

Maciej PILCH

Lech ZNAMIROWSKI

REALIZACJA PROGRAMU ASIC/DI W INSTYTUCIE TECHNOLOGII ELEKTRONOWEJ I POLITECHNIKACH

Streszczenie. Realizację programu ASIC/DI (Application Specific Integrated Circuits/Design and Implementation) stanowiącego analog MOSIS (MOS Implementation Service) w USA oraz EUROCHIP (kraje EWG) w Europie rozpoczęto w czerwcu 1992 roku. Celem programu jest stworzenie warunków do rozwinięcia kooperacji na linii zespół projektantów układów ASIC (Politechniki, Instytuty Badawcze) oraz Instytut Technologii Elektronowej jako Silicon Foundry. W pracy przedstawiono pewne wnioski wynikające z realizacji pierwszej rundy Programu.

PROGRAM ASIC/DI IMPLEMENTATION IN INSTITUTE OF ELECTRON TECHNOLOGY AND UNIVERSITIES

Summary. ASIC/DI (Application Specific Integrated Circuits/Design and Implementation) program, the Polish analog for MOSIS (MOS Implementation Service) in U.S.A. and EUROCHIP (CEC countries) in Europe had started on June 1992. The goal of a Program is to make a possibility to implement in a standard form students and research ASIC projects from Universities and other research institutions in Institute of Electron Technology as a Silicon Foundry. The paper presents some conclusions resulting from first round of a Program.

L'ACTION ASIC/DI DANS L'INSTITUTE DE TECHNOLOGIE ELECTRONIQUE ET DES POLITECHNIQUES

Résumé. ASIC/DI (Application Specific Integrated Circuits/Design and Implementation) le programme polonaise analogique pour MOSIS (MOS Implementation Service) aux États-Unis et EUROCHIP (CEC pays) en Europe, a commencé au juin 1992, pour stimuler le processus de la coopération les projecteurs (l'Uni-

versites et l'Institut des Recherches) des circuits ASICs avec l'ITE comme le "Silicon Foundry". Dans ce travail, il a été présenté quelques conclusions obtenu dans premier tour du Programme.

1. Wprowadzenie

Celem programu ASIC/DI (Application Specific Integrated Circuits/Design and Implementation) [PILCH1, PILCH2] jest stworzenie warunków do rozwinięcia kooperacji na linii zespół projektantów układów ASIC (Politechniki, Instytuty Badawcze lub inne instytucje zainteresowane) oraz Instytut Technologii Elektronowej jako realizatora projektów (Silicon Foundry).

W 1991 r. została stworzona dokumentacja umożliwiającą realizację projektów powstałych poza Instytutem Technologii Elektronowej, w tym:

- 1) Parametry dla symulatora SPICE tranzystorów w cyfrowych układach wykonywanych w technologii CMOS;
- 2) Reguły projektowania układów CMOS;
- 3) Biblioteka komórek peryferyjnych układów CMOS.

Zdecydowano, że od 1 czerwca 1992 r. zostanie rozpoczęta próbna realizacja projektów dostarczonych z Instytutu Mikroelektroniki i Optoelektroniki Politechniki Warszawskiej w postaci projektu wielostrukturalnego. Zdecydowano się również na dołączenie do tej próby projektów z Instytutu Informatyki Politechniki Śląskiej, mimo że stwarzało to pewne komplikacje wykonawcze, ponieważ projekty były przygotowane dla technologii NMOS, która od ok. 4 lat nie jest wykorzystywana przez ITE.

W pracy przedstawiono uwagi związane z realizacją pierwszej rundy Programu z punktu widzenia realizatora projektów oraz jednego z uczestników jako zgłaszającego projekt układów ASIC dla ich realizacji.

2. Pierwsza runda realizacji studenckich i badawczych projektów ASIC

2.1. Projekty przyjęte do realizacji

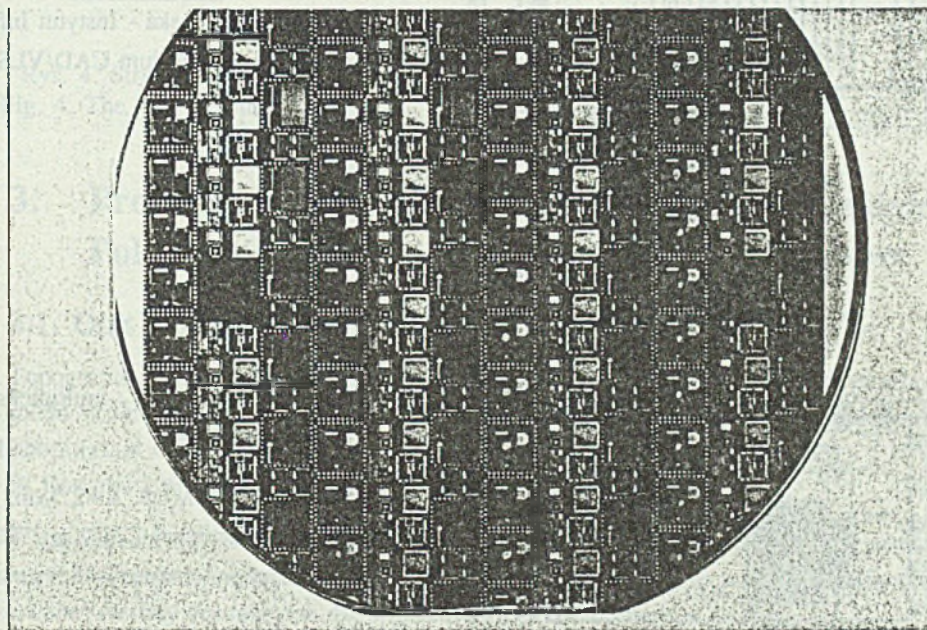
W ramach zadania weryfikacji i wdrożenia projektów studenckich sprawdzono i wykonano komplet masek dla następujących 3 grup układów:

- 3 układy NMOS opracowane w Instytucie Informatyki Politechniki Śląskiej o nazwach PROJECT5, AF4 i ST2,
- 3 układy CMOS opracowane w Instytucie Mikroelektroniki i Optoelektroniki Politechniki Warszawskiej o nazwach LEDXCHIP, ADDYF i KRAS,
- 1 układ CMOS opracowany w Instytucie Elektroniki Politechniki Śląskiej oparty na matrycy bramkowej GA 500,
- 1 układ służący do weryfikacji procesu CMOS (struktura próbna).

Ze względów ekonomicznych, z uwagi na bardzo wysokie koszty masek, postanowiono z powyższych ośmiu układów stworzyć tylko jeden komplet masek, umieszczając je obok siebie na zasadzie struktury multi-chip i wykorzystując je do dwóch odrębnych partii: z jednej, po procesie CMOS, otrzymując układy CMOS, z drugiej, po procesie NMOS otrzymując układy NMOS. Mimo zwiększonego zużycia płytek, koszty pozyskania pojedynczych układów do weryfikacji są, dzięki oszczędności na maskach, najniższe.

2.2. Krótki opis wybranych projektów

W toku realizacji pierwszej rundy Programu zrealizowano strukturę multi-chip na jednej płytce krzemowej. Mikrofotografię przedstawiającą ogólny obraz płytki krzemowej multiprojektu pokazano na rys. 1.

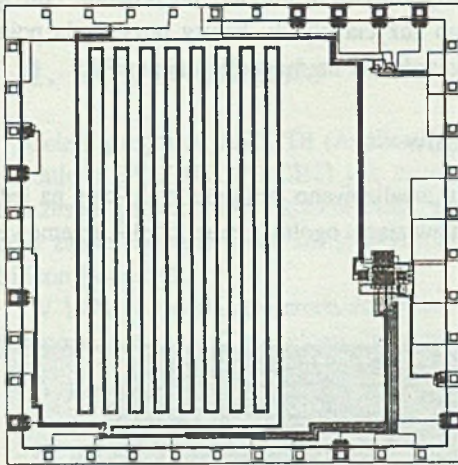


Rys. 1. Ogólny obraz płytki krzemowej pierwszego multiprojektu
Fig. 1. Photograph of first multiproject wafer

Spośród projektów realizowanych w pierwszej rundzie Programu poniżej przedstawiono krótkie opisy trzech projektów studenckich i badawczych: AF4, PROJECT5 oraz ST2, nadesłanych do Instytutu Technologii Elektronowej z Instytutu Informatyki Politechniki Śląskiej celem ich realizacji.

Projekt I

Rysunek zbiorczy maski



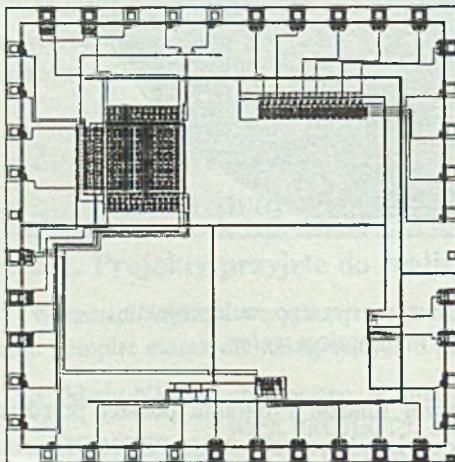
Rys. 2. Struktura AF4
Fig. 2. The AF4 structure

Krótki opis

- Multiline transmisyjne, sumator czterobitowy, struktury testowe.
- Technologia NMOS. $\lambda=2.5\mu\text{m}$.
Wymiary struktury $2750\lambda \times 2750\lambda$.
Powierzchnia struktury 47.3 mm^2 .
- Autorzy projektu:
Grupa studencka - IV rok Informatyki/L. Znamirowski.
Politechnika Śląska - Instytut Informatyki, Laboratorium CAD/VLSI.

Projekt II

Rysunek zbiorczy maski



Rys. 3. Struktura PROJECT5
Fig. 3. The PROJECT5 structure

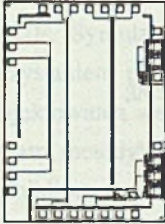
Krótki opis

- Pamięć, dekodery, multiplexer, struktury testowe.
- Technologia NMOS. $\lambda=2.5\mu\text{m}$.
Wymiary struktury $2750\lambda \times 2750\lambda$.
Powierzchnia struktury 47.3 mm^2 .
- Autor projektu:
S. Warecki - IV rok Informatyki.
Politechnika Śląska - Instytut Informatyki, Laboratorium CAD/VLSI.

Projekt III

Rysunek zbiorczy maski

Krótki opis



Rys. 4. Struktura ST2
Fig. 4. The ST2 structure

- a. Struktury testowe.
- b. Technologia NMOS. $\lambda=2.5\mu\text{m}$.
Wymiary struktury $1050\lambda \times 1450\lambda$.
Powierzchnia struktury 9.52 mm^2 .
- c. Autor projektu:
L. Znamirowski.
Politechnika Śląska - Instytut Informatyki, Laboratorium CAD/VLSI.

3. Projekty ASIC nadesłane z Instytutu Informatyki Politechniki Śląskiej

3.1. Opis topografii układu scalonego i biblioteki

Topografia projektów opisana została w formacie CIF [MEAD1] z wykorzystaniem biblioteki struktur elementarnych [NEWK1, MALY1] zaimplementowanej i rozbudowanej w Laboratorium CAD/VLSI Instytutu Informatyki Politechniki Śląskiej [ZNAM3].

W Dodatku 2 przedstawiono opis formatu CIF.

W projektach przyjęto, że wykonane będą w technologii NMOS z kontaktami pomiędzy rejonami aktywnymi, polikrzemem i metalizacją, realizowanymi poprzez metalizacje. Stosowane warstwy abstrakcyjne dla projektowania przedstawiono w Dodatku 1.

3.2. Format nadesłanych via FD/INTERNET plików CIF opisujących topografię projektów dla realizacji w ITE

Projekty do realizacji w ITE zostały nadesłane w formacie CIF na nośniku magnetycznym (FD). Istnieje też możliwość nadesłania ich za pośrednictwem sieci INTERNET (mpilch@VAX5.ITE.WAW.PL).

Przyjęto, że projektanci dostosowują format plików CIF opisujących projekty do formatu ogólnego przyjętego przez ITE.

Struktura pliku CIF opisująca kilka projektów ma postać następującą:

(Projekt I - AF4 - Wielolinie transmisyjne i struktury testowe);
 (Politechnika Slaska - Instytut Informatyki, Laboratorium CAD/VLSI);
 (Grupa studencka - IV rok Informatyki/L. Znamirowski);
 (Lambda=2.5 microns);
 (Powierzchnia 47.3 sq mm);
 DS 2 125/1;
 9 PADCAP;
 L NM; B 200 16 100,8; B 108 108 100,106;
 DF;
 B 500 14000 575000,253250; B 500 14000 566500, 253250;
 B 500 14000 558000,253250; L NC; L NG;

 (Nazwa projektu II);
 DS
 9 KOM1PII;
 .
 . opis komórki (lub wielu)
 .
 DF;

 (Nazwa projektu III);
 DS
 9 KOM1PIII;
 .
 . opis komórki (lub wielu)
 .
 DF;

 .
 .
 E

Oczywiście, w przypadku multiprojektu nadsyłanego przez jednego autora, wszystkie podprojekty (np. indywidualne chipy) mogą wykorzystywać tę samą bibliotekę struktur elementarnych (np. kontakty, sumatory, multipleksery), a numerację wprowadzanych w kolejnych podprojektach symboli można ograniczyć przez wielokrotne wykorzystanie komendy `rozkaz_dfdelete` z określeniem jej argumentu liczbą większą niż największy numer struktury elementarnej występującej we wspólnej bibliotece.

3.3. Przygotowanie projektów

Przy projektowaniu układów NMOS stosowano geometryczne reguły projektowania przyjęte za [MEAD1] (Dodatek 1). Reguły te niewiele odbiegają od reguł projektowania

obowiązujących w ITE, ustalono więc z wykonawcą, że w przypadku gdy wykryte zostaną naruszenia reguł nie wymagające przeprojektowania układu, poprawki zostaną nanesione przez wykonawcę układów (pkt. 4.2).

Edytorem masek był AutoCAD [AUTO2] współpracujący z translatorami formatów CIF/DXF oraz DXF/CIF [PAWL1]. Translatory wykorzystywane były wprost do wprowadzania danych z plików CIF do edytora masek, a z drugiej strony uzyskiwania z edytora plików CIF. Symulację układową realizowano symulatorem ISSPICE [NAGE1, INTU1] z wykorzystaniem parametrów modeli tranzystorów uzyskanych z ITE. Geometryczne reguły projektowania weryfikowano programem postprocesora plików CIF [ZNAM3].

Przyjęto metodykę projektowania układów ASIC typu "Bottom-up", bazującą na rozwijanej bibliotece kolejno opracowywanych makrokomórek struktur [GAJS1, GOTO1, ZNAM1, ZNAM2].

4. Realizacja programu w Instytucie Technologii Elektronowej

4.1. Wprowadzenie układów do systemu projektowania ITE

Opisy źródłowe układów w formacie CIF dostarczono na dyskietkach, załączając informacje o skali i zawartości poszczególnych warstw.

Układy zostały wczytane do użytkowanego w ITE systemu projektowania MIDAS. Stwierdzono dużą dowolność, jeśli chodzi o stosowane w każdej z trzech grup układów skale, nazewnictwo i numerację warstw, co miało, jak się potem okazało, poważne implikacje w fazie łączenia układów w multi-chip.

W ramach tego etapu w pełni zweryfikowano możliwość swobodnego i poprawnego wprowadzania do użytkowanego w ITE systemu projektowania układów zaprojektowanych poza Instytutem, korzystając z opisu w języku CIF.

4.2. Weryfikacja reguł projektowania

Dla układów NMOS z Gliwic (oraz CMOS) przeprowadzono pełną weryfikację reguł projektowania. Przygotowano odrębne zbiory sterujące do programu MASKAP-PDRC, uwzględniające wszelkie reguły projektowania oraz typowe błędy konstrukcyjne. Zbiory te przygotowano bardzo pieczołowicie, uwzględniając wszelkie możliwe i znane z wcześniejszych doświadczeń sytuacje, tak aby stworzyć uniwersalne narzędzie do obróbki dalszych układów przekazywanych do Instytutu do weryfikacji i (lub) do wykonania.

W wyniku sprawdzenia reguł projektowania wykryto pewne błędy, które jednak nie dyskwalifikowały układów i uznano, że najszybszą metodą będzie poprawienie ich we własnym zakresie, w ITE, bez odwoływania się do autorów projektów układów. Weryfikacja była jednak bardzo czasochłonna, ponieważ program wykazywał po kilkadziesiąt, a nawet po kilka tysięcy formalnych naruszeń reguł projektowania, z których każdy musiał być sprawdzony, po czym okazywało się, że tylko pojedyncze z nich wymagają ingerencji w projekcie. Oczywiście, układ idealnie zaprojektowany nie powinien wykazać żadnego naruszenia.

4.3. Normalizacja układów ze względu na numerację i nazewnictwo warstw, skale i technologie

Stworzenie układu typu multi-chip, zawierającego wszystkie układy (8 układów) na zasadzie stworzenia jednego kompletu negatywów pośrednich, okazało się niemożliwe ze względu na:

- a) zbyt duże wymiary poszczególnych układów, czyli zbyt dużą powierzchnię na negatywie pośrednim, której już nie objąłby układ optyczny do redukcji obrazu z negatywu pośredniego na maskę,
- b) różne skale (jednostka projektu/ilosc mm na masce) przyjęte przez poszczególnych projektantów; doprowadzenie do wspólnej skali powodowałoby przekroczenie maksymalnego zakresu współrzędnych w stosowanym w ITE systemie do projektowania.

Ostatecznie postanowiono rozdzielić układy na 3 grupy, tak aby wykonać 3 komplety negatywów, z których złożony zostałby 1 komplet masek. W ramach każdej grupy dokonano standaryzacji numeracji i nazewnictwa warstw.

Pierwszy komplet negatywów (układ MONSTER) zawierał układy CMOS. Pozostałe 2 układy zbiorcze zawierają układy NMOS z Instytutu Informatyki Pol. Śl.:

- układ 2 (MA) zawiera układy AF4 i 2 x ST2,
- układ 3 (PRO) zawiera układ PROJECT5.

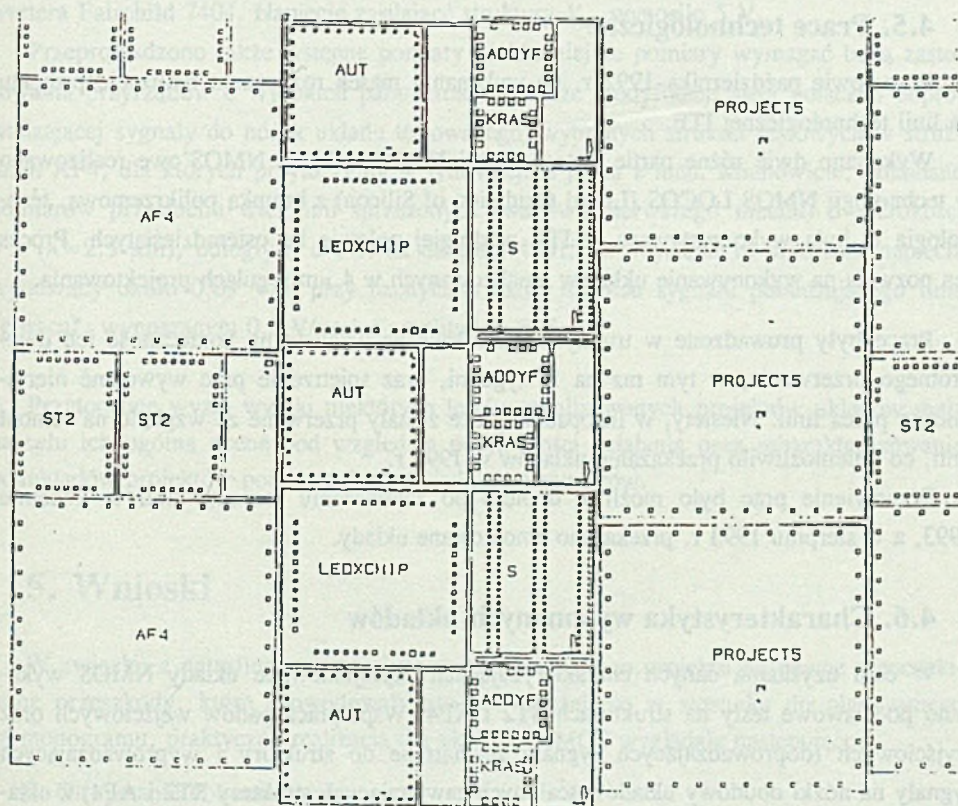
W układach tych przeniesiono odpowiednie warstwy, dodano znak centrujący i przeskalowano w trakcie wypuszczania taśmy na PG (Pattern Generator) tak, aby dopasować skalę do układu MONSTER.

4.4. Topografia struktury zbiorczej "EDUK"

Jak wspomniano wcześniej, przyjęto koncepcję stworzenia struktury zbiorczej typu multi-chip. Przyczyny podziału na trzy grupy (moduły) struktur podano w pkt 4.3. Kolejnym zadaniem było optymalne rozmieszczenie układów w ramach każdej grupy oraz rozmieszczenie tych grup względem siebie.

Projektując topografię struktury zbiorczej "EDUK" starano się osiągnąć kompromis między wymogami maksymalnego wykorzystania krzemu a pewnym uporządkowaniem rozmieszczenia struktur ze względu na późniejsze pomiary i cięcie struktur. Stąd regularne rozmieszczenie poszczególnych modułów struktur na zasadzie pasów pionowych, co zapewnia łatwe do określenia skoki przesunięć wkładki ostrzy pomiarowych i piły do cięcia.

Rozmieszczenie poszczególnych układów na masce/plytce przedstawiono na rys. 5.



Rys. 5. Rozmieszczenie układów MA, PRO oraz MONSTER
Fig. 5. The placement of MA, PRO and MONSTER circuits

Łączna powierzchnia zrealizowanych modułów wyniosła ponad 194 milimetry kwadratowe.

ZESTAWIENIE MODUŁÓW STRUKTUR:

		X	Y
MA		7060	x 10870
w tym:	AF4	7060	x 7060
	ST2	2810	x 3810
PRO		7060	x 7060
w tym:	PROJECT5	7060	x 7060
MONSTER		8200	x 8300
w tym:	LEDXCHIP	4880	x 4690
	ADDYF	2150	x 2000
	KRAS	1750	x 1750
	AUT	3610	x 3610
	S	3320	x 4550

4.5. Prace technologiczne

W połowie października 1992 r. po wykonaniu masek rozpoczęto realizację projektu na linii technologicznej ITE.

Wykonano dwie różne partie w technologii NMOS. Układy NMOS'owe realizowano w technologii NMOS LOCOS (Local Oxidation of Silicon) z bramką polikrzemową, technologia ta była wykorzystywana w ITE w drugiej połowie lat osiemdziesiątych. Proces ten pozwala na wykonywanie układów zrealizowanych w 4 μm regułach projektowania.

Prace były prowadzone w trudnych warunkach ze względu na konieczność ich dwukrotnego przerwania, w tym raz na 18 tygodni, oraz spiętrzenie prac wywołane nierytmiczną pracą linii. Niestety, w listopadzie prace zostały przerwane ze względu na remont linii, co uniemożliwiło przekazanie układów w 1992 r.

Wznowienie prac było możliwe dopiero po zakończeniu remontu linii w kwietniu 1993, a w sierpniu 1993 r. przekazano zmontowane układy.

4.6. Charakterystyka wykonanych układów

W celu uzyskania danych charakteryzujących zaprojektowane układy NMOS wykonano podstawowe testy na strukturach ST2 i AF4. Współpraca padów wejściowych oraz wyjściowych (doprowadzających sygnały zewnętrzne do struktury i wyprowadzających sygnały na nóżki obudowy układów scalonych zawierających struktury ST2 i AF4) z układami TTL jest realizowana bez zastrzeżeń w zakresie napięcia zasilającego struktury V_{DD} od 2.6 V do 5 V. Pady kontaktowe wejść i wyjść struktury testowano przy współpracy z bramkami inwertera Fairchild 7404.

Pad kontaktowy zegara generujący na krzemie dwie rozłączne fale impulsów taktujących $\Phi 1$ oraz $\Phi 2$ pracuje poprawnie dla częstotliwości do 5 MHz zewnętrznego sygnału taktującego. W strukturze AF4 zaprojektowano czterobitowy sumator/subtraktor, który także dla tej częstotliwości (wykorzystuje falę impulsów $\Phi 1$) poprawnie realizuje 19 założonych operacji arytmetyczno-logicznych na dwu czterobitowych słowach danych.

W strukturze ST2 zaprojektowano trzy inwertery o skalowanej geometrii w dwu kierunkach płaszczyzny projektowania. Jeśli określić stosunek liniowych wymiarów geometrycznych jak 1:2:4, to czasy propagacji inwerterów określają odpowiednio stosunki (dla czasu t_{PHL}) 1:1.17:1.44 oraz (dla czasu t_{PLH}) 1:1.28:1.78. Czasy propagacji na krzemie najmniejszego inwertera (o geometrii zdefiniowanej w formacie CIF w [ZNAM2]) wynoszą odpowiednio: t_{PHL} około 4.1 nsek oraz t_{PLH} około 6.0 nsek. Testowane struktury inwerterów od strony padów pobudzane były (a także odpowiednio obciążone) bramkami inwertera Fairchild 7404. Napięcie zasilające struktury V_{DD} wynosiło 5 V.

Przeprowadzono także wstępne pomiary (dokładniejsze pomiary wymagać będą zastosowania przyrządów o wysokich parametrach, a także modyfikacji sieci połączeń doprowadzającej sygnały do nóżek układu testowanego) wybranych struktur testowych w strukturze AF4, dla których przytoczymy w tym miejscu jeden z nich. Mianowicie, dokonano pomiarów przesłuchu wieloliniowej sprężonych (warstwa pierwszego metalu) o szerokości 3λ ($\lambda = 2.5\ \mu\text{m}$), odległych o 3λ na długości 40015λ . Wystąpił tu przesłuch napięcia wynoszący około 0.09 V/V przy nachyleniu czoła impulsu sygnału pobudzającego linię "gorącą", wynoszącym 0.5 V/nsek i amplitudzie 5 V.

Przytoczone wyżej wyniki niektórych testów zrealizowanych projektów układów mają na celu ich ogólną ocenę pod względem poprawności działania oraz scharakteryzowanie podukładów projektów pomiarami odpowiednich parametrów.

5. Wnioski

W związku z natrafieniem w trakcie wykonywania tego projektu na pewne nieoczekiwane przeszkody, które spowodowały znaczne opóźnienie w stosunku do planowanego harmonogramu, praktyczna realizacja dla układów NMOS wyglądała następująco:

0. Otrzymanie dokumentacji	0
1. Wykonanie taśmy magnetycznej	6 tydzień
2. Wykonanie masek	8 tydzień
3. Zakończenie prac technologicznych	20 tydzień

- | | |
|--|------------|
| 4. Wykonanie pomiarów kwalifikacyjnych | 21 tydzień |
| 5. Przekazanie próbek zmontowanych | 22 tydzień |

Trzeba jednak pamiętać, że policzono czas realizacji nie uwzględniający remontów oraz przestoju wynikłych z realizacji innych pilnych prac w ITE (kalendarzowy czas był ponad dwukrotnie dłuższy), a także należy pamiętać, że wykonano poprawki w layoutach układów. Mamy nadzieję, że realizacja następnej grupy takich projektów, którą rozpoczynamy w połowie sierpnia 1993 r. w technologii CMOS, nie przekroczy 22-25 tygodni kalendarzowych.

W celu usprawnienia wykonywania projektów wielostrukturalnych należy:

- w przypadku nietypowych elementów, jak np. rezystory, umieścić w strukturze właściwej element testowy, który można zmierzyć ostrzowo,
- przed przesłaniem projektu należy uzgodnić z ITE numerację warstw, nazwy warstw i komórek oraz stosowaną skalę projektu.

Wykonana praca dowodzi, że w Instytucie Technologii Elektronowej można z powodzeniem wytwarzać układy zaprojektowane w innych ośrodkach. Mogą to być układy edukacyjne, gdzie liczy się szybkie sprawdzenie i przetworzenie projektu studenckiego, szybkie otrzymanie masek i niewielkiej ilości układów do praktycznej weryfikacji działania układu.

Minimalizacja kosztów wymaga łączenia projektów w struktury multi-chip. Tworzenie takich zbiorczych projektów niesie za sobą komplikacje, np. z powodu różnych skal albo niestandardowych numeracji i nazewnictwa warstw; trudno jest jednak narzucać sztywne standardy różnym ośrodkom, kiedy panują w tym względzie różne upodobania lub standardy. Natomiast ścisłych ustaleń wymagają zawsze przyjęte reguły projektowania ze względu na weryfikację projektu w ITE, tym bardziej gdy układ ma być wykonywany na linii ITE.

6. Dodatek 1 - Geometryczne reguły projektowania

6.1. System graficzny stosowany w prezentacji reguł projektowania

6.1.1. Warstwy abstrakcyjne - proces NMOS

Symbole warstw stosowane w projektach realizowanych w procesie NMOS mają znaczenie następujące:

- ND - dyfuzja (rejon aktywny)
- NI - implantacja (depletion)

- NP - polikrzem
- NC - kontakty ND/NP do metalizacji
- NM - metalizacja
- NG - pasywacja (otwory montażowe) (overglass).

6.1.2. Warstwy abstrakcyjne - proces CMOS

Symbole warstw stosowane w projektach realizowanych w procesie CMOS mają znaczenie następujące:

- PWE - wyspa typu P
- NWE - wyspa typu N
- POA - obszary aktywne typu P
- NOA - - obszary aktywne typu N
- POLY - polikrzem
- CONT - kontakty POA/NOA/POLY do pierwszej metalizacji
- MET1 - pierwsza metalizacja
- VIA - kontakty między pierwszą a drugą metalizacją
- MET2 - druga metalizacja
- PVX - otwory montażowe.

W przypadku gdy projekt nie wymaga drugiego poziomu metalizacji, w opisie topografii nie używa się warstw VIA i MET2.

6.2. Geometryczne reguły projektowania - proces NMOS

Projektowanie struktur, ich edycja i modyfikacja wymagają spełnienia kilku zasad zapewniających poprawną realizację w procesie technologicznym. Ponieważ proces jest niezależny od realizowanych projektów, zbiór zasad zwanych regułami projektowania zapewnia uzyskanie projektów poprawnych w zakresie geometrii. Należy prócz tego dokonać sprawdzenia i ewentualnych korekt dla parametrów elektrycznych [MEAD1], co czasem wymaga korekty geometrycznej.

W bibliotece struktur i projektach układów NMOS stosowano reguły projektowania przyjęte za [MEAD1]. Trzeba podkreślić, że chociaż niewiele, to jednak reguły te różnią się od stosowanych w ITE. W związku z tym, chociaż uniknięto przeprojektowywania układów, to jednak wykonawca zmuszony był wnieść odpowiednie poprawki usuwające naruszenia reguł (pkt. 4.2).

Reguły projektowe dla technologii NMOS stosowane w realizowanych projektach zebrano na rys. 6 przedstawiając je jako parametry wymiarowania, które należy odczy-

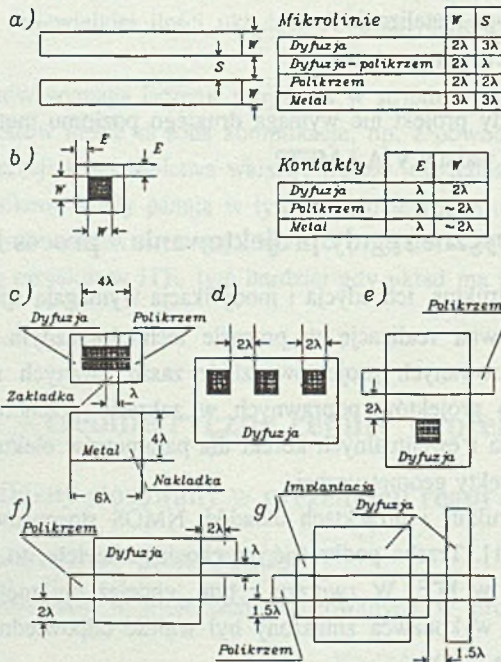
tywać jako wartości minimalne. Przykładowo dla sytuacji z rys. 6a reguła określa sytuację dla dwu mikrolinii, i z danych wynika:

w przypadku dwu linii utworzonych w procesie dyfuzji, minimalna szerokość W każdej ze ścieżek wynosi 2λ , a minimalny odstęp S linii nie powinien być mniejszy niż 3λ . Podobnie dla innych przypadków ujętych w tablicy (tablice "Mikrolinie" i "Kontakty" w górnej części rys. 6).

Reguła z rys. 6b dotyczy minimalnych "zakładek" dla dyfuzji, polikrzemu i metalu w sytuacji projektowania kontaktu polikrzem-polikrzem, dyfuzja-metal, natomiast reguła z rys. 6c wykonania "przejścia" dyfuzja-polikrzem z wykorzystaniem nakładki typu metal.

Reguły z rys. 6b i 6c przedstawiają warunki realizacji kontaktów typu Butting Contacts występujących w przytaczanej bibliotece.

Reguły dla projektowania tranzystorów przedstawiono na rys. 6e, f oraz g, natomiast sytuację i odległości krytyczne-minimalne, w przypadku gdy należy wykonać połączenie pomiędzy większymi obszarami metalu i dyfuzji, przedstawiono na rys. 6d.



Rys. 6. Reguły projektowania (proces NMOS [MEAD1])
Fig. 6. Design rules (The NMOS process [MEAD1])

6.3. Geometryczne reguły projektowania - proces CMOS

Pełny zestaw geometrycznych reguł projektowania obowiązujących dla projektów, które mają być wykonane w procesie HCOO2 (LOCOS CMOS) realizowanym w ITE, przedstawiono w [GROD1].

7. Dodatek 2 - Opis semantyczny i syntaktyczny formatu CIF

7.1. Opis semantyki formatu CIF

Tekstowy format opisu struktur CIF [MEAD1] charakteryzuje się tym, że zapisane struktury zajmują stosunkowo mało pamięci. Mechanizm formatu jest tak pomyślany, że stanowi on rodzaj programu generującego obraz wykorzystując struktury programowe o nazwie transformacji, które działając na wcześniej definiowanych substrukturach w sposób przejrzysty generują złożone obrazy.

Przyjęty zbiór prymitywów graficznych ukierunkowany na potrzeby projektów struktur VLSI daje możliwość efektywnej edycji zarówno "ręcznej", jak i na podstawie przetwarzania plików CIF.

Tablica 1 przedstawia szkieletowe elementy konstrukcji opisu struktury interpretowane jak komendy. Drukiem wytłuszczonym zaznaczono komendy najczęściej stosowane.

W formacie CIF wymiary i odległości przedstawiane są jako liczby całkowite. Wprowadzony sposób przeliczania wartości liczbowych przedstawianych w strukturach opisu symboli pozwala redukować liczby do małych liczb całkowitych, co znakomicie umożliwia oszczędzać pamięć i czyni pliki czytelnymi. Podobnie jak w procesie technologicznym integralną częścią opisu są warstwy, w ramach których opisuje się lokalizację prymitywów.

Dalej przedstawiono ścisły formalnie opis formatu CIF.

Składnia dopuszcza blanki przed i po komendzie, blanki lub inny rodzaj separatora (wyjątki w definicji duża_litera i blank).

Składnia odzwierciedla fakt, że definicje symboli nie mogą być zagnieżdżone.

Plik w formacie CIF składa się z ciągu rozkazów oddzielonych od siebie znakiem końca rozkazu, którym jest znak średnika ";". Plik może być dowolnie dzielony na wiersze - nie jest to konieczne, ale ułatwia analizę pliku przez człowieka.

Skrótowa prezentacja formatu rozkazów przedstawiona w tablicy 1 ma swoje odpowiedniki w szczegółowej notacji syntaktycznej formatu CIF przedstawionej dalej,

Tablica 1

Konstrukcja opisu struktury (CIF)

Komenda	Format
B (Box - prostokąt) z podaną długością, szerokością i kierunkiem (przyjmuje się kierunek (1,0), jeśli go nie podano)	B integer integer point point
P (Polygon - wielokąt) z podaną ścieżką	P path
R (Round flash - okrąg) z podaną średnicą i środkiem	R integer point
W (Wire - linia łącząca) z podaną szerokością i ścieżką	W integer path
L (Layer) ustawienie warstwy	L shortname
DS (Definition Start) rozkaz rozpoczęcia definicji symbolu z numerem, wartościami współczynników a i b (przyjmowanymi jako 1 przy pominięciu)	DS integer integer integer
DF (Definition Finish) rozkaz końca definicji	DF
DD (Delete Definition) kasowanie definicji symbolu	DD integer
C (Call) rozkaz wstawienia symbolu	C integer transformation
Rozkaz rozszerzenia	digit userText
Komentarz	(commentText)
Znacznik końca pliku CIF	E

zgodnie z następującym przyporządkowaniem:

B	▲ rozkaz_box
P	▲ rozkaz_polygon
R	▲ rozkaz_rflash
W	▲ rozkaz_wire
L	▲ rozkaz_layer
DS	▲ rozkaz_dfstart
DF	▲ rozkaz_dffinish
DD	▲ rozkaz_dfdelete
C	▲ rozkaz_call
digit	▲ cyfra
userText	▲ tekst_user_ext
(commentText)	▲ rozkaz_comment
E	▲ rozkaz_cifend
oraz:	
integer	▲ liczba_dodatnia
point	▲ punkt

path	▲ ścieżka
shortname	▲ nazwa_warstwy
transformation	▲ transformacje

Poniżej omówimy znaczenie i akcje wszczynane rozkazami dostępnymi w formacie CIF (opis syntaktyczny formatu CIF przedstawiono w pkt. 7.2).

7.1.1. rozkaz_layer (L)

Rozkaz ustawienia warstwy. W formacie CIF wszystkie elementy graficzne (P, B, W, R) muszą znaleźć się na określonej warstwie. Podczas "interpretacji" pliku CIF rozkazy "działają" na warstwie "aktualnej", którą ustawia właśnie rozkaz L. Parametrem rozkazu L jest nazwa warstwy, która może się składać z dużych liter lub cyfr. Długość nazwy warstwy wynosi od 1 do 4 znaków. Każdy inny znak jest interpretowany jako koniec nazwy warstwy. Jedyłą zastrzeżoną nazwą warstwy, której nie powinno się używać, jest "ZZZZ". Próba umieszczenia elementu graficznego na tej warstwie jest traktowana jako błąd. Warstwa ta jest automatycznie ustawiana na początku pliku, a także niejawnie ustawi ją rozkaz DS (początek definicji symbolu). Rozkaz L powinien poprzedzać każdą listę wyszczególnionych elementów graficznych. Ustawienia warstwy aktualnej nie wymagają inne rozkazy CIF (DS, DF, DD, C, rozkaz_user_ext, rozkaz_comment oraz rozkaz E).

7.1.2. rozkaz_polygon (P)

Rozkaz tworzący wielokąt. Parametrami rozkazu są pary liczb - współrzędne x i y kolejnych wierzchołków wielokąta. W formacie CIF zakłada się, że wielokąt jest zamknięty, a więc automatycznie na rysunku ostatni punkt łączony jest z pierwszym. Lista punktów powinna zawierać przynajmniej dwa punkty (warunek zgodny z syntaktyką CIF).

7.1.3. rozkaz_box (B)

Jest najczęściej używanym elementem graficznym formatu CIF. Rozkaz definiuje prostokąt. Jego parametry kolejno definiują długość prostokąta, szerokość oraz punkt centralny. Punkt centralny zawiera odpowiednio współrzędne x i y . Są to parametry obowiązkowe. Ograniczenia dotyczą wartości długości i szerokości, które muszą być liczbami większymi od 0. Opcjonalnym parametrem rozkazu rozkaz_box jest określenie kierunku. Kierunek definiowany jest jako para współrzędnych wektora, wyznaczającego kierunek, wzdłuż którego liczona jest długość prostokąta. W przypadku pominięcia kierunku domyślnie przyjmowany jest wektor $[1, 0]$ - długość liczona jest wzdłuż osi X, natomiast szerokość wzdłuż osi Y.

Praktyka generacji biblioteki podstawowej VLSI wskazuje na fundamentalne znaczenie prostego prymitywu Box w budowie, sprawdzaniu, stosowaniu w procesie technologicznym, przetwarzaniu, translacji struktur. Omówimy ten prymityw.

Zgodnie z syntaktyką komendy mamy jej postać tekstową:

rozkaz_box = "B" liczba_dodatnia separator liczba_dodatnia separator punkt
[separator punkt],

co można w sensie jej znaczenia (w pełnym rozwinięciu) zapisać w postaci:

B długość(L) szerokość(W) $x_c(C)$ $y_c(C)$ a_x a_y .

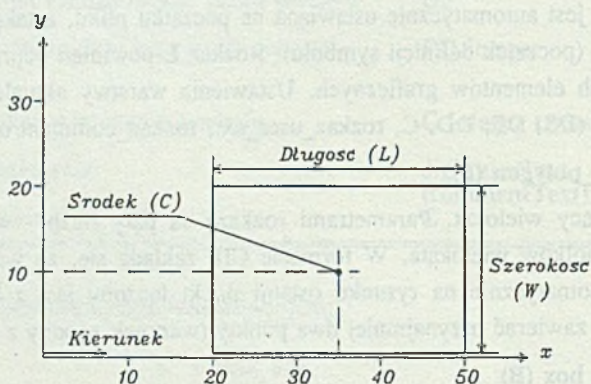
W relacji tej przyjęto oznaczenia:

x_c - współrzędna x środka Box'u,

y_c - współrzędna y środka Box'u,

a_x - współrzędna rzutu wektora kierunkowego Box'u na oś X ,

a_y - współrzędna rzutu wektora kierunkowego Box'u na oś Y .



Rys. 7. Struktura Box
Fig. 7. Box representation

Przykład 1

Strukturę Box, którą zapiszemy w postaci:

B 30 20 35,10;

w postaci graficznej przedstawia rys. 7.

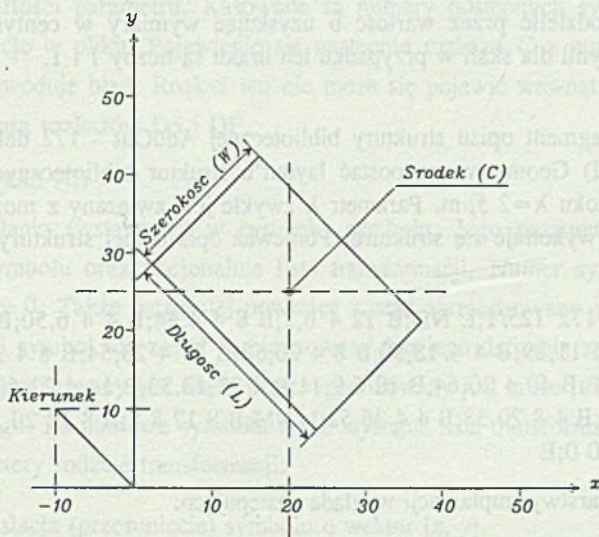
Jeśli uwzględnimy możliwość obrotu Box'u w jego definicji przez wprowadzenie wektora kierunkowego (np. 135°) i przyjmiemy, że teraz środek leży w punkcie (20,25), to postać struktury Box'u obróconej dla poprzednio przyjętego prostokąta można przedstawić następująco:

B 30 20 20,25 -10,10.

Odpowiednią postać graficzną struktury przedstawiono na rys. 8.

7.1.4. rozkaz_rflash (R)

Rozkaz tworzy na rysunku okrąg. Parametrami rozkazu R są kolejno: średnica, która musi być określona liczbą większą od 0 oraz środek okręgu wyznaczony przez współrzędne x i y .



Rys. 8. Struktura Box obrócona
Fig. 8. Rotated Box representation

7.1.5. rozkaz_wire (W)

Kolejny rozkaz "rysunkowy". Tworzy na rysunku ścieżkę o określonej szerokości, przechodzącą przez dane punkty. Parametrami rozkazu są: szerokość ścieżki, która musi być wyrażona liczbą większą od 0 oraz ciąg par współrzędnych x i y wyznaczających punkty, przez które przechodzi ścieżka. Ilość punktów ścieżki powinna być równa lub większa od dwóch.

7.1.6. rozkaz_dfstart (DS)

DS jest rozkazem początku definicji symbolu. Parametrami rozkazu są: numer symbolu oraz opcjonalnie dwie liczby określające wartość skali. Symbole w formacie CIF identyfikowane są przez swój numer, określony liczbą większą lub równą 0. Numer ten (dla danego symbolu) jest także parametrem rozkazu C (wywołanie danego symbolu) oraz

DD (kasowanie definicji symboli). Format CIF wymaga wprowadzenia definicji symbolu przed jego pierwszym wywołaniem - jest to zabezpieczenie przed rekurencyjnym wywołaniem symboli, co w formacie CIF jest niedopuszczalne.

Jednostką długości stosowaną w opisie geometrii jest centymikrometr (1/100 mikrometra).

Dwie liczby, określone tu symbolami a oraz b, umożliwiają zredukowanie rozmiarów plików CIF. W momencie wywołania danej definicji należy wymiary zawarte w każdym elemencie graficznym (współrzędne, długości, szerokości, a także wartości współrzędnych zawarte w liście transformacji rozkazów C) przeskalać, tj. przemnożyć przez wartość a, a następnie podzielić przez wartość b uzyskując wymiary w centymikronach. Wartościami domyślnymi dla skali w przypadku ich braku są liczby 1 i 1.

Przykład 2

Rozważmy fragment opisu struktury bibliotecznej AddOut - 172 definiujący warstwę implantacji (L NI) Geometryczna postać layout'u struktur bibliotecznych przedstawiona jest w siatce o skoku $\lambda = 2.5\mu\text{m}$. Parametr λ zwykle jest związany z możliwościami technologii, w jakiej wykonuje się strukturę. Ponieważ opis pełnej struktury AddOut ma postać:

```
(AddOut);DS 172 125/1;L ND;B 12 4 6,2;B 8 4 4,54;B 4 4 6,50;B 8 6 8,7;B 16 6
12,45;B 14 26 11,29;B 4 4 18,50;B 8 4 20,58;B 24 4 28,54;B 8 4 36,50;L NI;B 12
12 28,54;L NP;B 40 4 20,64;B 10 6 9,11;B 4 38 12,33;B 14 4 23,60;B 4 10 28,53;L
NC;B 4 8 8,8;B 4 8 20,58;B 4 4 36,52;L NM;B 8 12 8,8;B 8 12 20,58;B 8 68 36,34;
;DF;C 172 T 0 0;E
```

wyłączony opis warstwy implantacji wygląda następująco:

```
DS 172 125/1;L ND;B 12 4 .....L NI;B 12 12 28,54; ...;DF;.
```

Dla sytuacji, gdy zastosujemy technologię o wyższej rozdzielczości (mniejsze λ), czyli gdy schodzimy w dół z wymiarami struktury, mamy przykładowo:

dla $\lambda = 2\mu\text{m}$:

```
DS 172 100/1;L ND;B 12 4 .....L NI;B 12 12 28,54; ...;DF;.
```

natomiast dla $\lambda = 1.5\mu\text{m}$:

```
DS 172 75/1;L ND;B 12 4 .....L NI;B 12 12 28,54; ...;DF;.
```

Widać więc, że wprowadzenie skali (liczby a oraz b) pozwala zachować obraz layout'u (dla siatki λ) oraz opis struktury nie zmienione, z wyjątkiem tylko tych właśnie liczb a oraz b w opisie CIF.

Rozkaz DS powoduje zarejestrowanie definicji symbolu o danym numerze. W przypadku gdy wcześniej został zdefiniowany symbol o takim samym numerze, to sygnalizowane jest ostrzeżenie o redefinicji symbolu i nowa definicja zastępuje starą. Początek definicji symbolu nie może się pojawić wewnątrz definicji innego, tzn. między parą rozkazów DS i DF, format CIF nie dopuszcza bowiem zagnieżdżania definicji symboli.

7.1.7. rozkaz_dffinish (DF)

Rozkaz końca definicji symbolu. Rozkaz zamyka definicję, a symbol jest dostępny pod numerem, jaki wystąpił w rozkazie DS. Wszystkie elementy graficzne zawarte między rozkazami DS i DF stanowią definicję symbolu.

7.1.8. rozkaz_dfdelete (DD)

Rozkaz kasowania definicji symboli. Parametrem jest numer symbolu - liczba większa lub równa 0. Rozkaz ten powoduje skasowanie definicji symboli o numerach większych bądź równych wartości parametru. Kasowane są numery dostępnych symboli, a nie fizycznie ich definicje w pliku. Pojawienie się następnie rozkazu C z numerem skasowanego symbolu spowoduje błąd. Rozkaz ten nie może się pojawić wewnątrz definicji symbolu, tj. między parą rozkazów DS i DF.

7.1.9. rozkaz_call (C)

Rozkaz wywołania (wstawienia w rysunek) symbolu. Jego parametrami są: numer wywoływanego symbolu oraz opcjonalnie lista transformacji. Numer symbolu musi być większy lub równy 0. Także wcześniej powinien zostać zarejestrowany symbol o danym numerze. Gdy taki symbol nie został zarejestrowany lub jego definicja została skasowana rozkazem DD, sygnalizowany jest błąd. Błąd sygnalizowany jest także przy próbie wywołania rekurencyjnego. Po numerze symbolu może wystąpić lista transformacji.

Dostępne są cztery rodzaje transformacji:

T x y translacja (przesunięcie) symbolu o wektor $[x, y]$,

R x y rotacja (obróć) wokół punktu $(0, 0)$ definicji symbolu, wektor $[x, y]$ wyznacza kierunek "nowej" osi X,

M X odbicie lustrzane symbolu względem osi Y,

M Y odbicie lustrzane symbolu względem osi X.

Kolejne transformacje w liście nie są rozdzielane żadnym specjalnym separatorem (dla zwiększenia czytelności wygodnie jest rozdzielać transformacje spacjami). Symbol jest przetwarzany kolejno przez transformacje występujące na liście. Należy zaznaczyć, że wszystkie te transformacje są izometryczne, czyli nie zmieniają wymiarów symbolu, lecz jego usytuowanie na płaszczyźnie. Skalowanie symboli można zrealizować tylko w samej definicji symbolu (rozkaz DS). W przypadku gdy w ciele symbolu występuje rozkaz C (zagnieżdżone wywołania), najpierw przetwarzany jest symbol wewnętrzny, a następnie dodatkowo poddawany jest transformacji z zewnętrznego rozkazu C. Dodatkowo w momencie wywołania symbolu zapamiętywana jest aktualna warstwa, tak aby po zakończeniu

"rozwijania" symbolu można było ją odtworzyć. Gdy pominięta zostanie lista transformacji przyjmowana jest transformacja neutralna, czyli $T 0,0$.

7.1.9.1. Wywołania i transformacje

Odwolując się do definicji syntaktycznej komendy wywołania C mamy:

rozkaz_call	= "C" liczba_dodatnia transformacje.
transformacje	= {{blank} ("T" punkt "M" {blank} "X" "M" {blank} "Y" "R" punkt)}.
punkt	= współrzędna separator współrzędna.
współrzędna	= {separator} ["-"] ciąg_cyfr.
ciąg_cyfr	= cyfra {cyfra}.
separator	= duża_litera blank.
cyfra	= "0" "1" "2" "3" "4" "5" "6" "7" "8" "9".
duża_litera	= "A" "B" "C" ... "Y" "Z".
blank	= dowolny znak ASCII wyjąwszy cyfra, duża_litera, "-", "(", ")", lub ":", ";

Interpretacja semantyczna jest następująca: wywołanie C sprowadza się do wstawienia w realizowany rysunek makrostruktury o numerze określonym parametrem typu liczba_dodatnia występującym za symbolem wywołania C, przy czym makrostrukturę poddaje się transformacji "transformacje".

Przykład 3

Komenda:

C 217 T 0,0;

poleca wstawić symbol (makrostrukturę) 217 jej bazą w punkt (0,0, 0,0) rysunku bez żadnych operacji na symbolu.

Jeśli wartości współrzędnych punktów elementów struktury przed transformacją oznaczyć przez x oraz y , a po transformacji x' oraz y' , ogólnie transformację można zapisać relacją:

$$[x', y', 1] = [x, y, 1] T,$$

gdzie T jest macierzą transformacji.

W przypadku kilku transformacji w komendzie wywołania wyliczenie współrzędnych punktów po transformacji sprowadza się do obliczenia macierzy:

$$T = T_1 \cdot T_2 \cdot T_3 \cdot T_4,$$

gdzie przez T_1 , T_2 , T_3 , T_4 oznaczono kolejne elementarne transformacje. Obliczona transformacja T prezentuje więc macierzową postać transformacji złożonej.

7.1.10. rozkaz_user_ext

Jest to specjalny rozkaz formatu CIF przeznaczony na dowolne rozszerzenia wprowadzane przez użytkownika. Format CIF wprowadza tylko dwa ograniczenia dotyczące tekstu zawartego w tym rozkazie: tekst musi rozpoczynać się od cyfry dziesiętnej oraz nie może zawierać znaku średnika ";", który jest znakiem końca rozkazu.

7.1.11. rozkaz_comment

Jest pomocniczym rozkazem w pliku CIF. Może zawierać dowolny tekst ograniczony nawiasami okrągłymi: otwierający komentarz "(" oraz zamykający komentarz ")". Komentarze mogą być zagnieżdżane - wtedy części zagnieżdżone także muszą być ograniczone nawiasami.

7.1.12. rozkaz_cifend (E)

Ostatnim rozkazem w pliku CIF jest rozkaz (znacznik) końca pliku E. Jest rozkazem bez parametrów, a obecność znaków znaczących (duże litery, cyfry, znaki: "-", ";", "(", ")") w pliku za rozkazem E jest sygnalizowana jako ostrzeżenie.

7.2. Opis syntaktyczny formatu CIF

7.2.1. Notacja opisu syntaktyki

W celu opisu syntaktyki formatów CIF oraz DXF przyjęto zmodyfikowaną notację Backusa-Naura:

ZNAK	ZNACZENIE
=	jest zdefiniowane jako
	lub
.	koniec formuły
[x]	0 lub jednokrotne powtórzenie symbolu x
{x}	0 lub wielokrotne powtórzenie symbolu x
(x y.. z)	jeden z symboli x, y, ..., z
"x"	symbol terminalny
male_litery	symbol nieterminalny.

7.2.2. Opis syntaktyczny

plik_cif	= {{blank} [rozkaz] średnik} rozkaz_cifend {blank}.
rozkaz	= rozkaz_primowany rozkaz_dfdelete rozkaz_dfstart średnik {{blank} [rozkaz_primowany] średnik} rozkaz_dffinish.
rozkaz_primowany	= rozkaz_polygon rozkaz_box rozkaz_rflash rozkaz_wire rozkaz_layer rozkaz_call rozkaz_user_ext rozkaz_com- ment.
rozkaz_polygon	= "P" ścieżka.
rozkaz_box	= "B" liczba_dodatnia separator liczba_dodatnia separator punkt [separator punkt].
rozkaz_rflash	= "R" liczba_dodatnia separator punkt.
rozkaz_wire	= "W" liczba_dodatnia separator ścieżka.
rozkaz_layer	= "L" {blank} nazwa_warstwy.
rozkaz_call	= "C" liczba_dodatnia transformacje.
rozkaz_user_ext	= cyfra tekst_user_ext.
rozkaz_comment	= "(" tekst_komentarza ")".
rozkaz_dfdelete	= "D" {blank} "D" liczba_dodatnia.
rozkaz_dfstart	= "D" {blank} "S" liczba_dodatnia [separator liczba_dodatnia separator liczba_dodatnia].
rozkaz_dffinish	= "D" {blank} "F".
rozkaz_cifend	= "E".
transformacje	= {{blank} ("T" punkt "M" {blank} "X" "M" {blank} "Y" "R" punkt)}.
ścieżka	= punkt {separator punkt}.
punkt	= współrzędna separator współrzędna.
współrzędna	= {separator} ["-"] ciąg_cyfr.
liczba_dodatnia	= {separator} ciąg_cyfr.
ciąg_cyfr	= cyfra {cyfra}.
nazwa_warstwy	= znak [znak] [znak] [znak].
znak	= cyfra duża_litera.
tekst_user_ext	= {znak_user_ext}.
tekst_komentarza	= {znak_komentarza} tekst_komentarza "(" tekst_komentarza ")" tekst_komentarza.
średnik	= {blank} ";" {blank}.
separator	= duża_litera blank.
cyfra	= "0" "1" ... "8" "9".
duża_litera	= "A" "B" ... "Y" "Z".

- blank = każdy znak ASCII oprócz cyfr, dużych liter i znaków: "-", "(",
")" oraz ";".
- znak_user_ext = każdy znak ASCII oprócz znaku ";".
- znak_komentarza = każdy znak ASCII oprócz znaków: "(" oraz ")".

LITERATURA

- [GAJS1] Gajski D. D.: *Silicon Compilation*, Addison-Wesley, Reading, Massachusetts 1988.
- [GOTO1] Goto S. (Ed.): *Design Methodologies*, Advances in CAD for VLSI, Vol. 6, North-Holland, Amsterdam 1986.
- [GROD1] Grodner M.: Reguły projektowania i pomocnicze informacje do projektowania układów scalonych CMOS wytwarzanych w ITE, ITE, Warszawa 1993.
- [MALY1] Maly W.: *Atlas of IC Technologies: An Introduction to VLSI Processes*, The Benjamin/Cummings, Menlo Park 1987.
- [MEAD1] Mead C., Conway L.: *Introduction to VLSI Systems*, Addison-Wesley, Reading, Massachusetts 1980.
- [NAGE1] Nagel L. W.: SPICE2: A Computer Program to Simulate Semiconductor Circuits, University of California at Berkeley, ERL-M520, May 1975.
- [NEWK1] Newkirk J. A., Mathews R.: *The VLSI Designer's Library*, Addison-Wesley, Reading, Massachusetts 1983.
- [PAWL1] Pawłowski R., Znamirowski L.: Translacja formatów opisu struktur VLSI, *Prace Naukowe Politechniki Warszawskiej, ELEKTRONIKA*, Nr 101, Warszawa 1994.
- [PILCH1] Pilch M.: Doświadczenia z realizacji studenckich projektów układów scalonych, Referat na Seminarium Informatyki, Politechnika Śląska, Gliwice 1993.
- [PILCH2] Pilch M., Znamirowski L.: Program ASIC/DIP in Institute of Electron Technology and Polish Universities, Poster, *WORKSHOP - Design Methodologies for Microelectronics and Signal Processing*, Silesian Technical University, Gliwice-Cracow, 20-23 October 1993.
- [ZNAM1] Znamirowski L.: Komputerowe wspomaganie kreślenia masek struktur ASIC, *Zeszyty Naukowe Politechniki Śląskiej, INFORMATYKA*, z. 20, Gliwice 1993.
- [ZNAM2] Znamirowski L.: Komputerowe wspomaganie generacji złożonych masek struktur ASIC, *Zeszyty Naukowe Politechniki Śląskiej, INFORMATYKA*, z. 20, Gliwice 1993.
- [ZNAM3] Znamirowski L.: Projektowanie systemów VLSI, w *Laboratorium CAD/CAE systemów mikroelektronicznych*, Skrypt Politechniki Śląskiej (w przygotowaniu).

- [AUTO2] *AutoCAD Release 10*, Reference Manual, Autodesk, Inc., Oakland 1989.
[INTU1] *ISSPICE User's Guide*, Intusoft, San Pedro 1992.

Recenzent: Prof. dr hab. inż. Stanisław Nowak

Wpłynęło do Redakcji 31 stycznia 1994 r.

Abstract

Clear separation of a designing and a fabrication through readable rules for "interface" Designer-Silicon Foundry, provides facilities to constitute organization like MOSIS and EUROCHIP for chip design realization. New program for ASIC Design and Implementation in Institute of Electron Technology (Warsaw) and Universities has been started.

Project has started in 1992. In July ITE had ASIC projects from three different sources:

- 1) three CMOS projects from Institute of Microelectronics and Optoelectronics (Warsaw University of Technology),
- 2) one CMOS gate array project for Institute of Electronics (Silesian University of Technology),
- 3) three NMOS projects from Institute of Informatics (Silesian University of Technology).

All design were supplied in form of CIF files. Then projects that belonged to the first and third group were processed. This work included:

- 1) layer transfer,
- 2) merging of I/O cells with project, if necessary,
- 3) design rule checking,
- 4) project merging into the reticles,
- 5) files creation for pattern generator.

In the beginning of October set of photolithographic masks containing 7 project (area 194 sq. mm) was ready. Since December till end of April ITE lab line was refurbished and improved, so it was not able to start wafer processing. In the end of April the NMOS batch has began, work was finished in June and in August assembled chips were sent to Gliwice.

Now, ITE plans the next run (CMOS) which should start in September and we hope to have CMOS IC's after 22-25 weeks after receiving input files.

The general goal of the Program i.e. to bridge a gap for student/research projects prepared during the Lab course of ASIC/VLSI design at Universities and standard imple-

mentation at Silicon Foundry was fulfilled. As a stimulated result of this action is developing and mastering of the "interface" between designer and Silicon Foundry. This interface is represented with a projects transfer method, file transfer formats (CIF file for projects), design rule formalization and necessity to regulate the CAD tools for DRC, ERC, simulation and HDL of projects description for ASIC/VLSI designers, assembled around the ITE as a Silicon Foundry.

ALGORYTMY OPTYMALIZACJI W ZADANIACH MODELOWANIA

Streszczenie: W artykule przedstawiono opis procedur umożliwiających rozwiązywanie zadań optymalizacji parametrów powojt i spruch ich wykorzystanie w zadaniach modelowania układów dynamicznych ciągłych z problemami pakowania symulacyjnego DNF. Omówiono również wyniki badań prowadzonych opracowanych algorytmów do rozwiązywania typowych zadań optymalizacji parametrycznej.

THE ALGORITHMS OF OPTIMIZATION IN THE MODELLING PROBLEMS

Summary: Several algorithms of optimization required in DNF project are presented. These algorithms for calculating the unconstrained optimization and constrained optimization are addressed. The efficiency of optimization procedures is briefly discussed.

DIE ALGORITHMEN DER OPTIMIERUNG IN DEN MODELLIERUNGS AUFGABEN

Zusammenfassung: Im Artikel wurden sieben Algorithmen der Optimierung dargestellt, die im Fabrik DNF implementiert wurden. Diese Algorithmen ermöglichen die Aufgaben der Parameteroptimierung sowohl mit den Begrenzungen als auch ohne sie zu lösen. Den Algorithmen wurden ebenfalls die Ergebnisse der Experimente mit beigelegt, die ihre Effizienz bewerten.