowane narzędzia programistyczne i jednocześnie zaobserwować można najszybszy ich rozwój.

Rys. 3 przedstawia przebieg projektowania układów ASIC. Dąży się do tego, aby cały proces projektowania zintegrować w jednym środowisku, tak aby projektant przechodząc między poszczególnymi etapami nie musiał wychodzić na poziom systemu operacyjnego, ale wybierał poszczególne opcje z systemu menu posługując się np. myszą.

Projektowanie systemu cyfrowego rozpoczyna się od projektu jego architektury. Obecnie nie istnieją metody syntezy architektury układów cyfrowych. Pierwszy etap projektowania musi więc być dokonany przez projektanta przy użyciu jego wiedzy i intuicji. Następnie przechodzi się do bardziej szczegółowego projektowania poszczególnych bloków - syntezy wysokiego poziomu. Tu możliwa jest już pewna automatyzacja, choć metody syntezy wysokiego poziomu,





Rys. 3. Projektowanie układów ASIC  
Fig. 3. ASIC design flow

mimo prowadzenia w tym kierunku intensywnych prac, należy uznać za niedoskonałe. Większość narzędzi wspomagających projektowanie pokrywa dopiero jego następne etapy. W wyniku syntezy wysokiego poziomu otrzymujemy uszczegółowienie projektu w postaci podziału na bloki funkcjonalne. Działanie narzędzi wspomagających projektowanie rozpoczyna się od opisu działania poszczególnych bloków.

Zatrzymamy się przy tym problemie trochę dłużej. Większość projektantów przyzwyczajona jest do opisywania swoich projektów w postaci schematu. Jak już jednak wspomniano, aby można było przedstawić układ elektroniczny w postaci schematu, jego struktura musi być już określona. Wynika z tego, że schemat dobrze nadaje się do celów dokumentacyjnych dla układu już zaprojektowanego. Poza tym obecnie działanie układu elektronicznego coraz częściej określone jest nie tylko przez samą strukturę połączonych ze sobą elementów, ale także przez np. program wykonywany przez mikroprocesor lub programy konfigurujące układy programowalne. A zatem schemat przestaje być formą wystarczającą nawet do samych tylko celów dokumentacyjnych. Nie jest to natomiast forma wygodna do opisu układu, który dopiero ma być zaprojektowany. W tym wypadku chcielibyśmy przenieść myślenie na wyższy poziom, skoncentrować się nie na tym, jak urządzenie jest zbudowane, ale co ono ma robić. Do tego właśnie celu opracowano języki opisu sprzętu. Wywodzą się one od języków programowania w wysokim poziomie, takich jak Pascal, C lub Ada, zawierają jednak wiele specyficznych konstrukcji składniowych pozwalających wyrażać zależności charakterystyczne dla sprzętu. Języki opisu sprzętu umożliwiają opisywanie układów cyfrowych w sposób behawioralny, czyli zorientowany na zachowanie się układu, a nie na jego wewnętrzną budowę. Opisując działanie projektowanego układu w języku wysokiego poziomu projektant może się uwolnić od wielu rutynowych i żmudnych etapów syntezy logicznej, które mogą być wykonane automatycznie przez komputer. Język opisu sprzętu może też być uzupełniającą w stosunku do schematu formą dokumentacji projektu. Wreszcie jest to bardzo wygodny format do przenoszenia informacji projektowych między różnymi programami i komputerami. Tekst jest przecież w świecie komputerów najbardziej uniwersalnym formatem danych.

Przykładami języków opisu sprzętu mogą być Verilog, ABEL-HDL lub VHDL. Język VHDL jest obecnie standardem międzynarodowym. Jest on bardzo rozbudowany. Zawiera bogactwo struktur składniowych pozwalających na opisywanie zarówno zachowania się układu cyfrowego, jak i jego wewnętrznej struktury. Język ABEL-HDL został rozwinięty na początku lat osiemdziesiątych w firmie Data IO dostarczającej programatory układów PLD i przeznaczony był do opisu właśnie tego typu układów [6]. Stał on się szybko popularny do tego stopnia, że w niektórych katalogach posługiwano się nim do opisu działania zamieszczanych tam układów. Do dziś, po pewnych przeróbkach stanowi on podstawę oprogramowania CAD oferowanego przez Data IO. Język ABEL-HDL nie jest tak rozbudowany jak VHDL. Pozwala on na opisywanie projektu w sposób behawioralny na poziomie przestań międzyrejestrowych [6]. Poniżej przedstawiono prosty przykład programu w języku ABEL-HDL, opisującego transkoder z kodu BCD na kod wyświetlacza siedmiosegmentowego. Zachowanie układu zdefiniowane zostało za pomocą struktury składniowej przeznaczonej do zapisu tablic prawdy.



```

module BCD
title 'Dekoder z kodu BCD na kod wyświetlacza 7-segmentowego
Walter Bright Data I/O Corp Redmond WA'
"
  a
" --- Dekoder podobny do 7449
" f|g|b
" --- identyfikacja segmentów
" e|d|c
" ---
D3,D2,D1,D0 pin;
a,b,c,d,e,f,g pin istype 'com';
bcd = [D3,D2,D1,D0];
led = [a,b,c,d,e,f,g];
L,H,X,Z = 0,1,,X,,Z.;
"Wyświetlacze ze wspólną anodą
(bcd - [ a, b, c, d, e, f, g ])
  0 -> [ 0, 0, 0, 0, 0, 0, 1 ];
  1 -> [ 1, 0, 0, 1, 1, 1, 1 ];
  2 -> [ 0, 0, 1, 0, 0, 1, 0 ];
  3 -> [ 0, 0, 0, 0, 1, 1, 0 ];
  4 -> [ 1, 0, 0, 1, 1, 0, 0 ];
  5 -> [ 0, 1, 0, 0, 1, 0, 0 ];
  6 -> [ 0, 1, 0, 0, 0, 0, 0 ];
  7 -> [ 0, 0, 0, 1, 1, 1, 1 ];
  8 -> [ 0, 0, 0, 0, 0, 0, 0 ];
  9 -> [ 0, 0, 0, 0, 1, 0, 0 ];
test_vectors (bcd -> [ a, b, c, d, e, f, g ])
  0 -> [ 0, 0, 0, 0, 0, 0, 1 ];
  1 -> [ 1, 0, 0, 1, 1, 1, 1 ];
  2 -> [ 0, 0, 1, 0, 0, 1, 0 ];
  3 -> [ 0, 0, 0, 0, 1, 1, 0 ];
  4 -> [ 1, 0, 0, 1, 1, 0, 0 ];
  5 -> [ 0, 1, 0, 0, 1, 0, 0 ];
  6 -> [ 0, 1, 0, 0, 0, 0, 0 ];
  7 -> [ 0, 0, 0, 1, 1, 1, 1 ];
  8 -> [ 0, 0, 0, 0, 0, 0, 0 ];
  9 -> [ 0, 0, 0, 0, 1, 0, 0 ];
 10 -> [ X, X, X, X, X, X, X ];
 11 -> [ X, X, X, X, X, X, X ];
 12 -> [ X, X, X, X, X, X, X ];
 13 -> [ X, X, X, X, X, X, X ];
 14 -> [ X, X, X, X, X, X, X ];
 15 -> [ X, X, X, X, X, X, X ];
end;

```

Człowiek potrafi jednak znacznie łatwiej ogarniać informację podaną w sposób graficzny niż tekstowy. Dlatego mimo wszystkich przedstawionych wyżej zalet języków opisu sprzętu firmy dostarczające oprogramowanie wspomagające projektowanie układów elektronicznych starają się wzbogacić możliwości specyfikacji projektu o formy graficzne, takie jak graf stanów automatu synchronicznego lub diagram ścieżki danych. Formą graficznego opisu projektu jest też schemat. Schemat jako forma specyfikacji projektu może być użyteczny w przypadku przenoszenia do układu ASIC starego projektu wykonanego z elementów gotowych, lub jako ilustracja połączenia ze sobą kilku bloków funkcjonalnych zdefiniowanych w inny sposób [5]. Niebagatelne znaczenie dla producentów narzędzi CAD mają też zapewne przyzwyczajenia klientów. Zaznaczyć trzeba, że schemat jako forma opisu układu ASIC jest czymś znacząco różnym od schematu opisującego układ zbudowany z elementów katalogowych. W tym ostatnim przypadku z każdym elementem bibliotecznym związany był fizyczny układ (np. 7400) z charakterystycznymi dla siebie parametrami czasowymi i wymiarami przestrzennymi [9]. W przypadku układów ASIC symbol biblieczny oznacza wyłącznie zależność logiczną pomiędzy sygnałami wejściowymi i wyjściowymi (np. iloczyn logiczny) i może nie mieć nic wspólnego ze strukturą układu otrzymaną w wyniku syntezy. Układ zbudowany z bramek może w wyniku redukcji logicznej lub dekompozycji zmienić liczbę warstw, może też zostać wykonany na bazie struktury w ogóle nie zawierającej bramek logicznych (np. PROM) [8].

Projektant powinien mieć możliwie szeroki wybór postaci specyfikacji projektu, aby wybrać tę, która w danym wypadku jest najwygodniejsza. Istotne jest, aby można było łączyć w jedną całość fragmenty projektu opisane na różne sposoby [1]. Najbogatsze pod tym względem możliwości ze wszystkich znanych autorowi programów daje system Compass [4]. Najnowszy produkt firmy Data IO - pakiet Synario pozwala wprowadzać projekty w postaci schematów i programów w języku ABEL-HDL. Pakiet zawiera też program State CAD, umożliwiającą definiowanie automatów synchronicznych przez rysowanie grafów stanów. Na podstawie takiego rysunku generowany jest automatycznie program w języku ABEL-HDL [5].

Po dokonaniu specyfikacji projektu następuje tłumaczenie opisów poszczególnych bloków na jeden format. Generowany jest behawioralny opis działania całego bloku ASIC w języku wysokiego poziomu. Dysponując pełnym opisem działania bloku, odpowiednie programy mogą dokonać estymacji parametrów projektowanego układu przy założeniu realizacji w konkretnej technologii. Dobrym przykładem takiego programu może być Design Navigator wchodzący w skład systemu Compass [4]. Nowsze wersje oprogramowania dostarczanego przez firmę Data IO pod nazwą ABEL zawierają program SmartPart, wyszukujący z bazy danych układy programowalne, na podstawie których można zrealizować wprowadzony projekt i spełniające przy tym pewne warunki (np. czas propagacji, maksymalna liczba wprowadzeń) zadane przez projektanta [7]. Dopiero teraz może zapaść decyzja o wyborze konkretnej technologii. Dąży się do tego, aby tak długo jak to możliwe nie uszczegóławiać pewnych parametrów, aby projektant jak najdłużej miał możliwość wyboru najlepszego z wielu wariantów. Kiedy taka decyzja zapadnie, dalsze etapy mogą przebiegać automatycznie. Dobre narzędzia CAD powinny umożliwiać projektowanie w pełni automatyczne oraz powinny pozwalać projektantowi na ingerencję w dowolnym momencie procesu projektowania, aby dzięki swojemu doświadczeniu mógł on wymuszać optymalniejsze rozwiązania.

Aby projektowanie przebiegało efektywnie, niezbędne jest wprowadzenie sprzężenia zwrotnego w postaci symulacji [1]. Powinna ona być sukcesywnie powtarzana po zakończeniu każdego etapu - w miarę uszczegóławiania parametrów. Jest to zapewnione poprzez generowanie



po każdym kroku opisu w języku opisu sprzętu, który to opis może być użyty do symulacji. Np. system Synario generuje na podstawie zredukowanych równań logicznych modele projektowanego układu w języku Verilog. Modele te są danymi wejściowymi do symulatora Silos [5]. Podobnie może być w systemie Cadence (zależy to od wykorzystywanego symulatora). Opis w języku opisu sprzętu jako format bardzo uniwersalny może też być użyty do przeniesienia danych o projekcie do innego systemu. Proces projektowania powinien zakończyć się dokładną symulacją logiczno-czasową z uwzględnieniem parametrów fizycznych zaprojektowanej struktury [1]. Dobre symulatory powinny też umożliwiać generowanie raportów z ostrzeżeniami o naruszeniach parametrów czasowych (np. czasów hold i setup przerzutników). Współczesne narzędzia wspomagające projektowanie zapewniają tak silne sprzężenie zwrotne poprzez symulację, że zaprojektowanie wadliwego układu przy pozytywnym wyniku wszystkich kolejnych symulacji i spełnieniu reguł projektowych uważa się za bardzo mało prawdopodobne (mówi się o narzędziach zapewniających sukces za pierwszym razem - ang. First Time System Success).

Jednym z ostatnich etapów syntezy jest synteza fizyczna (Place & Route). W przypadku układów typu custom odpowiednie narzędzia zwykle stanowią część całego systemu. Synteza przeprowadzana jest na podstawie specyficznych dla danego producenta układów scalonych biblioteki komórek standardowych i reguł projektowych. W ostatnich latach udało się wypracować dość duże ujednoczenie technologii, reguły projektowania dla wielu producentów są podobne i nie jest zwykle problemem przeniesienie projektu do innej technologii w ramach tego samego oprogramowania. Odmienną sytuację mamy w przypadku układów programowalnych. Oprogramowanie wspomagające projektowanie złożonych układów logiki programowalnej (CPLD, FPGA) można podzielić, posługując się zwięzłą angielską terminologią, na dwie grupy: oprogramowanie typu Front End i oprogramowanie typu Back End. Zadaniem oprogramowania typu Front End jest umożliwienie projektantowi w wygodny sposób specyfikacji projektu. Wykorzystujemy je więc na początku procesu projektowania, co uzasadnia użyty na jego określenie termin. Oprogramowanie to wspomaga wszystkie etapy projektowania z wyjątkiem etapów ściśle związanych z syntezą fizyczną. Zadaniem oprogramowania typu Back End jest przeprowadzenie fizycznej syntezy projektu w konkretnym układzie FPGA, zwanej w angielskiej terminologii najczęściej "Place & Route". Oprogramowanie typu Back End jest zwykle dostarczane przez producentów układów FPGA, którzy najlepiej znają i potrafią wykorzystać ich możliwości. Współczesne układy FPGA projektowane są przy współpracy programistów opracowujących algorytmy syntezy fizycznej i w ten sposób oprogramowanie to powstaje niejako równoległe z nowymi układami. Niestety producenci układów FPGA dostarczają narzędzia do syntezy wyłącznie opierając się na produkowanych przez siebie układach. I tak np. firma Xilinx opracowała system XACT umożliwiający fizyczną syntezę projektów wykorzystujących układy z serii LCA2000, 3000 i 4000 tej firmy, firma Altera do swoich układów z serii Classic, Max5000, Max7000 i Flex8000 sprzedaje system Max II plus, firma Actel opracowała program pod nazwą Designer itd. Natomiast oprogramowanie typu Front End powstaje najczęściej niezależnie od producentów układów programowalnych. Producenci tego typu oprogramowania dążą do umożliwienia przejścia z projektem do maksymalnie wielu rodzin układów FPGA (co wynika także z metodologii projektowania) [1]. Przykładami oprogramowania typu Front End mogą być produkty firmy Data IO sprzedawane pod nazwami Abel i Synario. Żeby więc wykonać kompletny projekt, trzeba zakupić przynajmniej dwa pakiety oprogramowania. Warto tu wspomnieć o pakiecie FPGA Foundry firmy NeoCAD, który powstał niezależnie od producentów układów FPGA i jak na razie umożliwia fizyczną syntezę projektów na podstawie

układów firm Xilinx i AT&T, Motorola i QuickLogic. Być może pojawienie tego pakietu zapowiada przełom na rynku oprogramowania do syntezy fizycznej opierając się na strukturach FPGA i w przyszłości pojawią się systemy pozwalające na przeprowadzenie projektu do końca.

Wykonanie układu zaprojektowanego w technologii komórek standardowych zleca się firmom zajmującym się produkcją układów scalonych. Realizacja zamówienia trwa zwykle kilka tygodni. Wykonywanie tego typu układów jest stosunkowo drogie i zaczyna się opłacać od kilku tysięcy sztuk. Realizacja projektu na podstawie struktury programowalnej sprowadza się do zaprogramowania układu, co można zrobić w ciągu kilku minut posługując się tanim programatorem. Wynika stąd atrakcyjność oprogramowania pozwalającego realizować projekty zarówno w technologii komórek standardowych, jak i na podstawie układów programowalnych. Realizacja projektu w układzie FPGA umożliwi szybkie i tanie prototypowanie [2]. Sprawdzony projekt można następnie wykonać jako układ scalony z komórek standardowych, co opłaca się przy dłuższych seriach.

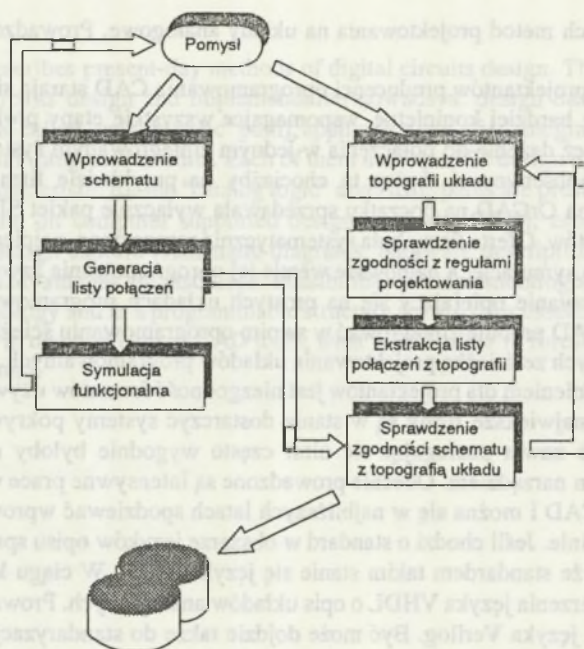
Na skutek ogromnej złożoności współczesnych systemów cyfrowych wielkie znaczenie ma testowanie. Rzeczą bezwzględnie konieczną jest projektowanie układów w taki sposób, aby ułatwić ich testowanie. Współczesne systemy wspomagania projektowania układów ASIC pozwalają zarówno na projektowanie układów łatwo testowalnych, jak też umożliwiają automatyczną generację testów. Biblioteki komórek standardowych w takich systemach, jak Compass lub Cadence, zawierają komórki umożliwiające realizację wbudowanych testów, czy np. ścieżki brzegowej. Także złożone układy programowalne, np. z serii LCA4000 mają wbudowane struktury umożliwiające realizację ścieżki brzegowej.

## ASIC Full-Custom

Narzędzia umożliwiające projektowanie układów z komórek standardowych zwykle umożliwiają także większe zbliżenie się do projektowania układu na krzemie w postaci Full-Custom. Zwykle nie projektuje się tą metodą całych układów, lecz wykorzystuje się ją do optymalnego projektowania tych części, dla których pewne parametry, np. czasowe są krytyczne albo też do projektowania sekcji analogowych. Można też pokusić się o projektowanie w ten sposób nowych komórek standardowych. Projektowanie układów w technice Full-Custom możliwe jest zwykle za pomocą tych samych narzędzi, które umożliwiają projektowanie z komórek standardowych. Daje to możliwość łączenia ze sobą w jednym projekcie fragmentów zaprojektowanych obiema metodami.

Rys. 4 przedstawia przebieg projektowania dla układu Full-Custom. Polega ono na niezależnym wprowadzaniu schematu elektrycznego układu oraz topografii na krzemie, a następnie sprawdzeniu ich zgodności ze sobą. Topografię projektuje się za pomocą odpowiedniego edytora graficznego. W procesie projektowania można wykorzystywać proste elementy biblioteczne: tranzystory MOS, rezystory, diody itp. zadając tylko ich pewne parametry (np. szerokość i długość kanału). Można posłużyć się metodą topografii symbolicznej, wreszcie można zejść na najniższy poziom nanosząc kształt obszarów n, p, metalizacji, polikrzemu itd. Podczas projektowania automatycznie sprawdzana jest zgodność projektu z pewnym zespołem reguł charakterystycznym dla wykorzystywanej technologii. Do symulacji zaprojektowanej sekcji można posłużyć się symulatorem obwodów analogowych, np. Spice, działającym w środowisku projekto-





Rys. 4. Projektowanie układów typu ASIC Full - Custom  
 Fig. 4. ASIC Full - Custom design flow

wym. Pozwala to np. na dokładne obliczenie czasów propagacji i czasów przełączania nowo zaprojektowanej komórki.

## Tendencje

Nieodłączną cechą rozwoju elektroniki jest stałe dążenie do miniaturyzacji. Można się więc spodziewać w najbliższej przyszłości dalszego rozwoju i poszerzenia zastosowań układów ASIC wykonywanych zarówno jako układy programowalne, jak i w technologii komórek standardowych. Ze względu na malejące z roku na rok ceny potrzebnego oprogramowania i sprzętu ścieżka ta staje się coraz bardziej dostępna nawet dla średnich i małych firm.

Wydaje się, że projektowanie z elementów gotowych coraz częściej sprowadzać się będzie do łączenia ze sobą i obudowywania elementami pomocniczymi kilku układów VLSI, w tym układów ASIC. Punkt ciężkości myślenia koncepcyjnego projektantów w dalszym ciągu przenosić się będzie na projektowanie układów scalonych.

Równocześnie z rozpowszechnieniem się układów specjalizowanych można się spodziewać coraz większego zautomatyzowania prac projektowych. Być może w niedługim czasie praca projektanta układów cyfrowych przypominać będzie pracę programisty komputerowego, opisującego swój projekt w języku wysokiego poziomu. Należy się też spodziewać rozszerzenia

zautomatyzowanych metod projektowania na układy analogowe. Prowadzone są już prace w tym kierunku.

Dla wygody projektantów producenci oprogramowania CAD starają się dostarczać oprogramowanie coraz bardziej kompletne, wspomagające wszystkie etapy projektowania. Charakterystyczne jest też dążenie do połączenia w jednym zintegrowanym systemie wielu ścieżek projektowania. Zaobserwować można to chociażby na przykładzie firm OrCAD, Aldec i Racal-Redac. Firma OrCAD na początku sprzedawała wyłącznie pakiet SDT, wspomagający rysowanie schematów. Oferta firmy była systematycznie rozszerzana, najpierw o projektowanie druku, następnie o symulację, a najnowsze wersje jej oprogramowania zawierają pakiet wspomagający projektowanie opierający się na prostych układach programowalnych [8]. W ten sposób firma OrCAD próbuje zintegrować w swoim oprogramowaniu ścieżkę projektowania z elementów gotowych ze ścieżką projektowania układów programowalnych.

Dużym utrudnieniem dla projektantów jest niezgodność formatów używanych przez różne programy. Tylko największe firmy są w stanie dostarczyć systemy pokrywające cały proces projektowania, ale nawet posługując się nimi często wygodnie byłoby na pewnym etapie posłużyć się innym narzędziem. Obecnie prowadzone są intensywne prace w kierunku standaryzacji narzędzi CAD i można się w najbliższych latach spodziewać wprowadzenia pewnych norm w tej dziedzinie. Jeśli chodzi o standard w obszarze języków opisu sprzętu, to wydaje się być przesądzone, że standardem takim stanie się język VHDL. W ciągu kilku lat można się spodziewać rozszerzenia języka VHDL o opis układów analogowych. Prowadzone są też prace nad standaryzacją języka Verilog. Być może dojdzie także do standaryzacji samych narzędzi CAD, podobnie jak np. ustandaryzowano oprogramowanie w dziedzinie sieci komputerowych. Umożliwiłoby to modułarną konstrukcję i wymiennność oprogramowania, dzięki czemu można by je w elastyczny sposób dobierać odpowiednio do potrzeb [3].

## LITERATURA

- [1] Schulze B.: Avoid pitfalls in selecting the right programmable-logic design tools, *Electronic Design*, November 7, 1991.
- [2] Knack K.: Prototyping with FPGAs, ASIC & EDA, December 1993.
- [3] Holley M., Kalpinsky C.: Streamline programmable-logic design with the proposed LPM standard, *Electronic Design* November 7, 1991.
- [4] Compass Design Automation Fundamentals I. Tools Fundamentals Utilities.
- [5] Synario User Manual.
- [6] Synario ABEL-HDL Reference.
- [7] ABEL Design Software User Manual.
- [8] OrCAD PLD User Guide.
- [9] OrCAD Schematic Design Tools User Guide.

Wpłynęło do Redakcji w czerwcu 1994 r.



## Abstract

The article describes present-day methods of digital circuits design. There are three basic paths of digital circuits design and implementation nowadays: design based on ready-made catalog elements (so called "catalog logic" path), application specified integrated circuits (ASIC) path and full-custom integrated circuits. Each of them uses specific design methodology. In the paper methods characteristic for the "catalog logic" and ASIC paths are presented. The author's interests concentrate on computer supported design methods (CAD), especially for ASICs. Various ways of design capture (schematic diagrams, hardware description languages), their advantages and disadvantages are discussed. Possibilities of implementing a design both in the standard cell technology and in a programmable structure are also discussed. The author tries to observe the trends in development of CAD tools, what would allow to foresee their most likely development in future.

## English translator of PLDs

**Abstract.** This paper presents the method of translating PLD logic equations into the logic equations of the resulting built in the tools of XILINX FPGA is described as well as the method of utilization of FPGA resources for simulation purposes.

## Zarys metody wydevelopmentu programowalnych struktur logicznych

**Przebieg** i sposoby wykorzystania zasobów strukturalnych wydevelopmentu logicznych struktur programowalnych (PLD) w celu wydevelopmentu układow programowalnych (FPGA) jest opisany, jak rowniez jest przedstawiona metoda wykorzystania zasobow układow programowalnych (FPGA) do celow symulacyjnych.

## 1. WPROWADZENIE

W projektowaniu urządzeń inżynierii cyfrowej istnieje już dwie metody realizacji algorytmów pracy – sprzętowa i programowa. Sprzętowa realizacja opiera się głównie na układach cyfrowych o małym stopniu scalenia MSI/MSL, które są powszechnie produkowane i prezentowane w katalogach. Metody te przez wypracowanie jakichkolwiek modyfikacji wiąże się z koniecznością bezpośredniej ingerencji w strukturę urządzenia i dlatego wylicza się ją względnie rzadko, gdzie natomiast algorytmy pracy jest strukturalnie nielimitowane. W przeciwnym przypadku stosuje się realizację programową. Nowoczesne układy scalone – na podstawie programowej metody realizacji cechule modularną budowę i detektywności konstrukcji, a modyfikacja algorytmu pracy polega jedynie na zmianie oprogramowania sterującego.

Nowe podejście do konstruacji urządzeń cyfrowych opiera się na układach ASIC (Application Specific Integrated Circuit). Jest to specjalizowany układ scalony urządzeń realizujący jedną, zwaną także stałymi systemu cyfrowego, a budowane z nich urządzenia, które się może w produkcji, pobierają mniej energii, szybciej pracują i mają mniejsze wymiary.