

Wykład Projektowanie układów ASIC

Plan wg planu studiów

Karta zajęć – wykład

Lp.	Zagadnienie	poziom					liczba godzin
		wiedzy			umiej.		
		A	B	C	D	E	
1.	Technologie produkcji układów scalonych, kroki technologiczne, reguły technologiczne, dostępne elementy i ich ograniczenia.		X				1
2.	Rozrzuty względne i bezwzględne i metody projektowania zapewniające dopasowanie elementów.		X				1
3.	Scalone przetworniki cyfrowo-analogowe, klasyfikacja i podstawowe właściwości.		X				1
4.	Realizacja scalona wybranych przetworników CA.		X				1
5.	Scalone przetworniki analogowo-cyfrowe, klasyfikacja i podstawowe właściwości.		X				1
6.	Układy próbkująco–pamiętające i scalona realizacja wybranych przetworników AC.		X				1
7.	Podstawowe układy cyfrowe: bramki i przerzutniki.		X				1
8.	Szacowanie czasów propagacji i wybór optymalnych wymiarów tranzystorów.		X				1
9.	Projektowanie ścieżek zegarowych. Generacja drzewa zegarowego.		X				1
10.	Zautomatyzowane projektowanie układów cyfrowych. Omówienie możliwości komercyjnych systemów projektowych. (np Cadence).	X					1
11.	Testowanie układów ASIC. Projektowanie układów z uwzględnieniem ich testowania.		X				1
12.	Omówienie interfejsu IEEE1149.1.		X				1
13.	Zastosowanie interfejsu IEEE1149.1 do testowania systemów cyfrowych.		X				1
14.	Problemy związane z projektowaniem systemów mieszanych analogowo-cyfrowych.		X				1
15.	Tendencje rozwojowe układów ASIC.	X					1
Razem							15

Plan skorygowany, dostosowany do przedmiotu z sem. 5, p.t. „Inżynieria Układów i Systemów Scalonych”

Karta zajęć – wykład

Lp.	Zagadnienie	poziom					liczba godzin
		wiedzy			umiej.		
		A	B	C	D	E	
1.	Technologie produkcji układów scalonych, kroki technologiczne, reguły technologiczne, dostępne elementy i ich ograniczenia.		X				0,5
2.	Rozrzuty względne i bezwzględne i metody projektowania zapewniające dopasowanie elementów.		X				0,5
3.	Projektowanie cyfrowych układów scalonych w technice komórek standardowych – synteza z użyciem Cadence RTL Compiler.		X				1
4.	Projektowanie cyfrowych układów scalonych w technice komórek standardowych – implementacja użyciem Cadence SOC RTL Compiler.		X				1
5.	Scalone przetworniki cyfrowo-analogowe, klasyfikacja i podstawowe właściwości.		X				1
6.	Realizacja scalona wybranych przetworników CA.		X				1

7.	Scalone przetworniki analogowo-cyfrowe, klasyfikacja i podstawowe właściwości.		X				1
8.	Układy próbkująco–pamiętające i scalona realizacja wybranych przetworników AC.		X				1
9.	Podstawowe układy cyfrowe: bramki i przerzutniki.		X				1
10.	Szacowanie czasów propagacji i wybór optymalnych wymiarów tranzystorów.		X				1
11.	Projektowanie ścieżek zegarowych. Generacja drzewa zegarowego.		X				1
12.	Testowanie układów ASIC. Projektowanie układów z uwzględnieniem ich testowania.		X				1
13.	Omówienie interfejsu IEEE1149.1.		X				1
14.	Zastosowanie interfejsu IEEE1149.1 do testowania systemów cyfrowych.		X				1
15.	Problemy związane z projektowaniem systemów mieszanych analogowo-cyfrowych.		X				1
16.	Tendencje rozwojowe układów ASIC.	X					1

Razem 15

0. Zasady zaliczenia i literatura.

1) Ocena końcowa = średnia ważona oceny z wykładu (waga 2), laboratorium (waga 1) i projektu (waga 1). Zarówno wykład, laboratorium jak i projekt muszą być zaliczone na ocenę co najmniej 3.0 – w przeciwnym przypadku brak zaliczenia przedmiotu.

2) Ocena z wykładu.

A) Zaliczenie zerowe:

- kolokwium 85pkt w czasie ostatniego wykładu,
- każda obecność na wykładzie 1pkt.,
- możliwe dodatkowe punkty w przypadku aktywności na zajęciach lub przygotowania prezentacji.

lub:

B) Egzamin w terminie podstawowym i poprawkowym:

- 100% punktów do uzyskania w czasie egzaminu.

Ustalenie oceny końcowej wg poniższej skali:

0 - 50pkt – 2.0

51 - 60pkt. – 3.0

61 – 70pkt. – 3.5

71 – 80pkt. – 4.0

81 – 90pkt. – 4.5

91 – 100pkt. – 5.0

Pytania do kolokwium jak również do egzaminu dostępne na stronie:

http://www.ue.eti.pg.gda.pl/~bpa/pua_inz/pua.html

3) Zaliczenie laboratorium: Ocenę końcową stanowi średnia uzyskana ze sprawozdań oraz odpowiedzi ustnych udzielanych prowadzącemu zajęcia. Należy obowiązkowo wykonać wszystkie przewidziane programem laboratorium ćwiczenia.

4) Zaliczenie projektu: Ocenę stanowi średnia uzyskana z prezentacji na ostatnich zajęciach projektowych oraz ocena ze sprawozdania przesłanego prowadzącemu zajęcia.

4) Literatura:

- 1) R. L. Geiger, P. E. Allen, N. R. Strader, „VLSI design techniques for analog and digital circuits“, McGraw-Hill 1990. (str. 19-27 uzysk produkcyjny, okno procesu, str. 39 – elektromigracja, str. - 61 zatraskiwanie się układów CMOS, str. 62 – zabezpieczanie wejść przed ładunkiem elektrostatycznym)
- 2) P. E. Allen, D. R. Holberg, „CMOS analog circuit design“, Sunders College Publishing, 1986.
- 3) P. R. Gray, R. G. Meyer, „Analysis and design of analog integrated circuits“, John Wiley & Son, Inc. 1993.
- 4) P. Gajewski, J. Turczyński, „Cyfrowe układy scalone CMOS“, WKŁ 1990. (zabezpieczenie przed ładunkiem elektrostatycznym str. 31-37).
- 5) „Matching properties of MOS transistors“, M. Pelgrom, A. Duinmaijer, A. Welbres, IEEE Journal of Solid-State Circuits, vol.. 24, no. 5, October 1988.
- 6) „Boundary-Scan Tutorial“, ASSET Inter Tech 2000, www.asset-intertech.com.
- 7) K. P. Parker, „The Boundary-Scan Handbook Second Edition Analog and Digital“, Kluwer Academic Publishers, 1997.

- 8) A. Grochowski, D. Bhattachara, T.R. Viswanathan, K. Laker, „Integrated Circuit Testing for Quality Assurance in Manufacturing: History, Current Status, and Future Trends”, IEEE Transactions on Circuits and Systems – II Analog and Digital Signal Processing, vol. 44, no. 8, pp. 610-633, August 1996.
- 9) “Analog VLSI: Signal and Information processing”, Chapter Design for Testability.
- 10) J. Izydorczyk, „PSpice komputerowa symulacja układów elektronicznych”, Helion, 1993.
- 11) C. Wai-Kai (editor), „The VLSI Handbook”, Taylor & Francis Group, 2006.
- 12) Dokumentacja systemu projektowania układów scalonych CADENCE – dostępne w Katedrze Systemów Mikroelektronicznych.
- 13) E.A. Vittoz, „Analog Layout Techniques”, Practical Aspects of Analog and Mixed-Mode IC Design, Portland, Oregon, USA, July 18-22, 1994.

1. Informacje wstępne.

1.1. TECHNOLOGIA [1,2,3,11]

Dobra znajomość procesu technologicznego zapewnia możliwość wszechstronnego wykorzystania technologii i optymalizacji projektu całego układu scalonego. Bazując na tej wiedzy, rzeczywisty kształt topografii układu scalonego może być rozważony pod względem elementów pasożytniczych a istnienie ich może być uwzględnione już w procesie projektowania. Nowoczesne techniki projektowe bardzo często opierają się na szczególnych właściwościach procesu produkcyjnego. Wiedza o charakterystycznych właściwościach procesu produkcyjnego umożliwia projektantowi oszacowanie uzysku produkcyjnego, stosowanie kompromisu między możliwościami, uzyskiem i prostotą układową. W niniejszym opracowaniu technologia produkcji układów scalonych rozważana jest z jakościowego punktu widzenia. Przedstawiono szczegółowo proces technologiczny CMOS.

1.1.1 Etapy technologiczne produkcji układów scalonych.

Przygotowanie krysztalu. Podłożem dla bipolarnych i MOS-owych układów scalonych jest najczęściej krysztal krzemu lekko domieszkowany domieszkami typu n- lub p-. Podłoże służy jako fizyczne medium dla produkcji układu scalonego jak również jako część obwodu elektrycznego. Podłoża te są cięte z dużego walca krzemowego o strukturze krystalicznej o długości do ok. 2m i średnicy od 1 do kilkunastu cali. Grubość płytek krzemu (wafer) wynosi 250μm do 400μm. Z elektrycznego punktu widzenia mogłyby być one cieńsze, ze względów praktycznych (pękanie) są one grubsze. Średnica płytek krzemowych używanych jako podłoża układów scalonych wzrasta (przykładowo kiedyś 4" teraz ponad 10"). Daje to możliwość produkcji większej ilości układów scalonych przypadających na jedną płytkę (wafer) lub produkcji większych układów scalonych.

Maskowanie. Maski układów scalonych są wysokokontrastowymi (czarno-białe) pozytywami lub negatywami fotograficznymi. Są one używane do selektywnego zabezpieczania powierzchni przed promieniami światła w czasie procesu fotolitografii. Maski są zazwyczaj wykonywane ze szkła powleczonego cienką warstwą nieprzezroczystego metalu. Czasami spotyka się też maski z materiałów mniej odpornych i oczywiście tańszych. Maski wytwarzane są z cyfrowego opisu kształtu maski (wygenerowanego z programu projektowego). Jest kilka metod wytwarzania masek. Pierwsza z nich polega na fotograficznym pomniejszaniu dużej kopii żądanego kształtu wydrukowanego lub narysowanego za pomocą drukarki lub plotera. Metoda ta była używana szeroko w przeszłości, obecnie została wyparta przez dwie następne. Druga metoda polega na wykorzystaniu promienia lasera do wytworzenia obrazu maski. Obie metody wymagają kroków o wysokiej rozdzielczości i kamery pomniejszającej do wytworzenia finalnych masek. Obraz pośredni nazywany jest *reticle* i jest zazwyczaj 5 lub 10 krotnie większy od rzeczywistego. Trzecia metoda wykorzystuje promień elektronów (E-beam) do wytworzenia obrazu od razu o wymiarach

rzeczywistych bezpośrednio na masce finalnej. Ta metoda daje maski o najlepszej jakości i jest używana szeroko dla technologii niskoskalowalnych. Niestety wymaga względnie dużego czasu i drogiego sprzętu.

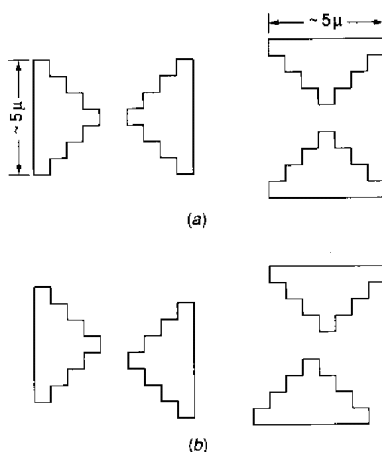
Proces fotolitograficzny Warstwa światłoczuła zwana fotomaską (photoresist) jest lepkim płynem. Jest ona nanoszona w postaci cienkiej jednorodnej warstwy na całą płytkę krzemową, (ok. 1µm poprzez obracanie) po uprzednim jej czyszczeniu. Następnie warstwa ta jest utwardzana poprzez wypalanie. Fizyczne właściwości warstwy światłoczułej mogą być zmienione poprzez naświetlanie przez przezroczyste powierzchnie masek (przez naświetlanie kontaktowe lub przez projekcję), poprzez projekcję światła przez maskę pośrednią (*reticle*) zawierającą tę samą informację (nazywane bezpośrednim działaniem na płytkę krzemową) lub przez wiązkę elektronów (E-beam), który przemiata żądane obszary. Po naświetlaniu warstwa światłoczuła jest wywoływana w celu selektywnego usunięcia z miejsc gdzie jest niepożądana. Często następnym krokiem jest ponowne utwardzenie pozostałej warstwy światłoczułej przez następne wypalanie. Możliwe jest stosowanie zarówno warstw światłoczułych negatywowych jak i pozytywowych. W przypadku fotomasek negatywowych nienaświetlone powierzchnie są usuwane w procesie wywoływania, w przypadku pozytywowych naświetlone regiony są usuwane. Warstwy negatywowe są bardziej odporne na działanie wytrawiaczy, lepsze rozdzielczości typowo uzyskuje się dla pozytywowych. Warstwa światłoczuła służy jako warstwa ochronna dla wielu wytrawiaczy, odczynników utleniających czy też jako bariera ochronna dla implantacji jonów.

W celu uzyskania prawidłowej pracy układu scalonego i dużego uzysku produkcyjnego, kolejne maski odpowiadające kolejnym warstwom materiału, powinny być ustawiane precyzyjnie względem siebie. W tym celu wykorzystuje się znaki pozycjonujące, które są zazwyczaj zawarte łącznie z opisem układu w momencie wytwarzania maski, tak więc znaki te są widoczne na płytce krzemowej podczas produkcji. Specjalna maszyna jest używana do wyrównania masek i do naświetlania płytek. Rys. 0.1 pokazuje typowe znaki pozycjonujące oraz możliwe ich przesunięcie. Fizyczne wymiary zależą od typu naświetlania użytego w czasie produkcji. Maszyny wyrównujące, które wykorzystują naświetlanie kontaktowe mają wiele kopii tego samego układu o wymiarach rzeczywistych. Takie maszyny mają dużą wydajność i są stosunkowo niedrogie. Niestety, z powodu zniszczeń podczas bezpośredniego styku z materiałem naświetlanym, duże maski mają krótki czas życia - typowo 3 do 10 naświetleń. To zwiększa efektywny koszt masek. Maszyny pozycjonujące wykorzystujące technikę bezpośredniego działania na płytkę krzemową typowo używają masek pośrednich (*reticle*) o 5-cio krotnie większych wymiarach od rzeczywistych. Typowo maska zawiera pojedynczą kopię układu, chociaż kilka kopii może być wykorzystanych dla małych układów scalonych. Obraz jest optycznie zmniejszany do wymiarów rzeczywistych podczas naświetlania. Płytkę krzemową jest przesuwana do kolejnych pozycji aż do naświetlenia całej powierzchni. Czas życia takiej maski jest bardzo długi gdyż jest brak fizycznego kontaktu z naświetlanym podłożem. Niestety przez ciągłe ruchy podłoża powtarzalność pozycjonowania maski jest gorsza oraz wydajność jest mniejsza. Z powodu precyzyjnego i powtarzalnego ruchu płytki krzemowej wymagane wyposażenie jest droższe. Oba typy pozycjonowania są obecnie szeroko stosowane w przemyśle.

Istnieje metoda (bardzo popularna i praktyczna), która łączy ekonomię pozycjonowania naświetlania kontaktowego i długi czas życia masek niekontaktowych. W tym przypadku cienka ochronna membrana (nazywa ang. *pelicle*) jest umieszczona pomiędzy chromową maską i emulsją światłoczułą. Chociaż membrana sama może być zabrudzona lub zadrapana jest ona umieszczona wystarczająco daleko od maski, która jest poprzez to chroniona i ma długi czas życia.

Czwarta metoda naświetlania nie używa masek. Zamiast tego wąski strumień elektronów (E-beam) jest selektywnie skupiany na płytce krzemowej, używając przesunięć płytki jak w metodzie bezpośredniej można naświetlić jego całą powierzchnię. Ta sama baza danych, która służy do opisu masek może być użyta do sterowania promienia elektronowego. Ta metoda daje największą rozdzielczość ale jest bardzo kosztowna ze względu na bardzo drogie wyposażenie i ma

mniejszą wydajność. Jej stosowanie jest uzasadnione tylko dla najbardziej wymagających zastosowań.



Rys. 0.1. Znaki wyrównujące.

Osadzanie W czasie procesu produkcyjnego układów scalonych istnieje potrzeba nanoszenia cienkich warstw rozmaitych materiałów na powierzchnię płytki krzemowej. Często warstwy te są bardzo cienkie (200Å lub mniej dla niektórych warstw SiO_2) lub ich grubości mogą dochodzić do $20\mu\text{m}$ dla układów grubowarstwowych. Nanoszone mogą być warstwy: izolatorów, rezystywne, przewodzące, półprzewodnikowe typu n lub p , domieszki, które są kolejno wciskane głębiej do podłoża. Stosowane są następujące techniki: fizyczne nanoszenie oparów (naparowywanie-*evaporation*, napyłanie katodowe-*sputtering*), CVD (osadzanie chemiczne z warstwy gazowej-*chemical vapor deposition*) oraz sitodruku (*screen printing*). Z wyjątkiem sitodruku wszystkie typy osadzania są nieselektywne i są wykonywane jednorodnie na całej warstwie płytki krzemu.

Naparowywanie odnosi się do naparowywania materiałów poprzez kontrolowanie temperatury i ciśnienia materiału nanoszonego (host material). Grubość warstwy zależy od czasu naparowywania oraz warunków przebiegu procesu.

Napyłanie katodowe. Używa się bombardowania materiału nanoszonego jonami o bardzo dużej energii w celu oderwania od niego molekuł, które są następnie osadzone na powierzchni płytki krzemowej (jak również do innych powierzchni). Często używa się jednoczesnego bombardowania dwóch różnych materiałów w różnym stopniu dla uzyskania pożądanej charakterystyki osadzanego materiału. Takie nanoszenie cienkich warstw nazywa się *cosputtering*. Przy użyciu pewnych materiałów napyłanie katodowe ma zaletę nad naparowywaniem w postaci lepszej integralności osadzanego materiału.

CVD jest otrzymywane na dwa różne sposoby: (1) poprzez reakcję chemiczną dwóch różnych gazów w pobliżu podłoża, reakcja powoduje powstawanie molekuł, które przylegają do podłoża, (2) poprzez dekompozycję pyrolityczną (powodowaną przez podgrzanie) pojedynczego gazu, który także uwalnia żądane molekuły.

Wytrawianie. Wytrawianie jest selektywnym usuwaniem niepotrzebnego materiału z powierzchni podłoża. Fotomaska i maski są używane do selektywnego naświetlania powierzchni podłoża. Po odwzorowaniu kształtu maski fizyczne właściwości powierzchni mogą być zmieniane przez wytrawianie. Pojedynczy układ scalony jest poddawany wielokrotnemu wytrawianiu w różnego typu wytrawiaczach. Związki chemiczne używane do wytrawiania są wybierane w ten sposób by usunąć niepotrzebny materiał na płytce krzemowej a pozostawić zabezpieczone regiony. Podsumowanie właściwości podstawowych substancji używanych w produkcji układów scalonych jest przedstawione w tabeli 1.1.

Są dwa typy wytrawiaczy: suche i mokre. *Wytrawiacze mokre*, często nazywane wytrawiaczami chemicznymi są cieczami, które nanosi się na powierzchnie płytek półprzewodnikowych. Chociaż były one szeroko stosowane w przeszłości wytrawiają one we wszystkich kierunkach z jednakową prędkością. Podtrawienie poprzeczne jest niepożądane.

Chociaż szerokość niewytrawionych powierzchni jest o rząd wielkości większa od grubości wytrawianych części, niejednorodność trawienia bocznego powoduje istotne zmiany w pożądanych charakterystykach wytwarzanych elementów. *Suche wytrawianie*, nazywane także wytrawianiem jonowym, jest kierunkowe i z tego powodu jest mniej podatne na niepożądane podtrawienie poprzeczne. Techniki trawienia suchego zawierają: trawienie przez rozpylanie jonowe, trawienie jonowe i trawienie plazmowe. Ponieważ nie używa się odczynników chemicznych istotne zmniejszenie kosztów jest związane z brakiem konieczności ich utylizacji. Prędkość trawienia suchego jest ogólnie mniejsza od trawienia mokrego.

Charakterystyki trawienia idealnego mokrego i idealnego kierunkowego są przedstawione na rys. 0.2. Wytrawianie niekierunkowe jest nazywane izotropowym. Kształt krawędzi przy takim wytrawianiu jest w przybliżeniu okręgiem o promieniu r i środkiem w punkcie A. Jeśli wytrawianie zostanie zatrzymane dokładnie w momencie odsłonięcia warstwy spodniej promień będzie wynosił T , czyli będzie równy grubości warstwy. W przypadku zatrzymania nieprecyzyjnego promień będzie wynosił T_1 czyli więcej niż grubość trawionej warstwy, więc zarówno efektywny otwór jak i podtrawienie będą większe od pożądanych.

Tabela 1.1. Charakterystyka najczęściej używanych materiałów
I. Materiały do produkcji układów scalonych

Cel	Materiał	Komentarz
Krzemowe podłoża krystaliczne	SiCl ₄ SiHCl ₄ SiO ₂ (piasek)	Źródło krzemu dla wzrostu monokryształu krzemu. Źródło krzemu dla wzrostu monokryształu krzemu. Źródło krzemu dla wzrostu monokryształu krzemu.
Warstwy krzemowe krystaliczne i polikrystaliczne	SiCl ₄ i H ₂ SiH ₄ SiH ₂ Cl ₂	Gaz wodorowy odcina atomy Cl w celu uformowania stałego krzemu. Podgrzewany powoduje uwolnienie (pirolizę) gazu H ₂ Podgrzewany powoduje uwolnienie (pirolizę) gazu HCl
Tlenki	O ₂ H ₂ O (para wodna) SiH ₄ i O ₂	Używany do wzrostu SiO ₂ poprzez podgrzewanie Używany do wzrostu SiO ₂ poprzez podgrzewanie Używany w CVD do osadzania SiO ₂ i wytworzenia warstwy szkła ochronnego SiO ₂
Warstwy azotków	Si ₄ i NH ₃ SiCl ₄ i NH ₃	Amoniak powoduje uwolnienie wodoru oraz osadzenie Si ₃ N ₄ Amoniak powoduje uwolnienie HCl oraz osadzenie Si ₃ N ₄
Wytrawiacze mokre	HF HF i HNO ₃	Kwas fluorowodorowy wytrawia SiO ₂ ale nie wytrawia Si ₃ N ₄ , fotomasek Wytrawia Si
Wytrawiacze suche	H ₃ PO ₄ CHF ₃ C ₃ F ₈ SF ₆ CF ₄ CCl ₄	Gorący kwas fosforowy wytrawia Si ₃ N ₄ ale nie wytrawia SiO ₂ , usuwa niektóre typy fotomasek Wytrawia SiO ₂ Wytrawia SiO ₂ Wytrawia krzem Wytrawia Si ₃ N ₄ Wytrawia aluminium
Oślanianie, odwzorowywanie	Fotomaska SiO ₂	Używany jako bariera dla implantacji jonów, używany także jako szablon do wytrawiania SiO ₂ ponieważ fotomaska nie jest trawiona przez HF - często używany do wytrawiania SiO ₂ . Służy do oślaniania przed niektórymi typami

	SiN ₄	domieszek typu p- i n-. Używany jako warstwa ochronna na krzemie lub SiO ₂ w celu ochrony przed termicznym wzrostem SiO ₂ . Służy także jako bariera dla niskoenergetycznej implantacji jonów, chociaż w przypadku implantacji wysokoenergetycznej cienkie warstwy przepuszczają domieszki. Służy także jako ochrona przed dyfuzją domieszkami takimi jak Ga, Al, Zn i Na.
--	------------------	---

II. Źródła domieszek

Typ domieszki	Domieszka	Materiał
n	Arsen Antymon Fosfor	As ₂ O ₃ , AsH ₃ Sb ₂ O ₃ , Sb ₂ O ₄ P ₂ O ₅ , POCl ₃ (płyn), PH ₃ (gaz-implantacja lub dyfuzja)
p	Gal Glin Bor	BN (stałe), BBr ₃ (ciecz), B ₂ O ₃ (gaz), B ₂ H ₆ (gaz), BF ₃ (do implantacji)

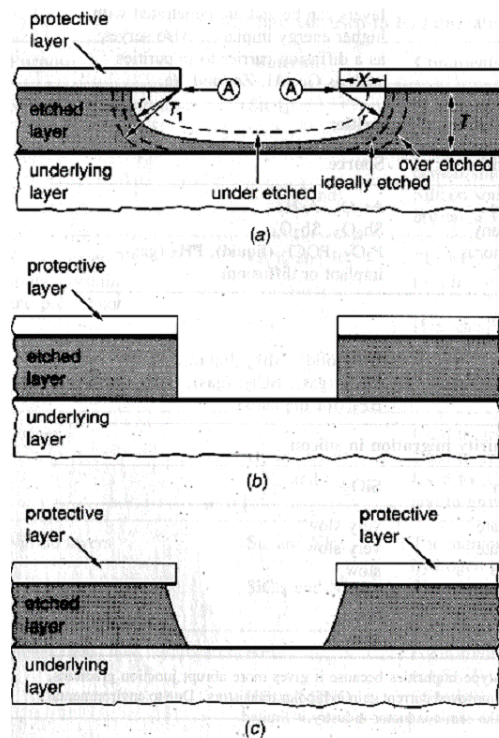
III. Migracja domieszek w krzemie

Domieszka	Krzem	SiO ₂
Arsen ¹	średnia	bardzo wolna
Antymon	średnia	bardzo wolna
Fosfor	szybka	wolna
Gal	średnia	szybka
Glin	szybka	szybka
Bor	szybka	wolna

Idealne wytrawienie kierunkowe nazywane jest anizotropowym. Należy zauważyć, że takie wytrawienie powoduje powstanie bardzo stromego zbocza warstwy wytrawianej i stwarza problemy dla jednorodnego nakładania kolejnych warstw w okolicy tej krawędzi.

Wytrawianie selektywne określa wytrawianie które ma różne prędkości w różnych kierunkach ale jest mniej kierunkowe niż anizotropowe. Taki typ wytrawiania przedstawiony jest na rys. 1.2(c). Wytrawianie selektywne może być dobrym rozwiązaniem w przypadku gdy kolejne nakładanie warstw na strome krawędzie jest trudne do wykonania.

¹ Arsen jest często preferowanym typem domieszek typu *n* gdyż daje większy gradient domieszkowania złącz półprzewodnikowych i w ten sposób uzyskuje się lepsze właściwości częstotliwościowe i zwiększone wzmocnienie prądowe tranzystorów bipolarnych. Jednak ze względu na ochronę środowiska używanie arsenu w przemyśle elektronicznym jest ograniczone.



Rys. 1.2. Charakterystyki wytrawiaczy: (a) izotropowe, (b) anizotropowe, (c) selektywne.

Dyfuzja. Dyfuzja w przypadku produkcji układów scalonych oznacza kontrolowaną migrację domieszek w głąb podłoża lub sąsiadujących materiałów. Wynikowy profil koncentracji domieszek, który gra kluczową rolę w parametrach układu scalonego, jest zniekształcony przez temperaturę i czas jak również zależność czasowo - temperaturową podczas poszczególnych kroków technologicznych. Kolejne dyfuzje w ogólności powodują dodatkową migrację wcześniej wykonywanych. W rzeczywistości proces dyfuzji trwa ciągle, ale dla normalnych temperatur pracy układów scalonych potrzeba dziesiątek lat aby dodatkowy ruch domieszek spowodował istotne zmiany.

Są różne techniki osadzania domieszek. Jako źródło domieszek może być wykorzystany materiał stały bądź gazowy. Domieszki mogą być także przyspieszane w celu bombardowania nimi podłoża, w ten sposób zostają wtrącone w strukturę podłoża w pobliżu jego powierzchni. Taka technika nazywana jest implantacją jonów i daje bardzo dokładną kontrolę koncentracji domieszek za cenę zniszczenia struktury krystalicznej w pobliżu powierzchni.

Po etapie wytrącenia lub osadzenia domieszek następuje proces dyfuzji. Jej celem jest migracja nośników ze stałej lub gazowej warstwy. Krok dyfuzji po implantacji jonów ma na celu naprawę zniszczonej wcześniej struktury krystalicznej w pobliżu powierzchni oraz dodatkową migrację domieszek.

Podobnie jak w przypadku wytrawiania, kierunek zachodzenia dyfuzji jest trudny do kontroli. Domieszki zazwyczaj dyfundują w sposób izotropowy, we wszystkich kierunkach z jednakową prędkością, podobnie jak wytrawianie na rys. 0.2(a).

Przewodniki i rezystory. Jako przewodniki dla połączeń elementów wewnątrz układu scalonego najczęściej stosuje się glin oraz rzadziej inne metale. Metale te są najczęściej najpierw osadzone, następnie ukształtowane (przez maskowanie) i wytrawione tak by pozostawić je tylko w miejscach pożądanых. Grubość warstwy najczęściej stosowanego glinu wynosi typowo 6000Å-8000Å ale może również dochodzić do 20,000Å dla procesów analogowych z pojedynczą warstwą metalu. Połączenia metalowe są szczególnie przydatne dla połączeń przez które płynie duży prąd. W takim przypadku ścieżki metalowe muszą być na tyle szerokie by uniknąć zjawiska migracji metalu (przemieszczania się atomów metalu) pod wpływem przepływającego prądu. Zjawisko to nazywane jest elektromigracją i może być porównane do erozji eolicznej lub spowodowanej zabrudzeniem. W

przypadku pojawienia się znaczącej migracji metalu przewodniki przerywają się i w rezultacie układ scalony ulega uszkodzeniu. Elektromigracja metalu jest pomijalna jeśli maksymalna gęstość płynącego prądu nie przekroczy pewnej wartości progowej. Dla aluminium wartość tego progu wynosi około $1\text{mA}/\mu^2$. Wartość progowa prądu dla innych materiałów mieści się w granicach od $0.05\text{mA}/\mu^2$ do $2\text{mA}/\mu^2$.

Przewodniki niemetaliczne są używane do połączeń w przypadku gdy płynący przez nie prąd jest mały. Typowo są one znacznie gorszymi przewodnikami niż metale i dlatego może na nich występować znaczny spadek napięcia. Takie materiały znajdują także ograniczone zastosowania do budowy rezystorów.

Polikrzem jest jednym z najpopularniejszych przewodników niemetalicznych. Polikrzem różni się od monokryształu krzemu, który jest używany jako podłoża układów scalonych tym, że zbudowany jest z wielu małych ułożonych bezładnie i przypadkowo ziaren krzystalu krzemu. Chociaż pod względem chemicznym jest identyczny jak krzem krystaliczny ich właściwości elektryczne różnią się znacznie. Mocno domieszkowany polikrzem jest dobrym przewodnikiem, lekko domieszkowany jest dobrym rezystorem. Polikrzem jest często używany jako materiał na bramki tranzystorów MOS i jako elektroda kondensatorów. Polikrzem może być nanoszony na warstwy SiO_2 . SiO_2 może być wytworzone na polikrzemie i jest często używany jako dielektryk w procesach z dwoma warstwami polikrzemu. Właściwości polikrzemu zależą od wielkości ziaren krzystalu. Może on być nanoszony na szeroki wachlarz materiałów a prędkość wzrostu warstwy może być całkiem duża. Warstwy polikrzemowe mają typowo grubość około 2000Å.

Krzemiany i metale wysokotopliwe są często używane na górnej warstwie lub zamiennie zamiast polikrzemu dla produkcji przewodników. Te materiały są często dużo lepszymi przewodnikami niż polikrzem.

Utlenianie. Utlenianie jest procesem podczas którego cząsteczki tlenu z gazu nad przetwarzaną płytką powodują wzrost warstwy tlenku na jej powierzchni. Ponieważ przeważnie podłożem jest krzem proces utleniania powoduje powstanie dwutlenku krzemu. Prędkość tworzenia się warstwy SiO_2 jest funkcją koncentracji domieszek i temperatury podłoża. Warstwy SiO_2 służą jako bardzo dobre izolatory pomiędzy innymi warstwami. Podczas wzrostu tlenku na powierzchni krzemowej następuje jego rozwijanie się w dół (47%) oraz ponad powierzchnię (53%).

Alternatywnie SiO_2 może zostać naniesiony przez CVD. Taką technikę stosuje się gdy dwutlenek krzemu musi pokryć powierzchnie inne niż krzem. Proces CVD jest wykonywany w niższych temperaturach niż zwykle utlenianie co zmniejsza wtórną dyfuzję wcześniej aplikowanych materiałów. Warstwy nanoszone metodą utleniania są jednak bardziej jednorodne.

Wykorzystuje się także inne typy tlenków. Przykładem może być szkło fosforowo-krzemowe (PSG). Jest ono często używane jako warstwa izolująca nanoszona na polikrzem.

W niektórych procesach azotek krzemu Si_3N_4 jest używany jako dielektryk pomiędzy dwoma warstwami polikrzemu. Stała dielektryczna azotku jest około 4 razy większa niż dwutlenku krzemu co umożliwia wykonywanie większych gęstości pojemności dla tych samych grubości izolatorów.

Jako izolatory stosuje się również poliamidy. Mają one dobre własności pokrywania warstw z ostrymi krawędziami. Często stosowane są pomiędzy warstwami metalicznymi.

Epitaksja. Wzrost epitaksjalny jest wykonywany zazwyczaj jako CVD. Jego skutkiem jest idealnie krystaliczne przedłużenie warstwy podłoża. Wzrost warstw epitaksjalnych jest na tyle wolny, że można łatwo dodać do nich domieszki, które formują krystaliczne domieszkowane warstwy epitaksjalne np. typu n lub typu p .

1.1.2 Pakowanie i testowanie.

Po zakończeniu procesu technologicznego układy scalone są testowane i pakowane. Najpierw następuje weryfikację procesu w celu upewnienia się, że proces leży w *oknie procesu*.

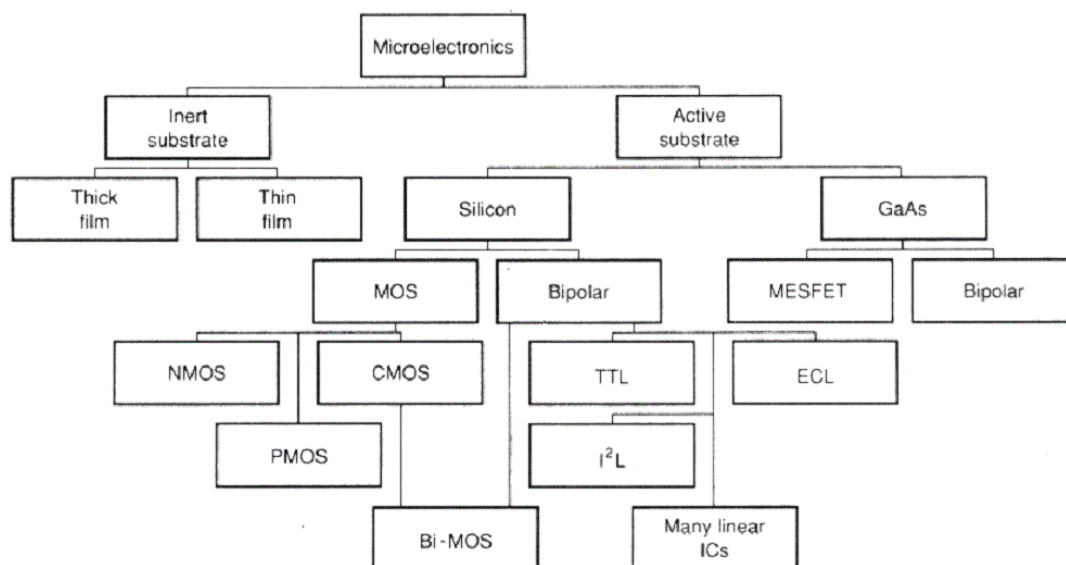
Aby to wykonać stosuje się *struktury testowe* (PCB, PCM) zawierające układy specjalnie przygotowane do testowania. Takie układy są umieszczane na powierzchni płytki krzemowej. Ostatnio modne jest umieszczanie struktur testowych na liniach trasujących podział pomiędzy chipami – w celu eliminacji użycia dodatkowej powierzchni. Próbnik dokonuje kontakt mechaniczny ze strukturami testowymi w celu pomiaru parametrów elektrycznych. Zakładając, że proces mieści się w oknie technologicznym poszczególne *die* są próbkowane i testowane oraz w razie ich niesprawności oznaczane specjalnym atramentem. Następnie płytka krzemowa jest trasowana i cięta przy użyciu piły. Poszczególne *die* są następnie przyklejane do podłoża lub bezpośrednio do obudowy a następnie wykonywane jest lutowanie (bonding) od poszczególnych pól kontaktowych struktury *die* do wyprowadzeń układu scalonego (ażura). Połączenia są zazwyczaj ze złota lub aluminium o średnicy typowo 1mil. Gdy lutowanie jest zakończone uformowane są obudowy (lub zamykane) oraz przeprowadzony jest elektryczny test końcowy.

Technologia pakowania nie zmieniła się znacznie od lat 70 tych. Obecnie mówi się, że pakowanie jest jednym z wąskich gardeł w rozwoju produkcji układów scalonych.

1.1.3 Procesy półprzewodnikowe.

Obecnie mamy do dyspozycji głównie trzy następujące procesy technologiczne: NMOS (lub PMOS), CMOS oraz proces bipolarny. Pierwsze dwa są nazywane procesami MOS (Metal Oxide Semiconductor) chociaż jak się za chwilę okaże nazwa ta jest obecnie historycznie uzasadniona – obecnie bowiem nie używa się metalu lecz warstw polikrzemu jako bramki. Czwarty typ procesu jest połączeniem procesów CMOS i bipolarnego w jeden nieco bardziej złożony i nazywany BiCMOS. Piątą metodą jest konstruowanie układów scalonych hybrydowych. Procesy te są przedstawione na rys. 0.3.

W tej części zostaną omówione ogólne procesy MOS, które są bardzo podobne do spotykanych w przemyśle – i rozpowszechnianych poprzez amerykańską firmę MOSIS. W dalszej części przedstawione są również reguły projektowe oraz podstawowe parametry procesów. Szczegóły dotyczące typowego procesu Bi-CMOS nie są przedstawione ale podstawowe zasady są takie same jak dla procesów MOS i bipolarnych.



Rys. 1.3. Główne procesy technologiczne stosowane w mikroelektronice.

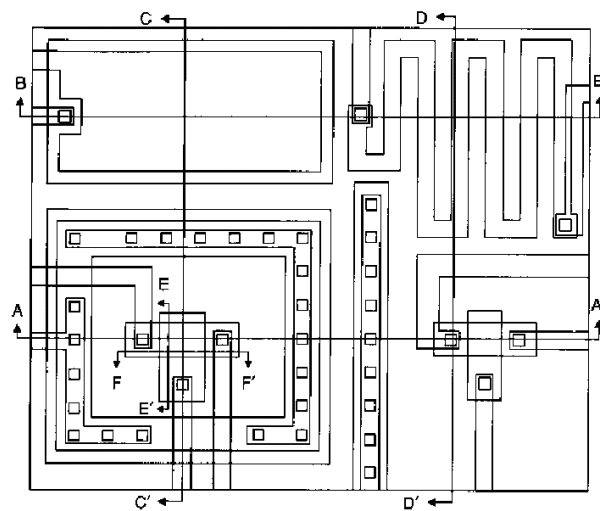
Procesy NMOS oraz CMOS są do siebie podobne w obu przypadkach elementami aktywnymi są tranzystory MOS. W procesie NMOS dostępne są tranzystory z kanałem typu N a w procesie CMOS dostępne są tranzystory z oboma typami kanałów. Proces CMOS w porównaniu z procesem NMOS (PMOS) daje możliwość uproszczenia projektu schematu elektrycznego za cenę zwiększenia ilości kroków technologicznych. Obecnie procesy PMOS i NMOS zostały niemal całkowicie wyparte przez CMOS ze względu na: zmniejszenie poboru mocy, możliwości projektowania układów analogowych, zmniejszenia wymiarów układów. Proces bipolarny zawiera

jako elementy aktywne tranzystory bipolarne BJT (ang. Bipolar Junction Transistor). Dla układów cyfrowych większe upakowanie możliwe jest jednak dla procesów CMOS. Proces bipolarny oferuje obecnie większe częstotliwości i jest częściej stosowany do układów analogowych.

Procesy hybrydowe zawierają cienko i grubowarstwowe elementy pasywne, które są na jednym lub więcej pasywnych podłożach oddzielonych od podłoża na którym wytworzone są elementy aktywne. Wszystkie podłoża przytwierdzone są do wspólnego nośnika. Układy takie są drogie. Są one stosowane dla układów wymagających precyzyjnych i niezależnych od temperatury wartości elementów pasywnych.

Terminologia. Symbol $n+$ oznacza silnie domieszkowany rejon półprzewodnikowy domieszkami typu n . Symbol $n-$ oznacza słabo domieszkowanie, podobnie $p+$ oraz $p-$ oznacza silnie / słabo domieszkowane rejon półprzewodnika domieszkami typu p . Nie będzie znaku $+$ lub $-$ w przypadku gdy dane miejsce jest domieszkowane z koncentracją pomiędzy $+$ i $-$, lub gdy koncentracja domieszek nie ma znaczenia.

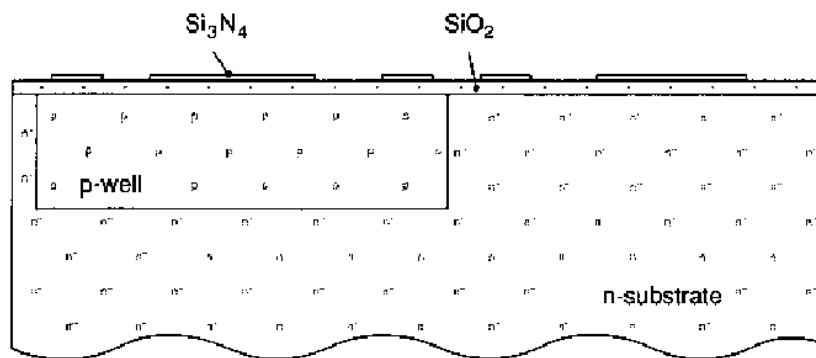
1.1.4 Przebieg typowego procesu p-well CMOS z przykładem prostego układu scalonego (tranzystor „n”, „p”, rezystor i kondensator)



Widok końcowy

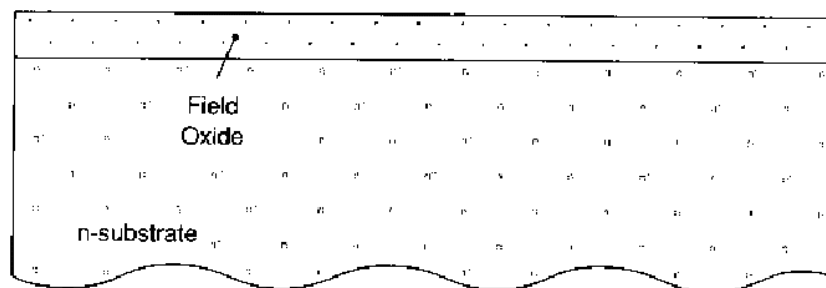
1. Czyszczenie płytki krzemowej (wafer)
2. Wzrost cienkiego tlenku
3. Naniesienie warstwy fotomaski
4. Odwzorowanie obszarów wyp
5. Wywołanie fotomaski
5. Dyfuzja domieszek typu „p”
6. Usunięcie fotomaski
7. Usunięcie cienkiego tlenku
8. Wzrost cienkiego tlenku
9. Naniesienie warstwy Si_3N_4
10. Naniesienie warstwy fotomaski
- 10.1. Odwzorowanie obszaru aktywnego (Si_3N_4) (MASKA #1)
- #2)
13. Wywołanie fotomaski
14. Wytrawienie Si_3N_4
15. Usunięcie fotomaski
- opcjonalne korygowanie napięcia progowego
- A.1 Naniesienie fotomaski
- A.2 Odwzorowanie antypodłoża (MASKA #A1)

- A.3 Wywołanie fotomaski
- A.4 Implantacja domieszek typu „n”
- A.5 Usunięcie fotomaski

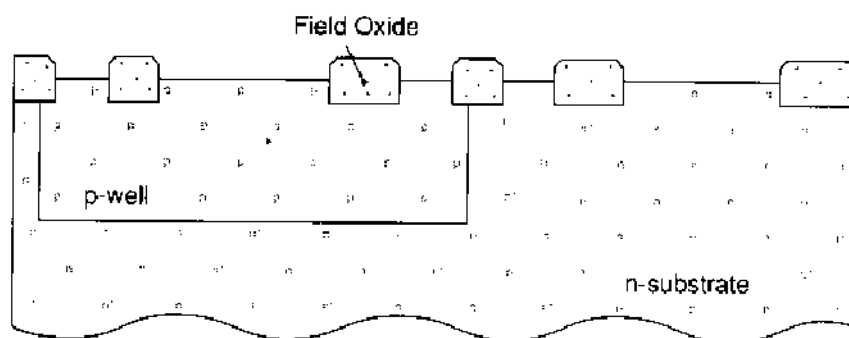


Przekrój A-A po kroku 15.

15. Wzrost tlenku grubego (polowego)



Przekrój B-B po kroku 15.



Przekrój A-A po kroku 15.

16. Usunięcie Si_3N_4
11. Usunięcie cienkiego tlenku
18. Wzrost tlenku bramkowego
15. Naniesienie warstwy polikrzemu (POLY1)
21. Naniesienie fotomaski
22. Odwzorowanie warstwy polikrzemu (POLY1) (MASKA #3)
23. Wywołanie fotomaski
24. Wytrawienie polikrzemu
12. Usunięcie fotomaski
 - opcjonalne kroki dla procesu z dwoma warstwami polikrzemu
 - B.1 Usunięcie cienkiego tlenku
 - B.2 Wzrost cienkiego tlenku
 - B.3 Naniesienie polikrzemu (POLY2)
 - B.4 Naniesienie fotomaski

-B.5 Odwzorowanie POLY2

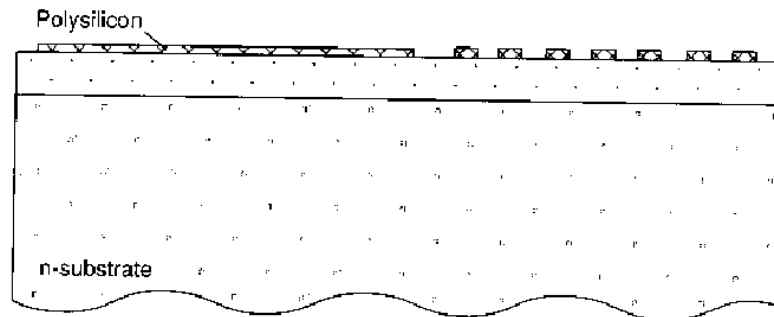
(MAKSA #B1)

-B.6 Wywołanie fotomaski

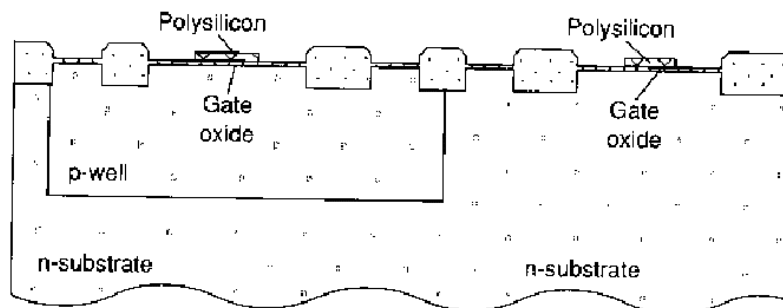
-B.7 Wytrawienie polikrzemu

-B.8 Usunięcie fotomaski

-B.9 Usunięcie cienkiego tlenku



Przekrój B-B po kroku 12.



Przekrój A-A po kroku 12.

12. Naniesienie fotomaski

13. Odwzorowanie obszarów: drenów i źródeł tranzystorów z kanałem typu „p”, pierścieni ochronnych P^+ (kontaktów omowych do wysp typu „p”) (MASKA #4)

14. Wywołanie fotomaski

28. Implantacja P^+

30. Usunięcie fotomaski

31. Naniesienie fotomaski

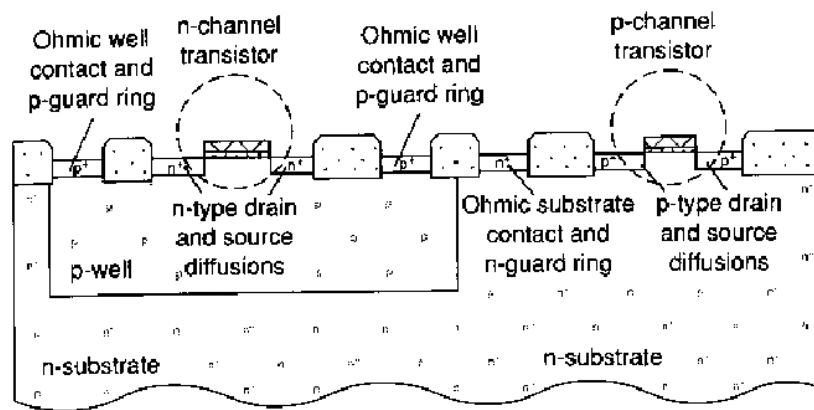
32. Odwzorowanie obszarów: drenów i źródeł tranzystorów z kanałem typu „n”, pierścieni ochronnych N^+ (kontaktów omowych do podłoża typu „n”) (MASKA #5)

33. Wywołanie fotomaski

34. Implantacja N^+

35. Usunięcie fotomaski

35. Usunięcie cienkiego tlenku

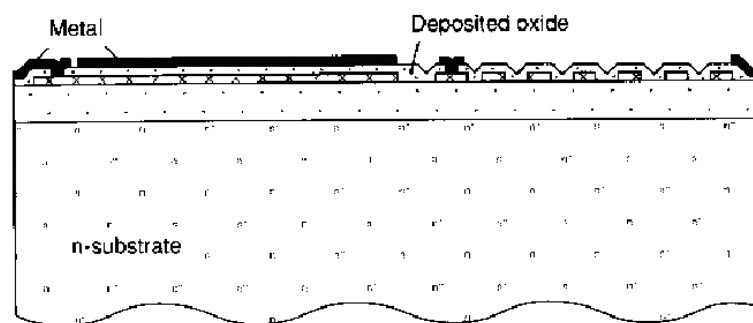


Przekrój A-A po kroku 35.

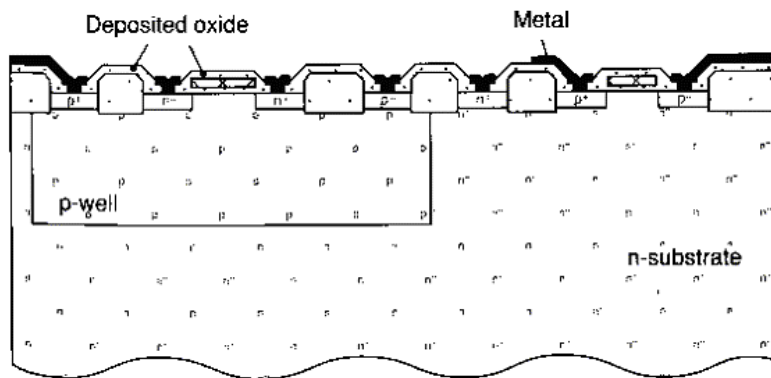
36. Wzrost tlenku
37. Naniesienie fotomaski
38. Odwzorowanie otworów kontaktów (MASKA #6)
40. Wywołanie fotomaski
41. Wytrawienie tlenku
42. Usunięcie fotomaski
43. Naniesienie metalu
44. Naniesienie fotomaski
45. Odwzorowanie metalu (MASKA #7)
45. Wywołanie fotomaski
46. Wytrawienie metalu
47. Usunięcie fotomaski

opcjonalne kroki dla procesu z podwójną warstwą metali

- C.1 Usunięcie cienkiego tlenku
- C.2 Naniesienie tlenku
- C.3 Naniesienie fotomaski
- C.4 Odwzorowanie przelotek (MASKA #C1)
- C.5 Wywołanie fotomaski
- C.6 Wytrawienie tlenku
- C.7 Usunięcie fotomaski
- C.8 Naniesienie warstwy metalu METAL2
- C.9 Naniesienie fotomaski
- C.10 Odwzorowanie metalu METAL2 (MASKA #C2)
- C.11 Wywołanie fotomaski
- C.12 Wytrawienie metalu
- C.13 Usunięcie fotomaski



Przekrój B-B po kroku 47.



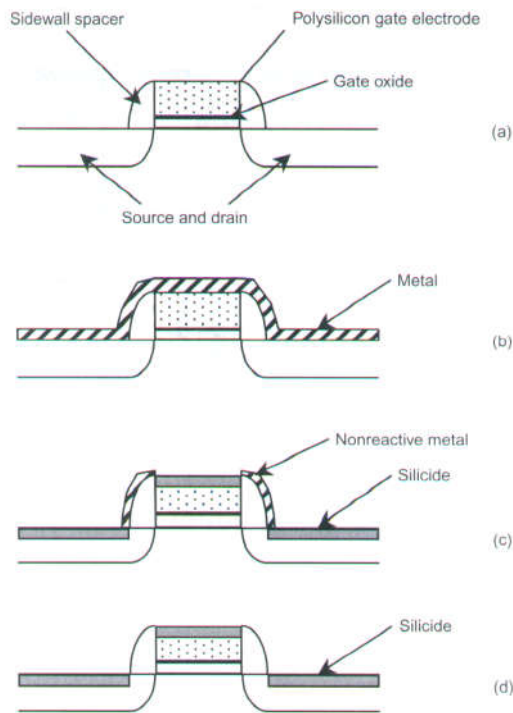
Przekrój A-A po kroku 47.

48. Naniesienie warstwy pasywacji
50. Naniesienie fotomaski
51. Odwzorowanie obszarów kontaktów (PAD OPENINGS) (MASKA #8)
52. Wywołanie fotomaski
53. Wytrawienie pasywacji
54. Usunięcie fotomaski
55. Złożenie, pakowanie i test

1.1.5 Modyfikacje procesu dla technologii niskoskalowanych [11]

1.1.5.1 Technika selektywnego wykorzystania krzemków (ang. *salicide – self aligned silicide*)

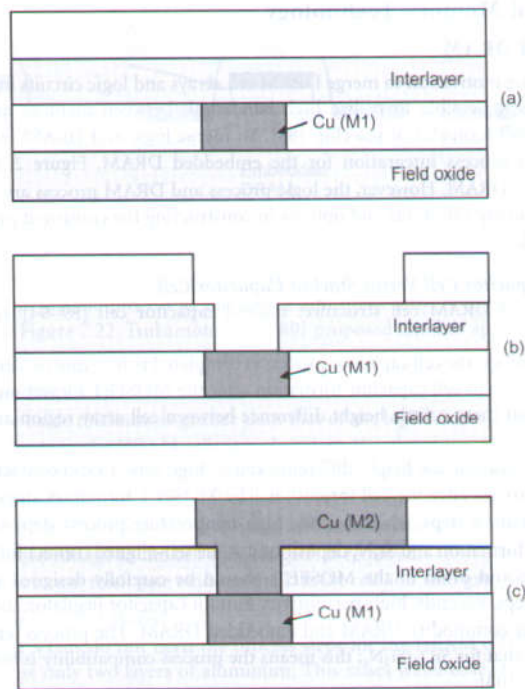
Wraz ze zmniejszaniem się wymiarów elementów tworzonych w strukturach układów CMOS zmniejszają się czasy propagacji elementów logicznych. Niestety, jak wynika z symulacji [11] wraz ze zmniejszaniem się wymiarów tranzystorów nie następuje proporcjonalny spadek czasów propagacji elementów logicznych a jest to spowodowane większymi rezystancjami cieńszych bramek polisilikonowych, cieńszych obszarów dyfuzji oraz powiększającymi się rezystancjami kontaktów. Aby uniknąć tego zjawiska stosuje się technikę selektywnego pokrywania krzemowych warstw przewodzących krzemkami, które mają wyższą przewodność od krzemu. Przedstawione to jest na rys. 0.4. Najpierw nieselektywnie nakładana jest warstwa metalu takiego jak np. Ti lub Co. Następnie warstwa ta jest wygrzewana metodą RTA (ang. Rapid Thermall Annealing) w środowisku neutralnym. W czasie wygrzewania cienka warstwa metalu, w miejscach gdzie jest styk z krzemem tworzy krzemki. Pozostałe miejsca pozostają metalem. Następnie miejsca metaliczne są wytrawiane pozostawiając krzemki tylko na powierzchniach krzemowych.



Rys. 1.4 Typowy przebieg procesu selektywnego nanoszenia krzemków stosowany w technologiach niskoskalowanych, (a) uformowanie tranzystora MOS, (b) naniesienie warstwy metalu, (c) utworzenie krzemku poprzez wygrzewanie RTA, (d) usunięcie metalu [11].

1.1.5.2 Metalizacja metodą damasceńską

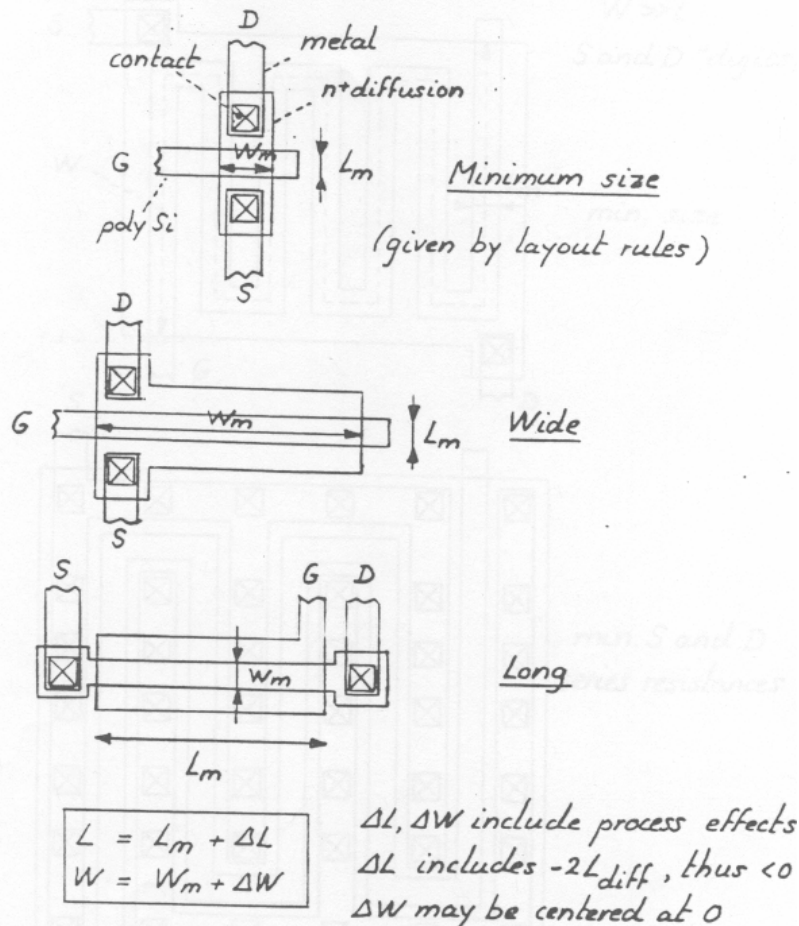
Jako metal połączeniowy szeroko stosowany jest glin (Al). Niestety w procesach niskoskalowanych, z powodu zjawisk elektromigracji (EM) oraz migracji naprężeń (SM) mogą następować przerwy ścieżek. Aby temu zapobiec stosuje się np. stop Al-Cu. Dodatkowo aby wykonać bardzo płaskie złącze często potrzebny jest materiał barierowy pomiędzy metalem i krzemem - taki jak np. TiN. W celu dalszego zmniejszenia rezystancji połączeń i szerokości ścieżek stosuje się jako metal miedź (Cu). W przypadku użycia miedzi metalizacja jest wykonywana metodą damasceńską. Spowodowane jest to trudnościami w wykonywaniu cienkich wzorów w Cu metodą wytrawiania RIE (ang. Reactive Ion Etching). Na rys. 1.4 przedstawiony jest proces podwójnej metalizacji metodą damasceńską. Po naniesieniu warstwy pośredniej wykonywane są otwory dla przelotek (via) oraz dla metalu na warstwie M2. Następnie poprzez wykorzystanie metody elektrolitycznej nanoszona jest warstwa miedzi (Cu). W kolejnym etapie wykonywana jest planaryzacja powierzchni metodą CMP (ang. Chemical Mechanical Polishing). Należy dodać że dla tego procesu bardzo ważne jest zastosowanie materiału barierowego takiego jak np. TiN (azotek tytanu).

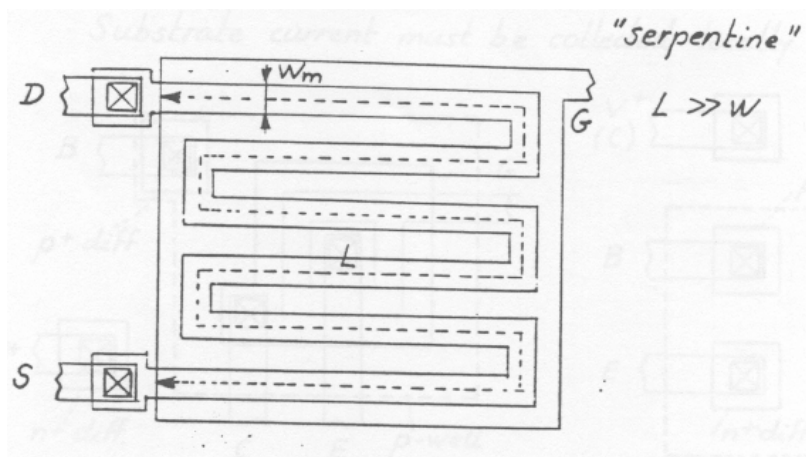
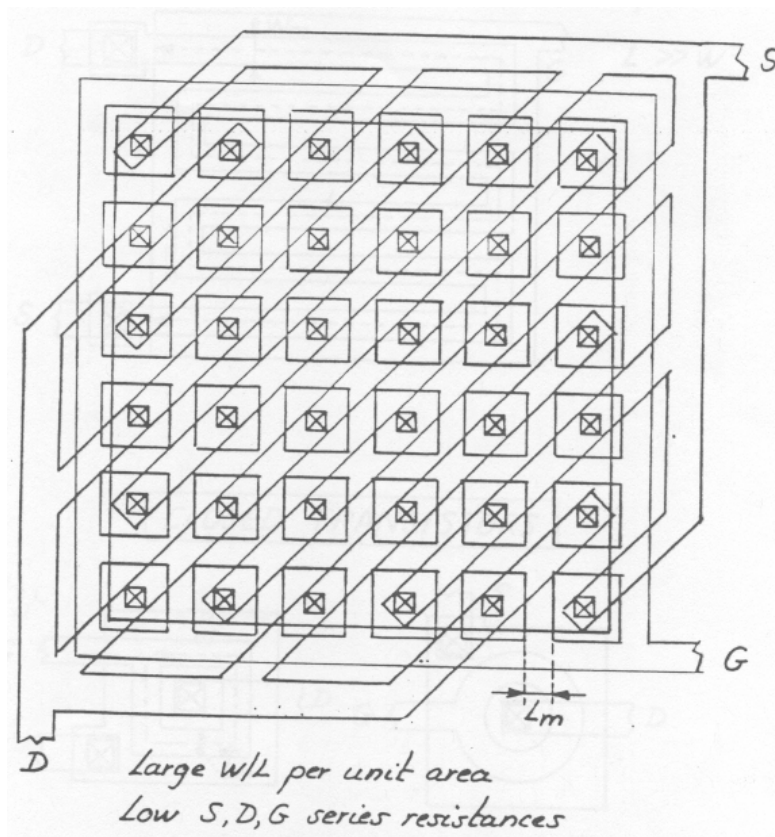
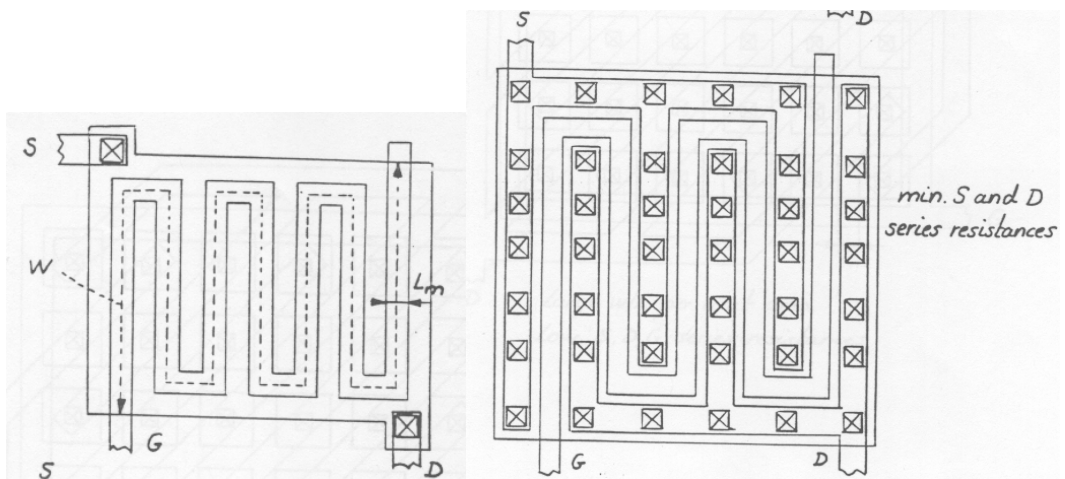


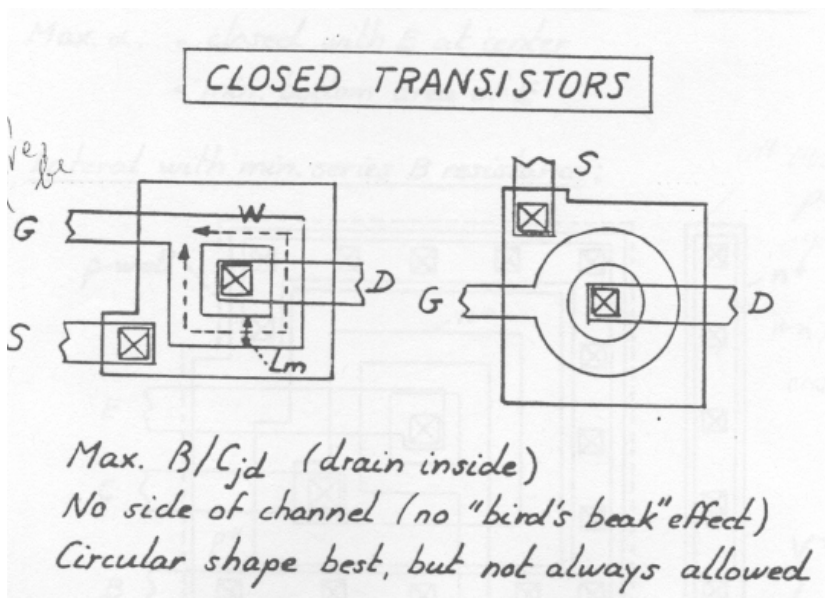
Rys. 1.4 Typowy przebieg procesu metalizacji metodą damasceńską, (a) nałożenie warstw pośrednich, (b) uformowanie obszarów przełotek (VIA), (c) nałożenie warstwy M2 (Cu) i planaryzacja [11].

1.2. Topografia niektórych elementów w technologiach scalonych (szybkie przypomnienie za pomocą rysunków).

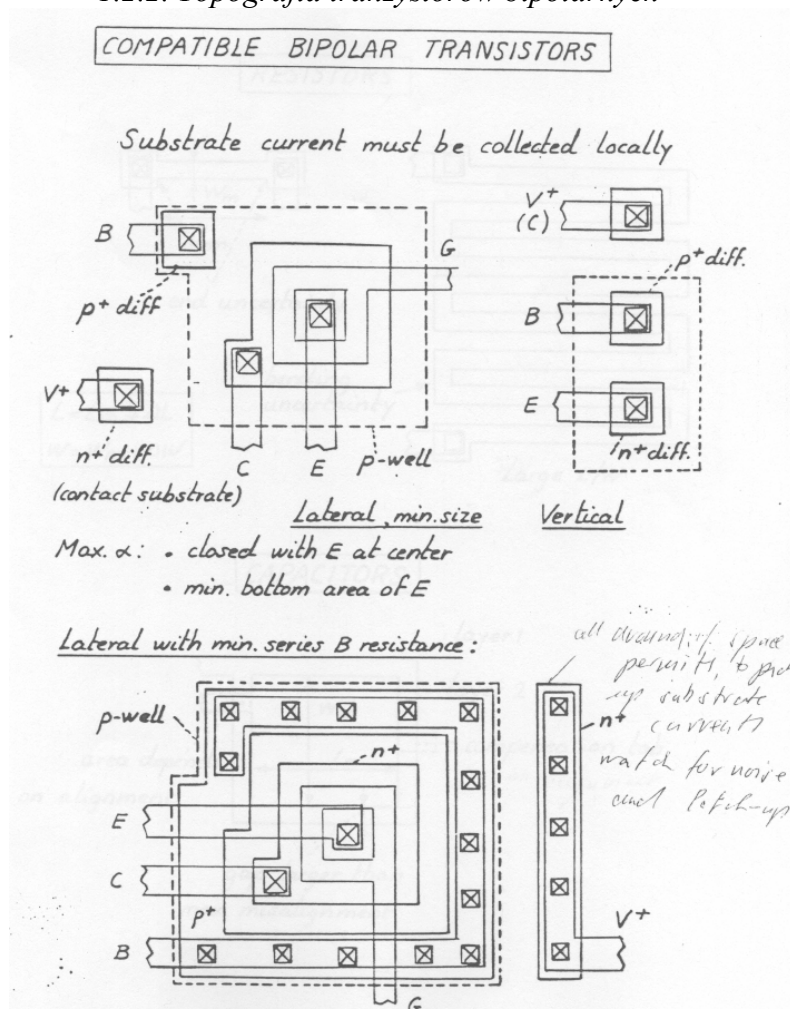
1.2.1. Topografia tranzystorów MOS:



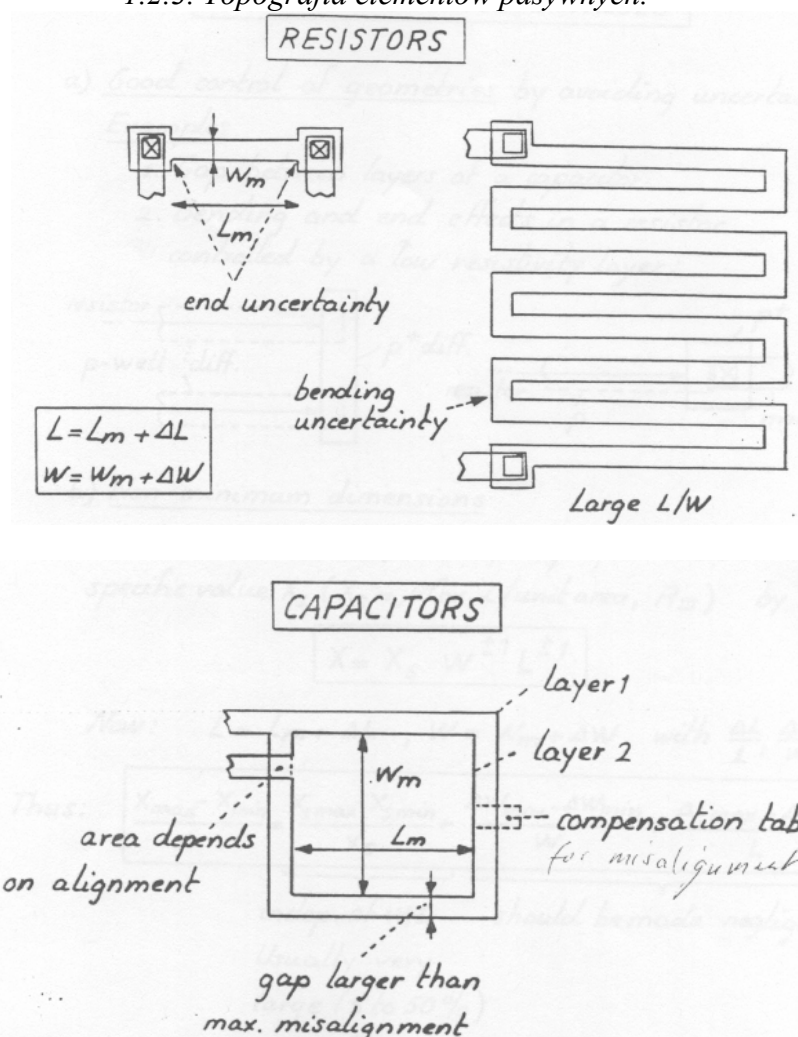




1.2.2. Topografia tranzystorów bipolarnych



1.2.3. Topografia elementów pasywnych.



1.3. Elementy pasywne – modele [1].

Pasywne elementy dyskretne są łatwe do modelowania. Rezystory i kondensatory mogą być generalnie modelowane poprzez idealną wartość rezystancji lub pojemności. Głównymi ograniczeniami w stosowaniu tych elementów są zazwyczaj tolerancje wykonania oraz zależności temperaturowe. Oba te składniki można zredukować do akceptowalnego poziomu poprzez rozsądny wybór elementów i ich parametrów.

Pasywne elementy monolityczne są dalekie od ideału. Mają zazwyczaj dość znaczące zależności temperaturowe i napięciowe. Zakres możliwych do zrealizowania wartości rezystancji i pojemności jest istotnie ograniczony przez możliwe do stosowania maksymalne wartości wymiarów elementów. Duże wartości rezystancji i pojemności są niepraktyczne. Udział efektów pasożytniczych jest również często bardzo istotny. Bezwzględna dokładność wykonania elementów jest bardzo słaba chociaż dokładność względna jest często bardzo dobra. Poniżej przedstawione są najważniejsze parametry rezystorów i kondensatorów scalonych.

Rezystory:

- rezystancja powierzchniowa warstwy (rezystancja na kwadrat),
- temperaturowy współczynnik rezystancji,
- napięciowy współczynnik rezystancji,
- dokładność bezwzględna,
- dokładność względna (dopasowanie).

Kondensatory:

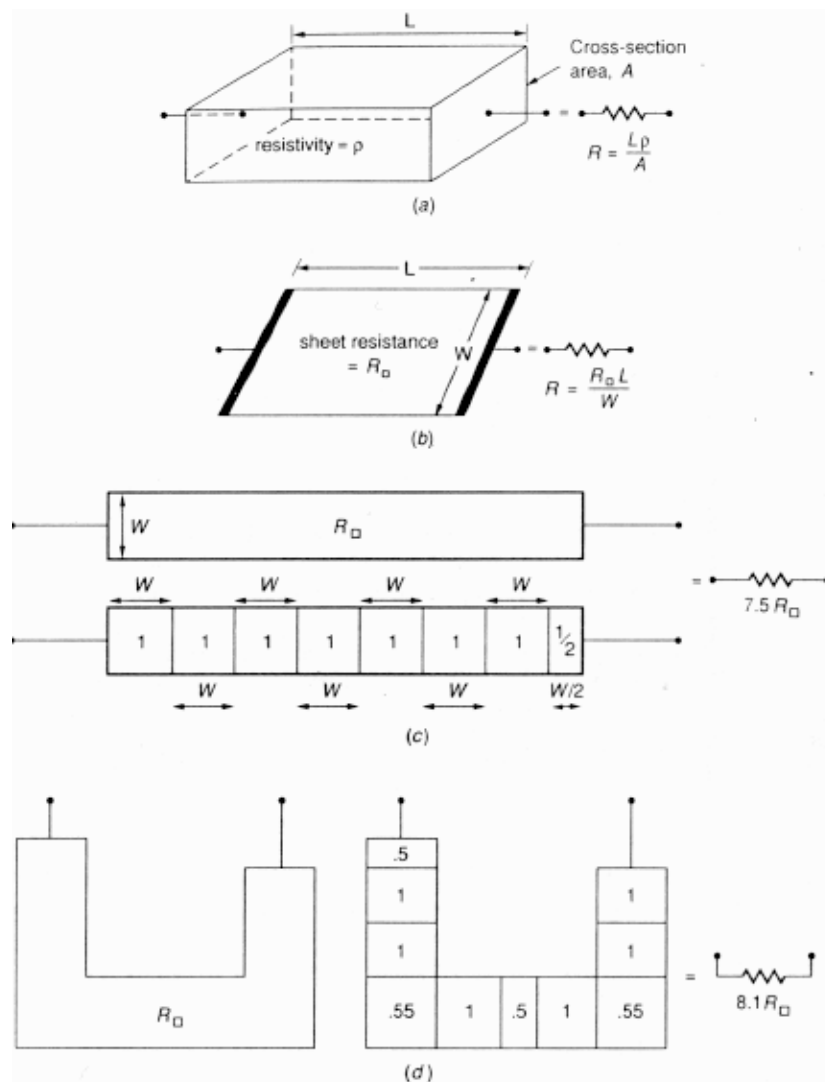
- gęstość powierzchniowa pojemności,
- współczynnik temperaturowy pojemności,

- współczynnik napięciowy pojemności,
- dokładność bezwzględna,
- dokładność względna (dopasowanie).

Rezystancja powierzchniowa jest to rezystancja kwadratu materiału widziana z jego przeciwległych boków. Jeśli rezystywność materiału wynosi ρ a grubość warstwy materiału z wówczas rezystancję na kwadrat można wyznaczyć na podstawie wzoru:

$$R_{KW} = \frac{\rho}{z}$$

Wartość rezystancji rezystora prostego złożonego z N kwadratów jest równa NR_{KW} . W przypadku stosowania zagięć rezystancja kwadratu narożnego wynosi $0.55R_{KW}$. Przykład obliczenia rezystancji przedstawiony jest na poniższym rysunku.



Rys. 1.4. Własności rezystywne materiałów a) rezystywność, b) rezystancja powierzchniowa (inaczej na kwadrat), c) i d) przykłady obliczenia rezystancji przy zastosowaniu rezystancji powierzchniowej.

Współczynnik zmian temperaturowych parametru x określony jest wzorem:

$$TC = \left(\frac{1}{x} \right) \left(\frac{dx}{dT} \right) \text{ w jednostkach } [1/^{\circ}\text{C}] \text{ lub wzorem}$$

$$TC = \left(\frac{1}{x} \right) \left(\frac{dx}{dT} \right) 10^6 \text{ w jednostkach } [\text{ppm}/^{\circ}\text{C}]$$

gdzie parametr x może być rozważaną rezystancją, pojemnością czy też dowolną inną wielkością fizyczną.

Jeśli wartość współczynnika temperaturowego jest niezależna od temperatury wówczas wartość parametru x w temperaturze T_2 można wyznaczyć na podstawie znajomości wartości w temperaturze T_1 poprzez wyrażenie:

$$x(T_2) = x(T_1)e^{[TC(T_2-T_1)]}$$

Równanie powyższe jest często aproksymowane zależnością:

$$x(T_2) = x(T_1)[1 + (T_2 - T_1)TC]$$

W przypadku stosowania współczynnika TC w jednostkach ppm/°C wartości TC użyte w powyższych wzorach powinny być podzielone przez współczynnik 10^5 . Spotyka się zarówno wartości dodatnie jak i ujemne współczynnika TC. Najczęściej wartość bezwzględna TC jest mniejsza niż 1000ppm/°C. Ponieważ oczekuje się, że układy scalone powinny pracować w szerokim zakresie temperatur (0-70°C dla zastosowań komercyjnych i od -55°C do +125°C dla zastosowań wojskowych) wpływ współczynnika TC może być bardzo znaczący.

Podobnie jak zmiany parametrów pod wpływem temperatury definiuje się zmianę parametrów pod wpływem przyłożonego napięcia:

$$VC = \left(\frac{1}{x}\right)\left(\frac{dx}{dV}\right)$$

Dokładność bezwzględna jest miarą jak dokładnie wykonywane są rzeczywiste wartości rezystancji i pojemności w stosunku do założonych. Dokładność względna jest miarą na ile powtarzają się wartości elementów w obrębie jednego układu scalonego.

Ze względu na trudność jednolitego scharakteryzowania, efekty pasożytnicze nie zostały tutaj opisane. Jako przykłady można jednak podać: rezystancję kontaktów, pojemności nakładkowe, pojemności rozłożone rezystorów, pojemności brzegowe i wiele innych.

Kondensatory monolityczne

Każda struktura zawierająca odseparowane od siebie warstwy przewodzące może być uznana za kondensator. Niektóre z możliwych kombinacji przedstawione są w poniższej tabeli. Najczęściej wykonywanym kondensatorem jest wykonany z warstw POLYI/POLYII. Taki rodzaj kondensatora jest prawie idealny, ma małą zależność temperaturową i napięciową. Głównym elementem pasożytniczym takiego kondensatora jest duża pojemność pomiędzy dolną okładką i podłożem. Pasożyty ograniczają możliwości zastosowań lecz rozsądne techniki projektowania mogą eliminować niekorzystne zjawiska pasożytnicze.

W przypadku gdy nie są dostępne kondensatory POLYI/POLYII można wykorzystać inne warstwy np.: metal/POLY, metal/dyfuzja lub POLY/dyfuzja wszystkie z dielektrykiem SiO₂. Te kondensatory mają zazwyczaj niższą gęstość powierzchniową pojemności oraz mogą mieć zwiększoną zależność napięciową i temperaturową oraz większą rezystancję okładek.

W każdym z procesów technologicznych można także wykorzystywać pojemności elementów aktywnych (złączowe, bramkowe) lecz te zazwyczaj są zależne od napięcia i temperatury.

Tabela 1.2. Właściwości kondensatorów monolitycznych.

Capacitor type	Process	Dielectric	Absolute accuracy	Ratio accuracy	Voltage characteristic	Temperature	Comments
Poly-Poly	MOS	SiO ₂	±20%	±0.06%	-5 ppm/V	25 ppm/°C	Most popular MOS, best characteristic
Poly-Diffusion	MOS	SiO ₂	±10%	±0.06%	-20 ppm/V	25 ppm/°C	Lower plate potential often fixed
Metal-Diffusion (with thin oxide)	MOS/Bipolar	SiO ₂	±10%	±0.06%	-20 ppm/V	25 ppm/°C	Most desirable bipolar
Moat-Substrate	MOS	Si					Voltage dependent Must be reverse biased, low density
Base-Collector	Bipolar	Si					
Base-Emitter	Bipolar	Si					Must be reverse biased
Conductor-Conductor	Thin film	Varies					Good characteristics
Screened	Thick film	Varies					Chip capacitors often preferred

Rezystory monolityczne

Rezystory monolityczne wykonywane są w dwóch technikach: jako pasywne warstwy o określonej długości/szerokości i kształcie oraz jako układy aktywne złożone z tranzystorów i innych elementów.

Podsumowanie przykładowych typów rezystorów przedstawione jest w poniższej tabeli.

Tabela 1.3. Właściwości rezystorów monolitycznych.

Device	Characterizing equation	Ideal resistance†	Temperature characteristic	Absolute accuracy	Relative accuracy	Comments
Poly Strip	$V = I R_C \frac{L}{W}$	$R = R_C \frac{L}{W}$	1500 ppm/°C	± 30%	± 2%	Linear, low resistance density
Diffusion	$V = I R_C \frac{L}{W}$	$R = R_C \frac{L}{W}$	1500 ppm/°C	± 35%	± 2%	Somewhat voltage dependent
MOSFET (Ohmic region, Fig. 3.4-2a)	$I = \frac{K' W}{L} [(V_{GS} - V_T) - \frac{V}{2}] V$	$R = \frac{L}{K' W (V_{GS} - V_T)}$	$(2 + d_1)$			To minimize distortion, $V < (V_{GS} - V_T)/2$
MOSFET (Saturated region, Fig. 3.4-2b)	$I = \frac{K' W}{2L} (V_{GS} - V_T)^2 (1 + \lambda V)$	$R_{SS} = \frac{2L}{\lambda K' W (V_{GS} - V_T)^2}$	$2d_2 + d_3)$			Small signal impedance only, high impedance values
MOS depletion (Fig. 3.4-2c)	$I = \begin{cases} \frac{K' W}{L} (-V_T) V & V < -V_T \\ \frac{K' W}{2L} V_T^2 (1 + \lambda V) & V > V_T \end{cases}$	$R = \frac{L}{K' W V_T }$ $R_{SS} = \frac{2L}{K' W V_T^2 \lambda}$	$1)$			Popular load device Quite nonlinear

1.4. Reguły technologiczne.

Projekt układu scalonego przekazywany producentowi zawiera kształty masek wygenerowane przez program do edycji topografii. W celu zapewnienia powodzenia i uzyskania sprawności układu scalonego, projektant musi zastosować się do wielu reguł projektowych. Część z nich można sprawdzić automatycznie przez program komputerowy. Reguły projektowe, ze względu na charakter można podzielić na dwie następujące grupy:

- reguły elektryczne,
- reguły geometryczne.

Reguły elektryczne stanowią zestaw ograniczeń na wartości napięć i prądów dopuszczonych do występowania w układzie. W ramach reguł elektrycznych można wymienić następujące: dopuszczalny zakres napięć zasilających, maksymalne gęstości prądu poszczególnych warstw przewodzących, dopuszczalna tracona moc w układzie. W większości przypadków ciężar dostosowania projektu pod względem reguł elektrycznych spada na projektanta.

Reguły geometryczne stanowią zestaw ograniczeń dotyczących możliwych do stosowania wymiarów i względnego położenia poszczególnych elementów układu, przykładami mogą być:

- minimalne i maksymalne wymiary poszczególnych warstw,
- minimalne odstępów między tymi samymi i różnymi warstwami,
- konieczne wartości nakładek warstw,
- maksymalna dopuszczalna odległość pomiędzy kontaktami do wysp i podłoża.

Większość reguł geometrycznych jest sprawdzana przez programy do edycji topografii co w znaczący sposób odciąża projektanta i istotnie zmniejsza ryzyko błędu projektowego.

1.5. Przykład fragmentu pliku parametrów technologicznych procesu.

MOSIS PARAMETRIC TEST RESULTS

RUN: N71V
TECHNOLOGY: SCNA20
INTRODUCTION: This report contains the lot average results obtained by MOSIS from measurements of MOSIS test structures on each wafer of this fabrication lot. SPICE parameters obtained from similar measurements on a selected wafer are also attached.
COMMENTS: Orbit Semiconductor 2.0 um n-well.

TRANSISTOR PARAMETERS W/L N-CHANNEL P-CHANNEL UNITS

MINIMUM 3/2

Vth		0.87	-0.99	Volts
SHORT	18/2			
Vth		0.79	-0.96	Volts
Vpt		9.0	-9.0	Volts
Vbkd		14.3	-15.9	Volts
Idss		132	-60	uA/um
WIDE	120/2			
Ids0		0.3	-0.2	pA/um
LARGE	50/50			
Vth		0.82	-0.95	Volts
Vjbkd		15.5	-16.1	Volts
Ijlk		-33.1	-1.3	pA
Gamma		0.21	0.66	V^0.5
Delta length		0.15	0.19	microns
(L_eff = L_drawn-DL)				
Delta width		0.00	0.00	microns
(W_eff = W_drawn-DW)				
K' (Uo*Cox/2)		12.5	-8.1	uA/V^2
POLY2 TRANSISTORS	W/L	N-CHANNEL	P-CHANNEL	UNITS
MINIMUM	6/4			
Vth		0.84	-1.34	Volts
SHORT	12/4			
Vth		0.81	-1.31	Volts
LARGE	36/36			
Vth		0.80	-1.32	Volts
Delta length		-0.05	-0.78	microns
(L_eff = L_drawn-DL)				
Delta width		0.36	0.49	microns
(W_eff = W_drawn-DW)				
K' (Uo*Cox/2)		21.1	-6.1	uA/V^2

FOX TRANSISTORS	GATE	N+ACTIVE	P+ACTIVE	UNITS
Vth	Poly	22.4	-13.6	Volts
BIPOLAR PARAMETERS	W/L	NPN		UNITS
2X1	2X1			
Beta		145		
V_early		62.0		Volts
Vce,sat		0.4		Volts
2X2	2X2			
Beta		145		
V_early		60.1		Volts
Vce,sat		0.2		Volts
2X4	2X4			
Beta		143		
V_early		57.3		Volts
Vce,sat		0.3		Volts
2X8	2X8			
Beta		142		
V_early		55.7		Volts
Vce,sat		0.6		Volts
BVceo		16.9		Volts
BVcbo		18.5		Volts
BVebo		7.5		Volts

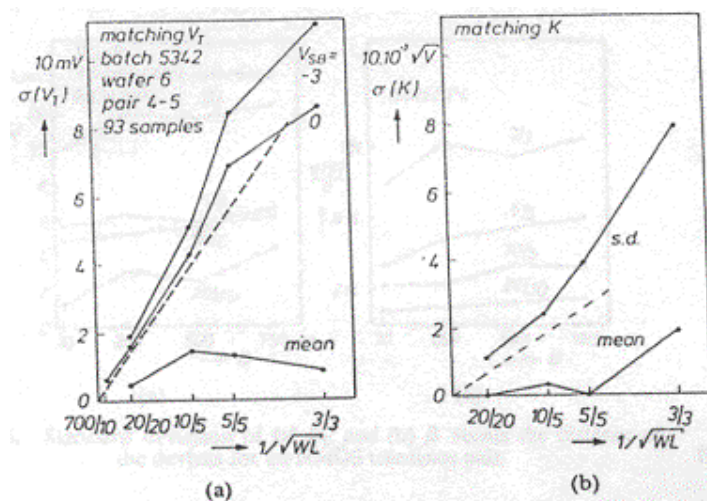
PROCESS PARAMETERS	N+DIFF	P+DIFF	POLY	POLY2	METAL1	METAL2	N WELL	UNITS
Sheet Resistance	12.3	55.6	24.3	15.1	0.05	0.03	2474	ohms/sq
Width Variation	0.41	0.08	-0.07	-0.01	-0.40	0.11		microns
(measured - drawn)								
Contact Resistance	13.5	130.1	8.5	8.3		0.04		ohms
Gate Oxide Thickness	410							angstroms

CAPACITANCE PARAMETERS	N+DIFF	P+DIFF	POLY	POLY2	METAL1	METAL2	UNITS
Area (substrate)	125	307	57		27	21	aF/um^2
Area (poly)				504	43	23	aF/um^2
Area (poly2)					43		aF/um^2
Area (metall1)						38	aF/um^2
Area (N+active)			841	669	47	27	aF/um^2
Area (P+active)			836	664			aF/um^2
Fringe (substrate)	548	400					aF/um
Fringe (N+active)			128				aF/um
Fringe (P+active)			145				aF/um

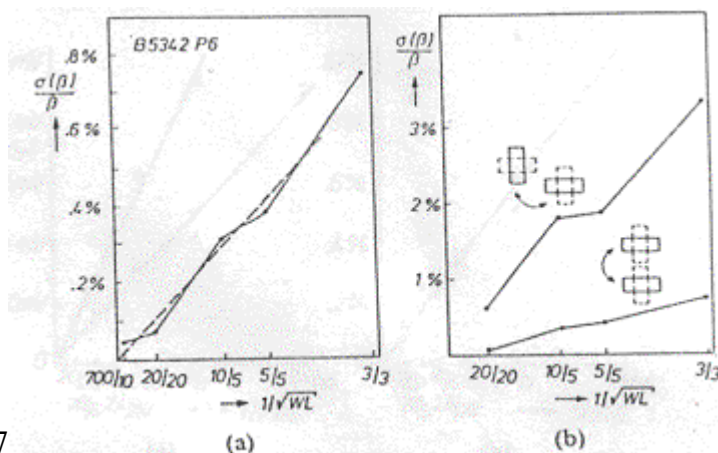
CIRCUIT PARAMETERS			UNITS
Inverters	K		
V _{inv}	1.0	2.08	Volts
V _{inv}	1.5	2.28	Volts
V _{ol} (100 uA)	2.0	0.27	Volts
V _{oh} (100 uA)	2.0	4.65	Volts
V _{inv}	2.0	2.43	Volts
Gain	2.0	-10.1.80	
Ring Oscillator			
MOSIS (31-stage, 5V)		28.46	MHz

2. Rozrzuty względne i bezwzględne i ich wpływ na pracę układów ASIC.

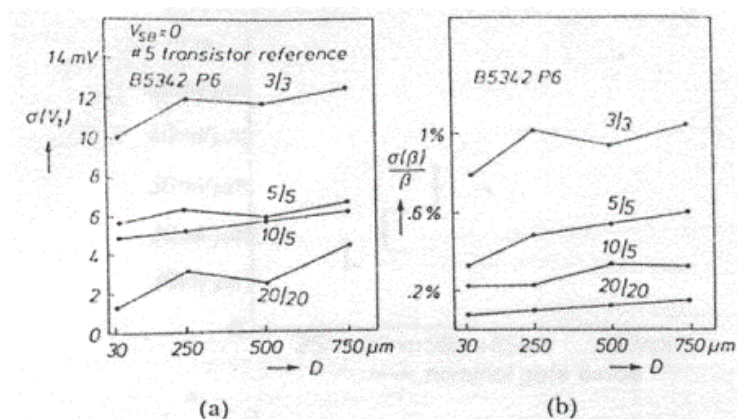
Dopasowanie tranzystorów MOS – wyniki pomiarów wg [5].



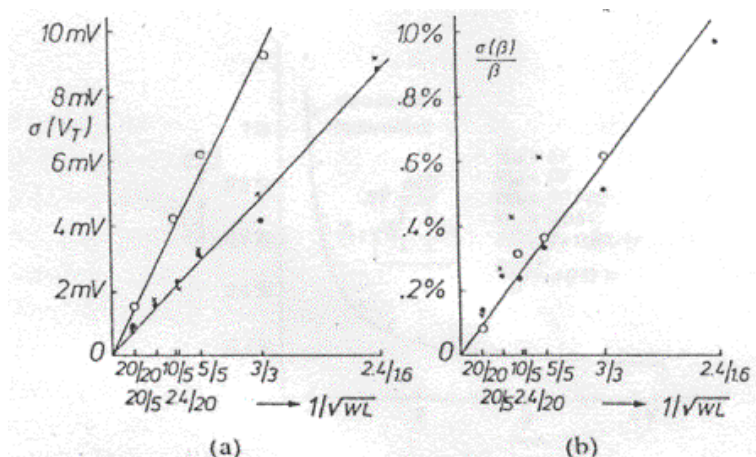
Rys. 2.1. Standardowa dewiacja napięcia progowego V_T (a) oraz współczynnika podłożowego K (b) [5].



Rys. 2.2. Standardowa dewiacja współczynnika transkonduktancyjnego β (a) w funkcji odwrotności powierzchni tranzystora oraz porównanie standardowej dewiacji współczynnika transkonduktancyjnego β dla elementów umieszczonych równolegle i prostopadle (b).



Rys. 2.3. Standardowa dewiacja napięcia progowego V_T (a) oraz współczynnika transkonduktancyjnego β w funkcji odległości elementów.



Rys. 2.4. Standardowa dewiacja (a) napięcia progowego oraz (b) współczynnika transkonduktancyjnego w funkcji odwrotności powierzchni pary tranzystorów MOS dla 25nm procesu (kropki) oraz 50nm procesu (okręgi).

Dopasowanie wynikowych parametrów elementów zrealizowanych geometrycznie jako prostokąty można zamodelować równaniem [5]:

$$\sigma^2(\Delta P) = \frac{A_P^2}{WL} + S_P^2 D_X^2 \quad (2.1)$$

gdzie: ΔP - rozpatrywany parametr, W, L - wymiary geometryczne rozpatrywanych elementów, σ - odchylenie standardowe, A_P - współczynnik proporcjonalności powierzchniowej, S_P - współczynnik proporcjonalności odległościowej.

Wartości parametrów A_P oraz S_P zmierzone dla technologii 2.5 μm przedstawione są w tabeli 2.1.

parameter	n-channel s.d.	p-channel s.d.	unit
A_{VT0}	30	35	mV μm
A_β	2.3	3.2	% μm
A_K	16×10^{-3}	12×10^{-3}	V ^{0.5} μm
S_{VT0}	4	4	$\mu\text{V}/\mu\text{m}$
S_β	2	2	$10^{-6}/\mu\text{m}$
S_K	4	4	$10^{-6}\text{V}^{0.5}/\mu\text{m}$

Tabela 2.1. Zmierzone wartości współczynników dopasowania par MOS w technologii 2.5 μm [5].

Dla czwórki elementów w topografii typu common-centroid wyrażenie powyższe można przybliżyć zależnością [5]:

$$\sigma^2(\Delta P) = \frac{A_p^2}{WL} + S_p^2 D_X^2 \frac{D_Y^2}{\text{średnica wafr}^2} \quad (2.2)$$

Dopasowanie pary tranzystorów MOS zasilanych wspólnym napięciem może zostać oszacowane na podstawie wyrażenia:

$$\frac{\sigma^2(I_D)}{I_D^2} = \frac{4\sigma^2(V_{T0})}{(V_{GS} - V_{T0})^2} + \frac{\sigma^2(\beta)}{\beta^2} \quad (2.3)$$

Model tranzystora MOS opisany jest poniższymi równaniami:

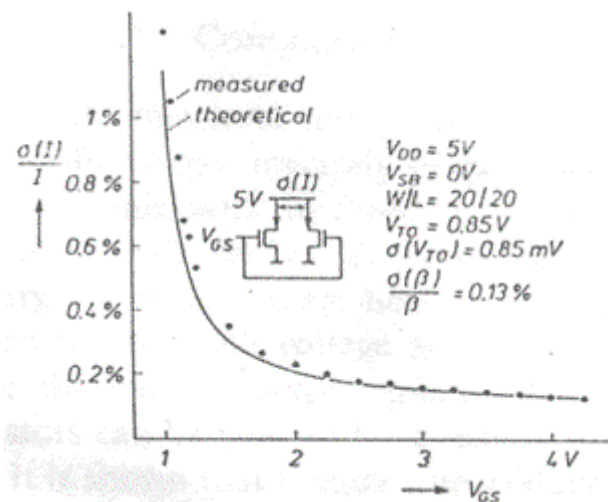
$$I_D = \beta \left[\frac{(V_{GS} - V_T - V_{DS}/2)}{1 + \theta(V_{GS} - V_T)} \right] \quad (2.4)$$

Prąd w zakresie omowym,

$$I_D = \frac{\beta}{2} (V_{GS} - V_T)^2 \quad (2.5)$$

Prąd drenu w zakresie nasycenia

gdzie: $\beta = C_{ox} \mu W / L$, $V_T = V_{T0} + K(\sqrt{|V_{SB}|} + 2\phi_F - \sqrt{2\phi_F})$



Rys. 2.5 Standardowa dewiacja prądu drenów pary MOS. Kropki oznaczają wartości zmierzone, linia ciągła wyznaczona została wzorem (3.3).

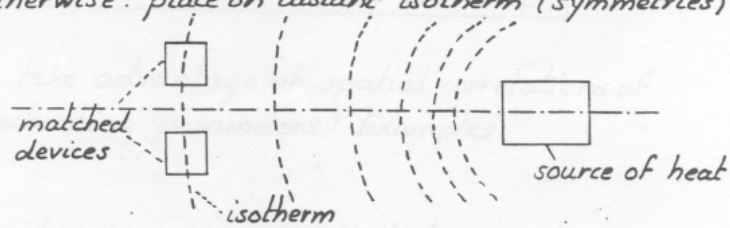
Zasady poprawy dopasowania elementów:

- stosowanie możliwie największych elementów,
- stosowanie identycznych kształtów i wymiarów elementów,
- taka sama temperatura elementów,
- umieszczanie elementów blisko siebie,
- rozdział elementów na mniejsze połączone szeregowo/równoległe i przemieszane między sobą,
- ustawianie elementów w tej samej orientacji geometrycznej,
- stosowanie możliwie największych napięć V_{GS} ,
- stosowanie topografii typu common-centroid,
- stosowanie struktur pustych (ang. dummy).

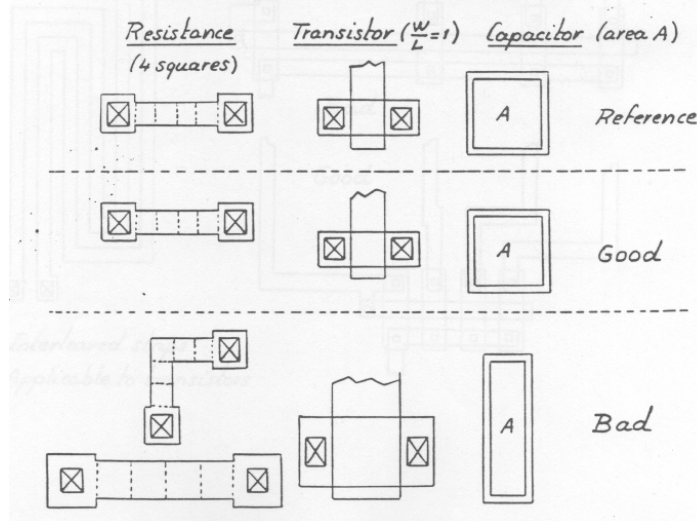
Poniższe rysunki przedstawiają graficznie metody poprawy dopasowania elementów.

MATCHING : SAME TEMPERATURE

No problem if power dissipation on chip is negligible.
Otherwise : place on distant isotherm (symmetries) :



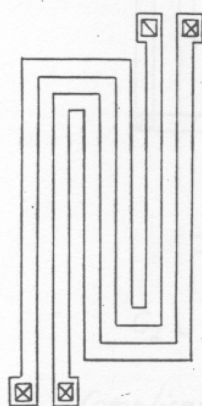
MATCHING : SAME SHAPE AND SIZE



MATCHING : MINIMUM DISTANCE

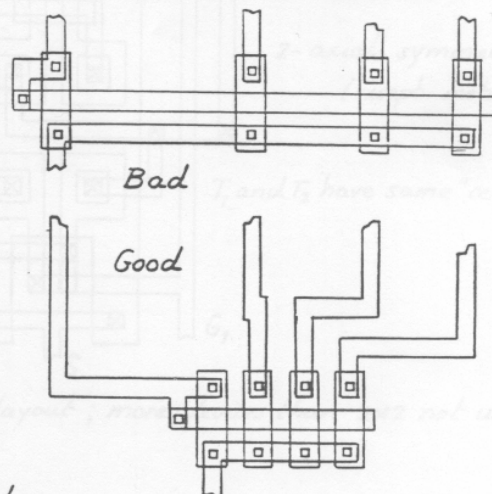
To take advantage of spatial correlations of
Fluctuating parameters : Examples:

Large resistors

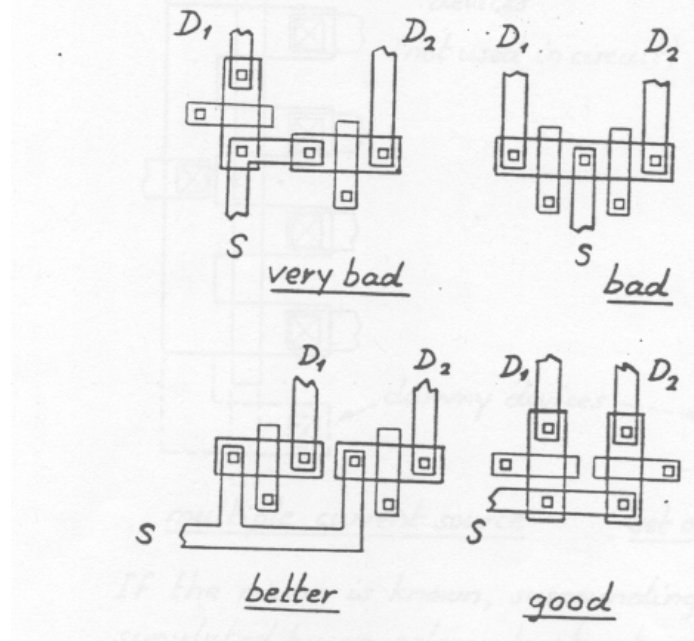


Interleaved strips
Applicable to transistors

Multiple current mirror



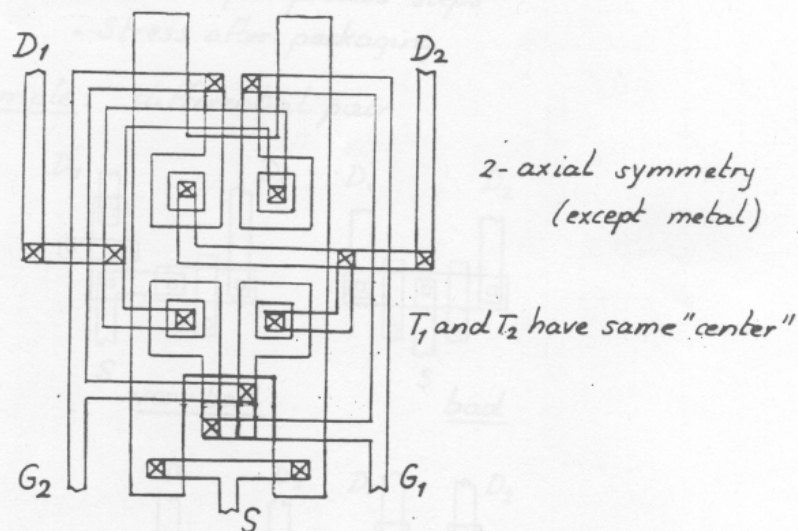
Example: differential pair



MATCHING: COMMON-CENTROID GEOMETRIES

Compensation of constant gradients

Example: quad of transistors

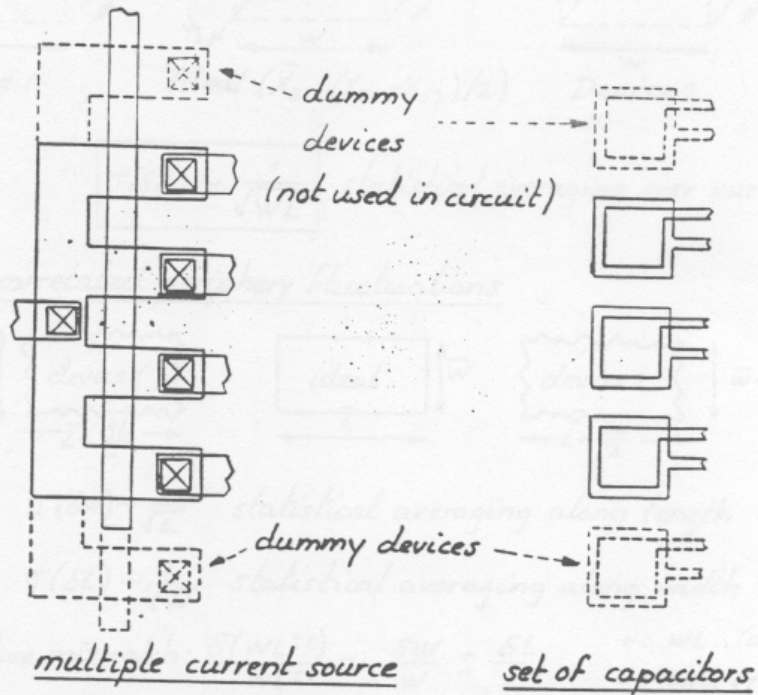


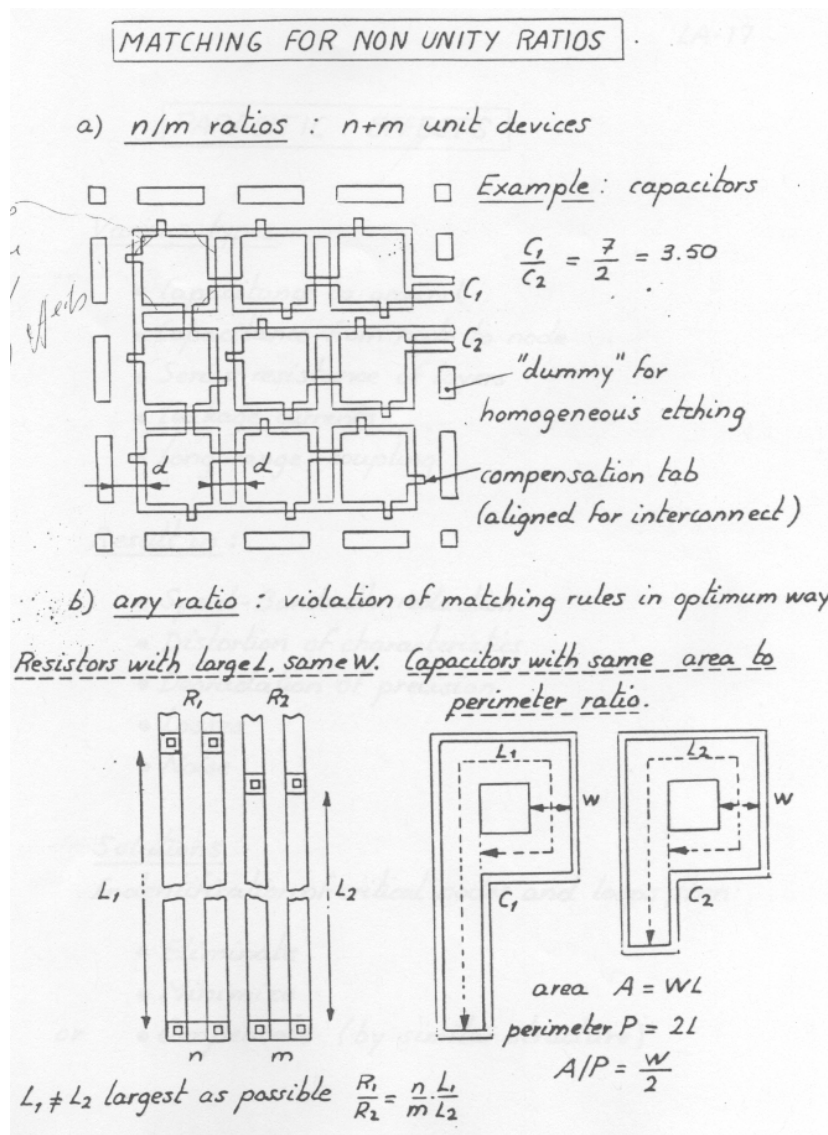
Complicated layout; more devices than 2x2 not useful

MATCHING: SAME SURROUNDINGS

- Various possible reasons, not always clear
- May be solved by implementing dummy devices

Examples





3. Projektowanie cyfrowych układów scalonych w technice komórek standardowych – synteza z użyciem Cadence RTL Compiler

Opis dostępny jest na stronie: http://www.ue.eti.pg.gda.pl/~bpa/pua_in/lab.pdf

4. Projektowanie cyfrowych układów scalonych w technice komórek standardowych – implementacja użyciem Cadence SOC RTL Compiler

Opis dostępny jest na stronie: http://www.ue.eti.pg.gda.pl/~bpa/pua_in/lab.pdf

5. Scalone przetworniki cyfrowo-analogowe, klasyfikacja i podstawowe właściwości [1].

Schemat blokowy przetwornika cyfrowo – analogowego przedstawiony jest na rysunku 5.1 a ogólne równanie opisujące wartość sygnału wyjściowego jest następujące:

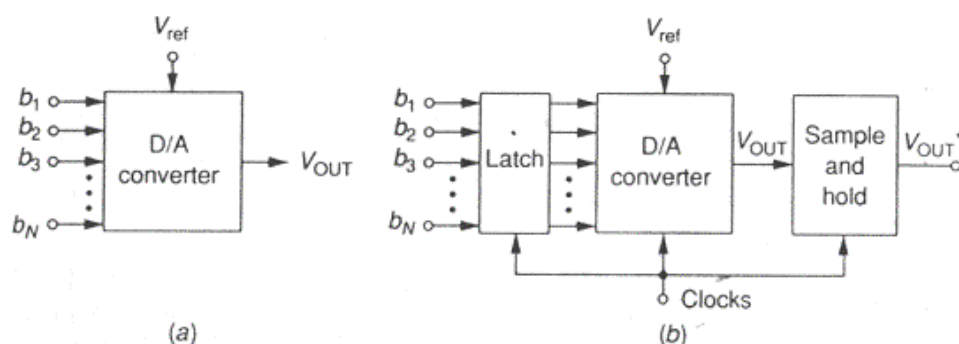
$$V_{OUT} = KV_{REF} D \quad (5.1)$$

gdzie: K – współczynnik skalujący, V_{REF} – napięcie odniesienia, D słowo cyfrowe dane równaniem:

$$D = \frac{b_1}{2^1} + \frac{b_2}{2^2} + \frac{b_3}{2^3} + \dots + \frac{b_N}{2^N} \quad (5.2)$$

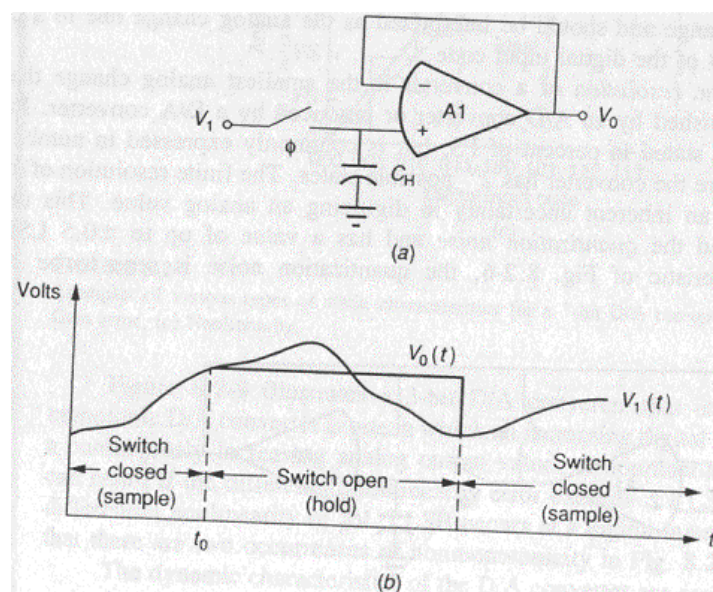
gdzie: N – liczba bitów przetwornika, $b_1 - b_N$ współczynniki słowa cyfrowego mogące przyjąć wartości 0 lub 1. Poprzez połączenie powyższych równań otrzymujemy:

$$V_{OUT} = KV_{REF} \left(\frac{b_1}{2^1} + \frac{b_2}{2^2} + \frac{b_3}{2^3} + \dots + \frac{b_N}{2^N} \right) = KV_{REF} \sum_{j=1}^N b_j 2^{-j} \quad (5.3)$$

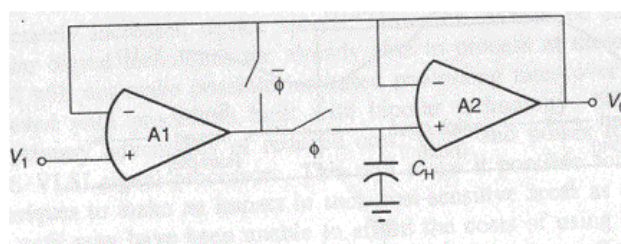


Rys. 5.1. Schemat blokowy przetwornika CA (a) oraz jego wersji zegarowanej z układem próbkująco – pamiętającym (b) [1].

Ze względu na czas potrzebny na konwersję oraz zabezpieczenie wyjścia układu przed pojawieniem się sygnałów niepożądanych stosuje się układy próbkująco-pamiętające. Przykład takiego układu przedstawiony jest na rysunku 5.2. W czasie gdy klucz jest zamknięty napięcie wejściowe jest doprowadzone do kondensatora C_H . W momencie otwarcia klucza poziom napięcia pozostaje zapamiętany. Ulepszona wersja układu próbkująco-pamiętającego przedstawiona jest na rysunku 5.3.



Rys. 5.2. Układ próbkująco-pamiętający (a) oraz przebiegi czasowe napięć węzłowych (b).



Rys. 5.3. Ulepszony układ próbkująco-pamiętający.

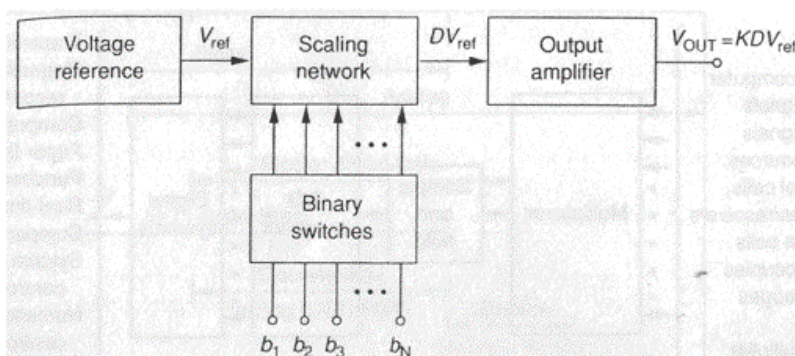
Ogólna struktura przetwornika cyfrowo - analogowego (bez układu próbkująco – pamiętającego) przedstawiona jest na rysunku 5.4. Przetwornik ten składa się z: generatora napięcia odniesienia, zestawu kluczy binarnych, sieci skalującej oraz wzmacniacza wyjściowego. Parametry opisujące przetworniki CA dzielą się na dwie grupy: statyczne i dynamiczne. Do zestawu

parametrów statycznych zalicza się: charakterystykę przejściową, szum próbkowania, zakres dynamiki wzmacnienie przesunięcie oraz nieliniowości.

Rysunek 5.5 przedstawia charakterystykę przejściową idealnego przetwornika CA. Jej kształt jest schodkowy, wartości analogowe przyjmują wartości nieparzyste z wielokrotności wyrażenia $FS/15$.

Rozdzielczość przetwornika jest to najmniejsza zmiana sygnału analogowego która może zostać rozróżniona w przetworniku AC lub wygenerowana w przetworniku CA. Wyraża się ją w liczbach bitów N co fizycznie odpowiada wartości $FS/2^N$. Skończona rozdzielczość przetworników powoduje powstanie niepewności co do rzeczywistej wartości sygnału analogowego. Ta niepewność nazywana jest szumem kwantowania i wynosi $\pm 0.5\text{LSB}$ (LSB to inaczej rozdzielczość bo zmiana najmniej znaczącego bitu to zmiana odpowiadająca rozdzielczości). Linia prosta na rysunku 5.5 odpowiada sytuacji przetwornika którego N dąży do nieskończoności.

Pelen zakres skali (FS) jest to różnica pomiędzy największą i najmniejszą wartością analogową generowaną przez przetwornik.



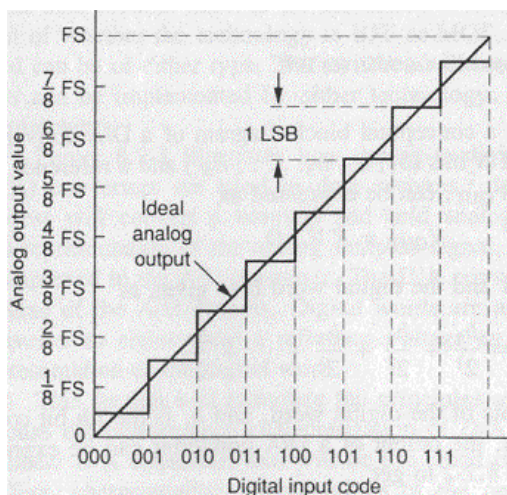
Rys. 5.4. Schemat blokowy przetwornika cyfrowo – analogowego.

Zakres dynamiki (DR) bezszumnego przetwornika jest to stosunek wartości FSR do jego rozdzielczości. Stąd można go zdefiniować jako:

$$DR = 2^N \quad (5.4)$$

lub w decybelach jako:

$$DR(db) = 20\log_{10}(2^N) = 6.02N \quad (5.5)$$

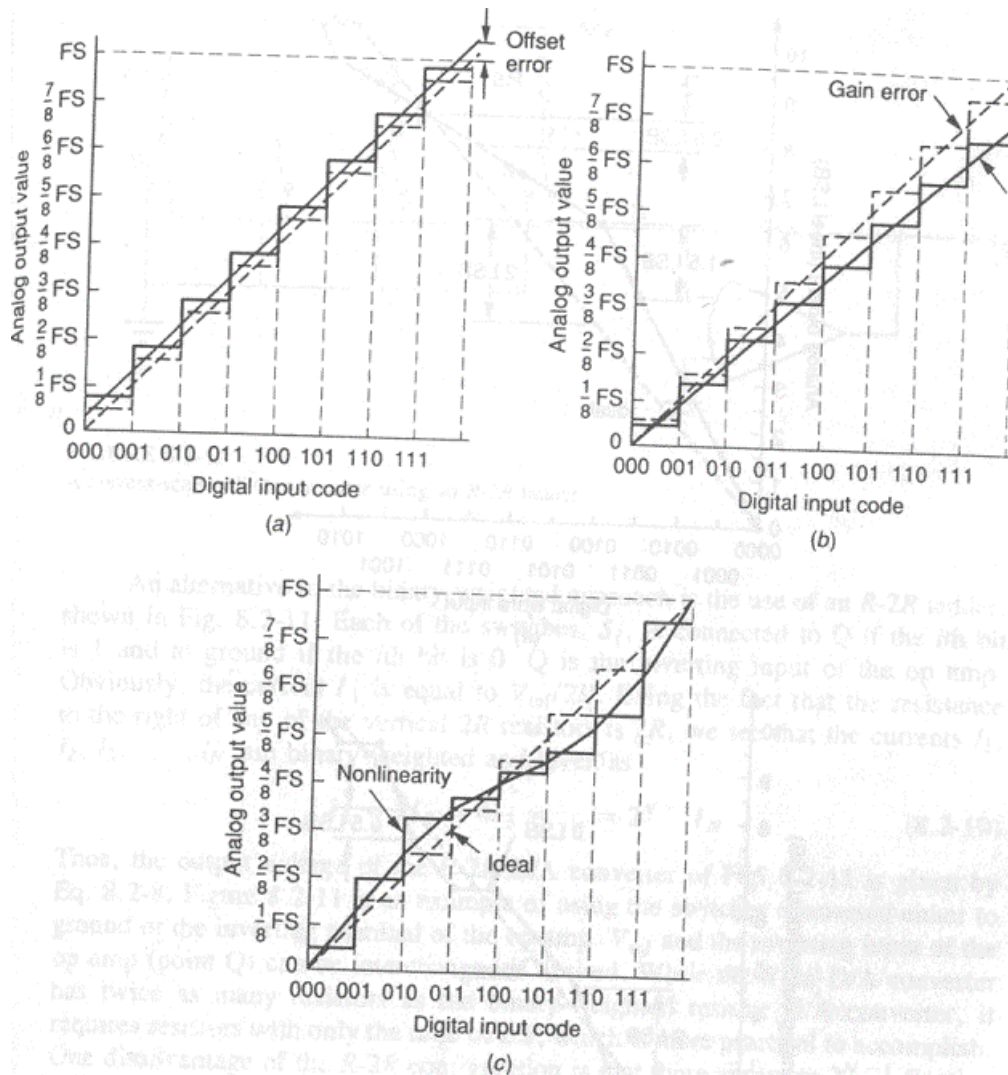


Rys. 5.5. Charakterystyki przejściowe idealnego przetwornika cyfrowo – analogowego.

Stosunek sygnał szum (SNR). Załóżmy, że przykładamy przebieg trójkątny do kaskady idealnych przetworników AC i CA. Jeśli następnie sygnał wyjściowy z przetwornika CA odejmiemy od sygnału wejściowego otrzymamy przebieg piłokształtny o amplitudzie $\pm 0.5\text{LSB}$. Ten przebieg reprezentuje szum kwantowania. Współczynnik SNR jest stosunkiem wartości skutecznej przebiegu

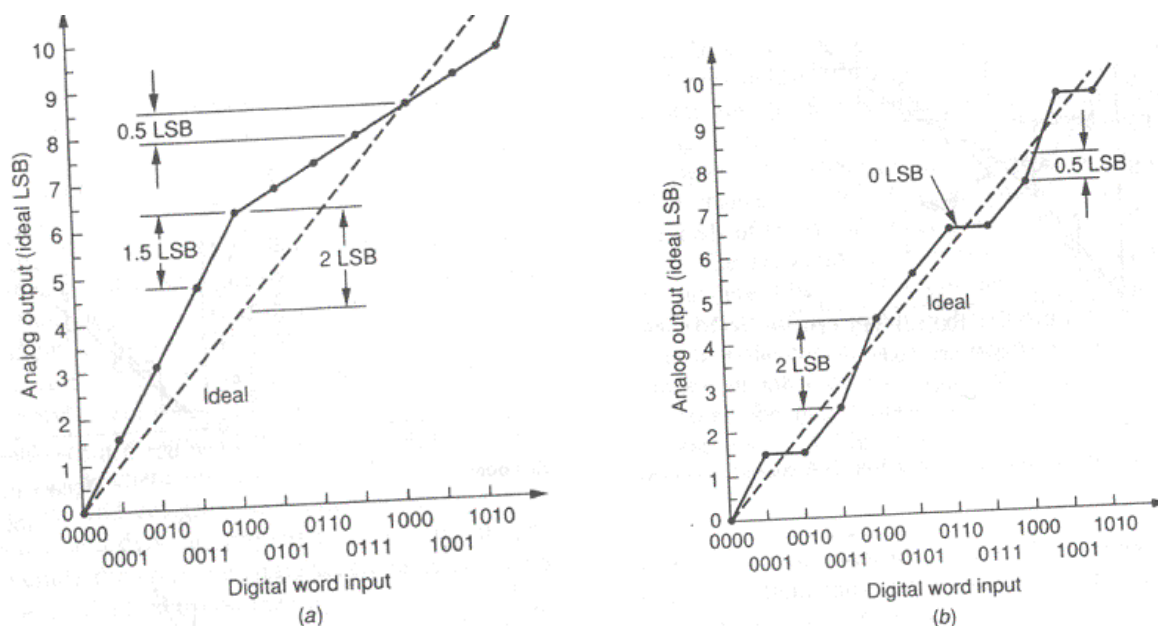
harmonicznego na wyjściu przetwornika o maksymalnej dostępnej amplitudzie do wartości skutecznej szumu kwantowania (a dla przebiegu jw. wynosi on $FS/(2^N\sqrt{12})$).

$$SNR = 20 \log_{10} \left(\frac{\frac{FS}{2\sqrt{2}}}{\frac{FS}{2^N\sqrt{12}}} \right) = 20 \log_{10}(2^N) + 20 \log_{10} \left(\frac{\sqrt{6}}{2} \right) = 6.02N + 1.76 \quad (5.6)$$

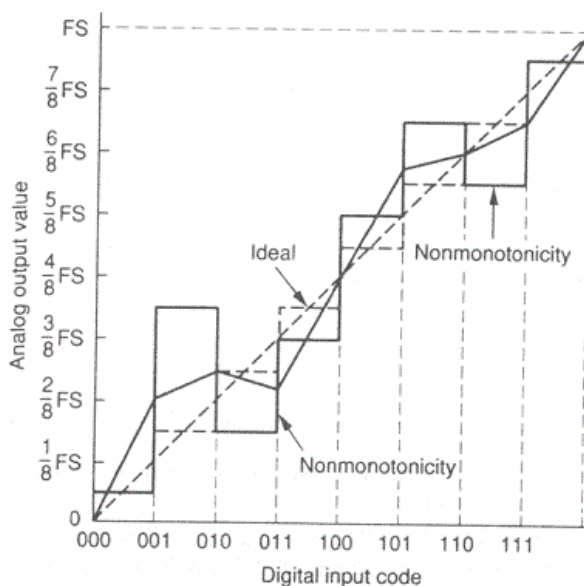


Rys. 5.5. Przykłady błędów przetworników CA, (a) przesunięcie, (b) błąd wzmacnienia, (c) nieliniowość.

Pozostałe parametry statyczne zawierają: błąd przesunięcia, błąd wzmacnienia, błędy nieliniowości i niemonotoniczność charakterystyki przejściowej. Parametry te są przedstawione na rysunkach 5.6-7.



Rys. 5.6. Różnica pomiędzy nieliniowością całkowitą oraz różnicową. (a) przetwornik z nieliniowością całkowitą 2LSB i nieliniowością różnicową 0.5LSB, (b) przetwornik z nieliniowością całkowitą 0.5LSB i nieliniowością różnicową 1LSB.



Rys. 5.6. Przykład konwertera CA z charakterystyką niemonotoniczną.

Parametry dynamiczne są związane ze zmianami cyfrowego słowa wejściowego. Czas niezbędny do pojawienia się prawidłowej wartości na wyjściu przy zmianie wejściowego słowa cyfrowego jest nazywany czasem ustalania.

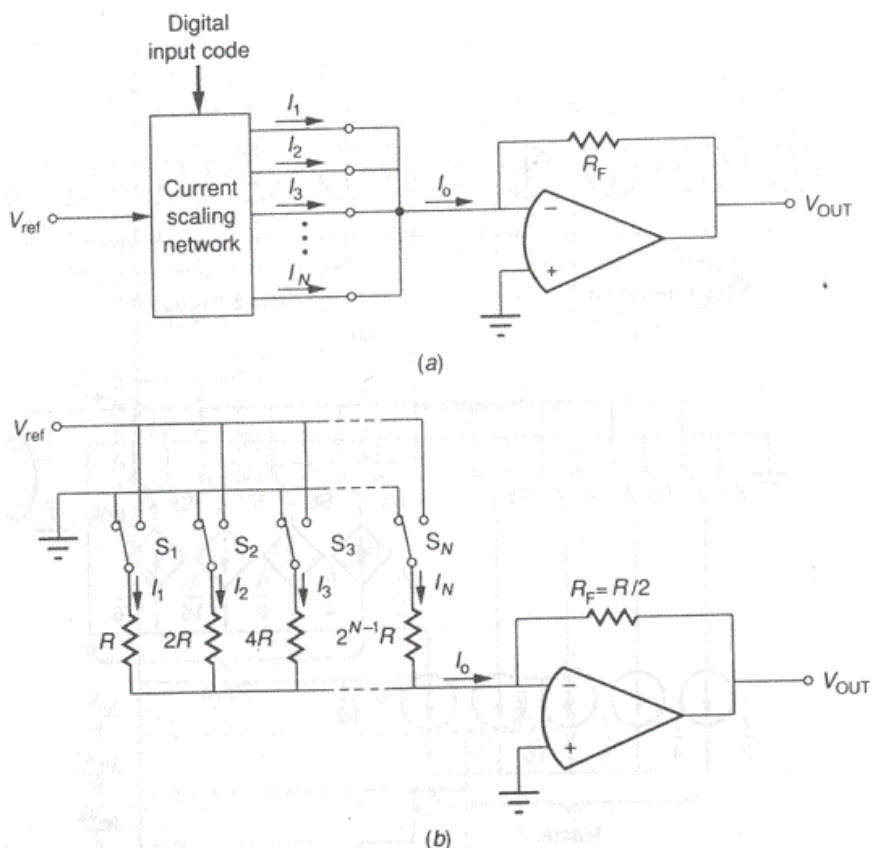
6. Realizacja scalona wybranych przetworników CA.

6.1 Przetworniki CA ze skalowaniem prądu.

Ogólna zasada działania przetwornika ze skalowaniem prądu jest przedstawiona na rysunku 6.1. Napięcie odniesienia jest przekształcone do binarnie skalowanych prądów a następnie przetworzone na wynikowe napięcie wyjściowe. Każdy z kluczy $S_1 - S_N$ jest połączony do V_{REF} jeśli odpowiadający bit jest równy 1 i do masy jeśli dany bit jest równy 0. Napięcie wyjściowe jest równe:

$$V_{OUT} = \frac{-R}{2} I_O = \frac{-R}{2} \left(\frac{b_1}{R} + \frac{b_2}{2R} + \frac{b_4}{4R} + \dots + \frac{b_1}{2^{N-1}R} \right) V_{REF} \quad (6.1)$$

W celu uzyskania dowolnego pożądanego wzmocnienia przetwornika można skalować rezystor sprzężenia zwrotnego R_F . Klucze można umieścić od strony napięcia odniesienia (jak na rysunku) lub przenieść na dolne wyprowadzenia rezystorów. To drugie rozwiązanie ma tę zaletę, że napięcie na kluczu jest zawsze równe zero i w konsekwencji jego pojemności pasożytnicze nie są ani ładowane ani rozładowywane w czasie zmian wejściowego słowa cyfrowego.

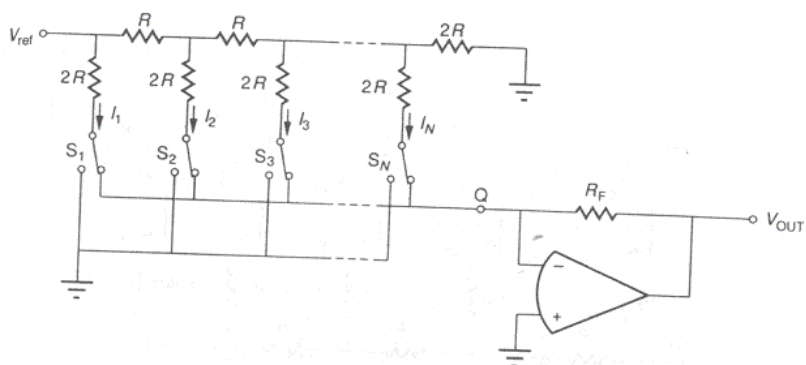


Rys. 6.1. Zasada działania przetwornika z podziałem prądu (a) oraz implementacja (b).

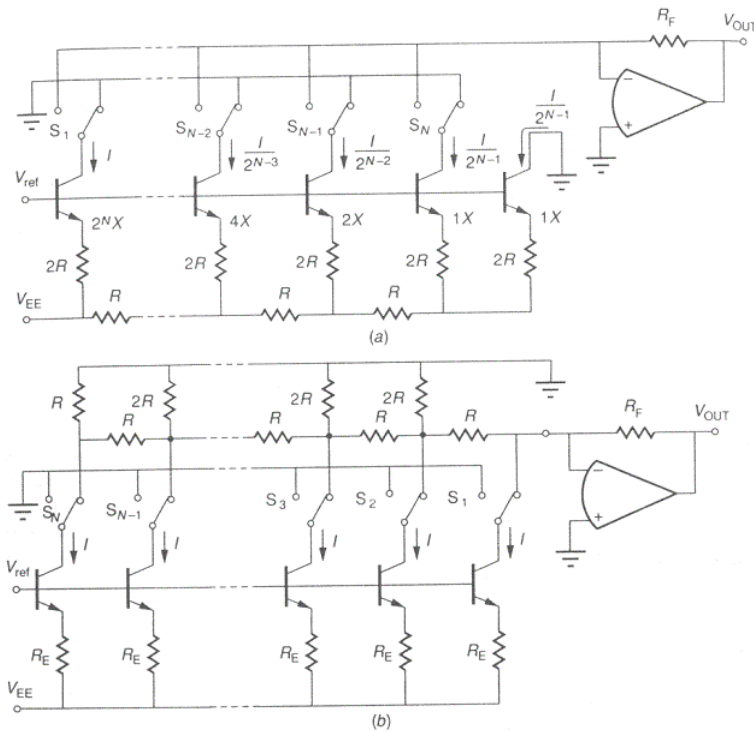
Stosunek wartości rezystancji w przetworniku z rys. 6.1 jest bardzo duży i wynosi :

$$\frac{R_{MSB}}{R_{LSB}} = \frac{1}{2^{N-1}} \quad (6.2)$$

Na przykład stosunek (6.2) dla przetwornika 8 bitowego wynosi 114. Bardzo trudno w takim układzie jest wykonać rezystory gdyż dokładność wykonania rezystora o największej wartości musi być lepsza niż wartość rezystora o minimalnej wartości. Dla przetwornika 8 bitowego dokładność ta musi być lepsza niż 0.78%. Taka dokładność jest osiągnięta w układach scalonych jedynie poprzez dostrajanie (laserem lub innymi technikami).

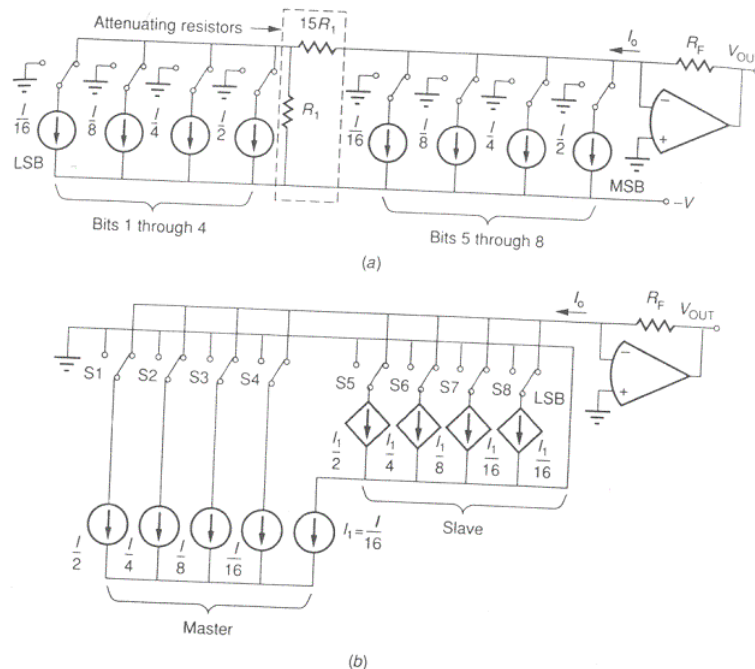


Rys. 6.2. Przetwornik ze skalowaniem prądu z drabinką R-2R.



Rys. 6.3. Dwie implementacje przetworników CA z drabinką rezystorową R-2R.

Alternatywnym rozwiązaniem jest zastosowanie drabinki rezystorowej typu R-2R, jak przedstawiono na rys. 6.2. Każdy z kluczy połączony jest z węzłem Q jeśli odpowiadający mu bit jest równy 1 oraz połączony do masy jeśli odpowiadający bit jest równy 0. Prąd I_1 jest równy $V_{REF}/(2R)$. Uwzględniając, iż rezystancja widziana z prawej strony drabinki w każdym jej węzle wynosi $2R$, prądy $I_1, I_2, \dots, I_N$ są skalowane binarnie a wynikowa wartość wyjściowa napięcia jest zgodna ze wzorem 6.1. Przetwornik z drabinką R-2R ma 2 razy większą liczbę elementów niż poprzednik, jednak wartości rezystancji są w stosunku $R/(2R)$ co jest dużo łatwiejsze do wykonania. Jego niekorzystną cechą jest fakt iż posiada 2^{N-1} węzłów wewnętrznych, które są czułe na pojemności pasożytnicze (mogą zwiększać czasy ustalania odpowiedzi). Przykładowe dwie architektury rozwiązań praktycznych przedstawione są na rysunku 6.3.

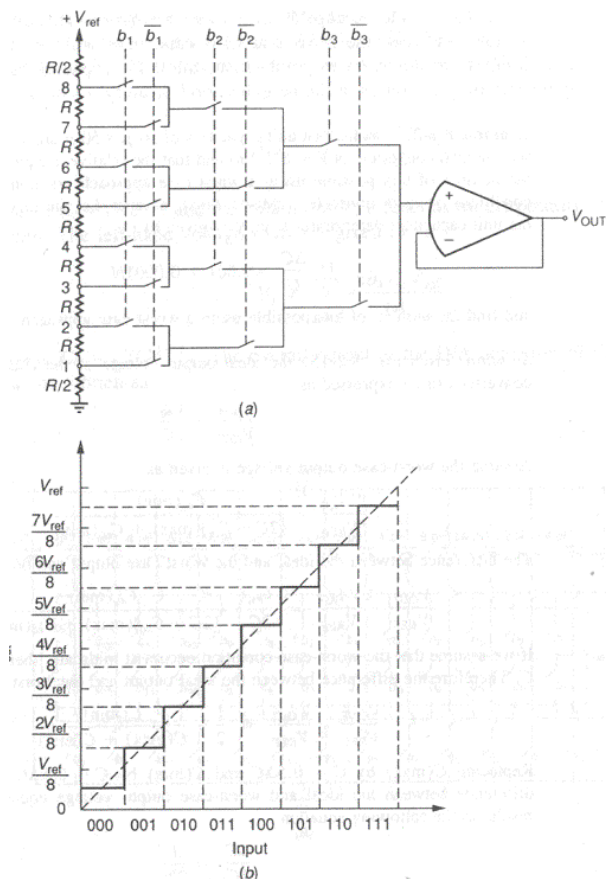


Rys. 6.4. (a) użycie dzielnika prądowego do kaskadowego połączenia dwóch źródeł prądowych oraz (b) technika master-slave.

Na rysunku 6.4 przedstawione są dwa inne rozwiązania zapewniające umiarkowany rozrzut wartości między rezystorami realizującymi prądy MSB i LSB. Pierwsze rozwiązanie (a) nazywane jest kaskadą. Zastosowanie dzielnika prądowego umożliwiło połączenie dwóch przetworników 4-ro bitowych w celu uzyskania jednego 8-io bitowego. Dokładność podziału dzielnika 1:16 musi być w zakresie wartości LSB całego przetwornika. Druga technika nazywana jest master-slave (b). W tym przypadku drabinka nadrzędna składa się z połowy bardziej znaczących bitów a drabinka podrzędna z połowy mniej znaczących bitów. Dokładność źródła prądowego $I_I = I/16$ musi być lepsza niż $\pm 0.5\text{LSB}$.

6.2 Przetwornik CA ze skalowaniem napięcia.

Przetwornik ze skalowaniem napięcia używa szeregowego połączenia rezystorów pomiędzy V_{REF} i masą w celu wytworzenia szeregu napięć będących progami zmiany wartości wyjściowej. Dla N bitowego przetwornika ciąg rezystorów składa się z 2^N sztuk rezystorów. W zależności od potrzeb wszystkie rezystory mogą być równe lub też rezystory końcowe mogą mieć wartości częściowe. Wzmacniacz operacyjny realizuje wyłącznie funkcję buforowania. Każdy węzeł ciągu rezystorów połączony jest z drzewem kluczy sterowanych wejściowym słowem cyfrowym. Jeśli dany bit jest równy 1 wówczas odpowiadający mu klucz b_i jest zamknięty a $\bar{b}_i$ otwarty. Jeśli jest równy zeru klucze są sterowane przeciwnie.



Rys. 6.5. Przetwornik CA ze skalowaniem napięcia.

Przykład 6.1 Znajdź wymaganą dokładność sieci rezystorowej N równych segmentów jako funkcję liczby bitów N . Ile wynosi największa liczba bitów dla 2% rezystorów przy założeniu, że dokładność ma być nie gorsza niż $\pm 0.5\text{LSB}$?

Rozwiązanie: W przypadku idealnym napięcie liczone od danego węzła do masy wzdłuż k rezystorów jest równe:

$$V_k = \frac{kR}{2^N R} V_{REF}$$

Największa zmiana tego napięcia wystąpi gdy założymy że wszystkie rezystory powyżej danego punktu będą miały mają maksymalną rezystancję a poniżej tego punktu rezystancję minimalną. Wówczas napięcie w węźle k jest równe:

$$V'_k = \frac{kR_{\min} V_{REF}}{(2^N - k)R_{\max} + kR_{\min}}$$

Różnica pomiędzy powyższymi napięciami wynosi:

$$\frac{V_k}{V_{REF}} - \frac{V'_k}{V_{REF}} = \frac{kR}{2^N R} - \frac{kR_{\min}}{(2^N - k)R_{\max} + kR_{\min}}$$

Założona różnica nie może przekroczyć 0.5LSB, stąd musi być spełniona nierówność:

$$\frac{kR}{2^N R} - \frac{kR_{\min}}{(2^N - k)R_{\max} + kR_{\min}} < \frac{0.5}{2^N}$$

Względna dokładność wykonania rezystorów może zostać wyrażona jako $\Delta R/R$, co daje maksymalną i minimalną wartość rezystancji równe $R_{\max} = R + 0.5\Delta R$, $R_{\min} = R - 0.5\Delta R$. Najgorszy przypadek pojawia się w środku drabinki rezystorowej, czyli dla $k=0.5 \cdot 2^N$. Zakładając dokładność rezystorów 2% powyższa nierówność sprowadza się do postaci:

$$|0.01| < 2^{-N}$$

Co daje w rezultacie największą liczbę naturalną równą $N=5$.

Struktura przetwornika ze skalowaniem napięcia jest regularna i dobrze pasuje do technologii CMOS. Zaletą jest to, że gwarantuje monotoniczność. Powierzchnia wymagana do realizacji jest duża dla dużej liczby bitów (większej niż 8). Przetwornik jest również wrażliwy na pojemności pasożytnicze w węzłach drabinki rezystorowej i sieci kluczy i dlatego nie jest bardzo szybki.

6.3 Przetworniki z podziałem ładunków.

Przetworniki z podziałem ładunków działają na zasadzie podziału ładunku na tablicy kondensatorów. Zazwyczaj, najpierw następuje rozładowanie wszystkich kondensatorów. Na rysunku 6.6 przedstawiono schemat blokowy takiego przetwornika. Użyto nie nakładający się zegar dwufazowy. Symbol ϕ_1 oznacza że dany klucz jest zamknięty w fazie 1 a symbol ϕ_2 oznacza że zamknięty w fazie 2. W czasie ϕ_1 wszystkie kondensatory są rozładowane. Następnie w czasie fazy ϕ_2 kondensatory, które są związane z bitem wejściowego słowa cyfrowego, którego aktualna wartość wynosi 1, są łączone do V_{REF} , a związane z bitem o wartości 0 są łączone do masy. Napięcie wyjściowe można oszacować poprzez odpowiednie porównanie wartości ładunków zgromadzonych na całkowitej pojemności C_{TOT} , daje to następujące równanie:

$$V_{REF} \left(b_1 + \frac{b_2}{2^2} + \dots + \frac{b_N}{2^{N-1}} \right) C = V_{REF} C_{EQ} = C_{TOT} V_{OUT} = 2C V_{OUT} \quad (6.3)$$

gdzie: C_{EQ} – suma pojemności ładowana napięciem odniesienia.

Z równania powyższego wynika, że:

$$V_{OUT} = (b_1 2^{-1} + b_2 2^{-2} + \dots + b_N 2^{-N}) V_{REF} \quad (6.4)$$

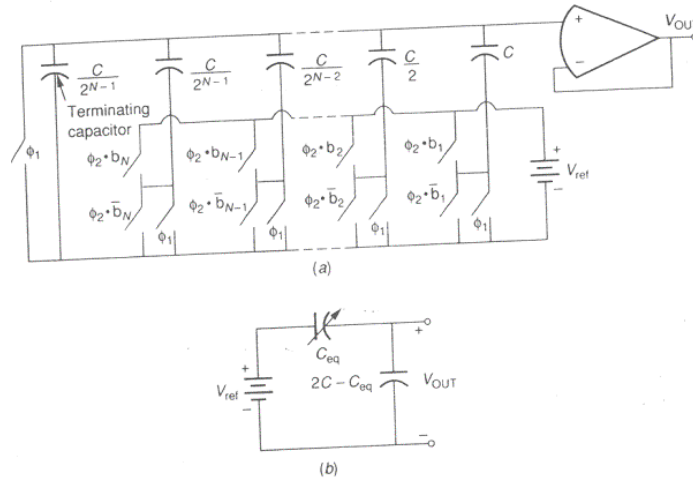
Przetwornik z rysunku 6.6a można łatwo zmodyfikować do generacji napięć zarówno dodatnich jak i ujemnych. Jeśli dolne okładki wszystkich kondensatorów zostaną przyłączone w fazie 1 do V_{REF} zamiast do masy oraz zostanie zanegowane znaczenie bitów przetwarzanego słowa cyfrowego wówczas:

$$V_{OUT} = -(b_1 2^{-1} + b_2 2^{-2} + \dots + b_N 2^{-N}) V_{REF} \quad (6.5)$$

Decyzja co do wyboru znaku + lub – wymaga dodatkowego bitu słowa cyfrowego. Jeśli dodatkowo napięcie V_{REF} jest bipolarne wówczas otrzymujemy przetwornik czteroćwiartkowy.

Maksymalna osiągnięta liczba bitów przetwornika ograniczona jest poprzez precyzję wykonania kondensatorów. Dokładność przetwornika CA zależy od względnych dokładności

wykonania pojemności i ewentualnych pojemności pasożytniczych. Dokładność kondensatorów o zbliżonych wartościach pojemności w technologii CMOS wynosi około 0.1% lub nawet lepiej. Jeśli wszystkie kondensatory miałyby taką dokładność wykonania prezentowany przetwornik byłby w stanie pracować z dokładnością do 10-ciu bitów. Jednakże wymaga to zastosowania kondensatorów o stosunku pojemności 1:1024 co nie jest korzystne z punktu widzenia użytej powierzchni.



Rys. 6.5. Przetwornik CA z podziałem ładunku.

Przykład 6.2. Załóżmy, że kondensatory jednostkowe o wymiarach 50 x 50um są użyte do budowy sieci przetwornika z podziałem ładunku wg rysunku 6.6(a). Względna dokładność wykonania poszczególnych kondensatorów wynosi 0.1%. Znajdź maksymalną liczbę bitów przetwornika jeśli dopuszczalny błąd wynosi $\pm 0.5\text{LSB}$ i przy założeniu, że występuje on w środku skali przetwornika tj. dla włączonego tylko bitu MSB. Następnie, zakładając, że dokładność wykonania kondensatorów pogarsza się wraz z liczbą realizowanych bitów wg zależności $\Delta C/C = 0.001 + 0.0001N$ znajdź liczbę realizowanych bitów przetwornika.

Rozwiązanie: Napięcie wyjściowe można określić na podstawie równania

$$\frac{V_{OUT}}{V_{REF}} = \frac{C_{EQ}}{2C}$$

Najgorszy przypadek daje napięcie równe

$$\frac{V'_{OUT}}{V_{REF}} = \frac{C_{EQ(\min)}}{[2C - C_{EQ}(\max)] + C_{EQ(\min)}}$$

Różnica pomiędzy napięciami będzie równa

$$\frac{V_{OUT}}{V_{REF}} - \frac{V'_{OUT}}{V_{REF}} = \frac{C_{EQ}}{2C} - \frac{C_{EQ(\min)}}{[2C - C_{EQ}(\max)] + C_{EQ(\min)}}$$

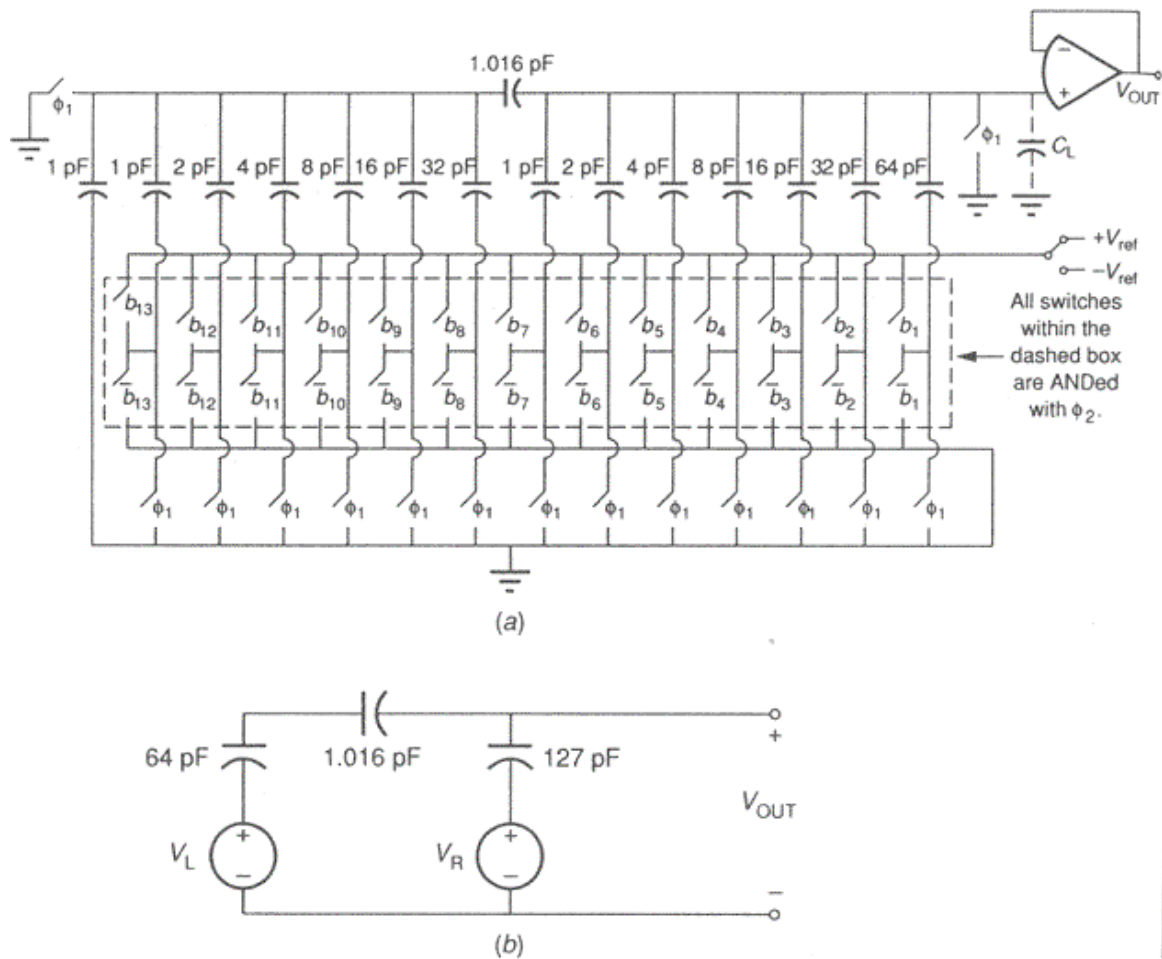
Jeśli założymy, że najgorszy przypadek wystąpi dla środka skali przetwornika wówczas $C_{EQ} = C$, a stąd różnica między wartością idealną a najgorszym przypadkiem wynosi

$$\frac{V_{OUT}}{V_{REF}} - \frac{V'_{OUT}}{V_{REF}} = \frac{1}{2} - \frac{C_{(\min)}}{C_{(\max)} + C_{(\min)}}$$

Zastępując $C_{(\max)} = C + 0.5\Delta C$ oraz $C_{(\min)} = C - 0.5\Delta C$ otrzymujemy

$$\frac{\Delta C}{2C} = \frac{1}{2^N}$$

co przy założeniu, że $\Delta C/C = 0.001$ daje wartość $N = 10$. Jeśli przyjmiemy ograniczenie $\Delta C/C = 0.001 + 0.0001N$ wynikowa liczba bitów wyniesie $N = 8$.



Rys. 6.6. 13-to bitowy przetwornik kaskadowy z podziałem ładunku.

Do układu z rysunku 6.6(a) można zastosować połączenie kaskadowe. Na rysunku 6.7 przedstawiono przykład praktyczny przetwornika 13-to bitowego. Kondensator o wartości 1.016 pF jest dzielnikiem o stosunku podziału 1:64. Układ równoważny przedstawiony jest na rysunku 6.7(b). Wydajności źródeł napięciowych przedstawionych na rysunku zależą od aktualnego położenia kluczy. Napięcie V_R wynosi

$$V_R = \sum_{i=1}^7 \frac{\pm b_i C_i V_{REF}}{127} \quad (6.6)$$

gdzie: $C_i = C/2^{(i-1)}$, a znak licznika zależy od polaryzacji napięcia odniesienia V_{REF} .

Napięcie z lewej strony wynosi

$$V_L = \sum_{k=8}^{13} \frac{\pm b_k C_k V_{REF}}{64} \quad (6.7)$$

gdzie: $C_k = C/2^{(k-1)}$. Całkowita wartość napięcia wyjściowego wynosi

$$V_{OUT} = \frac{\pm V_{REF}}{128} \left[\sum_{i=1}^7 b_i C_i + \sum_{k=8}^{13} b_k \frac{C_k}{64} \right] \quad (6.8)$$

Przetworniki z podziałem ładunku są wrażliwe na pojemnościowe obciążenie w węzle sumującym. Jeśli wartość tej pojemności oznaczmy C_L wówczas wyrażenie 6.8 przekształca się do postaci

$$V_{OUT} = \left(1 - \frac{C_L}{128} \right) \frac{\pm V_{REF}}{128} \left[\sum_{i=1}^7 b_i C_i + \sum_{k=8}^{13} b_k \frac{C_k}{64} \right] \quad (6.9)$$

Pojemność C_L wprowadza błąd wzmocnienia o wartości $[1 - (C_L/128)]$.

Dokładność tłumika pojemnościowego musi również być wystarczająco wysoka. Zmiany wartości pojemności z 1.016 wprowadza zarówno błędy wzmocnienia jak i liniowości. Jeśli założymy zmiany pojemności o wartości ΔC wówczas równanie 6.8 przekształca się do postaci:

$$V_{OUT} = \frac{\pm V_{REF}}{128} \left(1 - \frac{\Delta C_L}{128} \right) \left[\sum_{i=1}^7 b_i C_i + (1 + \Delta C) \sum_{k=8}^{13} b_k \frac{C_k}{64} \right] \quad (6.10)$$

Jeśli założymy 1.6% błąd wartości pojemności wówczas błąd wzmocnienia wyniesie

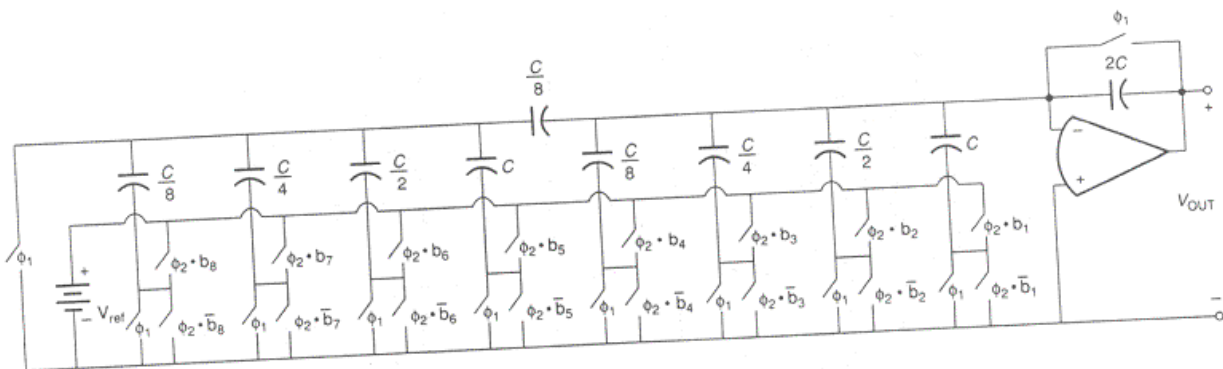
$$\text{Gain error} = 1 - \frac{\Delta C}{128} = 1 - \frac{1}{64 \cdot 128} \quad (6.11)$$

co jest wartością pomijalną. Błąd liniowości jest określony jako

$$\text{Linearity error} = \Delta C \sum_{k=8}^{13} \frac{b_k C_k}{64}$$

Najgorszy przypadek wystąpi dla współczynników b równych 1 i wyniesie ΔC . Utrzymywanie stabilnych wartości napięcia referencyjnego jest również bardzo ważne.

Nieco inny, dwustopniowy przetwornik CA z podziałem ładunków przedstawiony jest na rysunku 6.7. Wzmacniacz operacyjny jest użyty w konfiguracji odwracającej fazę z kondensatorem w sprzężeniu o wartości $2C$. Pojemności związane z wejściem wzmacniacza są wyeliminowane gdyż w takiej konfiguracji na wejściu odwracającym panuje masa pozorną.



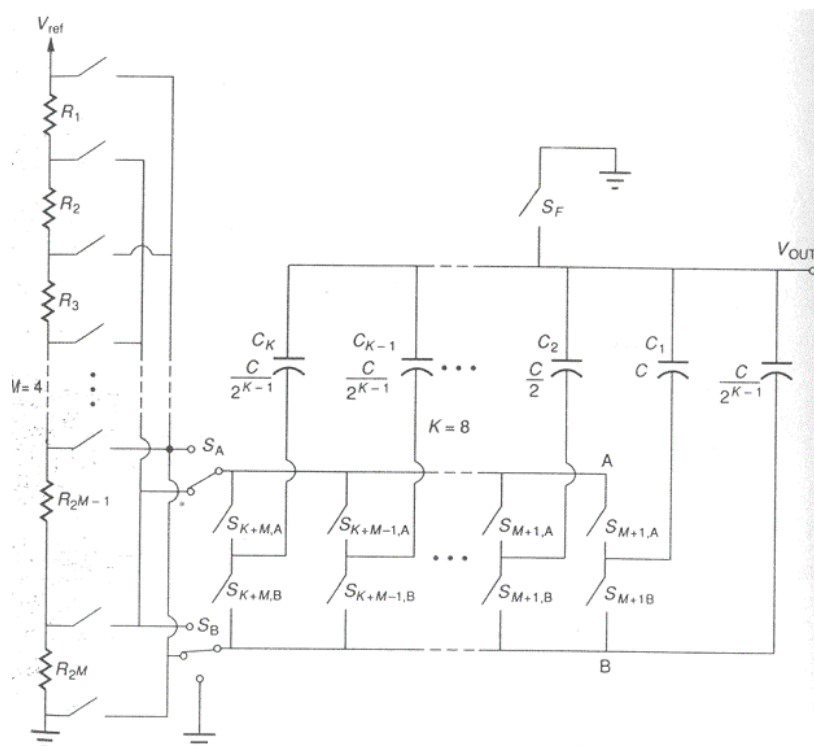
Rys. 6.7. Kaskadowy przetwornik CA z podziałem ładunku z użyciem wzmacniacza operacyjnego w konfiguracji odwracającej.

6.4 Przetworniki CA z użyciem mieszanych technik skalowania.

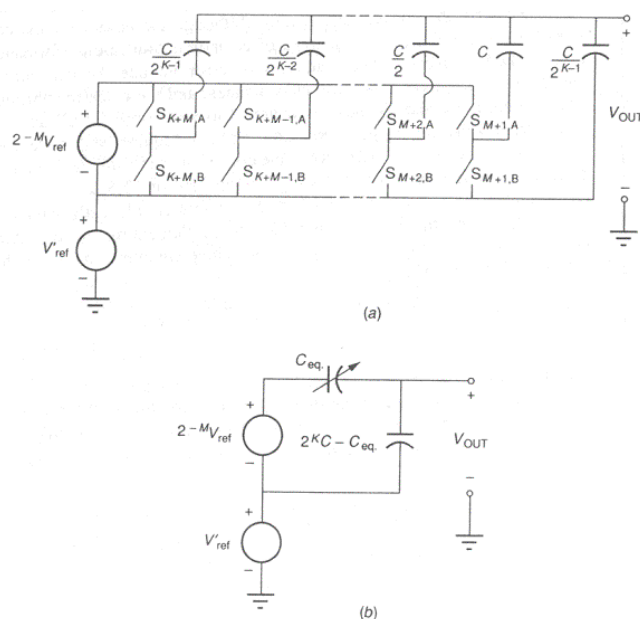
W celu zwiększenia rozdzielczości przetwornika stosuje się techniki łączenia różnych metod skalowania. Można użyć M -bitowy ciąg rezystorów oraz K -bitową tablicę kondensatorów do budowy przetwornika $N=(M+K)$ bitowego. Na rysunku 6.9 przedstawiono przykład takiego przetwornika, gdzie $M=4$ oraz $K=7$. Ciąg rezystorów od R_1 do R_{2^M} zapewnia monotoniczne wartości napięcia V_{REF} podzielonego na 2^M wartości. W celu dalszego podziału tych napięć na kolejne 2^K podprzedziałów użyto binarnie skalowaną tablicę kondensatorów $C_1 - C_K$. Praca przetwornika przebiega w następującym ciągu zdarzeń:

- Zamknięcie kluczy S_F , S_B oraz $S_{M+1,B} - S_{M+K,B}$, co powoduje uziemienie zarówno dolnych jak i górnych okładek kondensatorów $C_1 - C_K$.
- Po otwarciu klucza S_F szyny A oraz B są połączone do rezystorów drabinki określonych przez M bardziej znaczących bitów wejściowego słowa binarnego. Napięcia dołączone do dolnej i górnej okładki są równe odpowiednio V'_{REF} oraz $V'_{REF} + 2^{-M} V_{REF}$, gdzie: $V'_{REF} = V_{REF} (b_1 2^{-1} + b_2 2^{-2} + \dots + b_M 2^{-M})$. Po tym kroku układ można przedstawić jako równoważny jak na rys. 6.10(a).
- Ostatnim krokiem jest odpowiednie połączenie okładek kondensatorów do węzła A lub B w zależności od zestawu mniej znaczących bitów K słowa wejściowego. Po tym etapie układ równoważny jest jak na rys. 6.10(b). Napięcie wyjściowe jest wówczas równe

$$V_{OUT} = V_{REF} (b_1 2^{-1} + b_2 2^{-2} + \dots + b_M 2^{-M}) + V_{REF} \left[\frac{b_{M+1}}{2^{M+1}} + \frac{b_{M+2}}{2^{M+2}} + \dots + \frac{b_{M+K}}{2^{M+K}} \right] \quad (6.12)$$



Rys. 6.8. Przetwornik CA z wykorzystaniem podziału napięcia i ładunku.

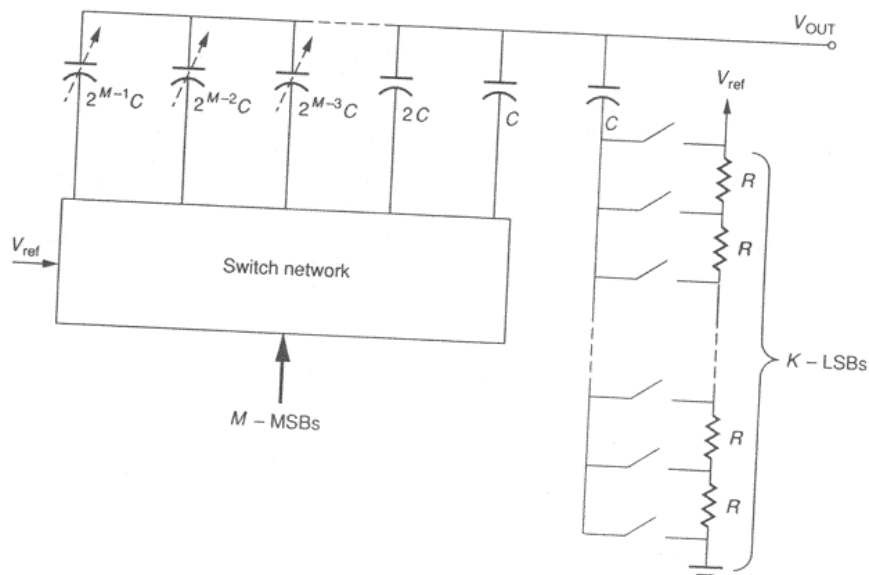


Rys. 6.9. Układ równoważny przetwornikowi z rys. 6.8.

Przetwornik z rysunku 6.9 ma tą zaletę, że ze względu na monotoniczność przetwornika rezystancyjnego M bitów MSB zawsze będzie monotoniczna niezależnie od dopasowania. To daje możliwość wykonania tablicy kondensatorów zawierającej mniej bo tylko K elementów bardzo silnie do siebie dopasowanych a cały przetwornik ma rozdzielczość będącą sumą rozdzielczości cząstkowych $N=K+M$. Szybkość przetwarzania sieci rezystancyjnej jest wyższa niż pojemnościowej, dodatkowo sieć rezystancyjna nie wymaga wstępnego rozładowania. Z tego względu, w zależności od potrzeb można stosować kompromis szybkość – dokładność. Do zwiększenia precyzji sieci rezystorowej można zastosować techniki dostrajania rezystorów z bezpiecznikami polikrzemowymi.

Możliwe są również inne połączenia technik podziału napięcia i ładunku. Najprostszym wariantem jest zamiana poprzez skalowanie bitów MSB tablicą pojemnościową a bitów LSB dzielnikiem rezystancyjnym. Rys. 6.11 przedstawia takie rozwiązanie. Kondensatory 3 – 4 są

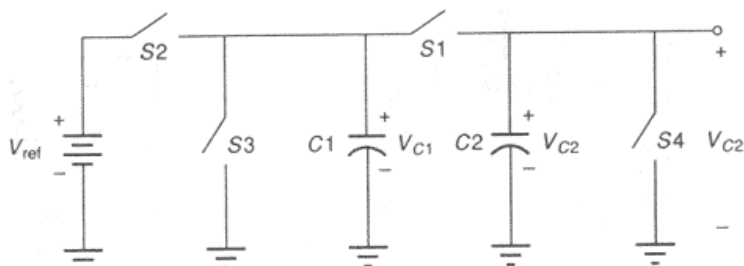
trymowane z użyciem techniki bezpieczników polikrzemowych, jest to oznaczone na schemacie przerywanymi strzałkami. Połączenie skalowania napięcia oraz ładunku umożliwia optymalizację parametrów układu. Wybrać można którą część wejściowego słowa bitowego przetwarza tablica kondensatorów a którą ciąg rezystorów. Jeśli znamy powierzchnie zajmowane przez rezystory oraz kondensatory jednostkowe oraz ich względne dokładności można dokonywać kompromisu ze względu na dokładność czy użytą powierzchnię.



Rys. 6.10. Przetwornik CA z podziałem ładunku dla bardziej znaczących bitów oraz podziałem napięcia dla mniej znaczących bitów.

6.5 CA przetworniki szeregowe.

Ostatnią zaprezentowaną kategorią są przetworniki szeregowe. Szeregowy przetwornik CA to taki przetwornik w którym konwersja następuje sekwencyjnie w kilku cyklach pracy. W najlepszym przypadku na jeden bit przetwarzanego słowa wejściowego przypada jeden cykl pracy przetwornika co daje łącznie minimum N cykli. Poniżej przedstawione zostaną dwa typy przetworników szeregowych, mianowicie z redystrybucją ładunku oraz przetwornik algorytmiczny.

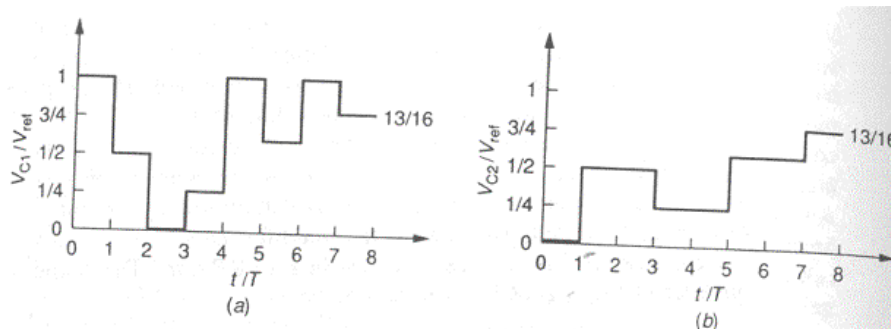


Rys. 6.10.1. Uproszczony schemat przetwornika redystrybucyjnego szeregowego.

Na rysunku 6.12 przedstawiono uproszczoną budowę szeregowego przetwornika CA z redystrybucją ładunku. Przetwornik składa się z 4 kluczy, 2 identycznych kondensatorów i źródła napięcia odniesienia. Klucz S1, nazywany kluczem redystrybucyjnym, ma na celu równoległe połączenie obu kondensatorów i w ten sposób wyrównanie wartości zgromadzonych na nich ładunków. Klucz S2 ma na celu naładowanie kondensatora C1 do wartości V_{REF} jeśli dany bit b_i jest równy 1 a klucz S3 służy wyzerowania C1 jeśli b_i jest równe zero. Klucz S4 służy do wstępnego rozładowania kondensatora C2 na początku procesu. Poniższy przykład omawia szczegółowo pracę przetwornika.

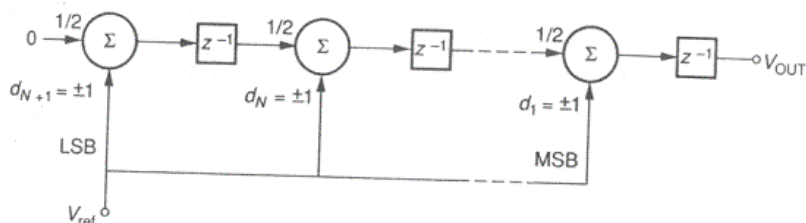
Przykład 6.3. Praca szeregowego przetwornika CA. Załóżmy, że $C1=C2$ a przetwarzane słowo ma wartość 1101. Należy wyznaczyć wartości napięć na kondensatorach C1 oraz C2.

Rozwiązanie: Proces przetwarzania rozpoczyna się wyzerowaniem kondensatora C2 czyli zamknięciem i otwarciem klucza S4 a więc $V_{C2}=0$. Ponieważ $b_4=1$, równocześnie zwierany jest klucz S2, co daje $V_{C1}=V_{REF}$. Następnie otwierany jest klucz S2 a zamykany S1 co daje napięcia równe $V_{C1}=V_{C2}=0.5V_{REF}$ co kończy fazę przetwarzania bitu b_4 . Następnie przetwarzamy bit b_3 czyli przy rozwartym kluczu S1 zwieramy klucz S3 a następnie rozwieramy S3 i zwieramy S1. Napięcia na kondensatorach są teraz równe $V_{C1}=V_{C2}=0.25V_{REF}$. Ponieważ dwa najstarsze bity są równe 1 nastąpi dwukrotne naładowanie kondensatora C1 i przekazanie ładunku do C2. Napięcie końcowe uzyskano po 9 cyklach kluczkowania a jego końcowa wartość wynosi $V_{C1}=V_{C2}=13/16V_{REF}$.



Rys. 6.13. Przebiegi napięć na kondensatorach dla przetwornika z rys. 6.12 przy przetwarzaniu słowa wejściowego 1101.

Z powyższego przykładu wynika, że przetwornik wymaga układu pomocniczego generującego ciąg sterujący poszczególnymi kluczami. Źródłami błędów w tym przetworniku są: pojemności pasożytnicze kondensatorów i klucze oraz przenikanie sygnału zegara. Kondensatory C1 i C2 muszą być identyczne z dokładnością lepszą niż 0.5LSB. Zaletami przetwornika są monotoniczność oraz bardzo mała zajmowana powierzchnia. Znane są praktyczne rozwiązania takiego przetwornika o rozdzielczości 8-iu bitów i czasie przetwarzania równym 13.5us.

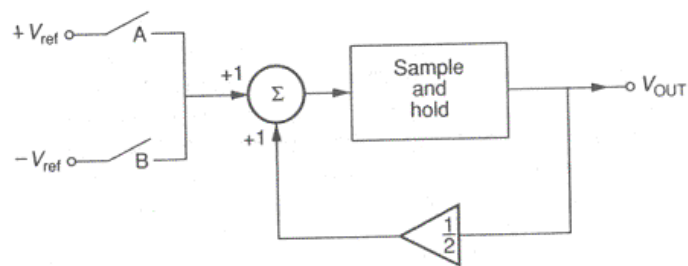


Rys. 6.14. Potokowy, algorytmiczny przetwornik CA.

Drugim prezentowanym przetwornikiem, o działaniu szeregowym, jest przetwornik algorytmiczny. Na rysunku 6.14 przedstawiono uproszczoną budowę. Składa się on z układów wprowadzających opóźnienia jednostkowe oraz sumatory. Można wykazać, że napięcie wyjściowe na wyjściu przetwornika wynosi

$$V_{OUT}(z) = (d_1 z^{-1} + 2^{-1} d_2 z^{-2} + \dots + 2^{-N+1} d_N z^{-N}) V_{REF} \quad (6.13)$$

gdzie: d_i jest zmodyfikowaną wartością bitu wejściowego b_i równą +1 dla stanu wysokiego i -1 dla stanu niskiego, oraz operator z^{-1} oznacza opóźnienie jednostkowe. Zaletą przetwornika jest to że składa się z podobnych bloków. Jeśli do pierwszego stopnia wprowadzimy 0 wówczas wszystkie stopnie są identyczne. Poprzez inteligentne podawanie ciągu przetwarzanych bitów można osiągnąć zjawisko polegające na tym, że czas przetwarzania będzie równy liczbie bitów przetwarzanego ciągu, jednakże kolejne wyniki, dla kolejnych słów wejściowych, można uzyskiwać już co jeden cykl zegarowy.

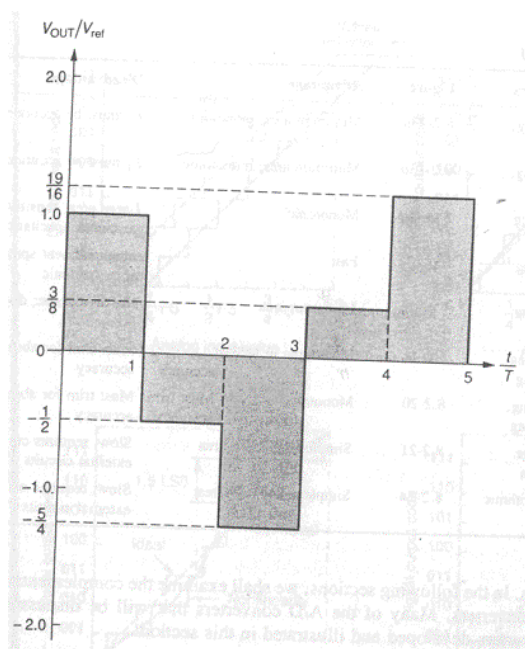


Rys. 6.15. Przetwornik algorytmiczny iteracyjny równoważny przetwornikowi z rys. 6.14.

Dzięki identyczności wszystkich stopni przetwornika, jego złożoność można zredukować poprzez zastosowanie przetwarzania iteracyjnego w miejsce potokowego. Schemat blokowy takiego rozwiązania przedstawiono na rysunku 6.15. Równanie opisujące pojedynczy stopień można podać w postaci

$$V_{OUT}(z) = \frac{d_i z^{-1} V_{REF}}{1 - 0.5z^{-1}} \quad (6.14)$$

Klucze w przetworniku na rysunku 6.15 są zwierane do $+V_{REF}$ jeśli dany bit jest równy 1 oraz do $-V_{REF}$ jeśli dany bit jest równy zero. Poniższy przykład ilustruje działanie przetwornika.



Rys. 6.15. Zmiany napięcia wyjściowego przetwornika z rys. 6.15 w czasie przetwarzania ciągu 11001.

Przykład 6.4. Przetwarzanie w przetworniku algorytmicznym iteracyjnym. Zakładając, że przetwarzany ciąg wejściowy wynosi 11001 należy określić wartość napięcia wyjściowego przetwornika z rysunku 6.15.

Rozwiązanie: Na początku procesu przetwarzania należy wyzerować wyjście (nie przedstawione na rysunku 6.15). Na rysunku 6.16 przedstawiony jest przebieg czasowy napięcia wyjściowego. T jest okresem przetwarzania pojedynczego bitu. Proces przetwarzania rozpoczyna się od bitu LSB, który w rozpatrywanym przypadku wynosi 1. Zamknięty jest klucz A a na wyjściu pojawia się wartość V_{REF} . W drugim kroku bit jest równy 0 na wyjście przechodzi napięcie równe $-V_{REF} + 1/2 V_{REF} = -1/2 V_{REF}$. Następny bit również jest równy zero więc napięcie wyjściowe osiągnie wartość $-1/4 V_{REF} - V_{REF} = -5/4 V_{REF}$. Postępując dalej uzyskujemy końcowe napięcie równe $19/16 V_{REF}$. Zakres odtwarzanych napięć wynosi od $V_{REF}(1 + 1/2 + 1/4 + 1/8 + 1/16) = 31/16 V_{REF}$ dla wejściowej wartości słowa cyfrowego 0000 do $+31/16 V_{REF}$ dla wejściowej wartości słowa 11110.

Przetwornik algorytmiczny ma zaletę, że nie jest zależny od stosunku pojemności, jednak w celu wytworzenia stosunku podziału $1/2$ mamy zależność stosunku pojemności lub rezystorów i musi ona być lepsza niż $\pm 0.5 \text{LSB}$. Przetwornik algorytmiczny zostanie jeszcze raz zaprezentowany w

części poświęconej przetwornikom AC. Przetworniki szeregowo są proste ale wymagają dłuższego czasu konwersji.

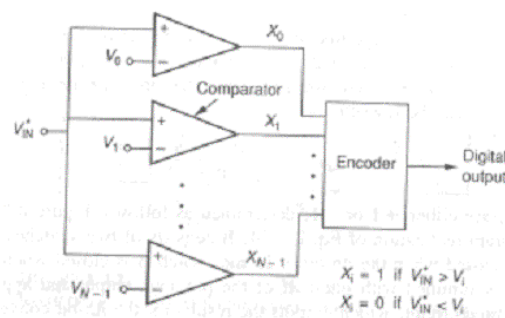
W tabeli 6.1 przedstawiono podsumowanie poszczególnych technik wykonania przetworników CA.

Tabela 6.1. Porównanie podstawowych właściwości przetworników CA.

Przetwornik CA	Rysunek	Zalety	Wady
Ze skalowaniem prądu - skalowany binarnie	6.1	szybki, nieczuły na elementy pasożytnicze	duży rozrzut wartości elementów, niemonotoniczny
Ze skalowaniem prądu - drabinka R-2R	6.2	szybki, niewielki rozrzut wartości elementów	niemonotoniczny, czuły na elementy pasożytnicze kluczy
Ze skalowaniem prądu – kaskadowy	6.4a	mała powierzchnia, monotoniczny	R_1 musi być dokładny
Ze skalowaniem prądu – master – slave	6.4b	mała powierzchnia, monotoniczny	I_1 musi być dokładne
Ze skalowaniem napięcia	6.5	monotoniczny	duża powierzchnia, wrażliwy na pojemności pasożytnicze
Ze skalowaniem ładunku	6.6	szybki	duży rozrzut wartości elementów, niemonotoniczny
Ze skalowaniem ładunku – kaskadowy	6.7	mała powierzchnia	niemonotoniczny, dzielnik musi być dokładny
Ze skalowaniem napięcia i ładunku	6.9	monotoniczny w zakresie MSB	musi być dostrajany
Ze skalowaniem ładunku i napięcia	6.11	monotoniczny w zakresie LSB	musi być dostrajany
Szeregowy z redystrybucją ładunku	6.12	prosty, mała powierzchnia	wolny, wymaga złożonego układu sterującego
Szeregowy algorytmiczny	6.15	prosty, mała powierzchnia	wolny, wymaga złożonego układu sterującego

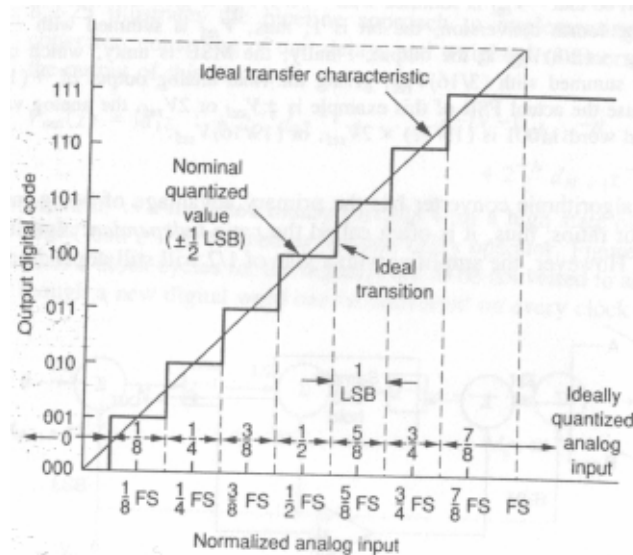
7. Scalone przetworniki analogowo – cyfrowe, klasyfikacja i podstawowe właściwości

W tym rozdziale zostaną przedstawione podstawowe właściwości przetworników analogowo - cyfrowych (AC). Celem działania przetwornika AC jest określenie na wyjściu cyfrowego słowa odzwierciedlającego wartość napięcia analogowego na jego wejściu. Przetwornik AC wymaga zazwyczaj układu próbkującego – pamiętającego na jego wejściu gdyż nie jest w stanie przetwarzać zmieniającego się napięcia wejściowego. Okazuje się, że niektóre rodzaje przetworników AC używają przetworników CA, dlatego przetworniki CA zostały przedstawione we wcześniejszym rozdziale. Przedstawione zostaną następujące rodzaje przetworników AC: z przetwarzaniem szeregowym, z kolejnym przybliżaniem, równoległy (flash) i przetworniki o polepszonych parametrach.



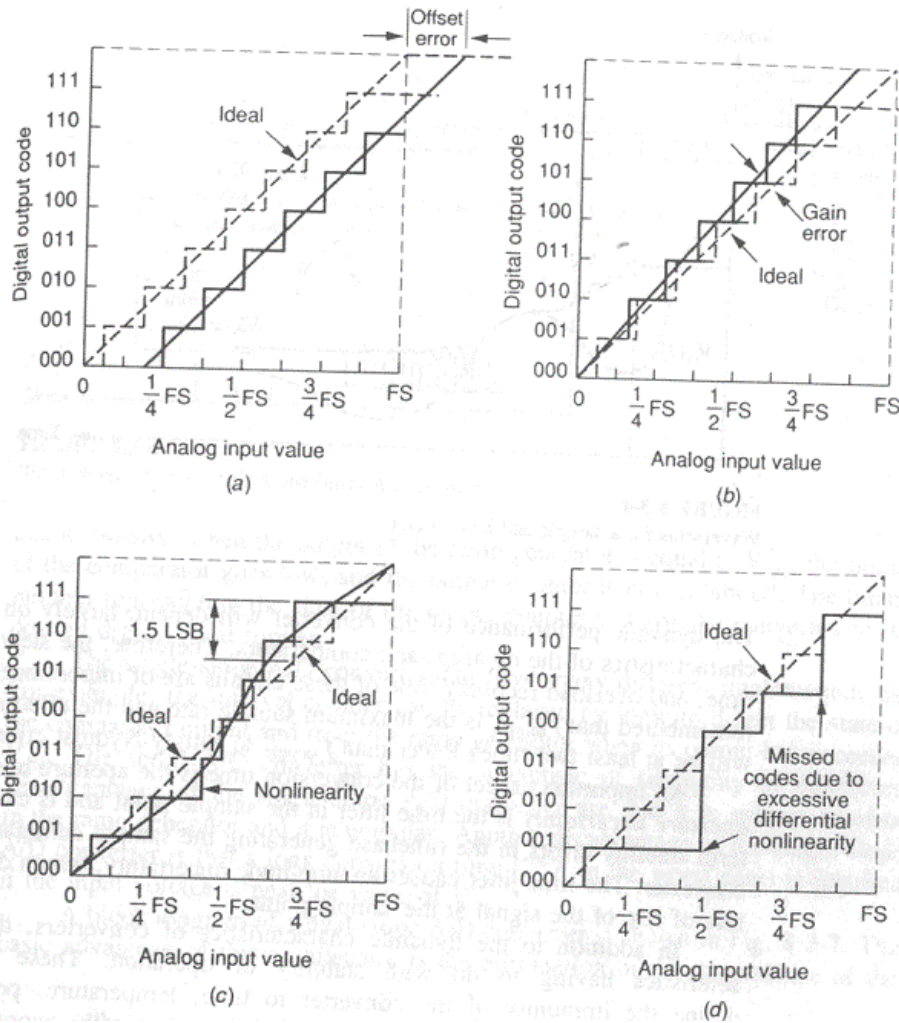
Rys. 7.1. Ogólny schemat blokowy przetwornika AC równoległego.

Na rysunku 7.1 przedstawiono podstawowy układ równoległego przetwornika AC. Napięcie oznaczone jako V_{in}^* jest napięciem po układzie próbkująco – pamiętającym. Napięcia $V_0 - V_{N-1}$ reprezentują napięcia odniesienia, które są proporcjonalne do ogólnego napięcia odniesienia V_{REF} . Po porównaniu napięć odniesienia z napięciem wejściowym sygnały $X_0 - X_{N-1}$ są dekodowane w układzie kombinacyjnym do właściwego kodu wyjściowego. Sposób przetwarzania może być zupełnie inny niż przedstawiony na rys. 7.1, jednakże zawsze następuje zamiana ciągłego zakresu napięć wejściowych na dyskretny, skończony zestaw wartości cyfrowych.



Rys. 7.2. Charakterystyka wejściowo – wyjściowa idealnego 3-bitowego przetwornika AC.

Parametry charakteryzujące przetworniki AC są niemalże identyczne jak parametry przetworników CA, zamienione są jedynie sygnały wejściowe i wyjściowe. Na rys. 7.2 przedstawiono statyczne charakterystyki przetwornika 3-bitowego. Napięcie wejściowe jest znormalizowane do wartości FS (oś pozioma), wyjściowe słowo cyfrowe przedstawione na osi pionowej. Przetwornik został tak zaprojektowany, że wyjście zmienia się na każdej nieparzystej wielokrotności wartości FS/15. Definicje parametrów takich jak rozdzielczość, zakres dynamiki, stosunek sygnał szum są takie same jak dla wcześniej przedstawianych przetworników CA. Błędy charakterystyk statycznych przedstawione są na rys. 7.3.



Rys. 7.3. Błędy statyczne przetwornika AC, (a) błąd przesunięcia, (b) błąd wzmocnienia, (c) błąd liniowości, (d) błędy braku kodów.

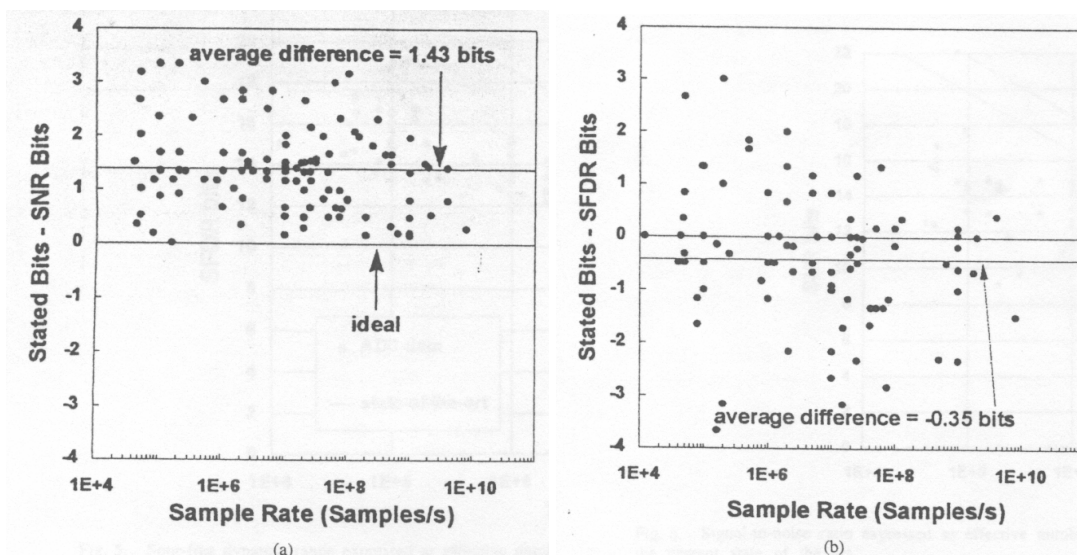
Błędy charakterystyk statycznych przetworników można równoważnie wyrazić poprzez zmniejszenie rozdzielczości przetwornika w stosunku do przetwornika idealnego. Przekształcając wzór 5.6 $SNR = 6.02N + 1.76$ do postaci wyrażającej liczbę bitów przetwornika, możemy znaleźć równoważną liczbę bitów przy założeniu iż parametr SNR został zmierzony dla rzeczywistego rozpatrywanego przetwornika:

$$N = \frac{SNR - 1.76}{6.02} \Rightarrow N_{EQ} = \frac{SNR_{MEASURED} - 1.76}{6.02} \quad (7.1)$$

gdzie: N_{EQ} jest równoważną liczbą bitów rzeczywistego przetwornika AC lub CA a $SNR_{MEASURED}$ jest zmierzoną wartością stosunku sygnał – szum dla wejściowego sygnału harmonicznego o maksymalnej amplitudzie. Innym parametrem określającym zniekształcenia nieliniowe przetwornika jest współczynnik SFDR (ang. spurious-free dynamic range). Jest to stosunek amplitud (wyrażony w [dB]) pojedynczego sygnału harmonicznego na wyjściu przetwornika do największej wartości amplitudy niepożądanego sygnału harmonicznego. Równoważna liczba bitów przetwornika AC lub CA biorąc pod uwagę współczynnik SFDR wynosi

$$N_{EQ} = SFDR / 6.02 \quad (7.2)$$

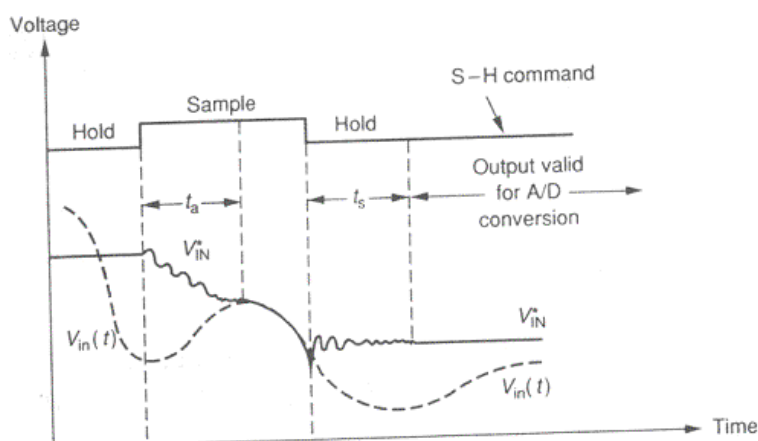
Na rysunku 7.4 przedstawiony jest rozrzut równoważnej liczby bitów przetworników wyznaczony wg powyższych wzorów dla kilkudziesięciu przetworników praktycznych.



Rys.7.4. Porównanie liczby bitów (idealnej rozdzielczości przetwornika AC) z (a) równoważną liczbą bitów wyznaczoną wg SNR, (b) równoważną liczbą bitów wyznaczoną wg SFDR.

Charakterystyki dynamiczne przetworników AC są związane głównie z szybkością przetwarzania. Czas konwersji jest czasem niezbędnym od pojawienia się wejściowego sygnału do pojawienia się prawidłowej odpowiedzi na wyjściu. Typowo przetworniki CA są szybsze niż AC (z wyjątkiem przetworników szeregowych). Dla układów przetworników AC układ próbkująco - pamiętający należy omówić bardziej szczegółowo niż było to przedstawione dla przetworników CA. Na rysunku 7.5 przedstawione są przebiegi sygnałów w układzie. Czas akwizycji t_a jest to czas (w stanie próbkowania, liczony od chwili wejścia w ten stan) niezbędny dla nadążenia układu próbkującego za napięciem wejściowym. Czas ustalania t_s jest to minimalny czas jaki musi upłynąć od momentu przełączenia w tryb pamiętania, niezbędny aby sygnał wyjściowy ustalił się na właściwym poziomie (bez wzbudzeń). Dlatego maksymalna częstotliwość pracy układu próbkująco - pamiętającego wynosi:

$$f_{sample} = \frac{1}{T_{sample}} = \frac{1}{t_a + t_s} \quad (7.3)$$

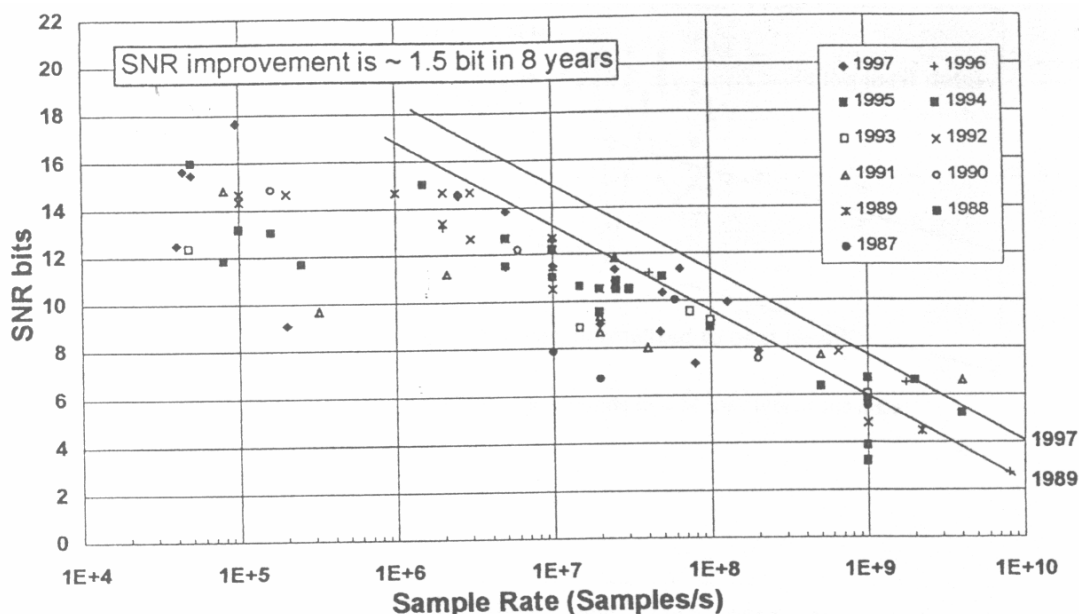


Rys. 7.5. Przebiegi czasowe napięć w układzie próbkująco - pamiętającym.

Właściwości dynamiczne układu próbkującego zależą do parametrów użytych wzmacniaczy, a w szczególności od ich współczynnika slew rate - SR.

Ważnym parametrem wpływającym bezpośrednio na czas konwersji jest niepewność szczeliny czasu próbkowania (ang. aperture uncertainty). Jest to niepewność momentu próbkowania spowodowana krótkoterminowymi błędami generacji zegara i poleceń do układu próbkująco - pamiętającego. Odchyłki czasowe punktów próbkowania (ang. time jitter) powodują niepewność w próbkowaniu chwilowej wartości sygnału a ta zależy głównie od szybkości zmian próbkowanego sygnału w punkcie próbkowania.

Postęp w dziedzinie przetworników AC nie jest tak szybki jak w innych dziedzinach rozwoju układów scalonych. Notuje się wzrost rozdzielczości przetworników o około 1.5 bita co 8 lat (dla danej częstotliwości). Na rysunku 7.6 przedstawiono kilkadziesiąt rozwiązań przetworników z podziałem na lata w których zostały opracowane.



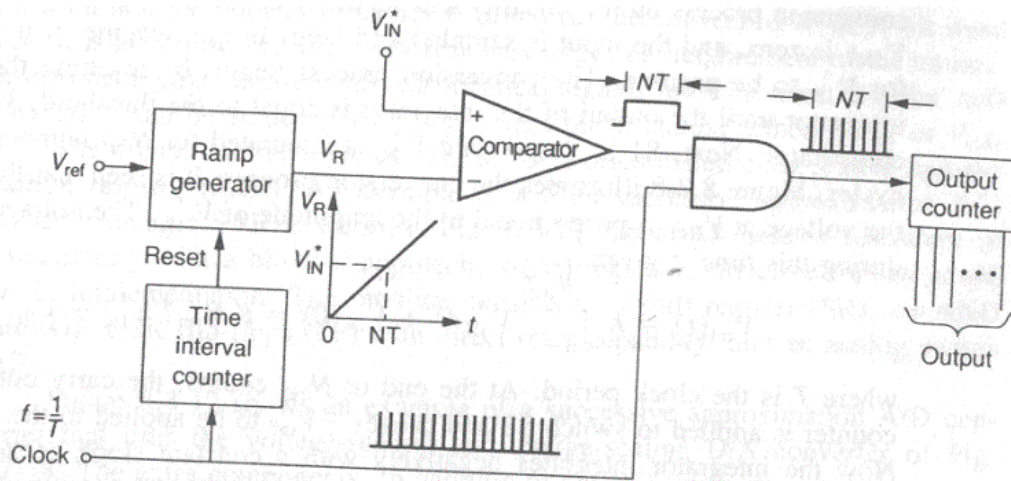
Rys. 7.5. Trend rozwojowy produkcji przetworników AC.

8. Scalona realizacja wybranych przetworników AC

8.1 Szeregowe przetworniki AC

Poniżej zostaną przedstawione dwa rodzaje przetworników szeregowych: z pojedynczym oraz z podwójnym całkowaniem. Na rysunku 8.1 przedstawiono układ przetwornika z pojedynczym całkowaniem. Zawiera on: generator przebiegu piłokształtnego (układ całkujący) licznik czasu, komparator, bramkę AND oraz licznik który wytwarza końcową wartość wyjściowego słowa binarnego. Na wstępie sygnał z układu próbkująco - pamiętającego dostarczony jest do nieodwracającego wejścia komparatora. Oba liczniki są wyzerowane. Dostarczany jest sygnał zegara i proces przetwarzania rozpoczyna się. Licznik wyjściowy zlicza sygnał o częstotliwości f tak długo jak długo napięcie wejściowe V_{IN}^* jest większe niż z generatora piłokształtnego. Po osiągnięciu napięcia na generatorze większego niż wejściowe liczba zliczonych impulsów odpowiada wartości wyjściowej słowa cyfrowego a cały proces można powtórzyć. Generator piłokształtny wykonany jest za pomocą integratora a napięcie odniesienia V_{REF} reguluje szybkość narastania sygnału wyjściowego tego generatora.

Przedstawiony przetwornik spotykany jest w wielu odmianach logiki sterującej jak również całkowanego sygnału (po modyfikacji może to także być V_{IN}^*). Jego niezaprzeczalną zaletą jest prostota, niestety ma dwie zasadnicze wady a mianowicie jest unipolarny oraz ze względu na błędy stałej czasowej integratora ma duże błędy przetwarzania. Dodatkową wadą jest długi czas przetwarzania wynoszący w najgorszym przypadku $2^N T$ dla dużych napięć wejściowych tj. równych FS.



Rys. 8.1. Schemat blokowy przetwornika AC z pojedynczym całkowaniem.

Przetwornik z podwójnym całkowaniem pozbawiony jest wady niedokładności przetwarzania związanych ze stałą czasową integratora. Jego schemat blokowy przedstawiony jest na rysunku 8.2. Wstępnie napięcie V_{INT} jest stawione na poziomie V_{TH} , następnie do wejścia integratora przyłożone jest napięcie z układu próbkująco-pamiętającego. Na przedstawionym schemacie napięcie wejściowe musi być dodatnie. Napięcie wyjściowe z integratora wzrasta proporcjonalnie do V_{IN}^* począwszy od wartości progowej V_{TH} przez czas równy $N_{REF}T$, gdzie T jest okresem zegara. Napięcie wyjściowe z integratora jest więc równe

$$V_{INT}(t) = K \int_{t=0}^{N_{REF}T} V_{IN}^*(t) dt + V(0) = KN_{REF}V_{IN}^* + V_{TH} \quad (8.1)$$

gdzie K jest odwrotnością stałej czasowej integratora. Na końcu okresu pierwszego całkowania otwarty jest klucz S1 a zamknięty klucz S2. W ten sposób do wejścia integratora dostarczone jest napięcie równe $-V_{REF}$. Integrator zmniejsza swoje napięcie proporcjonalnie do napięcia odniesienia. Równocześnie w tej fazie zliczana jest liczba okresów zegara N_{OUT} niezbędnych do obniżenia napięcia wyjściowego integratora ponownie do wartości V_{TH} . Napięcie wyjściowe w tej fazie jest równe

$$V_{INT}(t) = V(0) - K \int_{t=0}^{N_{OUT}T} V_{REF}(t) dt = +V_{TH} - KN_{OUT}V_{OUT} \quad (8.2)$$

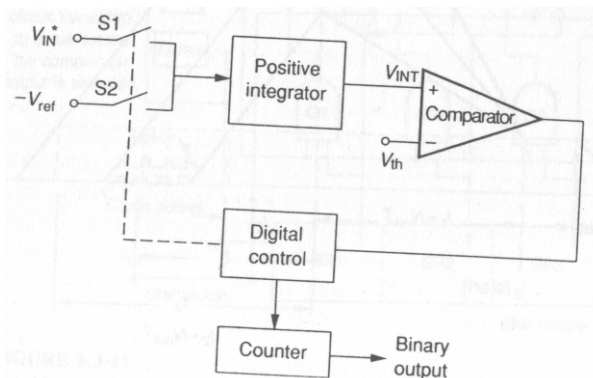
Porównując odpowiednie czynniki wyrażeń 8.1 i 8.2 otrzymujemy:

$$KN_{REF}TV_{IN}^* = KN_{OUT}V_{REF} \quad (8.3)$$

a stąd wyjściowa wartość słowa cyfrowego wynosi

$$N_{OUT} = \frac{V_{IN}^*}{V_{REF}} N_{REF} \quad (8.4)$$

Z powyższego równania wynika, że wartość N_{OUT} jest ułamkiem V_{IN}^*/V_{REF} wartości N_{REF} . Wartość wyjściowa N_{OUT} nie jest funkcją napięcia progowego, stałej czasowej integratora ani częstotliwości zegara. Z tego względu jest to bardzo dokładna metoda konwersji. Jedyną wadą jest fakt długiego czasu przetwarzania, który wynosi od $T2^N$ do $2T2^N$, gdzie N jest liczbą bitów przetwornika.

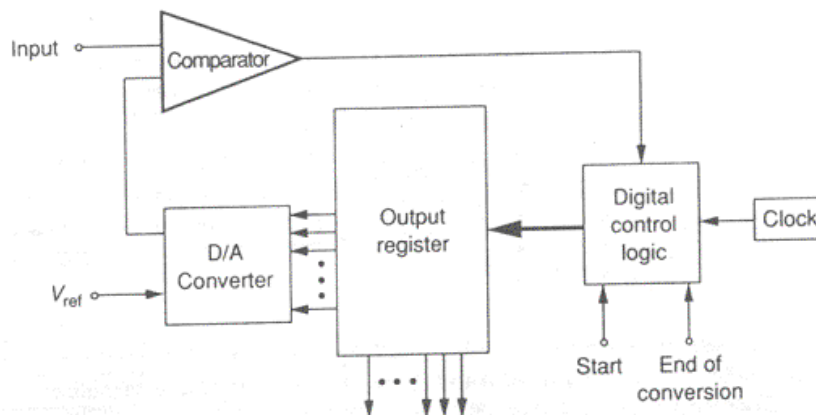


Rys. 8.2. Przetwornik AC z podwójnym całkowaniem.

Przedstawione przetworniki osiągają rozdzielczość 12 bitów przy częstotliwości przetwarzania około 100Hz.

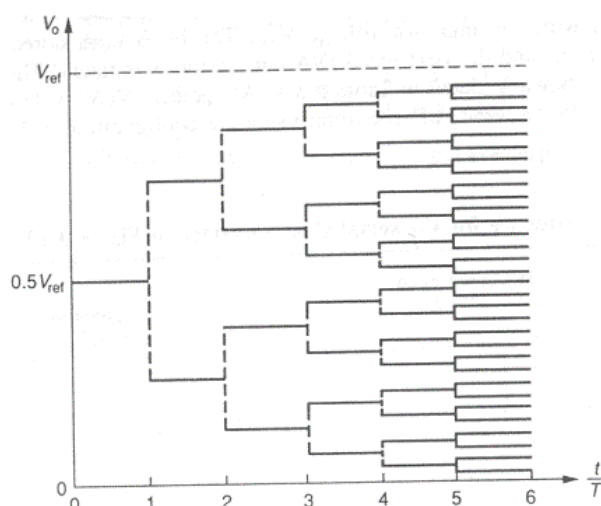
8.2 Przetwornik AC z sukcesywnym przybliżaniem.

Drugą rozpatrywaną kategorią są przetworniki AC z sukcesywnym przybliżaniem. Ten typ układu wymaga N cykli zegara do wykonania konwersji. Częścią składową takiego przetwornika jest przetwornik CA objęty pętlą sprzężenia zwrotnego.



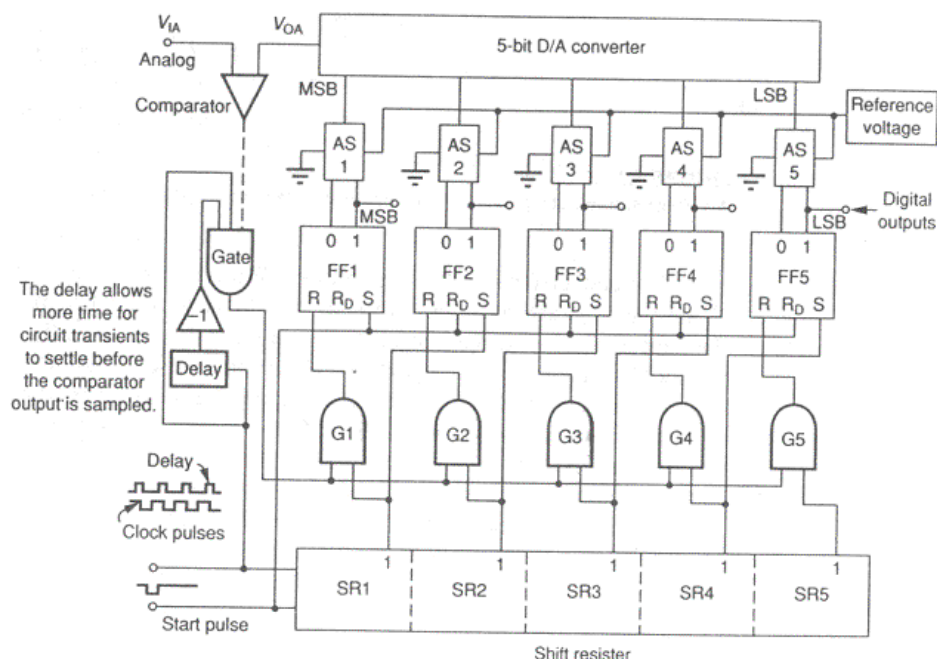
Rys. 8.3. Architektura przetwornika z sukcesywnym przybliżaniem.

Na rysunku 8.3 przedstawiono architekturę przetwornika AC z sukcesywnym przybliżaniem. Cały układ składa się z komparatora, przetwornika CA i cyfrowego układu sterującego. Celem układu cyfrowego jest ustalanie poszczególnych bitów wyjściowego słowa cyfrowego w oparciu o bieżące nastawy przetwornika CA i sygnał z komparatora. W celu ilustracji zasady działania założmy że przetwornik AC jest unipolarny. Cykl przetwarzania zaczyna się poprzez spróbkowanie i zapamiętanie sygnału w układzie S/H. W kolejnym kroku układ cyfrowy ustawia na wejściu przetwornika CA bit MSB równy 1 a pozostałe równe 0. Przetwornik CA wytwarza napięcie o wartości $0.5V_{REF}$, jest ono porównane z wartością wejściową V_{IN}^* . Jeśli wyjście komparatora jest w stanie wysokim wówczas bit MSB jest ustalony na 1, w przeciwnym przypadku na 0. Następnie kolejny bit (MSB-1) jest ustawiany na 1, wartość z przetwornika CA jest porównywana z wejściową i jeśli komparator daje stan wysoki bit MSB-1 jest zakwalifikowany jako 1 a w przeciwnym przypadku jako 0. Proces powtarza się aż do bitu LSB, czyli N -krotnie. Na rysunku 8.4 przedstawiono w sposób graficzny możliwe przejścia podczas procesu przetwarzania. Możliwa jest bipolarna wersja przetwornika AC przy zastosowaniu ujemnej wartości napięcia odniesienia V_{REF} .



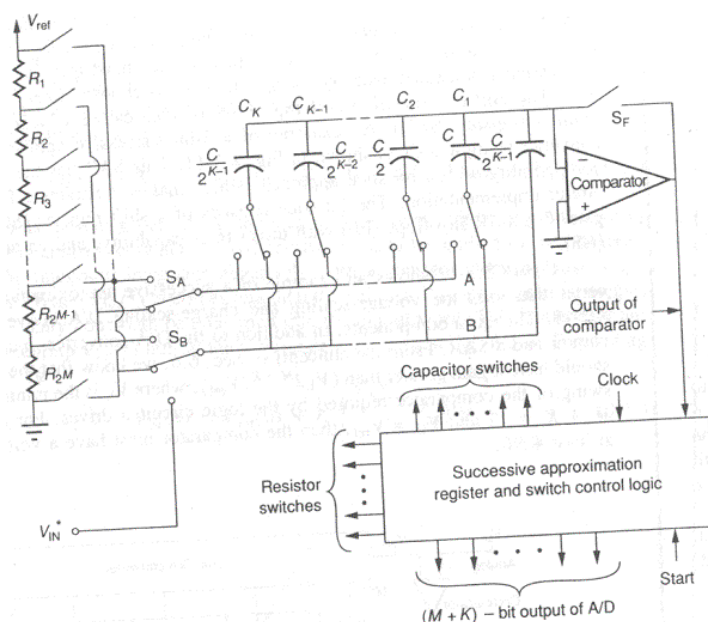
Rys. 8.4. Proces przybliżania wartości wyjściowej dla przetwornika z rys. 8.3.

Cyfrowy układ sterujący często nazywany jest rejestrem aproksymującym (SAR). Przykład przetwornika z 5-cio bitowym rejestrem SAR przedstawiony jest na rysunku 8.5. Rejestr SAR jest regularny i składa się z wielokrotnie powtarzanych zestawów: rejestru przesuwającego (SR), bramki AND (G), przerzutnika RS (FF) oraz analogowego klucza (AS).



Rys. 8.5. Pięciobitowy przetwornik z sukcesywnym przybliżaniem wykorzystujący rejestr aproksymacyjny.

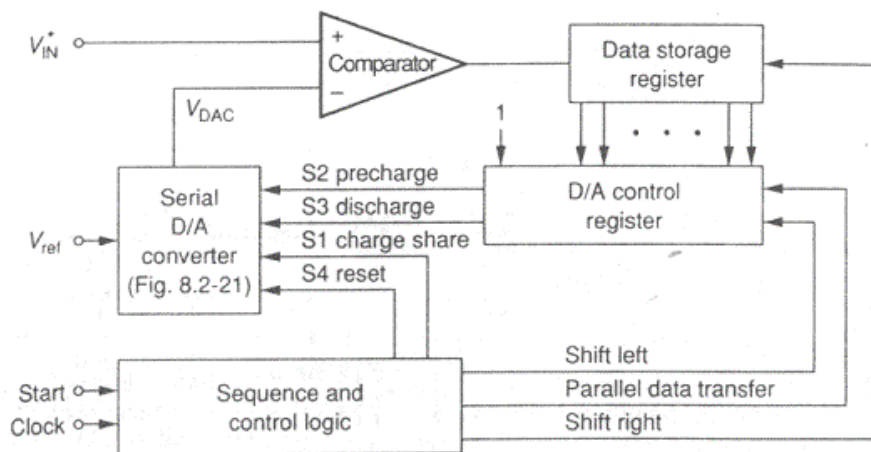
Na rysunku 8.6 przedstawiono przetwornik AC z użyciem przetwornika CA z rysunku 6.8. Dodatkowymi elementami w stosunku do samego przetwornika CA są komparator oraz rejestr aproksymacyjny. Komparator musi mieć wzmocnienie większe niż $V_L 2^{M+K}/V_{REF}$, gdzie V_L jest minimalną zmianą napięcia wyjściowego komparatora zapewniającą zmianę stanu logicznego dołączonego do niego układu cyfrowego. Na przykład dla przetwornika CA 12-to bitowego i $V_L = V_{REF}$ komparator musi mieć wzmocnienie większe niż 4096 [V/V].



Rys. 8.5. Przetwornik z sukcesywnym przybliżaniem wykorzystujący przetwornik CA z podziałem napięcia i ładunku.

Przebieg procesu przetwarzania układu z rysunku 8.6 jest następujący. Klucz S_F jest zwierany oraz równocześnie dolne okładki kondensatorów $C_1 - C_K$ są połączone przez klucz S_B do napięcia V_{IN}^* . Napięcie przechowywane na kondensatorach jest równe wartości aktualnej V_{IN}^* pomniejszonej o napięcie progowe komparatora, co w dalszym przetwarzaniu powoduje eliminację napięcia niezrównoważenia komparatora jako źródła dodatkowego błędu przetwarzania. Ze względu na zwarcie wejścia odwracającego komparatora z jego wyjściem, dla zapewnienia stabilności, komparator musi być skompensowany. Po otwarciu klucza S_F w ramach kolejnego przybliżania ustalane są bity z drabinki rezystorowej. Następnie szyny A oraz B są łączone z rezystorem określającym bity słowa wyjściowego ustalonego „rezystywnie”. W końcu następuje sukcesywne przybliżenie wartości wyjściowej słowa cyfrowego w oparciu o część z podziałem ładunku. Przetwornik daje możliwość uzyskania rozdzielczości w zakresie 12-tu bitów i czasów przetwarzania rzędu 50ms.

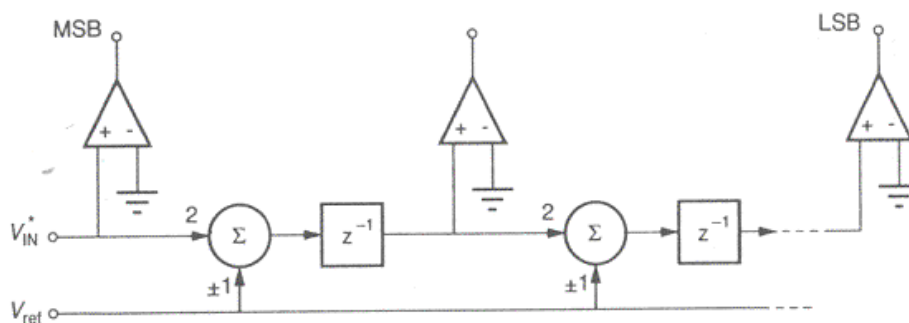
Na rysunku 8.7 przedstawiony jest przetwornik z sukcesywnym przybliżaniem wykorzystujący przetwornik CA szeregowy z redystrybucją ładunków (z rysunku 6.12). Szeregowy przetwornik z redystrybucją ładunków wymaga $2N$ cykli do przetworzenia dowolnego ciągu, jeśli jednak wiadomo, że P najmniej znaczących bitów jest równa zero można ich nie przetwarzać a całkowity czas niezbędny do przetworzenia będzie równy $2(N-P)$. W przypadku przetwornika AC z sukcesywnym przybliżaniem w pierwszym kroku tylko MSB jest równe 1 a pozostałe bity 0, w drugim kroku tylko dwa MSB są niezerowe, itd. Umożliwia to skrócenie ogólnego czasu przetwarzania przetwornika AC opartego na przetworniku z redystrybucją ładunku z początkowych $2N \cdot N$ cykli do $N(N+1)$. Wymaga to zastosowania inteligentnego sterowania cyfrowego procesem przetwarzania.



Rys. 8.6. Szeregowy przetwornik AC wykorzystujący szeregowy przetwornik CA z rys. 6.10.1.

8.3 Szeregowy algorytmiczny przetwornik AC.

Algorytmiczny przetwornik AC przedstawiony jest na rysunku 8.7. N bitowy przetwornik składa się z N stopni przetwarzających a każdy z nich zawiera mnożenie przez 2, sumator oraz komparator. Wyjścia komparatorów tworzą wyjściowy ciąg binarny. Znak odejmowania bądź dodawania jest wybierany odpowiednio dla wysokiego i niskiego poziomu wyjścia komparatora poprzedzającego dany sumator. Proces przetwarzania jest przedstawiony w przykładzie 8.1.



Rys. 8.7. Potokowa realizacja algorytmicznego przetwornika AC.

Przykład 8.1. Ilustracja procesu przetwarzania algorytmicznego przetwornika AC. Załóżmy, że $V_{IN}^* = 1.5V$, $V_{REF} = 5V$ a przetwornik jest 4-ro bitowy. Należy wyznaczyć wartość wyjściowego ciągu cyfrowego dla wyżej podanych parametrów oraz sprawdzić poprawność przetworzenia.

Rozwiązanie: Ponieważ $V_{IN}^* = 1.5V > 0$ więc MSB=1. Następny bit powstaje na skutek porównania wartości $2 \cdot 1.5 - 5 = -2$ z zerem czyli bit jest równy zeru. Kolejna porównywana wartość wynosi $-2 \cdot 2 + 5 = 1$ więc wartość odpowiadającego bitu jest równa 1. Następnie otrzymujemy $2 \cdot 1 - 5 = -3$ więc bit jest równy 0. Wartość wynikowa jest równa 109. Odpowiadająca temu wartość analogowa wynosi

$$V_{ANALOG} = V_{REF} (b_1 2^{-1} + b_2 2^{-2} + b_3 2^{-3} + b_4 2^{-4})$$

gdzie: b_i jest równe +1 jeśli odpowiadający bit jest równy 1 lub -1 jeśli odpowiadający bit jest zerowy. Podstawiając wartości liczbowe otrzymujemy:

$$V_{ANALOG} = 5V \left(\frac{1}{2} - \frac{1}{4} + \frac{1}{8} - \frac{1}{16} \right) = 1.5625V$$

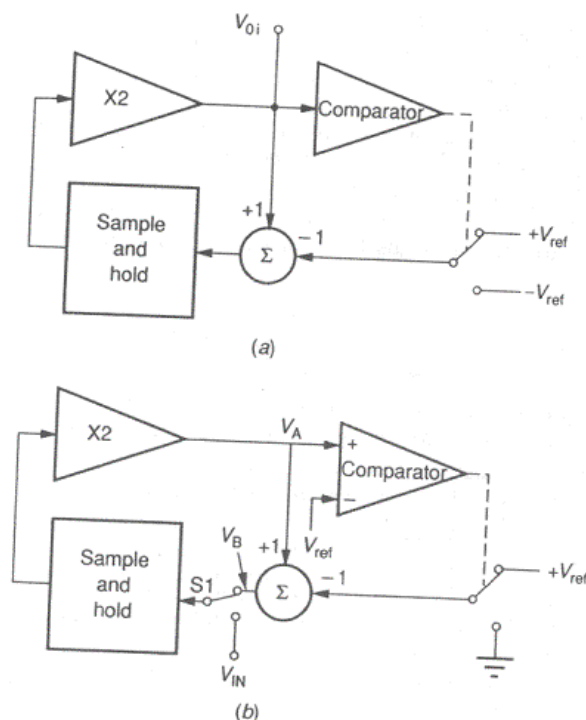
Przy zwiększaniu liczby bitów wartość V_{ANALOG} powinna dążyć do 1.5V.

Wadą przetwornika algorytmicznego jest potrzeba aż N cykli do uzyskania wyniku końcowego. W realizacji potokowej z rysunku 8.8 możliwe jest jednak ciągle podawanie wartości przetwarzanej na wejście przetwornika i ich odbieranie co jeden cykl z opóźnieniem co N cykli. Mnożenie przez 2 musi być wykonywane z dokładnością do $\pm 0.5LSB$ co jest główną trudnością w osiągnięciu dużych rozdzielczości.

Iteracyjna, zredukowana wersja przetwornika przedstawiona jest na rysunku 8.8. Analogowe napięcie wyjściowe z poszczególnych stopni można opisać równaniem

$$V_{O,i} = (2V_{O,i-1} - b_i V_{REF}) z^{-1} \quad (8.5)$$

gdzie: b_i jest równe +1 dla i -tego bitu równego 1 i równe -1 w przeciwnym przypadku. W realizacji z rysunku 8.9(b) napięcie $-V_{REF}$ zostało zastąpione masą.



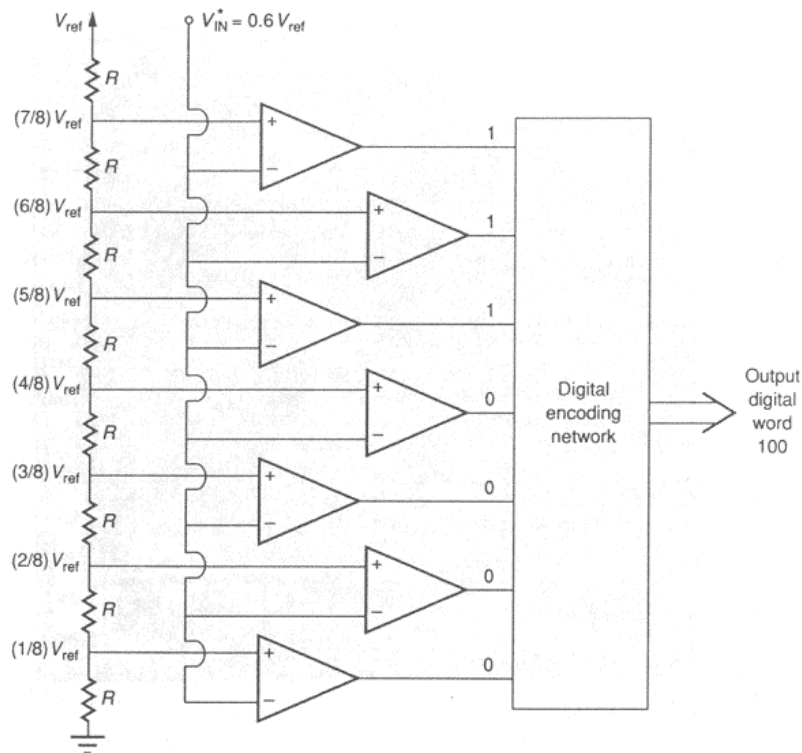
Rys. 8.8. Realizacja iteracyjnego algorytmicznego przetwornika AC (b) na podstawie bloku mnożąco – sumująco – opóźniającego (a).

Iteracyjne algorytmiczne przetworniki AC mogą być budowane przy użyciu małej liczby precyzyjnych podukładów. Użycie powierzchni jest więc nieduże. Źródłami błędów przetwarzania są: małe wzmocnienie WO, wstrzykiwanie ładunków z kluczy MOS, napięcia niezrównoważeń WO oraz zależności temperaturowe realizowanych kondensatorów. Praktycznie realizowane są przetworniki o rozdzielczości 12-tu bitów z różnicowym i całkowitym błędem liniowości równymi odpowiednio 0.8LSB oraz 1.5LSB dla częstotliwości przetwarzania około 4kHz (rok 1990).

Przetworniki z sukcesywnym przybliżaniem dają możliwość realizacji rozdzielczości w zakresie 8 – 12 bitów i częstotliwości przetwarzania rzędu $10^4 - 10^5$. Rozdzielczość można poprawić poprzez zastosowanie trybowania.

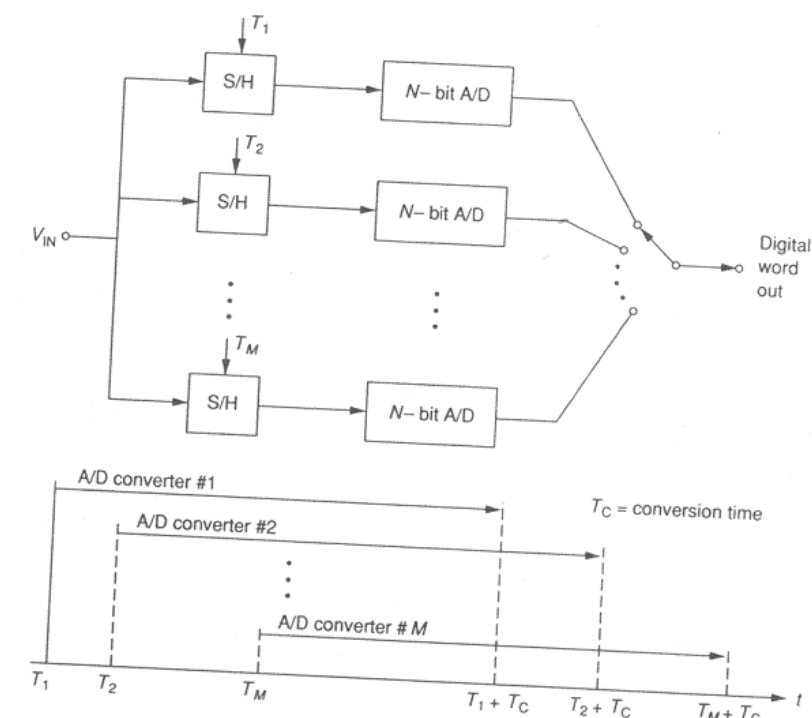
8.4 Równoległe przetworniki AC.

W wielu przypadkach niezbędny jest krótszy czas konwersji niż możliwy do zrealizowania przy użyciu poprzednio opisanych metod. To doprowadziło do rozwoju bardzo szybkich przetworników realizowanych w technice równoległej. Czas konwersji takiego przetwornika wynosi jeden cykl składający się zazwyczaj z czasu ustalania i czasu przetwarzania. Niektóre architektury wymagają użycia więcej niż jednego cyklu ale za to dają mniejsze zużycie powierzchni i w dalszym ciągu są znacznie szybsze od rozwiązań z sukcesywnym przybliżaniem. Inną techniką zwiększenia szybkości przetwarzania jest stosowanie elementów o lepszych parametrach czasowych. Poniżej zostaną przedstawione rozwiązania przetworników typu: równoległy, z przeplotem czasu oraz dwukrokowy.



Rys. 8.9. Trzybitowy równoległy przetwornik AC.

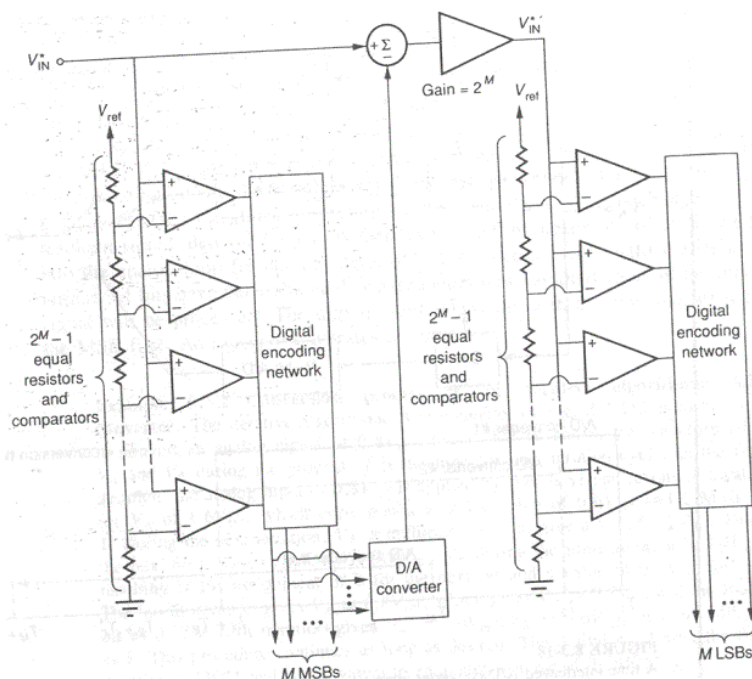
Na rysunku 8.10 przedstawiony jest 3 bitowy przetwornik równoległy (FLASH). Napięcie V_{REF} jest podzielone na 8 równych wartości, każda z nich jest podana na dodatnie wejście komparatora. Wyjścia komparatorów są dołączone do układu dekodującego. Analogowy sygnał wejściowy jest przetwarzany na równoważną postać cyfrową w jednym cyklu, zawierającym zwykle dwie fazy. W pierwszej fazie napięcie analogowe jest spróbkowane i dostarczone do wejścia przetwornika. Podczas drugiej fazy sieć cyfrowa dokonuje zdekodowania poziomów wyjściowych komparatorów do właściwej wartości słowa cyfrowego. Czas przetwarzania ograniczony jest powyższymi fazami. Typowe wartości częstotliwości próbkowania mogą dochodzić do 20MHz dla technologii CMOS i 100MHz dla BJT (1990) co daje odpowiednio czasy 50ns i 10ns. Z tego względu układ próbkujący – pamiętający może stać się głównym ograniczeniem szybkości przetwarzania. Innym problemem jest to iż z każdym wzrostem rozdzielczości o 1 bit podwaja się liczba niezbędnych komparatorów. To ograniczenie powoduje, że praktycznie spotykane są przetworniki FLASH o rozdzielczości do 8-iu bitów.



Rys. 8.10. Przetwornik z przeplotem czasowym.

Zmniejszenie czasu przetwarzania można uzyskać przez zastosowanie przeplotu czasowego jak na rysunku 8.10. M przetworników połączonych jest równolegle, każdy z nich przetwarza wejściowy sygnał spróbkowany w innej szczelinie czasowej.

Połączenie dwóch przetworników równoległych w sposób szeregowy, jak na rysunku 8.12, pozwala na zachowanie rozsądnej wartości użytej powierzchni oraz nieznacznego pogorszenia czasu konwersji. Takie rozwiązanie jest często nazywane przetwarzaniem potokowym, szczególnie jeśli liczba stopni połączonych szeregowo jest większa niż 2. W naszym przypadku przedstawiony jest przetwornik o wynikowej rozdzielczości równej $2M$, użyte tu są 2 przetworniki M -bitowe. Najpierw przetworzonych jest M bardziej znaczących bitów a następnie pozostała część M bitów LSB. Dla uzyskania rozdzielczości 2^{2M} niezbędnych jest $2^{M+2}-2$ komparatorów. Czas przetwarzania wydłuża się do 2 cykli.



Rys. 9-10.1. Równoległo - szeregowy przetwornik AC.

Do tego punktu zostały przedstawione główne techniki wykonania przetworników AC. Są to: szeregowy, sukcesywnego przybliżania oraz równoległy. W tabeli 8.1 przedstawiono porównanie parametrów poszczególnych rodzajów przetworników AC.

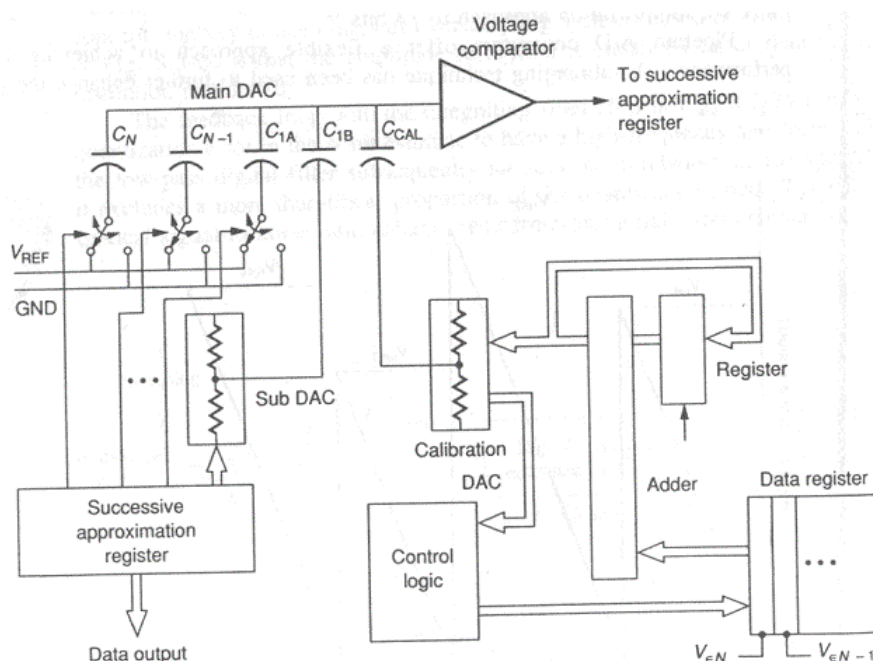
Tabela 8.1. Porównanie parametrów przetworników AC

Typ przetwornika	Charakterystyka
Szeregowy	1-100 przetworzeń/sek., 12-14 bitów dokładności, wymagane stabilne napięcie odniesienia
Sukcesywne przybliżanie	9.000-100.000 przetworzeń/sek., 8-10 bitów dokładności bez dostrajania, 12-14 z dostrajaniem
Równoległy	1.000.000 – 15.000.000 przetworzeń/sek., 7-8 bitów dokładności, duża powierzchnia
Zaawansowane przetworniki AC	7.000-1.000.000 przetworzeń/sek., 12-18 bitów dokładności

8.5 Zaawansowane rozwiązania przetworników AC .

Ostatnie postępy w konstrukcji przetworników AC zaowocowały poprawą ich jakości poprzez poprawę uzyskiwanej rozdzielczości, szybkości próbkowania oraz wykorzystywanej powierzchni. Poniżej przedstawione zostaną trzy rozwiązania przetworników AC a mianowicie: samokalibrujące, z przetwarzaniem potokowym oraz z nadpróbkowaniem.

Przetwornik samokalibrujący jest to taki przetwornik, który używa cyklu kalibracji w celu dostrojenia swoich charakterystyk do charakterystyk przetwornika idealnego. Schemat blokowy takiego przetwornika przedstawiony jest na rysunku 8.13. Składa się on z N-bitowego przetwornika CA z podziałem ładunku, M-bitowego przetwornika CA z podziałem napięcia (zwanego sub-przetwornikiem CA) oraz kalibrującego przetwornika CA. Rozdzielczość przetwornika kalibrującego musi być o kilka bitów lepsza niż sub-przetwornika CA. Układ cyfrowego sterowania zarządza kluczami kondensatorów podczas cyklu kalibrującego oraz zapamiętuje współczynniki korekcji nieliniowości w rejestrach danych. Uzyskiwane rozdzielczości są rzędu 15-16 bitów z błędem przesunięcia mniejszym od 1LSB, zakres dynamiki 90dB, nieliniowość mniejsza niż 0.25LSB czas przetwarzania około 50us.



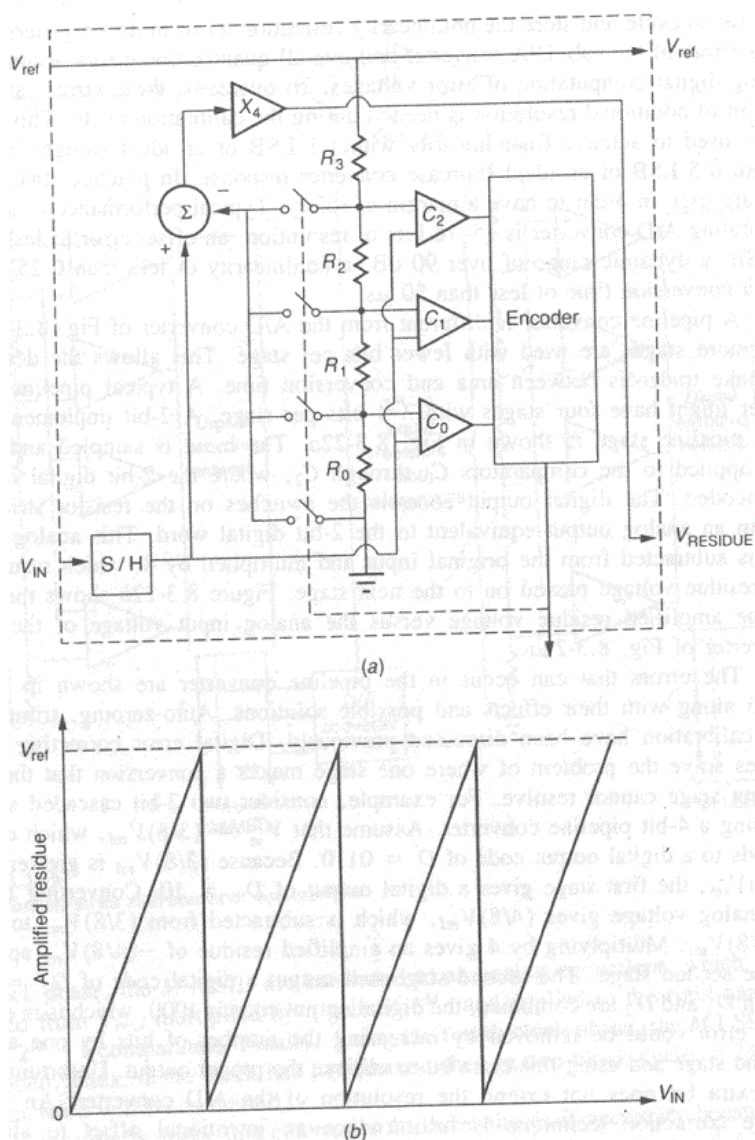
Rys. 8.13. Schemat blokowy samokalibrującego przetwornika AC.

Przetwornik potokowy zawiera wiele stopni realizujących identyczną funkcję. To umożliwia projektantowi uzyskanie kompromisu pomiędzy użytą powierzchnią oraz czasem przetwarzania.

Typowy przetwornik potokowy składa się z około 4 stopni, w każdym z nich jest 3 – 4 bitowy. Przykładowy dwubitowy blok przedstawiony jest na rysunku 8.14(a). Napięcie wejściowe jest dostarczone do komparatorów C_0 – C_2 , gdzie dokonuje się konwersji dwubitowej. Napięcie analogowe odpowiadające zdekodowanemu stanowi jest następnie odejmowane od napięcia wejściowego. Różnica jest następnie mnożona przez 4 i stanowi wejście do kolejnego stopnia przetwarzającego. Na rysunku 8.14(b) przedstawiona jest charakterystyka przejściowa omawianego stopnia, gdzie jako wyjście traktuje się wzmacnione napięcie pozostałe z działania układu odejmującego $V_{RESIDUE}$.

Błędy, które mogą się pojawić w przetworniku potokowym, oraz możliwe metody ich eliminacji przedstawione są w tabeli 8.2. W przypadku gdy błąd wprowadzany w danym stopniu jest niemożliwy do usunięcia w stopniu kolejnym można zastosować cyfrową korekcję błędów.

Parametry przetworników potokowych zależą od technologii wykonania. W CMOS uzyskuje się rozdzielczości rzędu 13-14 bitów, czas przetwarzania 5-10us, i około 10-30mW pobieranej mocy. Przetworniki bipolarne mają około 10 bitów rozdzielczości, czas konwersji około 200ns i 100-500mW niezbędnej mocy.

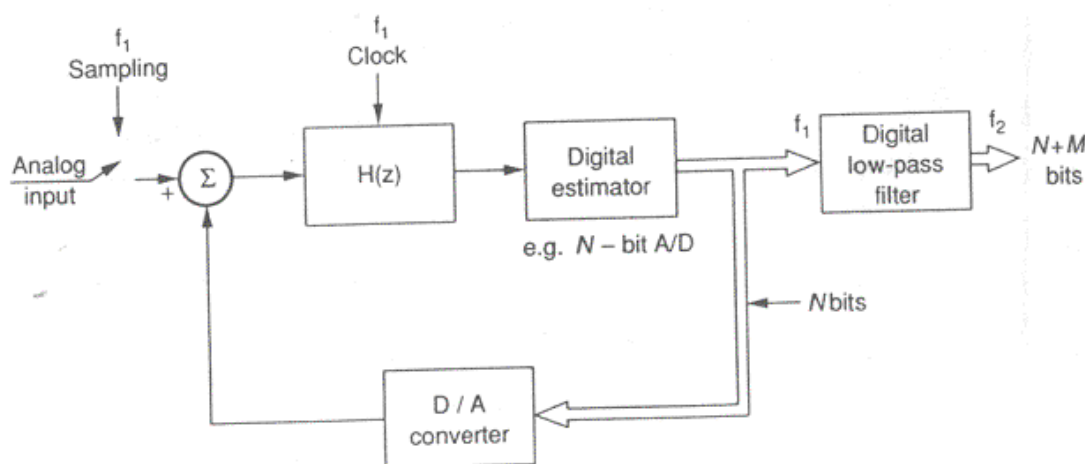


Rys. 8.14. Implementacja bwubitowego stopnia przetwornika potokowego (a) oraz idealna charakterystyka przenoszenia (b).

Tabela 8.2. Analiza błędów statycznych potokowych przetworników AC.

Układ	Błędy	Efekty	Rozwiązanie
Próbkująco - pamiętający	przesunięcie	przesunięcie, nieliniowość	auto – zerowanie cyfrowa korekcja
Przetwornik AC	przesunięcie nieliniowość	przesunięcie nieliniowość nieliniowość	auto – zerowanie cyfrowa korekcja cyfrowa korekcja
Przetwornik CA	nieliniowość	nieliniowość	dostrajanie/kalibracja
Wzmacniacz międzystopniowy	przesunięcie błąd wzmacnienia	nieliniowość nieliniowość	auto – zerowanie cyfrowa korekcja

Przetworniki z nadpróbkowaniem oferują możliwość wymiany rozdzielczości na czas przetwarzania oraz mają zmniejszoną liczbę precyzyjnych bloków analogowych. Podstawowa architektura przetwornika z nadpróbkowaniem przedstawiona jest na rysunku 8.15. W jej skład wchodzi: zegarowana ścieżka sprzężenia zwrotnego (która wytwarza zgrubną wartość napięcia zwrotnego oscylującą wokół wartości wejściowej) oraz filtr cyfrowy (który uśrednia wartość zgrubną w celu uzyskania dokładniejszej wartości reprezentacji cyfrowej). Uśrednienie przy użyciu filtru cyfrowego jest dokładniejsze dla częstotliwości o wartości niższej (decimated) niż częstotliwość pracy pętli.



Rys. 8.15. Architektura ogólna przetwornika AC z nadpróbkowaniem.

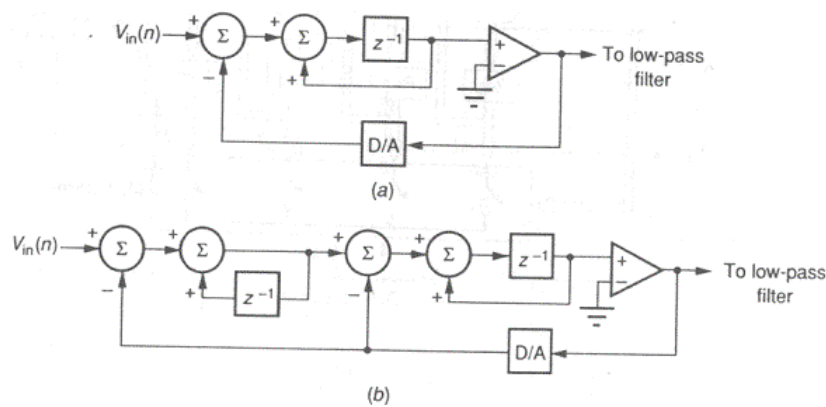
Pętla sprzężenia zwrotnego z filtrem $H(z)$ powoduje, że błąd próbkowania próbek N -bitowych zawiera widmo o szerokim paśmie. Podczas eliminacji tych składowych przy użyciu filtru cyfrowego wzrasta stosunek sygnał szum. Teoretyczne polepszenie rozdzielczości (w dB) przetwornika, z inicjalnych N bitów wynosi:

$$\Delta S/N = 9L - 5.2 \text{ dB} \quad (8.6)$$

gdzie: L jest liczbą oktaów wartości nadpróbkowania, $H(z)$ jest filtrem pierwszego rzędu. Jeśli $H(z)$ jest filtrem drugiego rzędu wówczas wzrost rozdzielczości wynosi

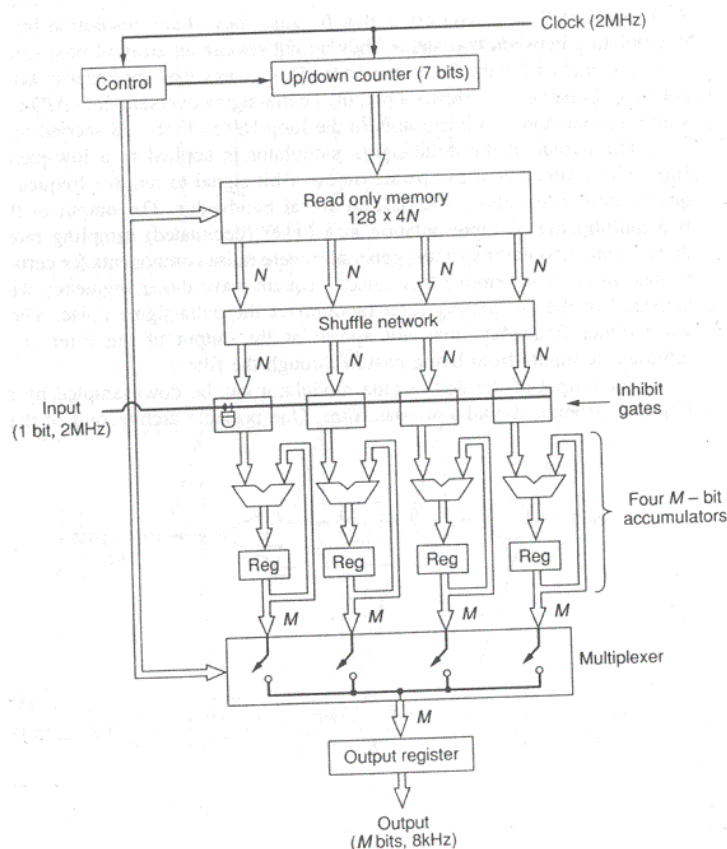
$$\Delta S/N = 15L - 13 \text{ dB} \quad (8.7)$$

Przy założeniu 128-krotnego nadpróbkowania wzrost dynamiki wynosi 56.8 dB dla układu pierwszego rzędu oraz 92 dB dla drugiego rzędu.



Rys. 8.15. Modulatory delta-sigma z wykorzystaniem jednego (a) i dwóch integratorów (b) w dziedzinie z .

Specjalnym przypadkiem jest przetwornik dla którego zgrubne przetwarzanie wykonywane jest jednobitowo, tj. dla $N=1$. Taki układ nazywany jest modulatorem delta-sigma i jest przedstawiony na rysunku 8.15. Istotną zaletą takiego rozwiązania jest całkowity brak precyzyjnie dopasowanych elementów w estymatorze i przetworniku CA. Oznacza to całkowity brak błędów liniowości.



Rys. 8.16. Architektura filtru o skończonej odpowiedzi impulsowej (1024impl). Wejście filtru jednobitowe wyjście M -bitowe (15), współczynniki filtru N -bitowe (6).

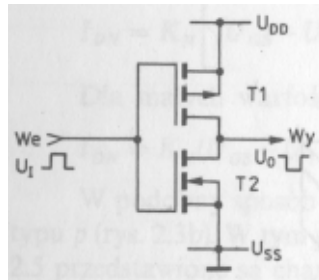
Wyjście modulatora delta – sigma jest dołączone do wejścia filtru cyfrowego. Filtr ten realizuje dolnoprzepustową funkcję przenoszenia. Sygnał wejściowy jest jednobitowy. Sygnał wyjściowy jest wielobitowy ale o mniejszej częstotliwości. Podstawowy układ modulatora delta – sigma generuje znaczne składniki szumowe dla pewnych wybranych wartości wejściowych. Aby tego uniknąć do sygnału wejściowego dodaje się falę prostokątną o częstotliwości leżącej w paśmie zaporowym filtru cyfrowego. Przykład realizacji takiego filtru przedstawiony jest na rysunku 8.16. Współczynnik decymacji wynosi 1:255. Filtr ma symetryczną 1024 punktową odpowiedź impulsową, w związku z tym w pamięci ROM jest przechowywanych tylko 512 współczynników, każdy z nich reprezentowany jest w postaci N bitowej. Współczynniki filtru muszą być tak wybrane

aby usuwać szum kwantowania oraz realizować filtrowanie antyaliasingowe. Ze względu na to, że realizowany filtr jest typu dolnoprzepustowego o skończonej odpowiedzi impulsowej, jego charakterystyka nie jest czuła na zaokrąglenia współczynników filtru i wybór $N=6$ jest wystarczający.

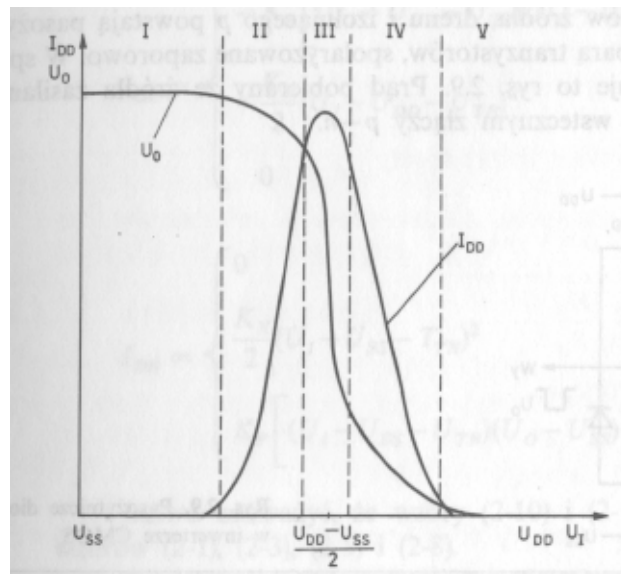
Typowe parametry przetworników AC CMOS delta-sigma to rozdzielczość rzędu 14-16 bitów, częstotliwość próbkowania 1-20kHz oraz pobór mocy rzędu 10-30mW. Osiągalny jest zakres dynamiki wynoszący ponad 96dB. Głównym czynnikiem ograniczającym parametry przetwornika jest szum.

9. Podstawowe układy cyfrowe: bramki i przerzutniki.

Bramki układów CMOS oparte są w swej budowie na inwerterze, którego schemat przedstawiony jest na rysunku 9.1.



Rys. 9.1. Schemat elektryczny inwertera CMOS [4].



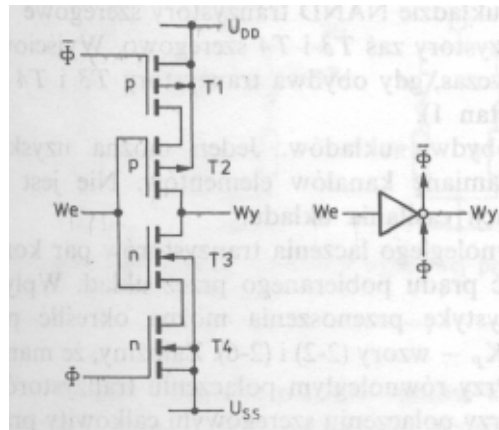
Rys. 9.2. Charakterystyki statyczne inwertera CMOS [4].

Nr obszaru	Zakres napięcia wejściowego	Stan tranzystora	
		T1	T2
I	$0 \leq U_I \leq U_{TN}$	nienasycony	zablokowany
II	$U_{TN} \leq U_I \leq U_O - U_{TP} $	nienasycony	nasycony
III	$U_O - U_{TP} \leq U_I \leq U_O + U_{TN}$	nasycony	nasycony
IV	$U_O + U_{TN} \leq U_I \leq U_{DD} - U_{TP} $	nasycony	nienasycony
V	$U_{DD} - U_{TP} \leq U_I \leq U_{DD}$	zablokowany	nienasycony

Tabela 9.1. Obszary pracy tranzystorów w inwerterze CMOS [4].

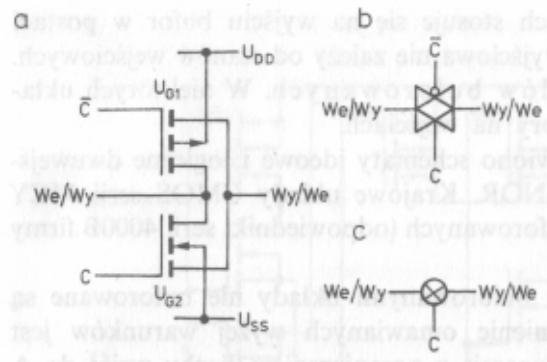
Zgodnie z tablicą 9.1 w czasie zmian napięcia wejściowego od wartości zerowej do równej napięciu zasilającemu tranzystory MOS przechodzą przez 5 kolejnych obszarów. W obszarze III występuje największe nachylenie charakterystyki przejściowej oraz największa wartość prądu zasilającego.

Inwerter kluczowany jest wzbogacony o dwa tranzystory MOS. Umożliwia to realizację wysokiej impedancji na wyjściu bramki, przez co nadaje się do sterowania szynami dwukierunkowymi.

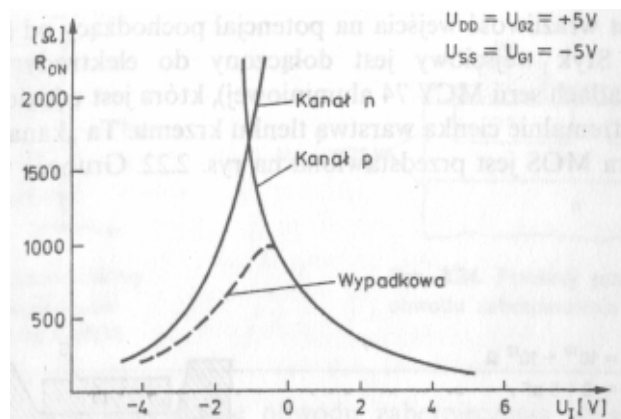


Rys. 9.3. Inwerter kluczowany i jego symbol graficzny [4].

Na rysunku 9.4 przedstawiona jest bramka transmisyjna oraz jej symbol graficzny. Bramka taka może zostać wykorzystana jako element sprzęgający zarówno sygnały analogowe jak i cyfrowe. Najważniejszym jej parametrem jest rezystancja szeregowa stanowiąca równoległe połączenie rezystancji tranzystora NMOS i PMOS. Wartość tej rezystancji zależy od poziomu napięcia, który jest do bramki dostarczany. Przykładowa zależność rezystancji bramki od wartości napięcia wejściowego przedstawiona jest na rysunku 9.4.

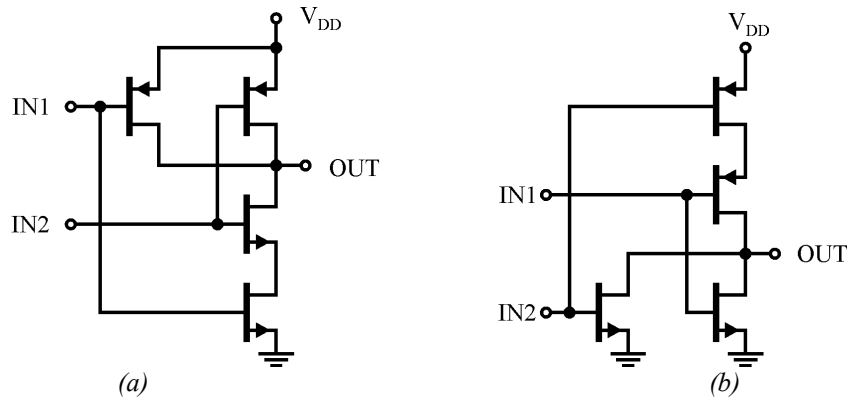


Rys. 9.4. Bramka transmisyjna (a) i stosowane jej symbole graficzne (b) i (c) [4].



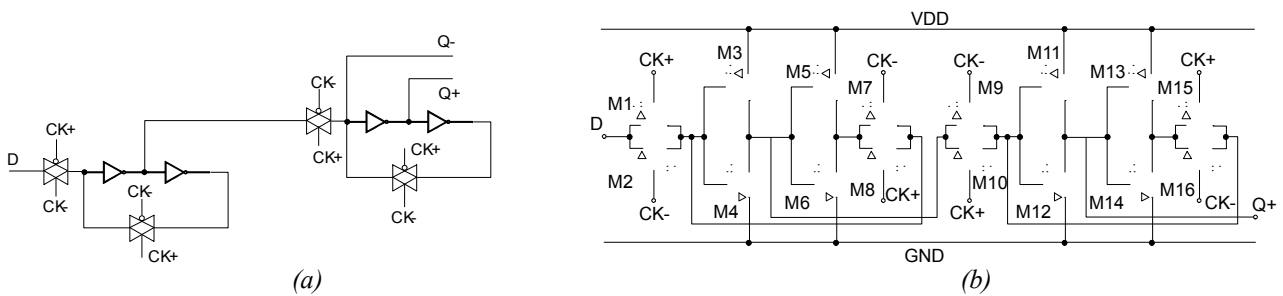
Rys. 9.5 Rezystancje poszczególnych tranzystorów MOS oraz wypadkowa rezystancja bramki transmisyjnej [4].

Wielowejsciowe bramki NAND i NOR powstają odpowiednio poprzez włączenie szeregowe bądź równoległe dodatkowych tranzystorów MOS jak to pokazano na rysunku 9.2. Tranzystory MOS połączone szeregowo zwiększają rezystancję ścieżki dla bramki NAND od masy do wyjścia a dla bramki NOR od zasilania V_{DD} do wyjścia. Rezystancja ta powoduje wzrost czasu propagacji, co w praktyce ogranicza stosowanie bramek wielowejsciowych do około pięciu wejść.



Rys. 9.5. Dwuwęściowa bramka NAND (a) oraz NOR (b).

Układy bardziej złożone takie jak przerzutniki, dekodery, liczniki itd. budowane są na podstawie bramek podstawowych. Poniżej, na rysunku 9.7 przedstawiony jest schemat blokowy jak również ideowy przerzutnika typu D sterowanego zboczem narastającym.



Rys. 9.7 Schemat blokowy (a) i szczegółowy (b) przerzutnika typu D sterowanego narastającym zboczem sygnału zegarowego.

10. Szacowanie czasów propagacji i wybór optymalnych wymiarów tranzystorów [1].

Podstawowe równania dla tranzystora MOS:

$$V_T = V_{T0} + \gamma(\sqrt{\phi - v_{BS}} - \sqrt{\phi})$$

$$K' = \mu C_{OX}$$

prąd drenu w obszarze omowym:

$$i_D = \frac{K'W}{L}(v_{GS} - V_T - v_{DS}/2)v_{DS}$$

prąd drenu w obszarze nasycenia:

$$i_D = \frac{K'W}{2L}(v_{GS} - V_T)^2(1 + \lambda v_{DS})$$

gdzie V_T – napięcie progowe [V], V_{T0} – napięcie progowe przy zerowym napięciu podłoże-źródło, t.j. dla $V_{BS}=0$ [V], γ - współczynnik podłożowy [$\sqrt{V}$], ϕ - potencjał powierzchniowy (około 0.7V), μ - ruchliwość nośników w kanale [$m^2/(sec V)$], C_{OX} – gęstość powierzchniowa pojemności bramki [F/m^2], λ - współczynnik modulacji długości kanału [$1/V$], W , L – odpowiednio szerokość i długość obszaru kanału [m].

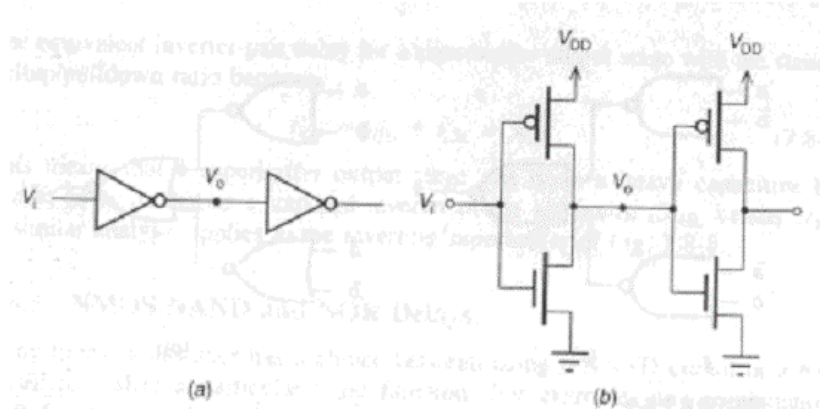
Stała czasowa procesu: Jest to stała wyznaczona jako iloczyn minimalnej pojemności bramki tranzystora oraz rezystancji dla zerowego napięcia v_{DS} przy sterowaniu bramki z napięcia zasilającego:

$$\tau_P = R_{SS}C_G \quad (10.1)$$

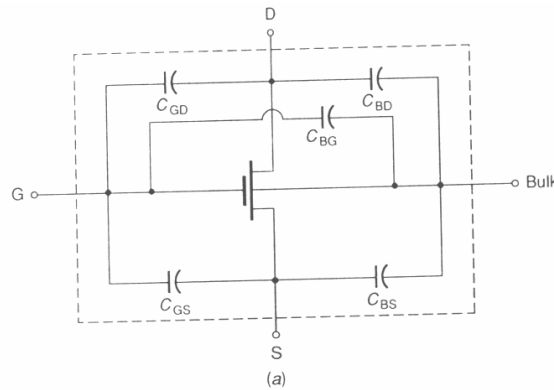
gdzie: $R_{SS} = \frac{L_1}{K'W_1(V_{DD} - V_{TN})}$, $C_G = W_1L_1C_{OX}$, stąd:

$$\tau_P = \frac{L_1^2 C_{OX}}{K'(V_{DD} - V_{TN})} \quad (10.2)$$

Analiza opóźnień w układach cyfrowych CMOS zostanie wykonana z przybliżeniem i w oparciu o bramkę inwertera obciążoną inną bramką inwertera. Taki przypadek przedstawiony jest na rysunku 10.1.



Rys. 10.1. Inwerter CMOS sterujący następnym inwerterem.



	Region		
	Cutoff	Ohmic	Saturation
C_{GD}	$C_{OX} W L_D$	$C_{OX} W L_D + \frac{1}{2} W L C_{OX}$	$C_{OX} W L_D$
C_{GS}	$C_{OX} W L_D$	$C_{OX} W L_D + \frac{1}{2} W L C_{OX}$	$C_{OX} W L_D + \frac{2}{3} W L C_{OX}$
C_{BG}	$C_{OX} W L$	0	0
C_{BD}	C_{BD1}	$C_{BD1} + \frac{C_{BC1}}{2}$	C_{BD1}
C_{BS}	C_{BS1}	$C_{BS1} + \frac{C_{BC1}}{2}$	$C_{BS1} + \frac{2}{3} C_{BC1}$

Rys. 10.2. Pojemności pasożytnicze tranzystora MOS. L_D – dyfuzja boczna kanału tranzystora MOS, pozostałe symbole wg zwyczajowego znaczenia.

Pojemność tranzystora z kanałem typu N wynosi (dla zakresu odcięcia i omowego)

$$C_{GN} = C_G \approx C_{OX} W_N L_N \quad (10.3)$$

Pojemność tranzystora typu P, oznaczona jako C_{GP} , jest o podobnej wartości. Jeśli tranzystor jest w zakresie odcięcia (czyli dla wysokiego stanu napięcia wejściowego) wówczas pojemność wynosi $C_{OX} W_P L_P$. W zakresie omowym wartość ta również wynosi $C_{OX} W_P L_P$. W zakresie nasycenia jest ona równa $2/3 C_{OX} W_P L_P$, jednakże tranzystor w tym stanie znajduje się tylko przez krótką chwilę podczas zmiany stanów. W związku z tym przyjmujemy, że w przybliżeniu pojemność bramki

tranzystora jest stała i wynosi $C_{OX}W_P L_P$. Pojemność wejściowa inwertera jest sumą pojemności tranzystorów składowych i wynosi:

$$C_{GC} = C_{GN} + C_{GP} \quad (10.4)$$

Uproszczony model inwertera z rysunku 10.1 przedstawiony jest na rysunku 10.4. Przybliżony czas narastania od 10% do 90% wartości sygnału wynosi około 2.0 stałe czasowe modelu idealnego z rysunku 10.4, stąd czas narastania i opadania wynosi odpowiednio:

$$t_{LH} = 2R_2 C_{GC} \quad (10.5)$$

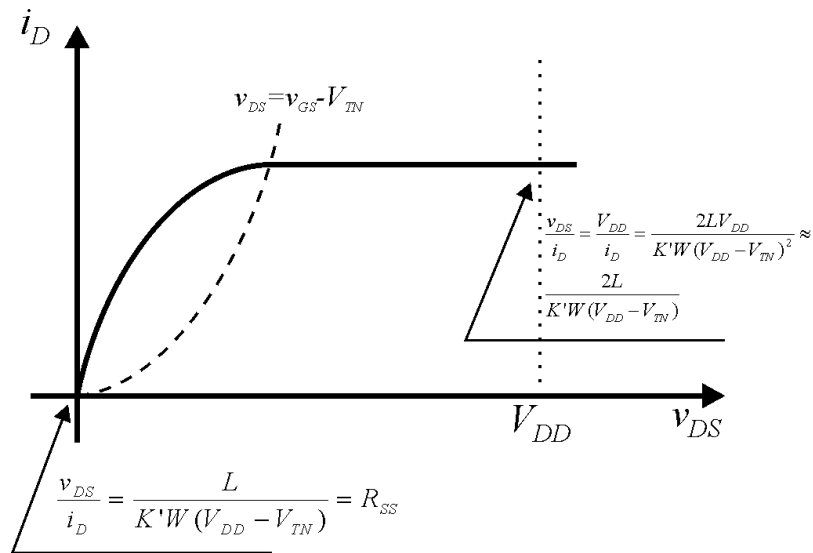
$$t_{HL} = 2R_1 C_{GC} \quad (10.6)$$

Wartość rezystancji R_1 można określić stosując przybliżoną analizę jako

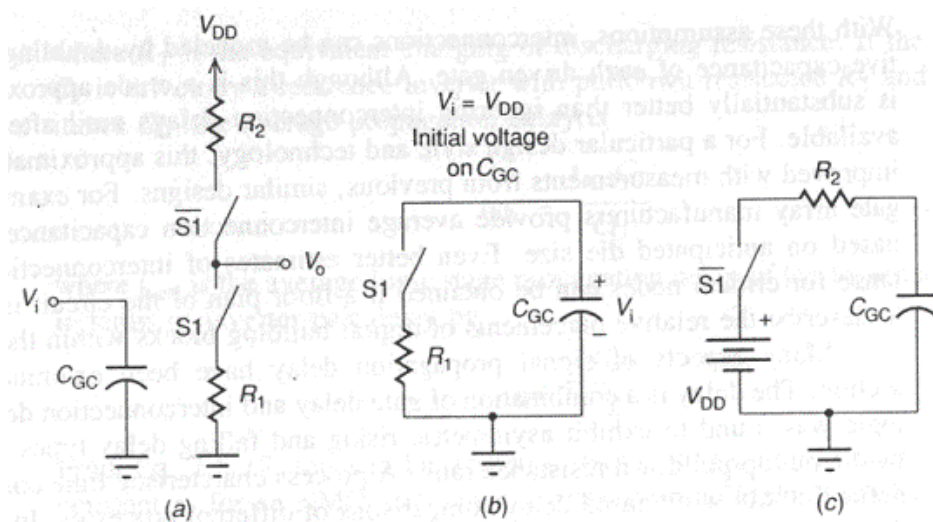
$$R_1 = \frac{2L_1}{K'W_1(V_{DD} - V_{TN})} = 2R_{SS} \quad (10.7)$$

Natomiast wartość R_2 jest równa:

$$R_2 = \frac{2L_2}{K'W_2(V_{DD} + V_{TP})}$$



Rys. 10.3. Charakterystyki tranzystora M1 inwertera CMOS.



Rys. 10.4. Układ równoważny do analizy opóźnień inwertera CMOS. (a) uproszczony model RC, (b) model w czasie zmiany stanu wyjścia z wysokiego na niski, (c) model dla zmian stanu niski – wysoki.

Przykład 10.1. Należy obliczyć opóźnienie kaskady inwerterów CMOS w technologii 2um. Dane: $K'_N = 45 \mu A/V^2$, $K'_P = 15 \mu A/V^2$, tranzystory dobrane dla symetrycznego sterowania wyjścia, napięcia

progowe $V_{TN}=1V$, $V_{TP}=-1V$, $C_{OX}=1fF/um^2$. Należy ustalić opóźnienia wyrażone w sekundach oraz w stosunku do stałej czasowej procesu.

Rozwiązanie: Dla jednakowego sterowania wyjściem w stanie niskim i wysokim musi być spełniony warunek

$$K'_N \frac{W_1}{L_1} = K'_P \frac{W_2}{L_2}, \text{ jeśli więc przyjmiemy, że tranzystor M1 będzie miał minimalne dostępne}$$

wymiary $W_1=2um$, $L_1=2um$ wówczas M2 będzie miał wymiary $W_2=6um$, $L_2=2um$. Pojemność wejściowa inwertera będzie równa

$$C_{GC} = C_{GN} + C_{GP} = C_{OX}L_1(W_1 + W_2) = 1fF * 2um(2 + 6)um = 16fF$$

stała czasowa procesu wynosi

$$\tau_P = \frac{L_1^2 C_{OX}}{K'(V_{DD} - V_{TN})} = \frac{(2um)^2 1fF / um^2}{45uA / V^2 (5V - 1V)} = 0.022ns, \text{ rezystancje } R_1 \text{ oraz } R_2 \text{ są równe}$$

$$R_1 = R_2 = \frac{2L_1}{K'W_1(V_{DD} - V_{TN})} = 11.1k\Omega. \text{ Czasy propagacji są więc równe}$$

$t_{LH} = 2R_2C_{GC} = t_{HL} = 2R_1C_{GC} = 0.355ns$, natomiast czas przejścia sygnału przez parę inwerterów będzie równy

$$t_{pid} = t_{LH} + t_{HL} = 0.71ns. \text{ Wartość ta w stosunku do stałej czasowej procesu wynosi } t_{pid} = 32\tau_P$$

W powyższym przykładzie rozważono istnienie wyłącznie pojemności bramkowych przy zaniechaniu wszystkich pozostałych. W rzeczywistości istotną rolę odgrywają również pojemności złączowe drenów i źródeł tranzystorów oraz pojemności połączeń.

Obciążenia pojemnościowe:

$$t_{dly} = \frac{t_{apd}C_T}{C_G} \quad (10.8)$$

t_{dly} – średni czas propagacji, C_T – pojemność obciążająca, C_G – pojemność bramki jednostkowej, t_{apd} – średni czas propagacji dla danej rodziny wyznaczony dla pary inwerterów jednostkowych

$$t_{apd} = \frac{t_{pid}}{2} \quad (10.9)$$

Sterowanie wieloma bramkami:

$$t_{stage} = t_{apd}f \quad (10.10)$$

gdzie t_{stage} – opóźnienie danego stopnia logicznego, f – równoważna liczba jednostkowych inwerterów, $f=C_T/C_G$. Stąd łatwo można oszacować opóźnienie danej ścieżki sygnałowej o N poziomach logicznych, o równoważnym obciążeniu na danym poziomie równym f_i , jako:

$$t_{path} = t_{apd} \sum_{i=1}^N f_i \quad (10.11)$$

Uwzględnienie pojemności połączeń:

Jeśli przyjmiemy że na danym poziomie logicznym obciążenie pojemnościowe jest równe sumie dwóch składowych tj. wynikających z wielokrotnych wejść bramek oraz pojemności połączeń wówczas opóźnienie w danym węźle będzie równe:

$$t_{node} = t_{apd}(f + m) \quad (10.12)$$

gdzie m odpowiada za część pojemności przypadającą na połączenia. Uwzględniając pojemności połączeń opóźnienie danej ścieżki jest równe:

$$t_{path} = t_{apd} \sum_{i=1}^N (f_i + m_i) \quad (10.13)$$

Zwiększenie wymiarów tranzystorów:

Zwiększenie stosunku W/L tranzystorów bramek powoduje proporcjonalną zmianę rezystancji a co za tym idzie zmniejszenie czasu propagacji. Jeśli jako θ oznaczymy wartość zwielokrotnienia tych

wymiarów wówczas otrzymamy odpowiednie skrócenie czasu propagacji oraz zwiększenie pojemności wejściowej takiej bramki (θC_G). Opóźnienie powiększonego inwertera wynosi więc:

$$t_{inv} = \frac{t_{apd} C_T}{\theta C_G} \quad (10.14)$$

Uwzględniając powiększoną wydajność bramek opóźnienie danej ścieżki sygnałowej można przybliżyć wzorem:

$$t_{path} = t_{apd} \sum_{i=1}^N \frac{f_i + m_i}{\theta_i} \quad (10.15)$$

10.1. Optymalizacja opóźnienia cyfrowych bloków wyjściowych [1].

Cyfrowe bramki logiczne konstruuje się zazwyczaj wykorzystując tranzystory o minimalnych wymiarach. Niewielkie wymiary są istotne dla osiągnięcia dużej gęstości scalenia bloków logicznych, niestety takie bramki mają niewielką siłę sterującą obciążeniem. Dodatkowo, często jest wymagane wytworzenie poziomów logicznych na odpowiednim poziomie napięciowym jak np. dla standardów przemysłowych TTL, LVCMOS, HSTL i inne. Przykłady obciążeń pojemnościowych często spotykanych w praktyce przedstawione są w tabeli 10.1.1

Tabela 10.1.1. Typowe wartości obciążeń pojemnościowych.

Obciążenie	C_T	C_T/C_G
Pojedynczy inwerter referencyjny	5.3fF	1
Dziesięć inwerterów	63fF	10
Szyna metalowa 4mm x 4.5um	0.450pF	71
Standardowy PAD 100um x 100um	0.25pF	40
Sonda oscyloskopu	9.0pF	1587
Wyprowadzenie adresowe chipa pamięci	5.0pF	794

W celu zmniejszenia opóźnienia wyprowadzeń sygnałów wychodzących z układów scalonych stosuje się kaskadowe łączenie buforów o coraz większej wydajności prądowej (i oczywiście wymiarach). Załóżmy, że mamy dostępny sygnał na wyjściu inwertera referencyjnego (o minimalnych wymiarach). Wówczas przy bezpośrednim sterowaniu obciążenia o pojemności C_L opóźnienie wyniesie:

$$t_{dir} = \frac{t_{apd} C_L}{C_G} \quad (10.1.1)$$

Dla liczby całkowitej n większej od zera definiujemy współczynnik α o wartości:

$$\alpha = \left(\frac{C_L}{C_G} \right)^{1/n} \quad (10.1.2)$$

Alternatywnie, n można wyznaczyć jako:

$$n = \frac{\ln C_L / C_G}{\ln \alpha} \quad (10.1.3)$$

Rozważmy strukturę jak na rysunku 10.1.1, która steruje obciążeniem o pojemności C_L . Jest ona złożona z kaskady inwerterów, każdy następny w ścieżce sygnałowej ma coraz większe wymiary geometryczne, większe są α razy szerokości kanałów przy czym długości kanałów pozostają stałe. W związku z tym obciążenie pojemnościowe k -tego stopnia będzie wynosiło:

$$C_{Lk} = \alpha^k C_G \quad (10.1.4)$$

Opóźnienie pierwszego inwertera będzie równe αt_{apd} , więc obciążenie ścieżki inwerterów będzie równe:

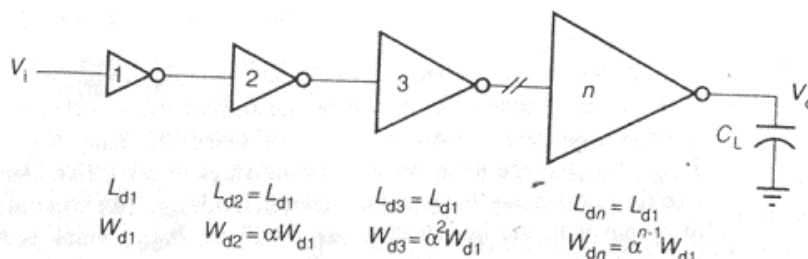
$$t_{cas} = n \alpha t_{apd} \quad (10.1.5)$$

Niech r będzie stosunkiem opóźnienia kaskady inwerterów w stosunku do bezpośredniego sterowania obciążeniem, wówczas:

$$r = \frac{t_{cas}}{t_{dir}} = \frac{n\alpha t_{apd}}{t_{apd} C_L / C_G} = \frac{n\alpha C_G}{C_L} \quad (10.1.6)$$

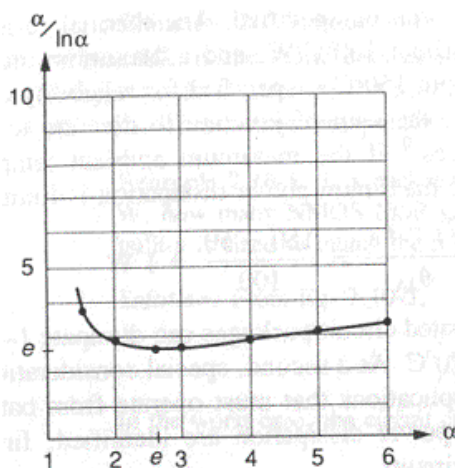
Naszym celem jest takie wybranie n a co za tym idzie również α , aby zminimalizować współczynnik r . Uwzględniając zależność 10.1.3 otrzymujemy:

$$r = \frac{\ln(C_L/C_G)}{C_L/C_G} \frac{\alpha}{\ln \alpha} \quad (10.1.7)$$



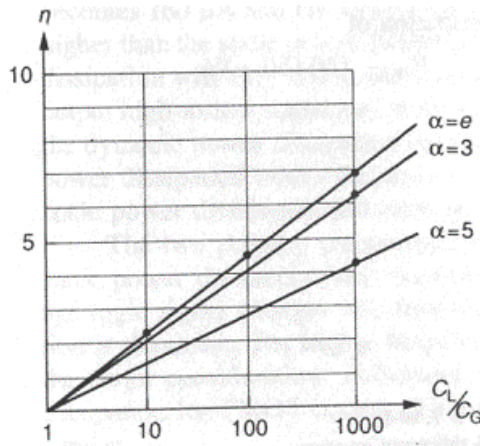
Rys. 10.1.1. Kaskada inwerterów o zwiększających się wymiarach i przez to sile sterowania.

Czynnik zawierający pojemności jest ustalony przez założenia obciążeniowe. Naszym celem jest więc takie dobranie α aby zminimalizowana była wartość r . Drugi czynnik powyższego wyrażenia przedstawiony jest na rysunku 10.1.2. Łatwo zauważyć, że minimum występuje dla $\alpha=e$. Wykres ilości niezbędnych stopni dla zwiększającej się wartości C_L/C_G przedstawiony jest na rysunku 10.1.3. Należy zauważyć, że n jest liczbą stopni więc musi zostać wybrane jako liczba całkowita. Skalowanie wymiarów tranzystorów również jest skokowe i nie może wynosić dokładnie e . W rzeczywistości skalowanie wybierane jest jako większe niż e co zmniejsza liczbę niezbędnych stopni. Z rysunku 10.1.2 widać, że dla α w zakresie od 2 do 4 zmiana od punktu minimum jest mniejsza niż 5%.



Rys. 10.1.2. Wykres wyrażenia $\alpha/\ln \alpha$ w funkcji α .

Na zakończenie należy wspomnieć o dwóch ważnych rzeczach: Po pierwsze, jeśli liczba inwerterów jest nieparzysta następuje odwrócenie polaryzacji sygnału wyjściowego. Po drugie, chociaż przyspieszenie pracy układu jest bardzo duże dla wysokich wartości n , zużycie powierzchni również wzrasta bardzo silnie. Na przykład dla $n=6$ i $\alpha=e$, powierzchnia jest $1+e^1+e^2+e^3+e^4+e^5=233$ razy większa niż inwertera referencyjnego.



Rys. 10.1.3. Liczba stopni w funkcji stosunku pojemności C_L/C_G .

Przykład 10.1.1 Należy wyznaczyć opóźnienie sygnału dla sterowania inwerterem jednostkowym (jak w przykładzie 10.1) oraz kaskadą optymalnie dobranych inwerterów PADa o wymiarach $100\mu\text{m} \times 100\mu\text{m}$ dla typowej technologii CMOS $2\mu\text{m}$ o pojemności powierzchniowej warstwy metalu wynoszącej $0.025\text{fF}/\mu\text{m}^2$. Do PADa dołączona jest sonda oscyloskopu o pojemności 10pF . Minimalny tranzystor $2 \times 2 \mu\text{m}$, $C_{OX}=1\text{fF}/\mu\text{m}^2$. Opóźnienie należy podać w stosunku do czasu propagacji inwertera jednostkowego t_{apd} oraz w sekundach dla danych jak w przykładzie 10.1. Należy również porównać zużycie powierzchni krzemu.

Rozwiązanie: Pojemność PADa wynosi:

$$C_{PAD} = 100\mu\text{m} * 100\mu\text{m} * 0.025\text{fF} / \mu\text{m}^2 = 0.25\text{pF}$$

Pojemność obciążenia jest więc równa

$$C_L = C_{PAD} + C_{SONDA} = 10.25\text{pF}$$

$$C_G = 3 * 2\mu\text{m} * 2\mu\text{m} * 1\text{fF} / \mu\text{m}^2 + 2\mu\text{m} * 2\mu\text{m} * 1\text{fF} / \mu\text{m}^2 = 16\text{fF}$$

Opóźnienie przy bezpośrednim sterowaniu wynosi:

$$t_{dir} = \frac{t_{apd} C_L}{C_G} = t_{apd} \frac{10.25\text{pF}}{16\text{fF}} = 640.625 t_{apd}$$

Wyrażając powyższe w bezwzględnych jednostkach czasu wynosi

$$t_{dir} = 640.625 t_{apd} = 640.625 \frac{t_{ipd}}{2} = 640.625 \frac{0.71\text{ns}}{2} = 227.4\text{ns}$$

(odpowiadająca częstotliwość $= 1/2/t_{dir} = 2.199\text{MHz} !!!$)

Dla sterowania kaskadą inwerterów wybieramy optymalne $\alpha=e$, wówczas

$$n = \frac{\ln C_L/C_G}{\ln \alpha} = 6.462, \text{ wybieramy najbliższą całkowitą wartość równą } 6. \text{ Współczynnik } \alpha$$

modyfikuje się do wartości

$$\alpha = \left(\frac{C_L}{C_G} \right)^{1/n} = 2.94$$

Wynikowe opóźnienie wynosi:

$$t_{cas} = n \alpha t_{apd} = 6 * 2.94 t_{apd} = 17.64 t_{apd} = 17.64 \frac{t_{ipd}}{2} = 17.64 \frac{0.71\text{ns}}{2} = 6.26\text{ns}$$

(odpowiadająca częstotliwość $= 1/2/t_{cas} = 78.872\text{MHz} !!!$)

Zwiększenie zużycia powierzchni będzie równe:

$$1 + \alpha + \alpha^2 + \dots + \alpha^5 = 332.4$$

Jeśli dla oszczędzenia powierzchni zostało by wybrane użycie 3 stopni wówczas

$$\alpha = \left(\frac{C_L}{C_G} \right)^{1/n} = 8.62$$

$$t_{cas} = n\alpha t_{apd} = 3 * 8.62 t_{apd} = 25.86 t_{apd} = 25.86 \frac{t_{ipd}}{2} = 25.86 \frac{0.71 ns}{2} = 9.18 ns$$

(odpowiadająca częstotliwość = $1/2/t_{cas}=54.466 MHz$)

Zwiększenie powierzchni w stosunku do inwertera jednostkowego wyniesie

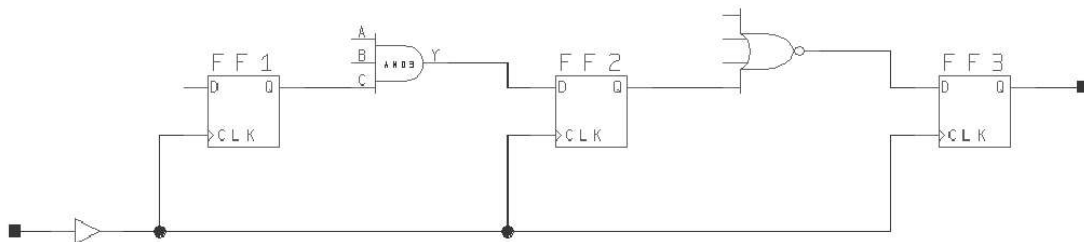
$$1 + \alpha + \alpha^2 = 82.92$$

11. Projektowanie ścieżek zegarowych. Generacja drzewa zegarowego.

Ścieżka zegara systemowego jest trudnym elementem projektowym, szczególnie dla układów VLSI. Wejściowy sygnał zegarowy musi dotrzeć do dużej liczby przerzutników z odpowiednim i zazwyczaj bardzo małym opóźnieniem. Ważniejszym jednak parametrem są maksymalne różnice w czasie dotarcia zegara do każdego z przerzutników zwane przesunięciem czasowym zegara (ang. clock skew). Jego wartość nie może przekraczać odpowiedniego progu aby nie doszło do hazardu w układzie. Z tego względu w pierwszej części przedstawiona zostanie szczegółowa definicja przesunięcia czasowego zegara oraz możliwe związane z tym problemy a następnie przedstawione zostaną architektury ścieżek zegarowych oraz algorytmy ich projektowania.

11.1 Definicje podstawowych parametrów.

Sąsiedztwo w sensie sekwencyjnym są to elementy zegarowane (rejestry, przerzutniki), które są oddzielone od siebie jedynie logiką kombinacyjną lub są ze sobą połączone bezpośrednio. Połączenie poprzez inny przerzutnik powoduje brak sąsiedztwa w sensie sekwencyjnym.

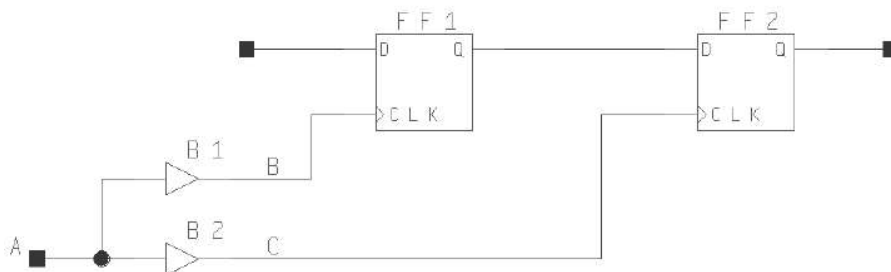


Rys. 11.1. Ilustracja sąsiedztwa w sensie sekwencyjnym. Przerzutniki FF1 oraz FF2 są sąsiadujące a FF1 i FF3 nie.

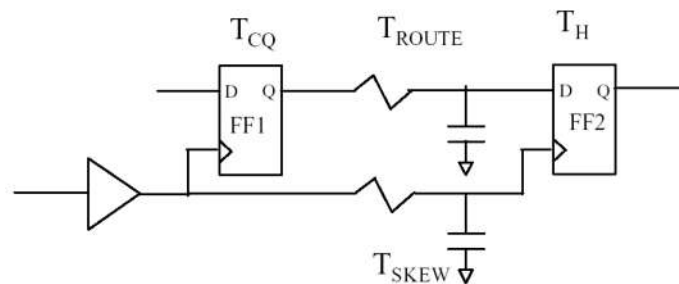
Przesunięcie czasowe zegara (ang. clock skew). Jeśli mamy dwa przerzutniki sąsiadujące w sensie sekwencyjnym, oznaczone symbolami i oraz j , oraz ścieżkę zegarową wychodzącą z tego samego źródła a dochodzącą do tych przerzutników, wówczas przesunięciem czasowym zegara nazywamy różnicę czasów dotarcia sygnałów zegarowych do wejść CLK przerzutników.

$$t_{SKEW} = t_i - t_j \quad (11.1)$$

gdzie: t_i oraz t_j są odpowiednio czasami opóźnień zegara dochodzącego do przerzutnika i oraz j .



Rys. 11.2. Przesunięcie czasowe zegara jest to różnica w czasie dotarcia sygnału A do przerzutnika FF1 i FF2.



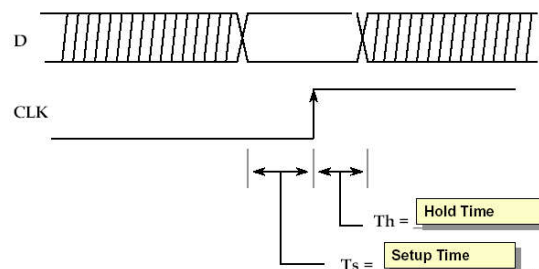
Rys. 11.3. Ze względu na rozłożony charakter linii długich przesunięcie czasowe powstaje nawet pomiędzy przerzutnikami z połączonymi bezpośrednio zegarami.

Na rysunku 11.2 przedstawiono przykład dwóch sekwencyjnie sąsiadujących przerzutników gdzie może powstać zjawisko przesunięcia czasowego zegara. Mimo, że sieć zegarowa dochodząca do przerzutników FF1 i FF2 składa się z identycznych elementów, w rzeczywistości następuje przesunięcie zegara gdyż: bufor B1 i B2 zawsze są nieidentyczne, wartości obciążeń sieci B oraz C są różne ze względu na różne długości oraz w końcu na fakt, że nawet na zwykłej linii metalicznej mamy do czynienia z opóźnieniami (ze względu na rozłożony charakter parametrów), co przedstawione jest symbolicznie na rysunku 11.3.

Czas setup jest to minimalny czas przed pojawieniem się sygnału zegara kiedy pozostałe wejścia przerzutnika muszą być o znanej i stabilnej wartości. W przeciwnym przypadku wartość wyjściowa przerzutnika jest nieznana.

Czas hold jest to minimalny czas po pojawieniu się sygnału zegara kiedy pozostałe wejścia przerzutnika muszą pozostawać w znanej i stabilnej wartości. W przeciwnym przypadku wartość wyjściowa przerzutnika może być przypadkowa.

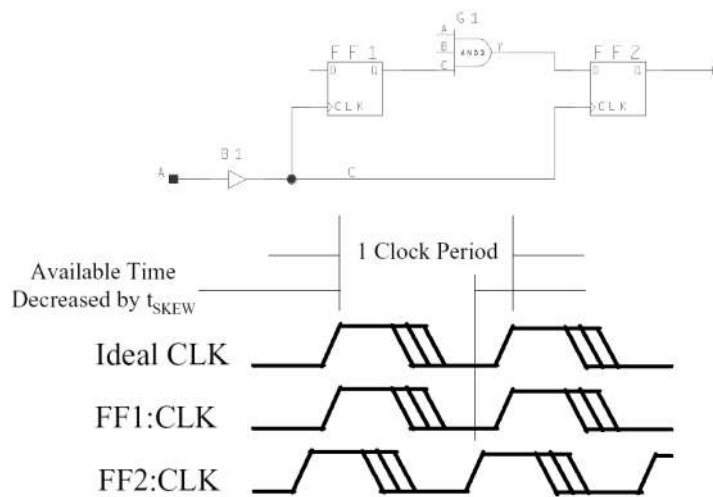
Edge-Triggered F-F Timing



Rys. 11.4. Graficzne przedstawienie czasów setup oraz hold.

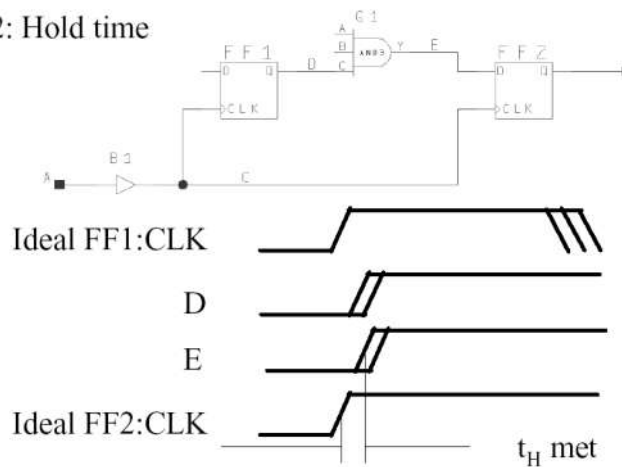
11.2 Wpływ przesunięcia czasowego zegara na spełnienie zadanych wartości czasów setup oraz hold.

Wpływ przesunięcia czasowego zegara przedstawiony jest graficznie na poniższych rysunkach. Powoduje on modyfikację wartości czasów setup oraz hold.

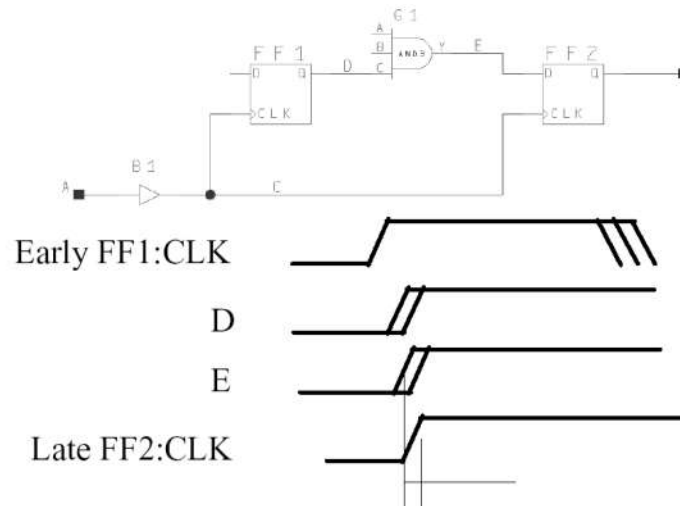


Rys. 11.5. Zmniejszenie możliwego dostępnego czasu setup spowodowane przesunięciem czasowym zegara.

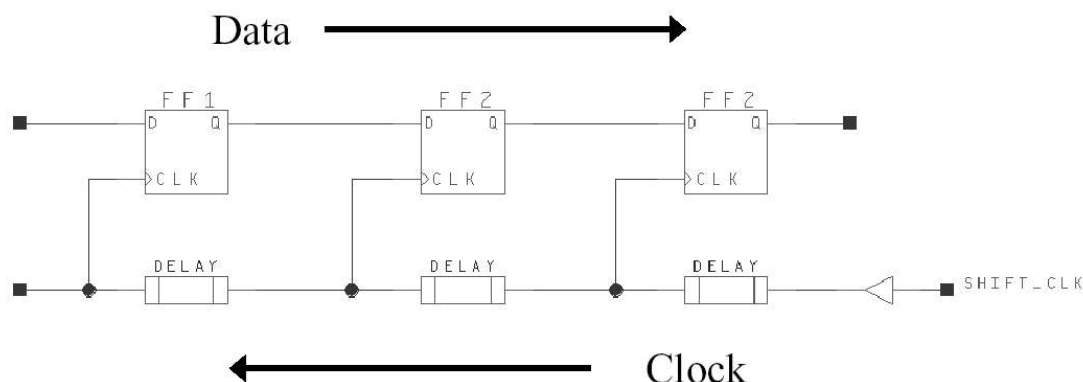
Case 2: Hold time



Rys. 11.5. Wpływ przesunięcia zegara na czas hold, przypadek idealny.



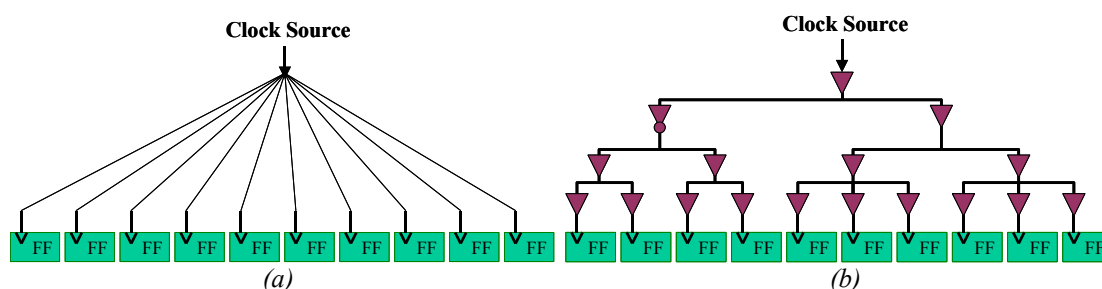
Rys. 11.6. Wpływ przesunięcia zegara na czas hold, przypadek niekorzystny. W przypadku opóźnień zegarów jak na rys. 11.5 czas hold zostałby poprawiony kosztem czasu setup.



Rys. 11.7. Celowo wprowadzane opóźnienia ścieżki zegarowej poprawiają czas hold kosztem pogorszenia czasu setup.

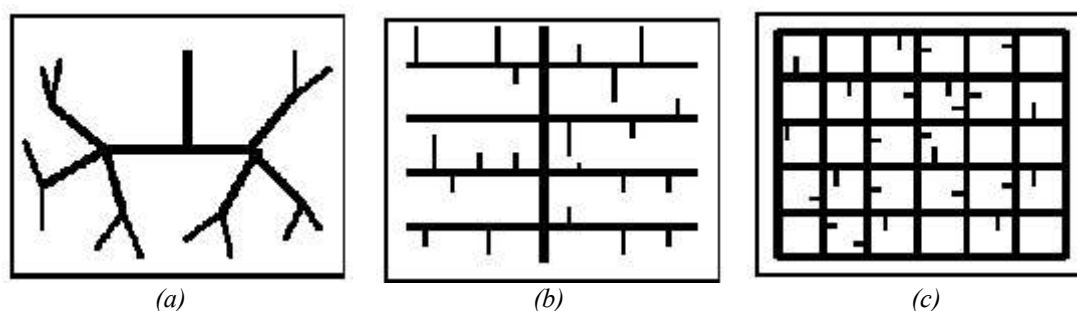
11.3 Architektury ścieżki zegarowej.

W celu zapewnienia równych czasów dotarcia sygnałów zegarowych jednocześnie do wszystkich przerzutników występujących w układzie stosuje się rozdział pojedynczej ścieżki na system ścieżek. Istnieją dwa rozwiązania rozproszania zegara: normalne oraz z dodanymi buforami w gałęziach. Drugie rozwiązanie zapewnia krótszy całkowity czas propagacji oraz skrócenie czasu narastania sygnału zegara. Niestety ze względu na zmiany parametrów technologicznych ma również większą wartość przesunięcia czasowego zegara. Przykład sytuacji wyjściowej i wynikowej ścieżki zegarowej przedstawiony jest na rys. 11.9



Rys. 11.8. Zestaw przerzutników sterowany pojedynczym sygnałem zegarowym (a) oraz przykładowe rozwiązanie drzewa zegarowego (b).

Oprócz tradycyjnej, drzewiastej ścieżki zegarowej spotyka się struktury zegara typu kufer (ang. trunk) oraz siatkowego (ang. mesh).



Rys. 11.9. Ścieżka zegarowa typu drzewo (a), kufer (b) oraz siatka (c).

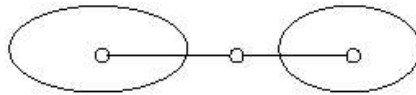
11.4 Problem generacji ścieżki zegarowej oraz algorytmy je realizujące.

Problem generacji ścieżki zegarowej można przedstawić następująco: Mamy daną lokalizację wejściowych wyprowadzeń sygnałów zegara przerzutników układu $\{CLK1, CLK2, \dots, CLKn\}$ oraz lokalizację źródła sygnału zegara $CLK0$. Mamy zadaną maksymalną wartość przesunięcia czasowego zegara B oraz ograniczenia dodatkowe takie jak np. minimalne opóźnienie, maksymalne opóźnienie, czas narastania sygnału zegara. Należy zbudować ścieżkę połączeniową spełniającą powyższe wymagania oraz zminimalizować koszt liczony w ogólnej długości połączeń, poborze

mocy oraz innych zadanych ograniczeniach drugoplanowych. Znanych jest kilka metod generacji drzewa zegarowego. Poniżej zostaną przedstawione 2 z nich.

Generacja drzewa zegarowego podejście TOP-DOWN, metoda środka masy.

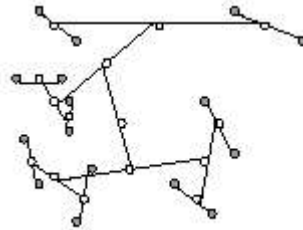
- podzieli wejścia zegarowe CLK na dwa zestawy CLK_L oraz CLK_R o równych rozmiarach,
- połącz środek CLK masy do zestawów CLK_L oraz CLK_R ,
- rekursywnie dokonuj podziałów w kierunku pionowym.



Jeśli mamy n wyprowadzeń zegarowych przerzutników wówczas metoda ta powoduje wzrost długości linii zegarowych proporcjonalnie do $3/2\sqrt{n}$ maksymalny wzrost przesunięcia czasowego zegara jest proporcjonalny do $1/\sqrt{n}$.

Generacja drzewa zegarowego podejście BOTTOM-UP, metoda dopasowania trasowania.

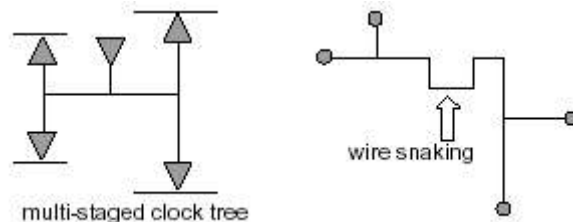
- rekursywnie dopasowuj każde poddrzewo na poszczególnych poziomach,
- minimalizuj przesunięcie czasowe zegara w każdym poddrzewie,



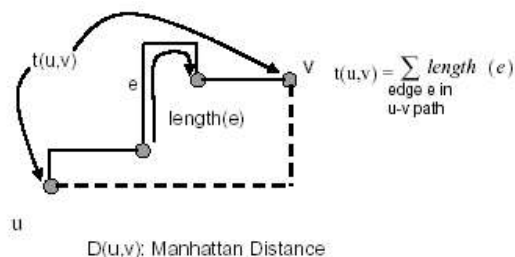
Na podstawie modelu opóźnień bazującego na długościach ścieżek zegarowych można osiągnąć zerową wartość przesunięcia czasowego zegara.

Możliwe modyfikacje metody

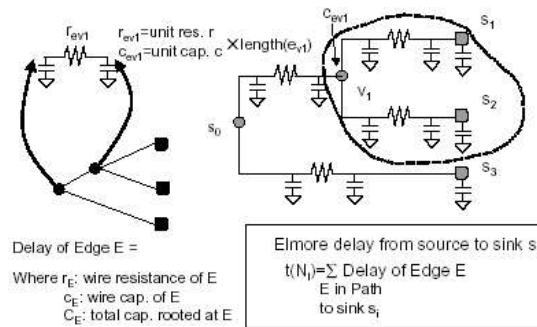
- użycie modelu opóźnień Elmora,
- użycie buforów w gałęziach,
- używanie zaginania ścieżek i przez to zwiększania pojemności (ang. wire snaking) w celu uzyskiwania dokładnie zerowego przesunięcia czasowego zegara.



Stosowane modele opóźnień ścieżek sygnałowych:



Rys 11.11 Model opóźnień bazujący na długościach ścieżek.



Rys. 11.12 Model opóźnień Elmora.

12. Testowanie układów ASIC [8].

12.1. Wstęp.

Testowanie układu scalonego jest powtarzane wielokrotnie po każdym etapie produkcji układu scalonego. Odpowiednie zaplanowanie testowania jest istotnym czynnikiem osiągnięcia zysku ekonomicznego. Jeśli jest wykonane błędnie, wówczas może się zdarzyć iż układ wadliwy jest zakwalifikowany jako sprawny lub układ sprawny jako wadliwy. Oba przypadki są niekorzystne dla producenta. W przypadku wysłania układu wadliwego do odbiorcy powstają dodatkowe koszty wymiany układu, strata reputacji i ewentualna strata udziałów w rynku. Odrzucenie układów sprawnych zmniejsza przychód producenta.

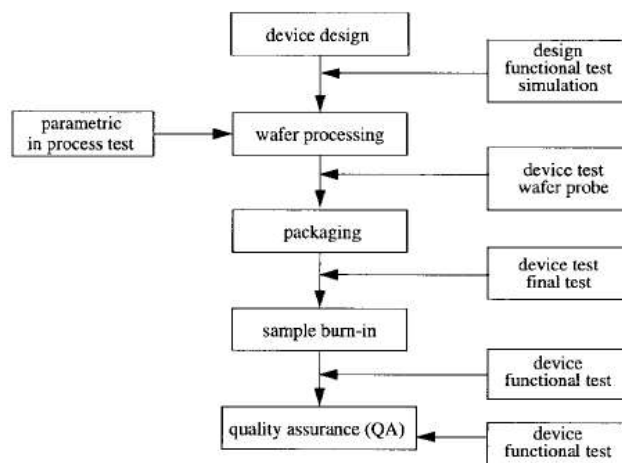
Główne kroki produkcji układu scalonego przedstawione są na rysunku 12.1. Po każdym z kroków dokonywane jest testowanie. Każdy z przedstawionych testów wykonywany jest w innym celu i z użyciem innych narzędzi. Początkowe testowanie przeprowadzane jest na etapie projektowania poprzez użycie oprogramowania CAD. Można wtedy zbadać funkcjonalność i przewidywane parametry układu scalonego.

Po przetworzeniu płytki półprzewodnikowej następuje testowanie przy użyciu specjalnych pól testowych na jej powierzchni. Te pola testowe mogą być umieszczone zamiast układów scalonych albo w miejscach przeznaczonych w czasie późniejszej produkcji do cięcia poszczególnych chipów (ang. scribe line). Ten test jest przeznaczony do szybkiej identyfikacji katastroficznych defektów procesu technologicznego. Jeśli płytka nie spełnia parametrów okna procesu technologicznego jest wyrzucana w całości. W tym momencie można także wykonać pomiary i ekstrakcję parametrów SPICE służących do symulacji.

Po przetestowaniu płytki półprzewodnikowej testowany jest następnie każdy chip umieszczony na tej płytce. W zależności od projektu / producenta prowadzony jest całkowity lub częściowy test każdego z układów. Układy wadliwe są oznaczane (elektronicznie) a po cięciu są odrzucane i nie są pakowane do obudowy. Następnym krokiem jest testowanie funkcjonalne po pakowaniu sprawnych chip'ów. Test końcowy sprawdza parametry układu w przewidzianym zakresie temperatur pracy.

Kilka układów poddawanych jest testom pracy w podwyższonej temperaturze przez dłuższy okres czasu (burn-in). Celem tej czynności jest eliminacja „słabych” elementów, które prawdopodobnie uległyby uszkodzeniu w początkowym okresie eksploatacji.

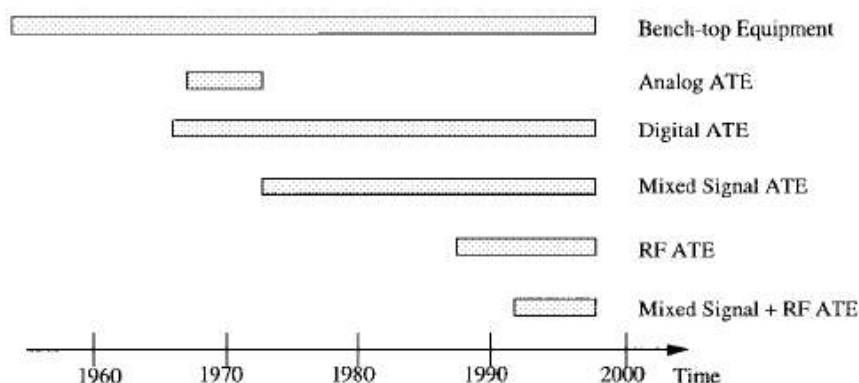
W końcu mniejsza próbka układów jest poddawana testom w celu zapewnienia jakości w odpowiedniej normie jakościowej (np. ISO9000).



Rys. 12.1. Testowanie układu scalonego na różnych etapach produkcji układu scalonego [8].

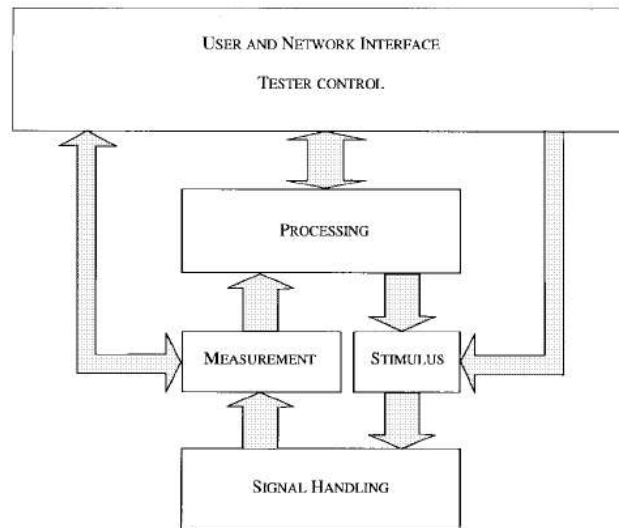
12.2. Sprzęt używany do testów.

Początkowo do testów układów scalonych wykorzystywano tradycyjną aparaturę kontrolno pomiarową, która była wcześniej używana przy projektowaniu i testowaniu układów prototypowych. Szybko jednak okazało się, że bardziej efektywne jest zastosowanie specjalizowanych urządzeń do automatycznego testowania ATE (ang. Automatic Test Equipment). W zależności od typu układu poddawanego testom taką aparaturę dzieli się na aparaturę przeznaczoną do badania układów cyfrowych, analogowych, mieszanych, wysokoczęstotliwościowych (RF) lub RF łącznie z sygnałami mieszanymi. Na rysunku 12.2 przedstawiono rozwój urządzeń w czasie. Analogowe ATE zostały wchłonięte przez mieszane systemy testowe, stąd na rysunku 12.2 testery analogowe zanikły około roku 1975.



Rys. 12.2. Rozwój testerów automatycznych ATE (ang. Automatic Test Equipement)[8].

System ATE generuje sygnały testowe, następnie podaje je do badanego układu, mierzy odpowiedzi, analizuje odpowiedzi i na tej podstawie daje raport końcowy stwierdzający (lub nie) sprawność badanego układu. Układ badany oznaczany jest skrótem DUT lub UUT (ang. Device/Unit Under Test). Schemat blokowy ATE przedstawiony jest na rysunku 12.3 a typowe parametry osiągane podsumowane są w tabeli 12.1.



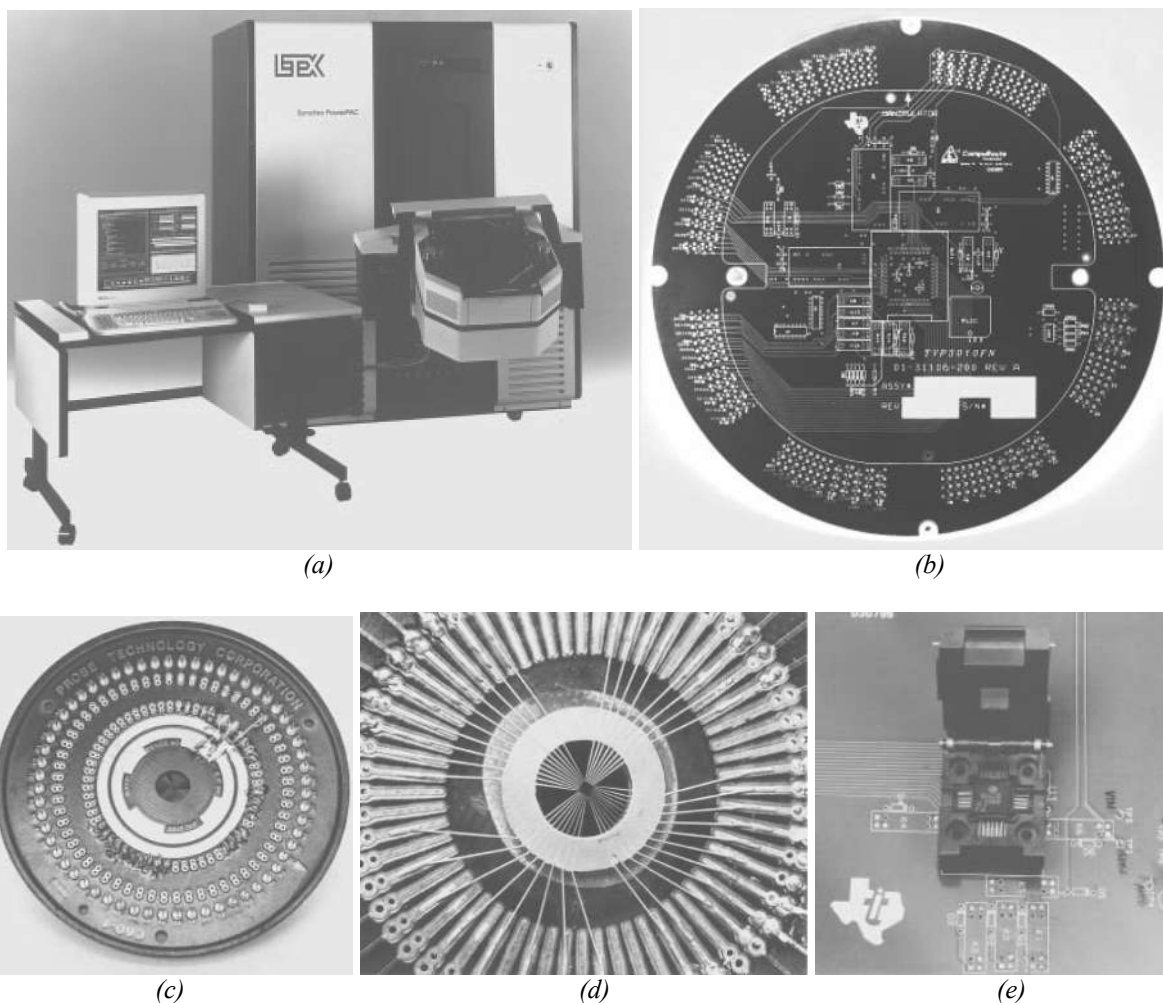
Rys. 12.3. Bloki funkcjonalne automatycznego testera [8].

Parameter	Typical Specification
Voltage force range resolution	$\pm 0.5\text{V}$ to $\pm 60.0\text{V}$ 12 to 14 bits
Voltage measure range resolution	$\pm 0.5\text{V}$ to $\pm 60.0\text{V}$ 12 to 16 bits
Current force range resolution	$\pm 1\text{nA}$ to $\pm 200\text{mA}$ 12 to 14 bits
Current measure range resolution	$\pm 3.7\text{nA}$ to $\pm 200\text{mA}$ 12 to 16 bits
Meter input impedance	up to 1 G Ω
Precision Waveform Sources and Digitizers Bandwidth Resolution Distortion	10 to 500 kHz 16 to 22 bits 75 to 110 dB
High Speed Waveform Sources and Digitizers Bandwidth Resolution Distortion	1 to 10 MHz 10 to 12 bits 40 to 60 dB
RF Sources and Measures Frequency Ranges I/O power 2 port S parameter measurement range	0.001 to 4GHz -130 to +30dBm -70 to +20dBm
Digital Input/Output Frequency Number of Formats Data Capture	up to 50MHz* 10 - 13 (drive and compare) up to 50MHz*

* - purely digital ATE's provide much higher digital rates
Some of the mixed-signal ATE vendors provide special modes which increase the digital rates and digital capture up to 400MHz.

Tabela 12.1. Typowe parametry sprzętu ATE [8].

Niezależnie od podziału funkcjonalnego, system ATE składa się zazwyczaj z komputera sterującego zestawem typowych urządzeń kontrolno pomiarowych połączonych z badanym układem poprzez specjalny interfejs DIB (ang. Device Interface Board) i próbnik ze złączami kontaktowymi. Na rysunku 12.4 przedstawiono przykładowe urządzenia ATE.

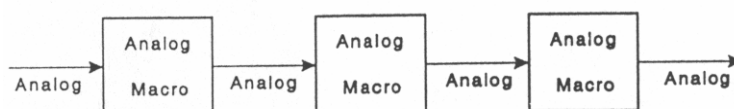


Rys. 12.4. System ATE LTX Syncho (a), interfejs połączeniowy DIB Teardyne A580 (b), próbnik o małej liczbie wyprowadzeń (c) i (d), podstawka przyłutowana bezpośrednio na DIB(e) [8].

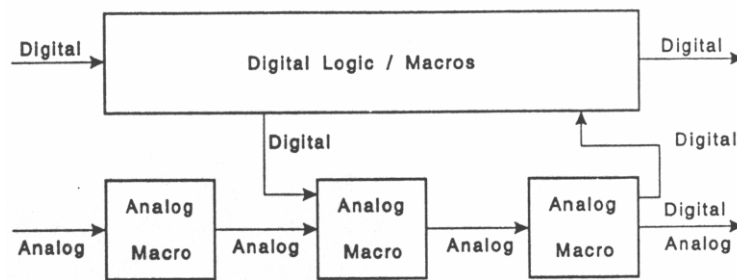
12.3 Projektowanie układów ASIC z uwzględnieniem ich testowania [9].

Ze względu na zwiększającą się złożoność układów scalonych VLSI ich testowanie staje się coraz trudniejszym zadaniem. Często czynność testowania jest głównym „wąskim gardłem” w czasie prototypowania, wdrażania, produkcji oraz utrzymania systemów mikroelektronicznych.

W dalszej części tego rozdziału zostanie wykorzystany podział układów na czysto analogowe oraz mieszane. Założono, że część analogowa układu scalonego jest podzielona na bloki funkcjonalne takie jak: przetwornik A/C lub C/A, wzmacniacz operacyjny, filtr i inne (rysunek 12.1). Założono, że cały projekt został wykonany w sposób hierarchiczny. Taki rodzaj syntezy analogowej umożliwia tradycyjny sposób testowania części analogowych. Część cyfrowa projektu może być również wykonana hierarchicznie ale nie jest to tutaj wymagane. Założono, że w układach mieszanych część cyfrowa jest oddzielona od części analogowej (do testowania) aczkolwiek część cyfrowa może sterować układami analogowymi (rysunek 12.2).



Rys. 12.1. Założony podział analogowego układu scalonego na bloki funkcjonalne [9].



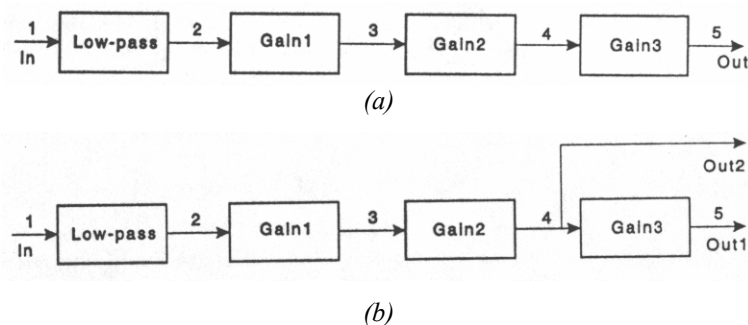
Rys. 12.2. Założony podział mieszanego układu scalonego na bloki funkcjonalne [9].

Projektowanie układu ASIC z uwzględnieniem jego testowania (ang. DFT Design for Testability) polega na takim dodaniu niezbędnych podukładów wykorzystywanych w czasie testowania, które umożliwią lub ułatwią wykrycie uszkodzenia układu scalonego. Dodane elementy powinny być zrealizowane jak najmniejszym kosztem (liczonym zwiększeniem czasu projektowania, zwiększeniem powierzchni układu scalonego jak również zmniejszeniem uzysku produkcyjnego) oraz powinny wprowadzać minimalne pogorszenie parametrów. Obecnie stosowane są 2 metody DFT. Pierwsza polega na takim zaprojektowaniu układu scalonego aby umożliwić obserwowanie jak również sterowanie wartościami sygnałów występujących w istotnych węzłach badanego układu. Druga technika polega na dodawaniu wbudowanych samotestów.

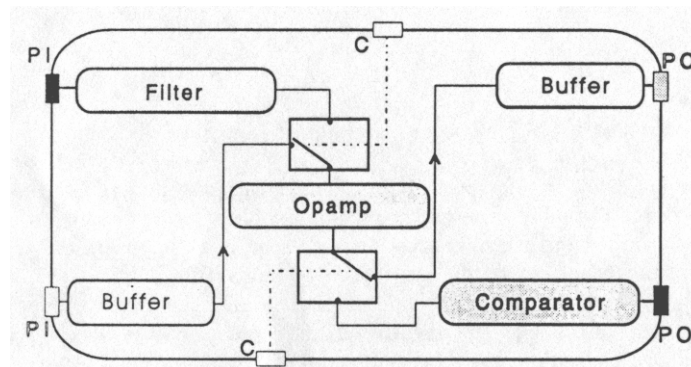
12.3.1 Układy analogowe i mieszane [9].

12.3.1.1. Zwiększanie możliwości sterowania i obserwacji istotnych węzłów sygnałowych.

W celu zwiększenia obserwowalności układu analogowego połączonego kaskadowo można dodać specjalne wyprowadzenia zewnętrzne w węzłach pośrednich jak to przedstawiono na rysunku 12.3. Takie rozwiązanie umożliwia dokładniejsze przebadanie stopnia GAIN3. Kosztem jest dodatkowe wyprowadzenie zewnętrzne oraz ewentualna degradacja parametrów całego toru przetwarzania analogowego ze względu na dodatkowe elementy w ścieżce sygnałowej. Często niezbędne jest stosowanie dodatkowych buforów sygnałowych „Out2”.

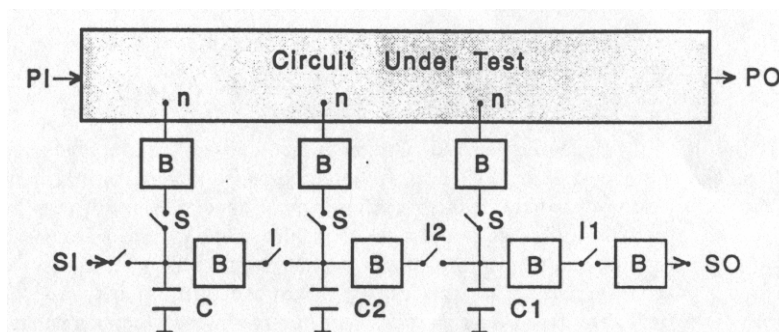


Rys. 12.3. Kaskada bloków analogowych (a) powiększona o dodatkowe wyprowadzenie kontrolne (b) [9].



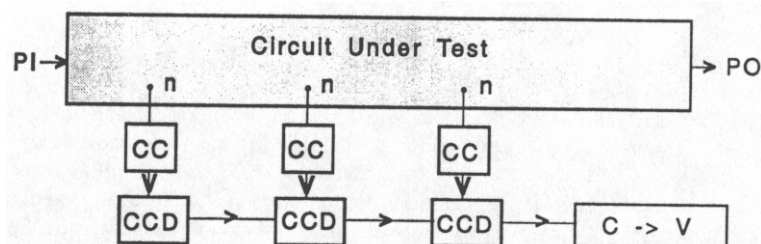
Rys. 12.4. Przykład zastosowania analogowych przełączników w celu umożliwienia sterowania i obserwacji wewnętrznym blokiem wzmacniacza operacyjnego. Wstawienie buforów jest niezbędne w celu zmniejszenia wpływu dodatkowego obciążenia [9].

Mniej bezpośrednią metodą ingerencji w wewnętrzne węzły układu jest zastosowanie multiplekserów analogowych sterowanych sygnałami cyfrowymi. Takie rozwiązanie przedstawione jest na rysunku 12.4. Sygnały są buforowane w celu umożliwienia podania ich do / z wnętrza rdzenia analogowego. Podawanie sygnałów z zewnątrz jest nieco trudniejsze do wykonania niż ich obserwacja gdyż zazwyczaj wymaga odcięcia wyjścia stopnia poprzedzającego. Multipleksery mogą działać w sposób inwazyjny (przecięcie ścieżki laserem) lub nieinwazyjny poprzez np. klucze analogowe.



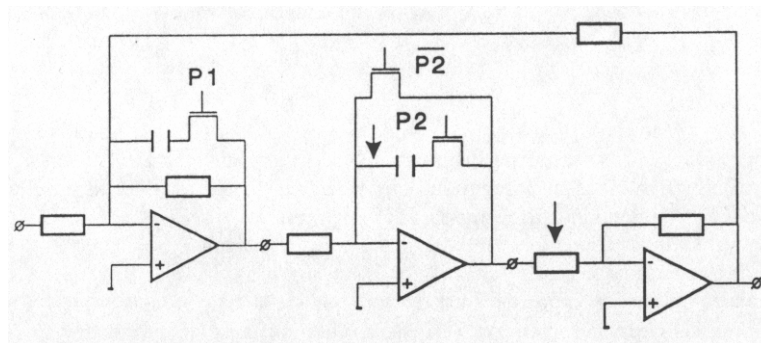
Rys. 12.5. Analogowe przesuwanie wartości sygnałów zapamiętanych w poszczególnych węzłach sygnałowych [9].

Techniki skanowania są szeroko wykorzystywane przy testowaniu układów cyfrowych. Główną zaletą takiego rozwiązania jest użycie tylko dwóch dodatkowych wyprowadzeń sygnałowych w celu zapisu i odczytu wartości sygnałów w rozpatrywanych węzłach oraz jednej lub kilku linii zegarowych. W przypadku układu analogowego należy zapamiętać i przesunąć wartość sygnału analogowego. Znane są dwa rozwiązania układów przesuwających sygnały analogowe a mianowicie: przełączane pojemności oraz elementy CCD. Układ wykorzystujący przełączane kondensatory łącznie z elementami dodatkowymi przedstawiony jest na rysunku 12.5. Bloki oznaczone literą B stanowią buforów sygnałowych (wzmocnienie jednostkowe, mała rezystancja wyjściowa, bardzo duża wejściowa). Układ ten pracuje w dwóch trybach: normalnym (wszystkie klucze otwarte) oraz trybie testowym. W trybie testowym najpierw są zamykane klucze S i w ten sposób na kondensatorach są zapamiętywane wartości napięć węzłowych. Następnie w celu wyprowadzenia sygnału na zewnątrz należy klucze I kolejno włączać.



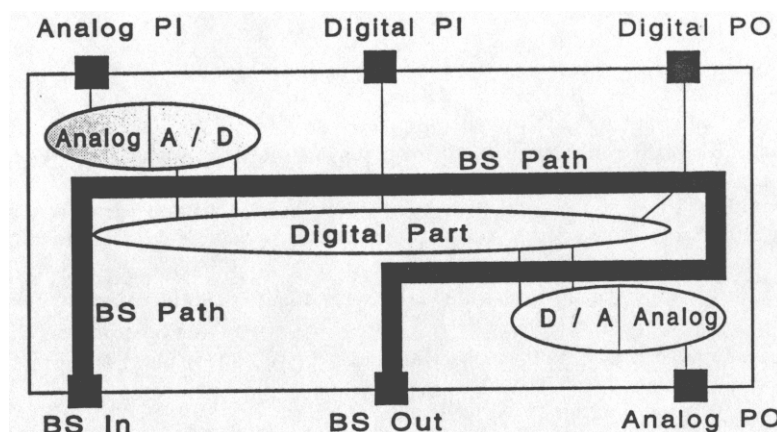
Rys. 12.5. Element CCD jako rejestr obserwacyjny wewnętrznych sygnałów analogowych [9].

Układ z wykorzystaniem łańcucha elementów CCD przedstawiony jest na rysunku 12.5. Elementy oznaczone CC są konwerterami sygnału obserwowanego na ładunek, który jest następnie przesuwany z wykorzystaniem elementów CCD. Na końcu łańcucha CCD jest umieszczony konwerter ładunek – napięcie (C->V)



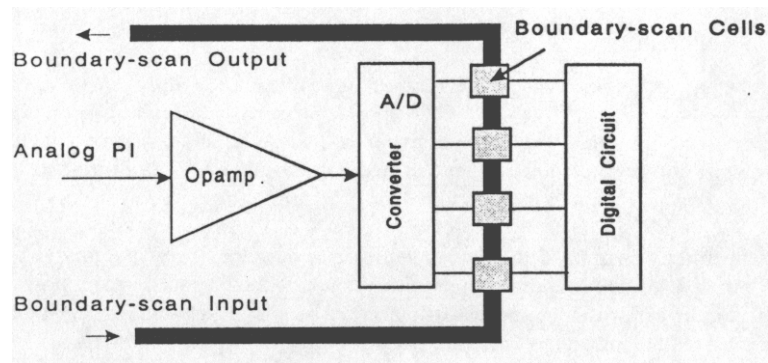
Rys. 12.6. Przykład filtru analogowego zmodyfikowanego o elementy umożliwiające testowanie uszkodzeń twardych i miękkich [9].

Na rysunku 12.7 przedstawiony jest przykład filtru analogowego wzbogaconego o klucze tranzystorowe umożliwiające testowanie pod względem funkcjonalnym. Filtr zawiera 3 stopnie przetwarzające. Aby przetestować stopień 2 należy poszerzyć pasmo stopni 1 i 3 aby ich charakterystyki nie wpływały na wynik pomiarów stopnia 2. Podobnie w celu pomiarów stopnia 1 należy poszerzyć pasmo stopnia 2 (bo stopień 3 ma szerokie pasmo). Klucze tranzystorowe P1 oraz P2 powodują odpowiednie zmiany charakterystyk częstotliwościowych.

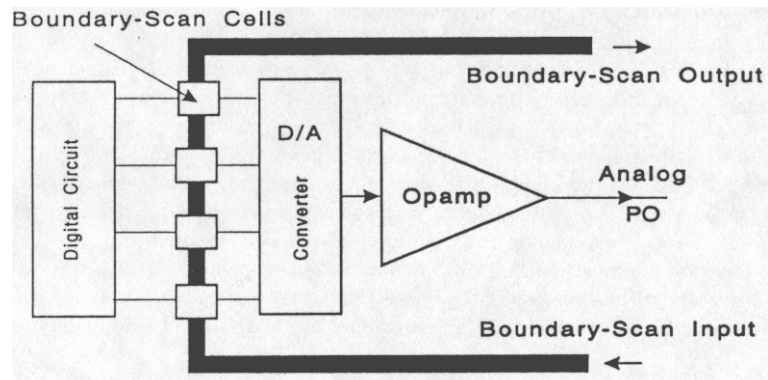


Rys. 12.7. Testowanie części analogowej układu mieszanego z wykorzystaniem rejestru brzegowego [9].

W układach mieszanych zawierających przetworniki A/C lub C/A pomiędzy częścią analogową i cyfrową można wykorzystać rejestr brzegowy w celu pomiaru lub ustawienia wartości sygnału analogowego. Można w układzie scalonym zawrzeć również specjalnie w tym celu dedykowane układy przetworników A/C i C/A oraz rejestrów cyfrowych. Ogólna struktura układu przedstawiona jest na rysunku 12.8 natomiast pomiar napięcia analogowego przedstawiono na rysunku 12.9 a jego ustawianie na rysunku 14.9.



Rys. 12.8. Testowanie sygnału wyjściowego z bloku analogowego za pomocą rejestru brzegowego [9].



Rys. 12.9. Ustawianie wartości sygnału wejściowego w testowanym bloku analogowym poprzez zastosowanie rejestru brzegowego [9].

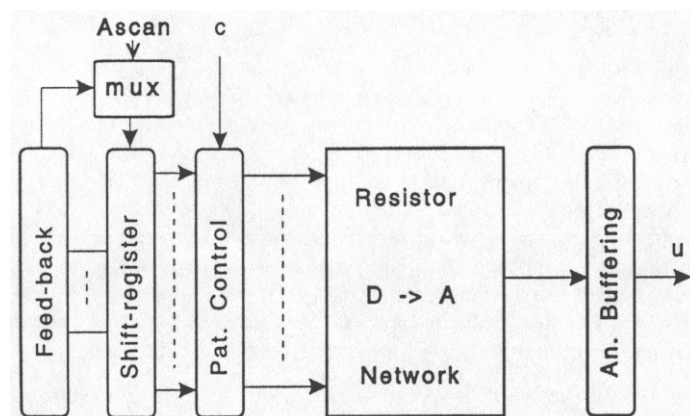
12.3.1.2. Wbudowane układy testujące.

Zupełnie odmienną techniką testowania układów scalonych jest umieszczenie pobudzeń jak również elementów pomiarowych wewnątrz struktury badanego układu zamiast stosowania oprzyrządowania na zewnątrz. Takie podejście ogranicza w sposób istotny zakres możliwych testów ale daje możliwość testowania układów w „terenach” poza fabryką a nawet układów już zamontowanych na płycie bez jej demontażu. W celu integracji samotestu do układu scalonego niezbędne jest dodanie następujących bloków:

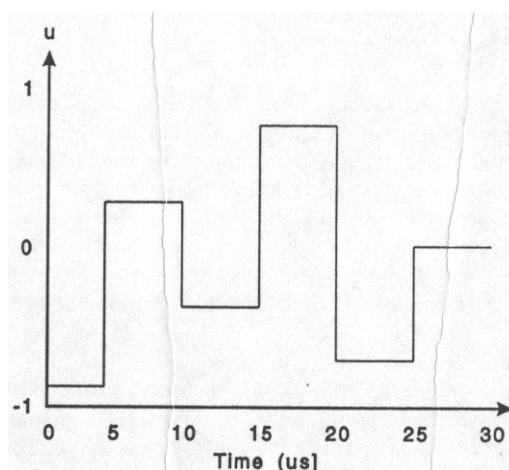
- analogowe multipleksery,
- generator sygnałów testowych,
- układ oceniający odpowiedzi bloków analogowych,
- układ sterujący testem.

Analogowy multiplekser jest używany do izolacji podbloku analogowego w celu jego przetestowania.

Generator sygnałów testowych wytwarza niezbędny przebieg sygnałowy podawany na wejście badanego podbloku. Przykład generatora pracującego jako przetwornik kolejnych słów cyfrowych na postać analogową przedstawiony jest na rysunku 12.11 natomiast przykładowy wytworzony przebieg na rysunku 12.10.1.



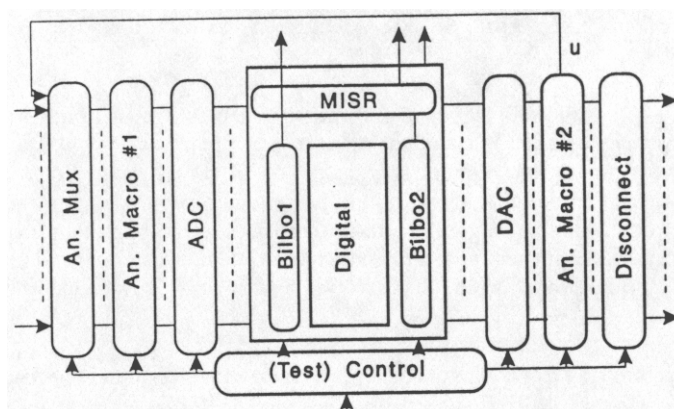
Rys. 12.10. Schemat blokowy układu generatora pobudzeń analogowych[9].



Rys. 12.10.1. Przykład sygnału generowanego w układzie z rys. 12.11 [9].

Układ oceniający odpowiedzi bloków analogowych ma na celu kwalifikację uzyskanej odpowiedzi do kategorii prawidłowych bądź błędnych. W przypadku uzyskania odpowiedzi błędnej można wykonać opcjonalną analizę miejsca i typu uszkodzenia. W odróżnieniu od układów cyfrowych kwalifikacja sygnału jako prawidłowego odbywa się poprzez porównanie z zakresem prawidłowych wartości a nie z jedną wartością.

Układ sterujący aktywuje rozpoczęcie testów i steruje pracą testera. Aktywacja powinna następować za pomocą sygnałów zewnętrznych lub poprzez włączenie zasilania lub zerowanie systemu.



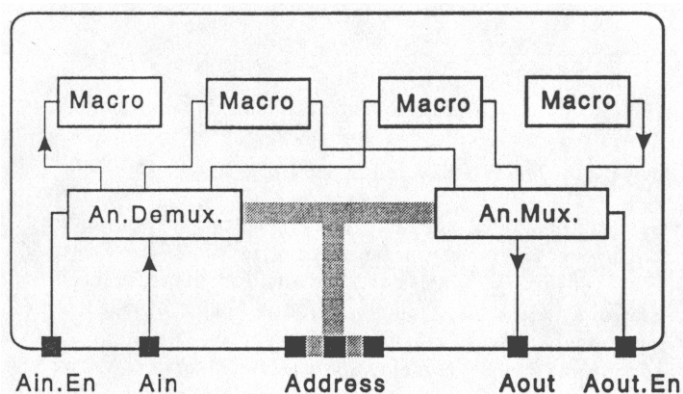
Rys. 12.13. Schemat blokowy mieszanego układu samotestującego [9].

Schemat blokowy układu scalonego zawierającego wbudowany samotest przedstawiony jest na rysunku 12.13. Sygnał oznaczony literą „u” jest analogowym pobudzeniem podawanym do badanego podbloku analogowego a wytwarzanym na podstawie wartości odczytywanych z bloku BILBO2 (bloki BILBO są pamięcią cyfrową). Sygnały wyjściowe bloku #2 są dostępne na

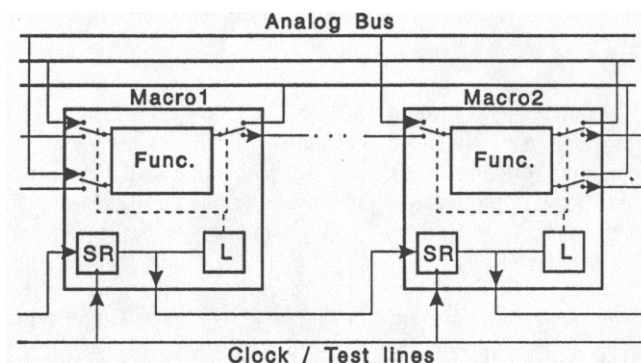
wyprowadzeniach zewnętrznych i mogą być pomierzone. Blok analogowy #1 odbiera sygnał „u” a jego odpowiedź jest przetwarzana na postać cyfrową i zapamiętywana w BILBO1. Blok BILBO1 jest włączony do rejestru szeregowego którego zawartość może zostać wyprowadzona w sposób szeregowy z układu scalonego. Blok „Disconnect” służy do odłączania obciążeń.

12.3.1.3. Multipleksowane analogowe szyny testujące [9].

Jedną z interesujących technik powiększenia możliwości testowania układów scalonych jest stosowanie analogowych szyn sygnałowych służących wyłącznie w celach testowych. Przykłady takich rozwiązań przedstawione są na rysunkach 12.14 i 12.15. Standard IEEE1149.4 definiuje multipleksowaną szynę analogową, która może służyć do pomiarów pojedynczego podbloku analogowego w danej chwili czasu (jednego bloku bo standard obsługuje tylko jedno wejście analogowe i jedno wyjście analogowe).



Rys. 12.14. Przełączana szyna analogowa [9].



Rys. 12.15. Przełączana szyna analogowa [9].

12.3.1.4. Testowanie przy zastosowaniu wiązki elektronów (ang. Electron-Beam Testability) [9].

Zastosowanie wiązki elektronów umożliwia pomiar wartości potencjału na powierzchni na której ten strumień operuje. Metoda pomiaru jest następująca: układ scalony jest umieszczany w próżni a następnie próbująca wiązka elektronów jest skierowana na punkt którego potencjał jest mierzony. Energia elektronów odbitych jest mierzona w spektrometrze. Wartość tej energii odzwierciedla napięcie panujące na podłożu od którego nastąpiło odbicie.

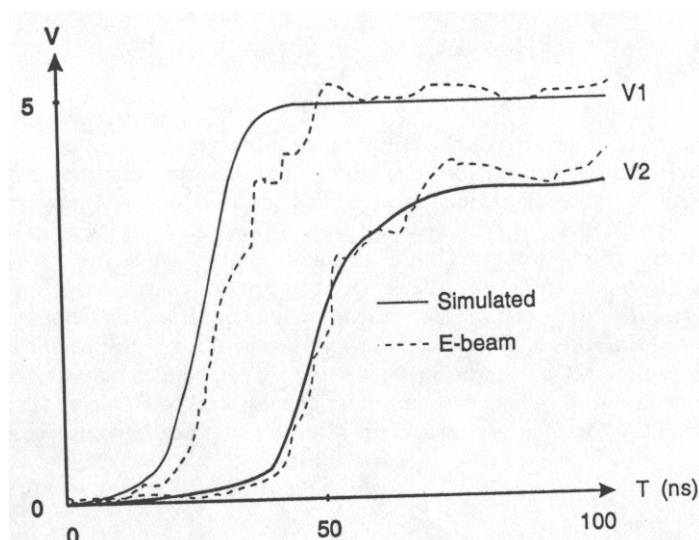
Powyższa technika pomiaru jest szeroko stosowana do testowania pamięci dynamicznych. Pomiar napięcia oraz wyniki symulacji w przykładowym układzie przedstawione są na rysunku 12.15.

Jednym z powodów dla którego ta technika nie jest bardzo szeroko stosowana jest dokładność uzyskiwanych pomiarów. Obecne możliwości systemów wykorzystujących wiązkę elektronów są następujące:

- zakres sygnałów $\pm 25V$,
- rozdzielczość mierzonego napięcia 5mV,

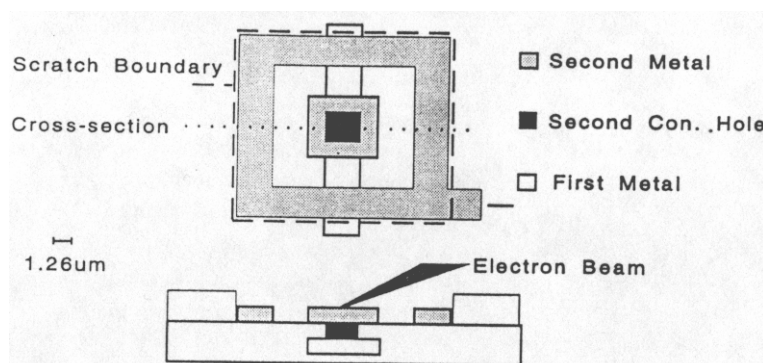
- dokładność pomiaru $<3\%$,
- częstotliwości pomiarowe do kilkuset MHz,
- średnica wiązki elektronów mniejsza niż $0.1\mu\text{m}$.

Tradycyjne testery analogowe uzyskują dokładności o rząd lub kilka rzędów lepsze. Polepszenie dokładności pomiaru E-BEAM wymaga wydłużenia czasu pomiaru.



Rys. 12.15. Porównanie pomiarów przy użyciu aparatury wykorzystującej wiązkę elektronów oraz symulacji SPICE [9].

Często zdarza się sytuacja, że dana ścieżka sygnałowa leży gdzieś głęboko i wtedy ścieżki wierzchnie zniekształcają wyniki pomiarowe. W związku z tym stosuje się wyciąganie takiego punktu pomiarowego na najwyższą warstwę przewodzącą. Pole pomiarowe ma wymiary około $3\mu\text{m}$ na $3\mu\text{m}$. Dla zabezpieczenia się przed przesłuchami i innymi efektami stosuje się linie ochronne. Dodatkowo w celu poprawy dokładności pomiaru nie stosuje się pasywacji powierzchni pola pomiarowego. Przykład topografii takiego punktu pomiarowego przedstawiony jest na rysunku 12.16.



Rys. 12.16. Punkt dostępu do pomiarów z wykorzystaniem wiązki elektronów [9].

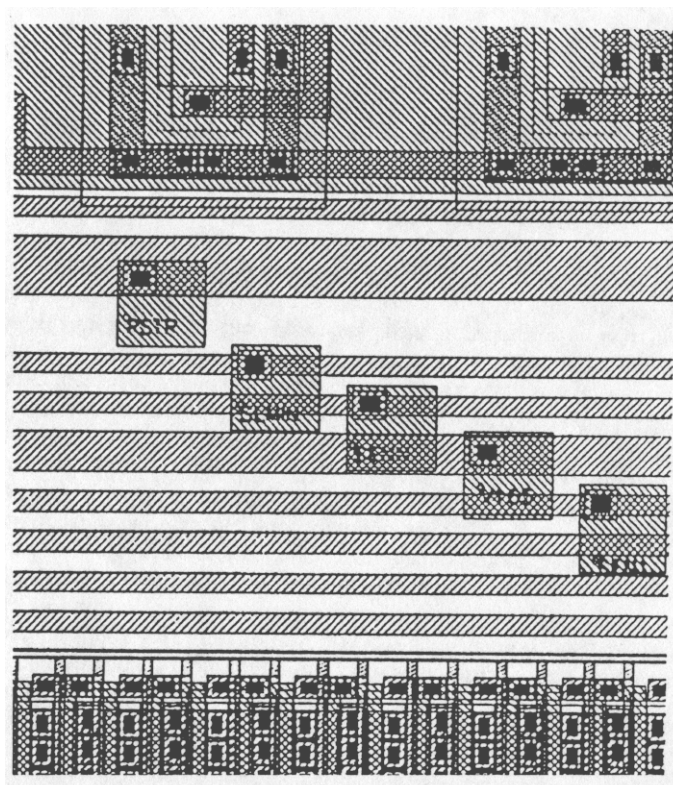
Wykorzystując programy symulacji komputerowej można określić kluczowe węzły sygnałowe i właśnie w nich umieścić punkty pomiarowe. Wstawienie tych punktów może odbywać się całkowicie automatycznie poprzez wykorzystanie odpowiedniego programu do edycji topografii. Przykład rozmieszczenia punktów pomiarowych w układzie o dużej skali integracji przedstawiony jest na rysunku 12.11.

Technika wiązki elektronów ma następujące zalety:

- bardzo mała i możliwa do uwzględnienia przy projektowaniu pojemność dołączana do badanych węzłów,
- bardzo duża rozdzielczość przestrzenna ($0.1\mu\text{m}$),
- technika niedestrukcyjna,
- możliwość automatycznego umieszczania pkt. pomiarowych.

Wady techniki są następujące:

- niska dokładność i rozdzielczość pomiarów,
- relatywnie długi czas pomiarów w porównaniu z technikami klasycznymi,
- duży koszt inwestycji w sprzęt.



Rys. 12.11. Przykład umiejscowienia punktów testowych w układzie scalonym[9].

12.3.2. Układy cyfrowe – na podstawie pakietu CADENCE.

W układach cyfrowych wykorzystuje się szeregowy rejestr przesuwany umożliwiając zarówno obserwację istotnych węzłów układu jak i ustawianie wartości tych węzłów do pewnej zadanej wartości. Jest to czynność standardowa i może zostać zautomatyzowana uwalniając w ten sposób projektanta od konieczności zwiększonego wysiłku i opisu w języku HDL dodatkowych struktur testowych. Sposób integracji struktur DFT jest omówiony poniżej na przykładzie komercyjnego pakietu CADENCE.

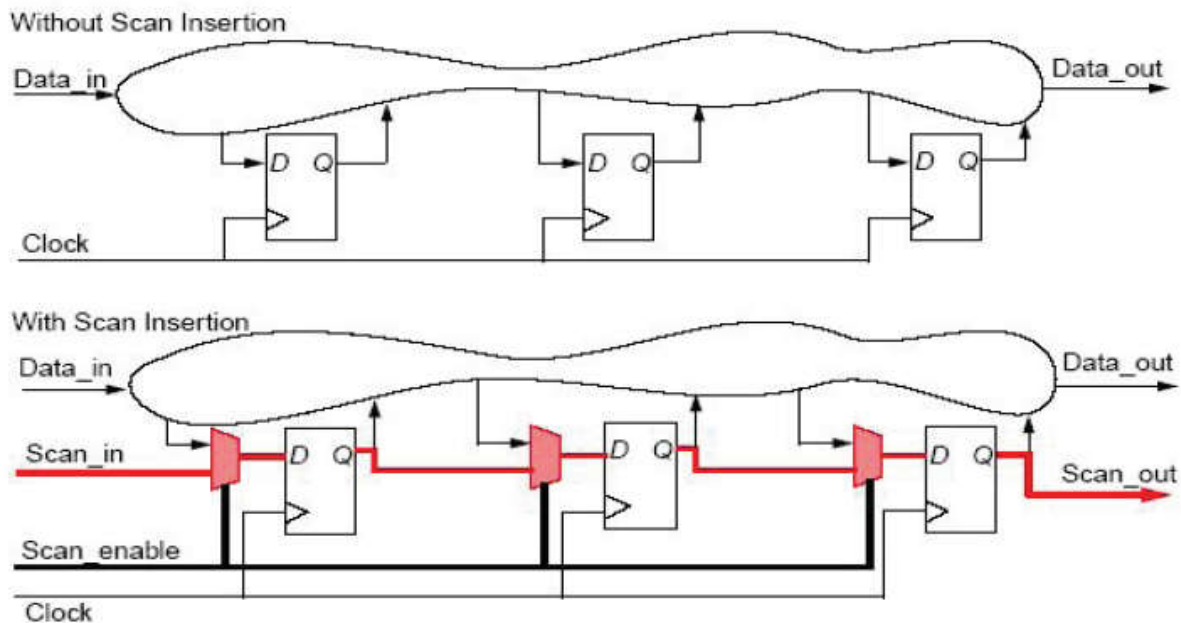
Synteza logiczna z uwzględnieniem testowania prowadzona jest w systemie CADENCE przed oraz w czasie optymalizacji. W jej ramach umożliwiające jest wykonanie automatycznej integracji logiki testowej (DFT) do projektu. Najpierw należy przeprowadzić specyfikację stylu DFT jako jeden z poniższych:

- Muxed Scan Style,
- Clocked Scan Style,
- Clocked Level Sensitive Scan Design (LSSD),
- Auxiliary Clocked-LSSD Scan Style.

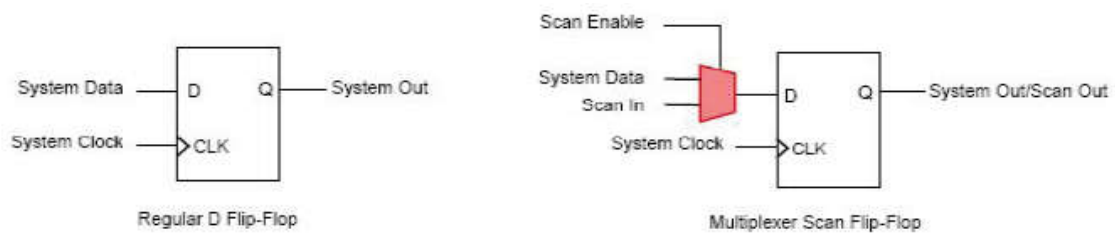
Najczęściej stosowanym stylem jest Muxed Scan Style. Niezależnie od wybranego stylu syntezer wykonuje następujące czynności:

- Zamienia wszystkie przerzutniki, które spełniają warunki DFT, układem zastępczym odpowiednim dla danego wybranego stylu DFT. W przypadku stylu Muxed Scan Style są to przerzutniki zawierające na wejściu danych multiplexer. W zależności od aktualnego trybu pracy układu multiplexer wybiera albo sygnał systemowy albo sygnał skanowany.
- Dodaje do modułu wyprowadzenia sterujące układem zastępczym.
- Dodaje do modułu wyprowadzenia wejściowe i wyjściowe łańcucha skanującego.

Schemat logiczny układu przed i po wstawieniu multiplekserów w stylu Muxed Scan Style przedstawiony jest na rysunku 12.18. Przerzutniki wstawiane w poszczególnych stylach przedstawione są na rysunkach 12.20-23.



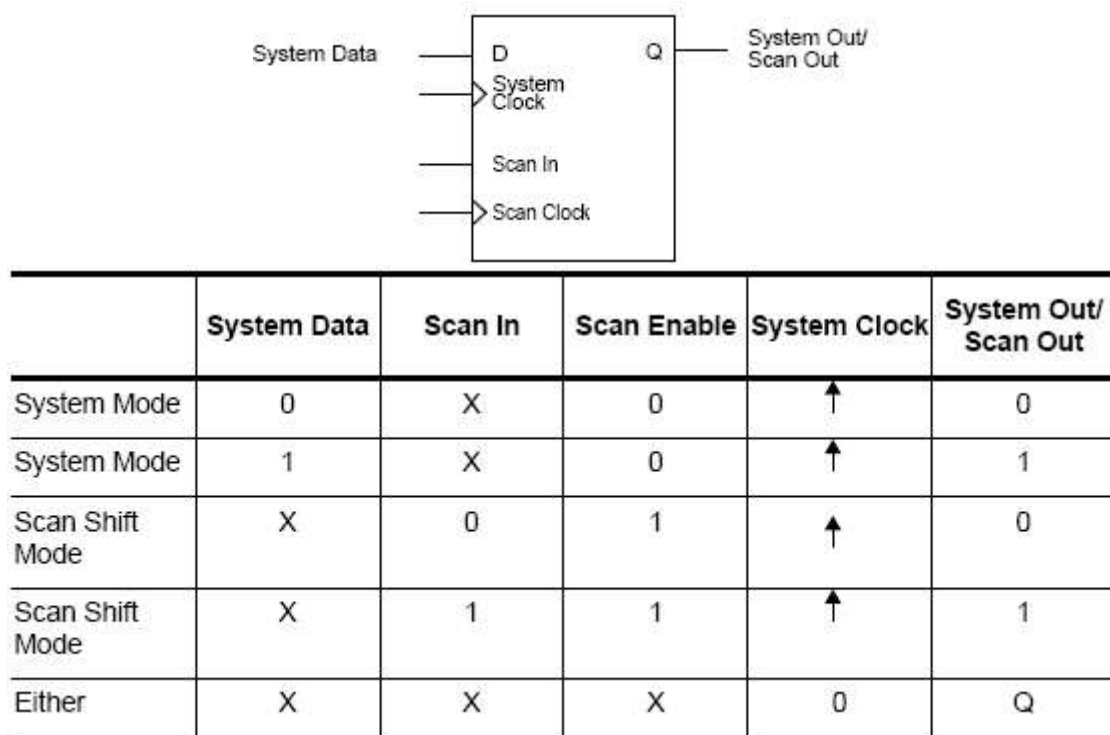
Rys. 14.18. Rdzeń modułu cyfrowego przed (góra) i po wstawieniu (dół) multiplekserów łańcucha skanującego.



	System Data	Scan In	Scan Enable	System Clock	System Out/ Scan Out
System Mode	0	X	0	↑	0
System Mode	1	X	0	↑	1
Scan Shift Mode	X	0	1	↑	0
Scan Shift Mode	X	1	1	↑	1
Either	X	X	X	0	Q

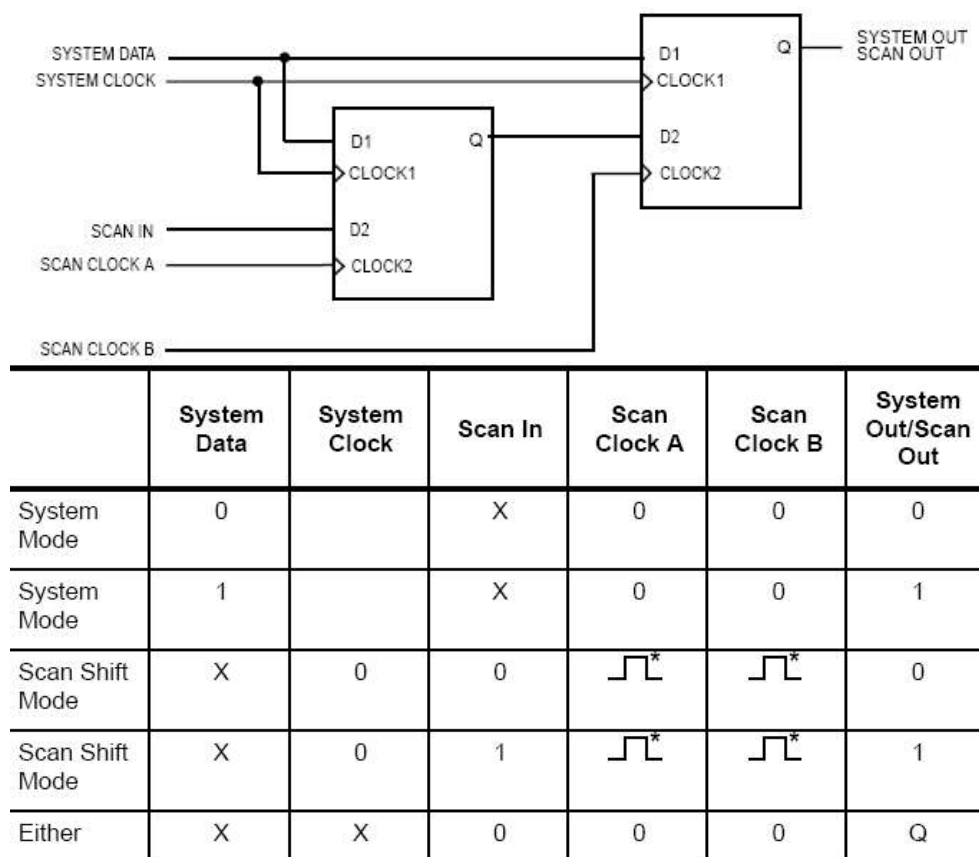
Rys. 12.15. Przerzutnik standardowy (na lewo) oraz zamiennik DFT w trybie Muxed Scan Style (strona prawa).

W stylu Clocked Scan Style przerzutnik ma dwa wejścia zegarowe: zegar systemowy oraz zegar skanowania. W czasie normalnej pracy zegar skanowania jest utrzymywany w stanie nieaktywnym a zegar systemowy pracuje normalnie. W czasie skanowania aktywność zegarów jest zamieniona.



Rys. 12.21. Przerzutnik DFT w stylu Clocked Scan Style.

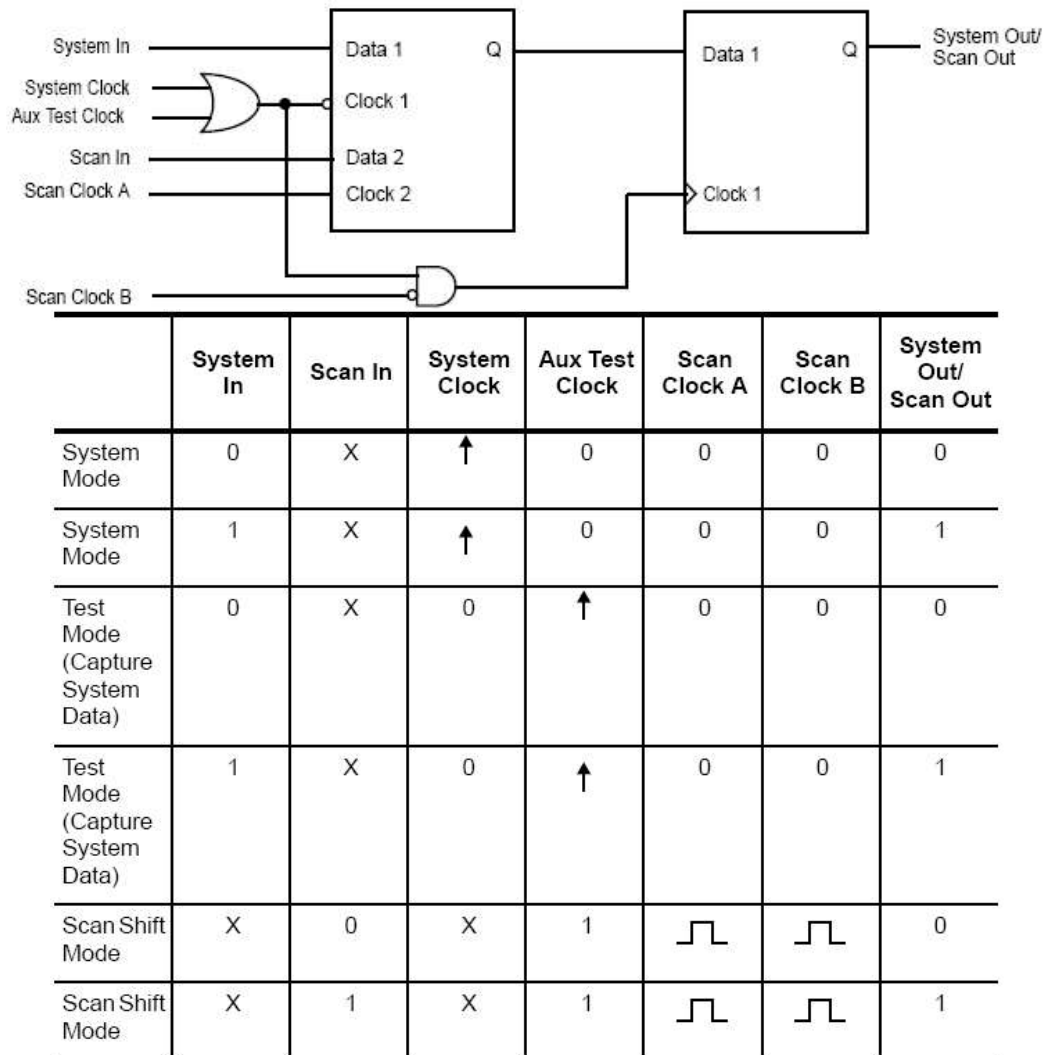
W stylu Clocked Level Sensitive Scan Design (LSSD) przerzutnik D jest zamieniany na komórkę LSSD która ma jeden zegar systemowy aktywowany zboczem oraz dwa zegary skanowania aktywowane poziomem. Schemat logiczny oraz tabela przejść przedstawione są na rysunku 12.22.



* Pulse on Scan Clock A precedes the pulse on Scan Clock B.

Rys. 12.22. Przerzutnik DFT w stylu Clocked LSSD Scan Style.

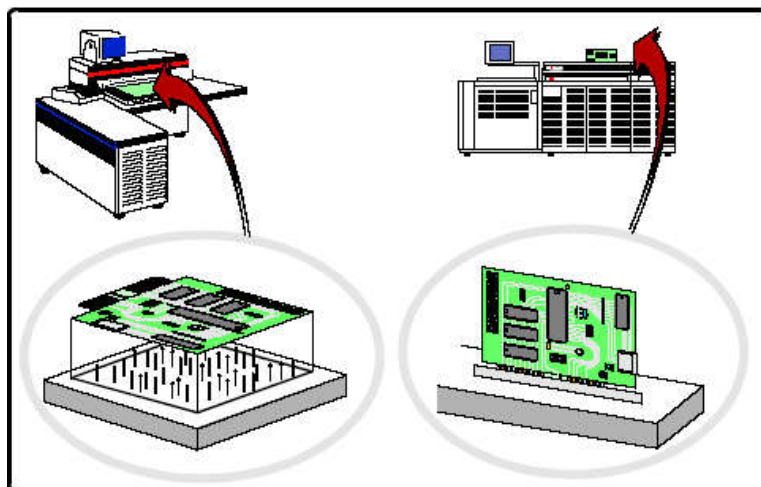
W trybie Auxiliary Clocked-LSSD Scan Cell przerzutnik zastępowany jest komórką przedstawioną na rysunku 12.23. W czasie normalnej pracy wszystkie 3 zegary skanowania są nieaktywne a zegar systemowy jest przekazywany do przerzutnika. W trybie skanowania aux_test_clock jest utrzymywany w stanie wysokim a zegary A i B są aktywowane naprzemiennie w celu szeregowego wyprowadzania danych. Dane na wejściu systemowym mogą zostać złapane poprzez impuls zegara systemowego albo poprzez impuls aux_test_clock w czasie gdy zegar systemowy jest w stanie niskim.



Rys. 12.23. Przerzutnik DFT w stylu Auxiliary Clocked Scan Style.

13. Interfejs IEEE 1149.1 (JTAG) [6,7].

13.1. Powstanie standardu 1149.1.

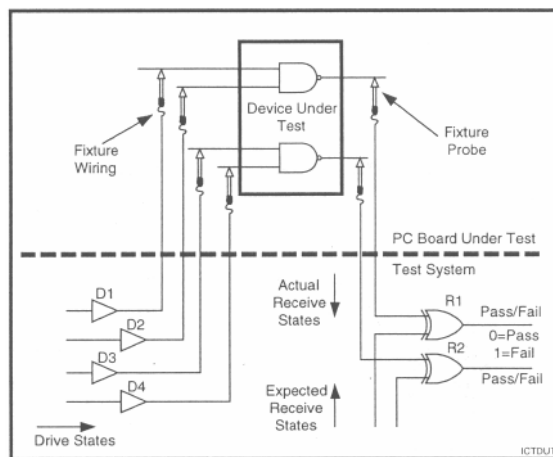


Rys. 13.1. Testowanie przy użyciu testera typu „łóże fakira” oraz testowanie funkcjonalne [6].

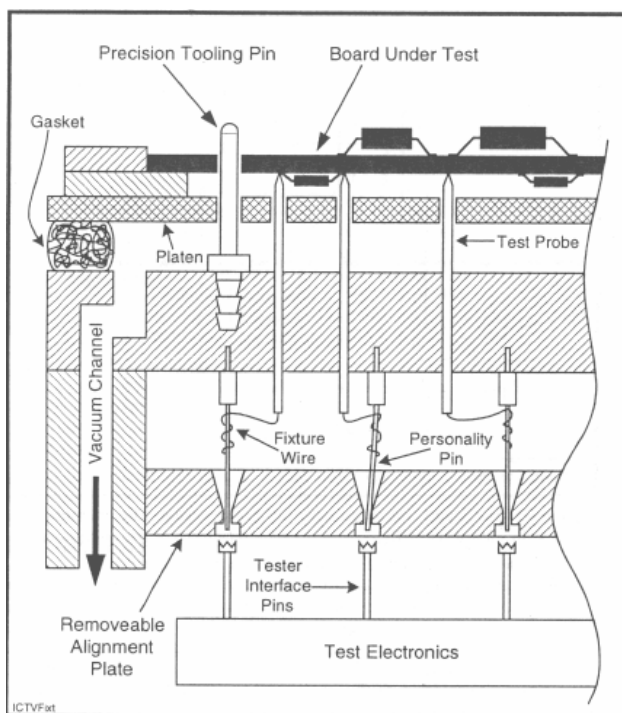
W latach 70-siątych testowanie zmontowanych układów odbywało się zazwyczaj poprzez wykorzystanie aparatury testującej oraz specjalnego złącza w postaci szeregu szpilek kontaktowych zwanych „łożem fakira”. W ten sposób można dotrzeć do każdego punktu znajdującego się na wierzchniej lub dolnej warstwie płytki drukowanej. Po dołączeniu badanej płytki do przystosowanego złącza następuje jej testowanie, które zazwyczaj odbywa się w dwóch fazach: testy z wyłączonym zasilaniem oraz testy z włączonym zasilaniem płytki drukowanej. Testy z wyłączonym zasilaniem sprawdzają kontakt z płytką drukowaną oraz wykonują pomiary impedancji i w ten sposób sprawdzają ewentualne wystąpienia zwarcień bądź przerw. Testy z włączonym zasilaniem polegają na podaniu pobudzeń do wybranego układu a następnie pomiarze i analizie odpowiedzi z tego układu. Inne elementy i układy w otoczeniu układu testowanego UUT (ang. unit under test) powinny być odporne na podawanie sygnałów z sond lub być wprowadzone w stan bezpieczny w którym nie nastąpi ich uszkodzenie. W ten sposób można sprawdzić obecność układu, jego orientację, oraz połączenia.

Powyższa technika opiera się na fizycznym kontakcie z badanym układem. W przypadku montażu przewlekłego taki dostęp jest fizycznie możliwy. W przypadku elementów montowanych powierzchniowo taki dostęp staje się utrudniony lub nawet niemożliwy (obudowy BGA). Obecny trend wymusza stosowanie coraz to mniejszych obudów i odstępów między wyprowadzeniami co powoduje, że technika testowania w oparciu o „łóże fakira” stała się nieefektywna i nieekonomiczna.

W latach 80-siątych grupa europejskich przedsiębiorstw założyła organizację nazwaną Joint European Test Action Group mającą na celu rozwiązanie problemów związanych z testowaniem coraz to większych układów. Później do grupy dołączyło kilka przedsiębiorstw ze Stanów Zjednoczonych co spowodowało zmianę nazwy organizacji na Joint Test Action Group (JTAG). Organizacja ta utworzyła standard IEEE1149.1 nazywany często również skrótem JTAG.

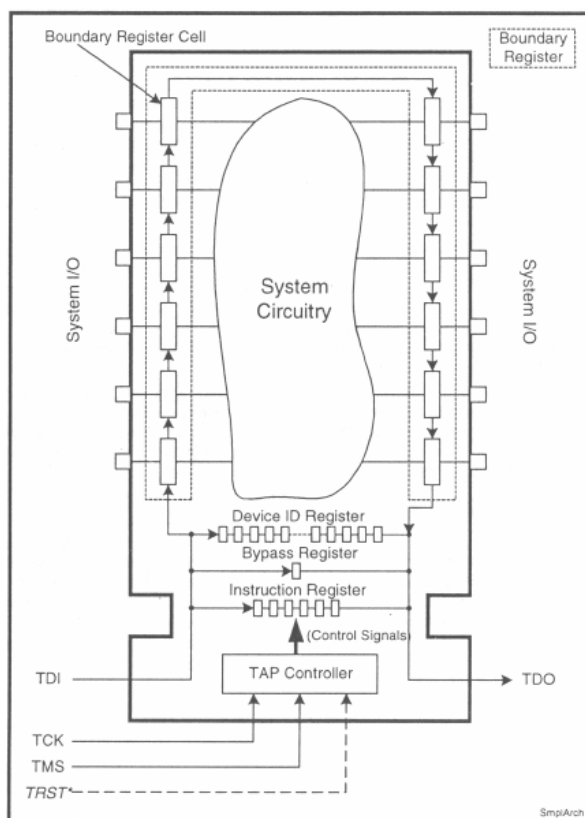


Rys. 13.2. Schemat blokowy testowania w układzie [7].



Rys. 13.3. Przekrój poprzeczny „łoża fakira” [7].

13.2. Architektura Boundary-Scan IEEE 1149.1.



Rys. 13.4. Ogólna architektura układu scalonego zgodnego ze standardem IEEE1149.1 [7].

Ogólna struktura układu scalonego zgodnego ze standardem IEEE1149.1 przedstawiona jest na rysunku 13.4. Układ taki w stosunku do układu bez części testującej Boundary-Scan, jest wzbogacony o następujące komponenty: kontroler TAP, rejestr przesuwany Boundary-Scan, rejestr instrukcji, rejestr obejścia (ang. bypass), rejestr identyfikacyjny – opcjonalny (ID), oraz zestaw rejestrów dodatkowych nieobowiązkowych i nieopisanych w IEEE1149.1.

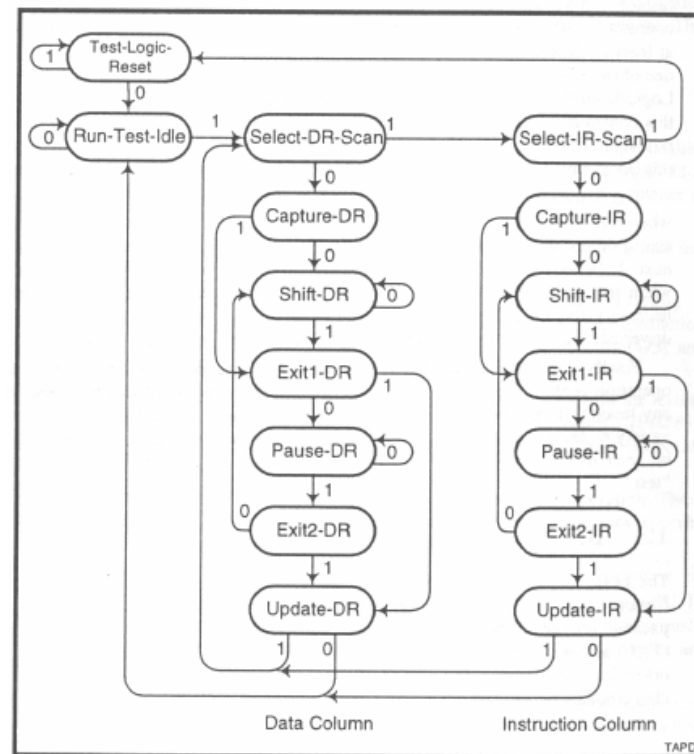
Każdy sygnał wyjściowy lub wejściowy jest wzbogacony o uniwersalną komórkę nazywaną komórką boundary-scan BSC (ang. Boundary-Scan Cell). Dodatkowo są umieszczone cztery (opcjonalnie 5) wyprowadzenia układu scalonego. Wyprowadzenia te dedykowane są do podłączenia do kontrolera TAP (ang. Test Access Port) i nie mogą być używane do innych celów. Wyprowadzenia te łącznie z pewnym protokołem służą do komunikacji z kontrolerem TAP i przez to z logiką testera Boundary-Scan.

Wyprowadzenie TRST* (reset kontrolera TAP, aktywny zerem) jest opcjonalne, nie podłączone powinno być nieaktywne. Sygnał TCK jest zegarem kontrolera, sygnał TMS jest wejściem sterującym kontroler TAP (stany 0 i 1 na rys. 13.5 odnoszą się do sygnału TMS). Sygnały TDI oraz TDO są odpowiednio wejściem i wyjściem aktywnego rejestru. Standard wymaga aby wejścia TMS, TDI oraz TRST* w przypadku niepodłączenia samoczynnie wchodziły do stanu wysokiego. To podnosi niezawodność systemu (co zostanie omówione później).

13.3 Kontroler TAP.

Kontroler TAP jest maszyną stanów o 16 stanach, schemat przejść przedstawiony jest na rysunku 13.5. Przejścia pomiędzy stanami następują na narastającym zboczach zegara TCK lub asynchronicznie przy aktywowaniu wejścia TRST* (podanie zera). Aktywowanie TRST* wprowadza kontroler w stan **Reset**. Sekwencja pięciu okresów zegara w czasie których sygnał TMS jest w stanie wysokim również wprowadza kontroler w stan **Reset** niezależnie od stanu początkowego. Każdy ze stanów kontrolera TAP ma swoją etykietę. Wejście do odpowiedniego stanu warunkowane jest wartością sygnału TMS. Wartości te widnieją obok strzałek wskazujących przejścia kontrolera, wartość ta musi pojawić się na TMS przed narastającym zboczem TCK. Na

rysunku 13.5 widoczne są dwie kolumny nazwane Data oraz Instrukcja. Każda z tych kolumn zawiera 7 stanów. Etykiety w obu kolumnach są podobne – różnią się wyłącznie typem rejestru do którego się odnoszą. Przejścia w obu kolumnach są identyczne.



Rys. 13.5. Schemat przejść maszyny stanów kontrolera TAP [7].

Funkcje poszczególnych stanów kontrolera TAP opisane są poniżej:

TEST-LOGIC-RESET: Jest to stan resetu kontrolera TAP. W tym stanie układ scalony pracuje jakby części testowej w nim nie było. Rejestr instrukcji jest inicjalizowany zawartością instrukcji IDCODE jeśli układ zawiera opcjonalny rejestr ID. Jeśli nie ma rejestru ID rejestr instrukcji jest zapelniany instrukcją BYPASS. Aktywowanie TRST* wprowadza asynchronicznie kontroler w stan reset. Sekwencja pięciu okresów zegara w czasie których sygnał TMS jest w stanie wysokim również wprowadza kontroler w stan reset niezależnie od bieżącego stanu maszyny stanów. Włączenie zasilania także powinno prowadzić do wejścia w ten stan.

RUN-TEST/IDLE: Po wejściu do tego stanu kontroler pozostaje w nim tak długo jak długo TMS jest w stanie niskim. Kiedy TMS jest w stanie wysokim następnym stanem kontrolera jest SELECT-DR-SCAN. W stanie RUN-TEST/IDLE aktywność logiki testującej występuje tylko wtedy jeśli pewne instrukcje są obecnie aktywne. Na przykład instrukcja RUNBIST powoduje wykonanie samotestowania układu scalonego. Inne instrukcje testujące mogą być także wykonywane w tym stanie jeśli były wcześniej w ten sposób zaprojektowane. Dla instrukcji które nie mają funkcji do wykonania w tym stanie, wszelkie rejestry wybrane przez bieżącą instrukcję pozostają niezmiennione.

SELECT-DR-SCAN: Ten stan jest stanem tymczasowym i przy kolejnym narastającym zboczu zegara TCK zmieni się albo na SELECT-IR-SCAN albo na CAPTURE-DR.

SELECT-IR-SCAN: Ten stan jest stanem tymczasowym i przy kolejnym narastającym zboczu zegara TCK zmieni się albo na TEST-LOGIC-RESET albo na CAPTURE-IR.

CAPTURE-IR: W tym stanie rejestr przesuwu (rejestry są zbudowane z dwóch zestawów: części przesuwnej (szeregowej) oraz części pamiętającej (równoległej), kiedy jest tu mowa o wyborze lub przesuwaniu rejestru tyczy się to części włączonej pomiędzy TDI a TDO czyli części przesuwnej) instrukcji IR ładuje w sposób równoległy wzorzec wartości na narastającym zboczu TCK. Dwa najmniej znaczące bity wypełniane są wartością „01”. W tym opracowaniu każda wartość jest przedstawiana jako MSB po stronie lewej i LSB po stronie prawej. Najmniej znaczący bit wchodzi

jako pierwszy do rejestru przesuwne. Każdy bit rejestru IR o większym priorytecie może otrzymać dowolną wartość. Całość wpisywanego słowa nie necessarily musi być instrukcją.

SHIFT-IR: W tym stanie rejestr instrukcji IR jest włączony pomiędzy TDI a TDO. Na każdym narastającym zboczu zegara TCK wpisywana i przesuwana jest nowa wartość do rejestru instrukcji. Jeśli TMS jest w stanie niskim do rejestru instrukcji można wpisać dowolnie długi ciąg. Jeśli TMS jest w stanie wysokim kontroler przechodzi do stanu EXIT1-IR. Jak można zobaczyć na rysunku 13.5, jest możliwy powrót do stanu SHIFT-IR poprzez przejście stanów EXIT1-IR, PAUSE-IR, EXIT2-IR. Jest to ważne jeśli zewnętrzny kontroler (nazywany master) ładuje bity instrukcji ale nie ma dostatecznie dużej pamięci aby wykonać pełne ładowanie w jednym ciągu. Można wówczas czynność przesłania dużej ilości danych rozbić na kilka ciągów a pomiędzy nimi wprowadzać TAP w stan PAUSE-IR aż do przygotowania kolejnej porcji danych.

EXIT1-IR: Jest to stan tymczasowy w którym należy podjąć decyzję czy kolejnym stanem będzie CAPTURE-IR czy też UPDATE-IR, który kończy proces skanowania instrukcji.

PAUSE-IR: W tym stanie kontroler jest chwilowo zatrzymany. Stan pomocny w oczekiwaniu na przygotowanie kolejnej porcji danych przekazywanych do rejestru instrukcji IR.

EXIT2-IR: Stan tymczasowy w którym podejmowana jest decyzja czy przejść do ponownego skanowania instrukcji czy też do zakończenia skanowania – stan UPDATE-IR.

UPDATE-IR: W tym stanie instrukcja wcześniej wprowadzona w sposób szeregowy (wskanowana) jest zapamiętywana w części pamiętającej rejestru - na opadającym zboczu TCK. W momencie zapamiętania instrukcji staje się ona bieżącą aktualną instrukcją ustawiającą nowy tryb pracy. Następnym stanem może być SELECT-DR-SCAN jeśli TMS jest w stanie wysokim lub RUN-TEST/IDLE jeśli TMS jest w stanie niskim.

CAPTURE-DR: W tym stanie dane mogą zostać w sposób równoległy wpisane do części szeregowej rejestru DR na narastającym zboczu TCK.

SHIFT-DR: W tym stanie rejestr wybrany przez aktualną instrukcję jest włączony pomiędzy TDI a TDO. Przy każdym narastającym zboczu TCK dane są przesuwane z TDI do TDO. Podobnie jak w kolumnie Instrukcja można czasowo wejść w stan PAUSE-DR w celu przygotowania nowej porcji danych a następnie ponownie wejść do stanu SHIFT-DR.

EXIT1-DR: Jest to stan tymczasowy w którym należy podjąć decyzję czy kolejnym stanem będzie CAPTURE-DR czy też UPDATE-DR, który kończy proces skanowania instrukcji.

PAUSE-DR: W tym stanie kontroler jest chwilowo zatrzymany. Stan pomocny w oczekiwaniu na przygotowanie kolejnej porcji danych przekazywanych do rejestru danych DR.

EXIT2-DR: Stan tymczasowy w którym podejmowana jest decyzja czy przejść do ponownego skanowania danych czy też do zakończenia skanowania – stan UPDATE-DR.

UPDTAE-DR: Niektóre z rejestrów danych mogą zawierać część równoległą – wówczas na zboczu opadającym TCK przepisywana jest zawartość części szeregowej do części równoległej. Dane zapamiętywane w części równoległej są zmieniane tylko w tym stanie. Następnym stanem może być SELECT-DR-SCAN jeśli TMS jest w stanie wysokim lub RUN-TEST/IDLE jeśli TMS jest w stanie niskim.

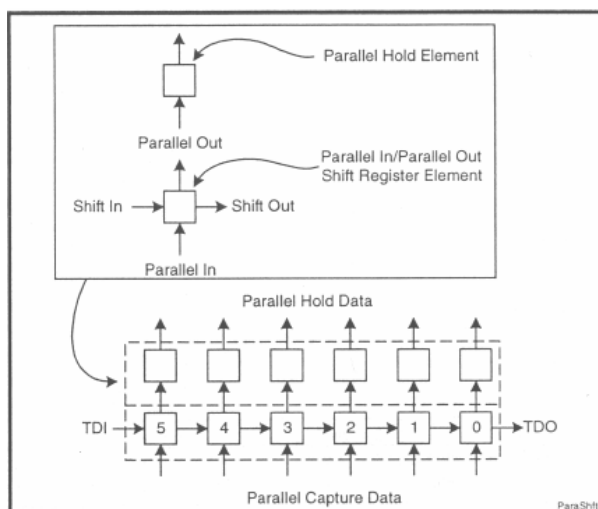
Dodatkowe wyjaśnienia:

- Dwa stany SHIFT-IR oraz SHIFT-DR w których następuje szeregowe wpisywanie odpowiednio do rejestrów instrukcji oraz danych aktywują wyprowadzenie TDO. To wyprowadzenie pozostaje aktywne aż do opadającego zbocza TCK w stanie EXIT1-IR lub EXIT1-DR. W pozostałym czasie sterownik TDO jest wyłączony co oznacza stan wysokiej impedancji na tym wyprowadzeniu.
- W obu stanach (t.j. UPDATE-IR oraz UPDATE-DR) proces aktualizacji części równoległej rejestru przeprowadzany jest na opadającym zboczu zegara TCK.
- W obu stanach (t.j. CAPTURE-IR oraz CAPTURE-DR) dane są wpisywane do szeregowej części rejestru na opadającym zboczu TCK. Ta operacja nazywana jest operacją odczytu. W połączeniu z operacją zapisu obie te operacje dają możliwość układowi Boundary-Scan do łącznego zapisu i odczytu pojedynczego bitu w czasie 2,5 okresu zegara TCK.

- Dane są wyprowadzane na wyjście TDO na opadającym zboczach TCK w obu stanach przesuwania danych. Dane są jednak wprowadzane do TDI na narastających zboczach TCK. To powoduje powstanie opóźnienia o wielkości 0,5 okresu TCK pomiędzy wyjściem TDO a „chwyceniem” danej przez TDI.

13.4 Rejestr instrukcji IR.

Schemat rejestru przedstawiony jest na rysunku 13.5. Rejestr instrukcji składa się z części szeregowej i równoległej. Część szeregową jest ładowana równoległe podczas stanu CAPTURE-IR i może być przesuwana pomiędzy wyprowadzeniami TDI a TDO w stanie SHIFT-IR. Część równoległa jest ładowana zawartością części szeregową w stanie UPDATE-IR. Standard IEEE 1149.1 definiuje wiele obowiązkowych i opcjonalnych instrukcji. Instrukcje te zostaną omówione w dalszej części materiału. Możliwe jest implementowanie instrukcji dodatkowych przez projektanta układu scalonego. Minimalna długość rejestru instrukcji wynosi 2.



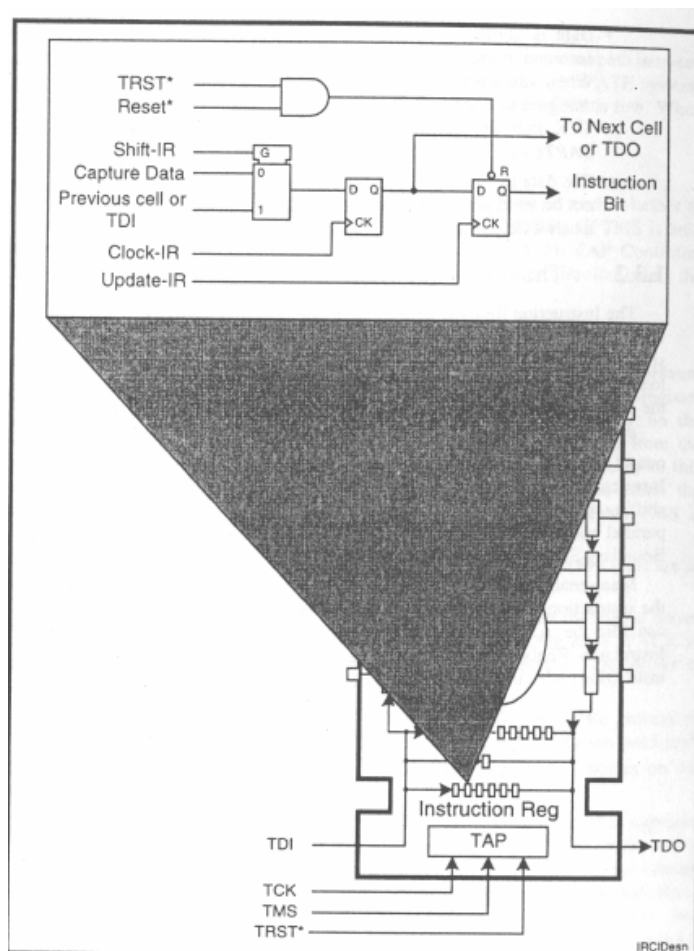
Rys. 13.5. Architektura rejestru instrukcji IR [7].

Dwa najmniej znaczące bity IR, muszą w stanie CAPTURE-IR zostać ustawione do wartości „01” (Bity te mogą zostać wykorzystane do testowania integralności logiki JTAG). Bardziej znaczące bity tego rejestru mogą wypełniać się zawartością zależną od konkretnego układu scalonego. Instrukcja wprowadzona w sposób szeregowy w stanie SHIFT-IR jest następnie w stanie UPDATE-IR przepisywana do części równoległej. Wartość wpisana do części równoległej ustala w następnym kroku tryb pracy oraz dołączony rejestr danych.

TAP Controller State	Shift Register Flip-Flops	Parallel Output Latches
TEST-LOGIC-RESET	Undefined	Set to give the IDCODE instruction if a Device Identification Register is present, or BYPASS if no Device ID Register exists
CAPTURE-IR	Load “01” into LSBs and design-specific data into any MSBs	Retain last state
SHIFT-IR	Shifts instruction bits towards the serial output	Retain last state
EXIT1-IR EXIT2-IR PAUSE-IR	Retain last state	Retain last state
UPDATE-IR	Retain last state	Latch data from shift register flip-flops into the parallel hold latches
All other states	Undefined	Retain last state

Tabela 13.1 Działanie rejestru instrukcji IR w poszczególnych stanach kontrolera TAP [7].

Kiedy przyłożony jest sygnał resetu do TRST* lub kiedy kontroler TAP wchodzi do stanu TEST-LOGIC-RESET, do rejestru instrukcji wpisywana jest jedna z 2 następujących instrukcji. Jeśli układ posiada rejestr ID (identyfikacji) wówczas do części równoległej wpisywana jest instrukcja IDCODE. W przeciwnym przypadku wpisywana jest instrukcja BYPASS. W tabeli 13.1 podsumowano zachowanie się rejestru instrukcji w każdym ze stanów kontrolera TAP.



Rys. 13.6. Przykład realizacji rejestru instrukcji IR. Rozwinięta komórka przedstawia szereg sygnałów generowanych przez kontroler TAP [7].

Na rysunku 13.7 przedstawiona jest przykładowa realizacja pojedynczej komórki rejestru instrukcji. Sygnały **Capture Data** oraz **Instruction Bit** są równoległym wejściem i wyjściem. Sygnały **Previous Cell or TDI** oraz **To Next Cell or TDO** są odpowiednio szeregowym wejściem i wyjściem. Sygnał **Clock-IR** jest wytworzony na podstawie TCK, sygnał **Update-IR** jest wytworzony na podstawie zanegowanego sygnału TCK. Sygnał **Shift-IR** jest jedynką w stanie kontrolera SHIFT-IR. Sygnał **Reset*** jest w stanie niskim jedynie dla TEST-LOGIC-RESET. Jeśli TRST*=0 wówczas rejestr równoległy zostanie ustawiony do instrukcji BYPASS lub IDCODE.

13.5 Rejestry danych.

Wszystkie instrukcje kontrolera TAP umieszczają odpowiadający im rejestr pomiędzy TDI a TDO. Rejestr taki nazywany jest rejestrem celu. To zapewnia fundamentalną właściwość układu Boundary-Scan, że pomiędzy TDI a TDO zawsze jest umieszczony rejestr. Funkcja realizowana przez dany rejestr jest podyktowana bieżącym rozkazem załadowanym do rejestru instrukcji IR. Ogólna architektura każdego z rejestrów może być identyczna jak rejestru instrukcji przedstawionego na rysunku 13.5.

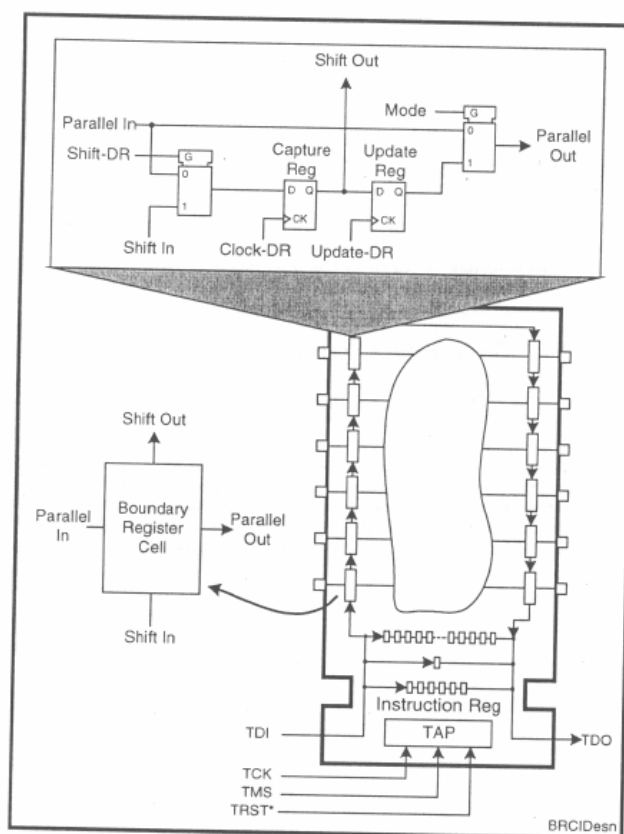
Rejestr obejścia (BYPASS). Jest to obowiązkowy rejestr. Składa się z jednej komórki ładowanej logicznym 0 w stanie CAPTURE-DR. Jest to użyteczne w czasie testowania integralności łańcucha JTAG.

Rejestr identyfikacji (Device_ID). Nieobowiązkowy rejestr. Jeśli jest zaimplementowany musi mieć długość 32 bitów. Rejestr zawiera dane identyfikujące element. Instrukcje odnoszące się do tego rejestru to IDCODE oraz USERCODE. Rejestr nie musi zawierać części równoległej. W stanie CAPTURE-DR do rejestru wpisywana jest 32 bitowa stała identyfikująca (jej format będzie omówiony później). Można używać tego rejestru do testowania integralności łańcucha testowego.

Rejestr Boundary-Scan. Najważniejszy rejestr układu. Zawiera komórki sąsiadujące z każdym wejściem i wyjściem układu cyfrowego (oprócz wyprowadzeń kontrolera TAP). Rejestr ten służy do ustawiania i obserwacji sygnałów na wyprowadzeniach zewnętrznych układu. Jest to rejestr obowiązkowy w standardzie IEEE 1149.1 i zostanie omówiony szczegółowo w dalszej części materiału.

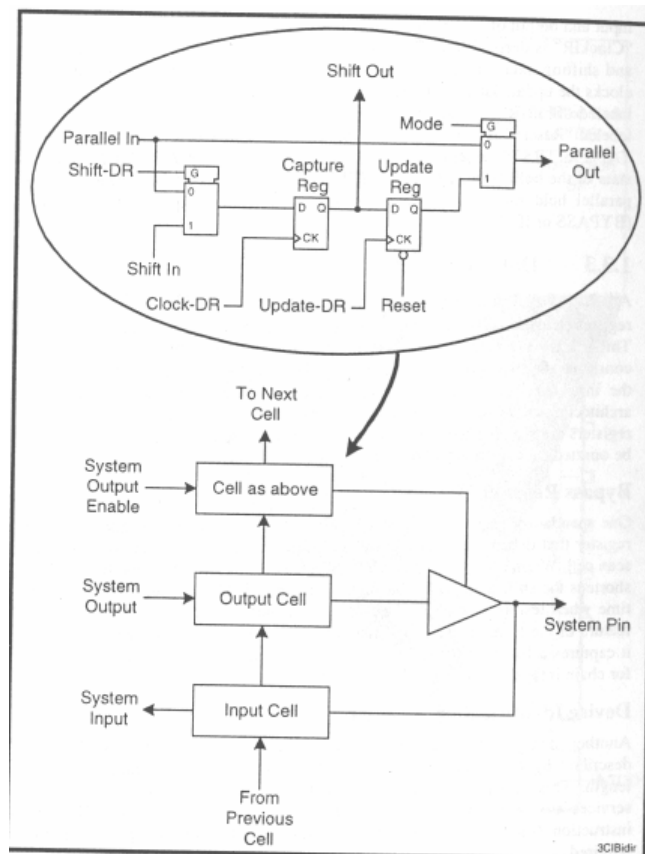
Rejestry użytkownika. Standard umożliwia dodawanie dowolnej ilości rejestrów użytkownika. Te rejestry w połączeniu z instrukcjami użytkownika, mogą zostać wykorzystane do własnych testów lub innych funkcji. W przypadku wybrania, rejestr użytkownika musi być włączony pomiędzy TDI a TDO.

13.6 Rejestr Boundary-Scan.



Rys. 13.7. Typowa budowa komórki rejestru Boundary-Scan [7].

Na rysunku 13.8 przedstawiona jest budowa typowej komórki rejestru Boundary-Scan. Projekt tej komórki jest na tyle uniwersalny, że może być użyty zarówno do wyprowadzeń wejściowych jak i wyjściowych. Sygnały **Parallel IN** oraz **Parallel Out** są połączone z wyprowadzeniem zewnętrznym lub siecią wewnętrzną układu w zależności od roli komórki. Na przykład, jeśli komórka jest użyta przy pinie wejściowym wówczas **Parallel IN** jest połączone do pinu układu scalonego natomiast **Parallel Out** do wnętrza układu scalonego. Dla wyprowadzenia wyjściowego układu połączenia powyższe są odwrócone. Na rysunku 13.8 sygnały **Shift In** oraz **Shift Out** są szeregowymi sygnałami części szeregowej rejestru i formują ścieżkę przesuwczą rejestru Boundary-Scan.



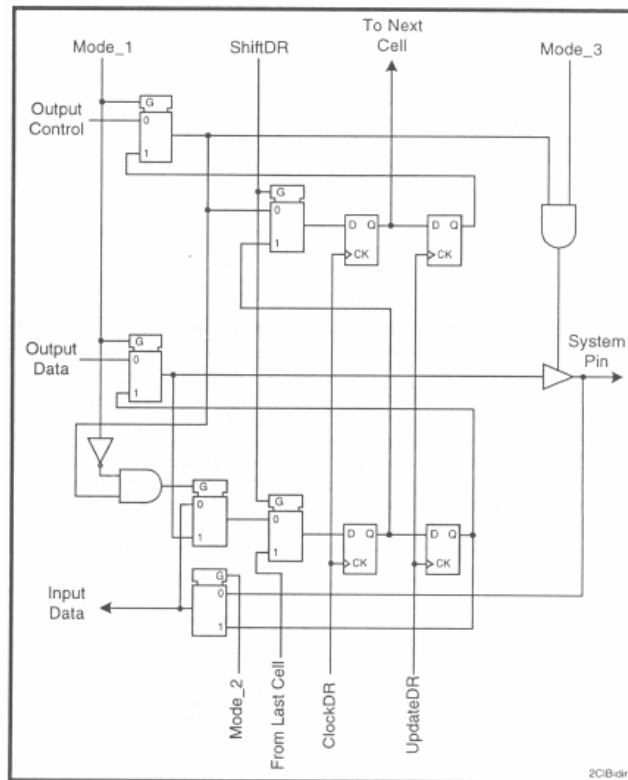
Rys. 13.8. Wyprowadzenie dwukierunkowe z oddzielnymi komórkami Boundary-Scan [7].

W celu umieszczenia linii dwukierunkowych w rejestrze Boundary-Scan można zastosować dwa podejścia. Pierwszym z nich jest zastosowanie 3 komórek w miejsce jednej jak to przedstawiono na rysunku 13.8. Drugim rozwiązaniem jest zastosowanie nieco bardziej złożonej komórki, której schemat jest przedstawiony na rysunku 13.9. W obu przypadkach stosowana jest dodatkowa komórka (jak w górnej części rysunku 13.9), która daje łańcuchowi Boundary-Scan kontrolę nad sygnałem sterującym bramką 3-stanową. Standard umożliwia sterowanie przez jedną dodatkową komórkę zestawem sygnałów dwukierunkowych, jednakże nakłada dodatkowe ograniczenie polegające na tym iż każda ze sterowanych bramek 3-stanowych powinna zachowywać się identycznie w stosunku do zawartości sterującej (wszystkie ON lub wszystkie Z).

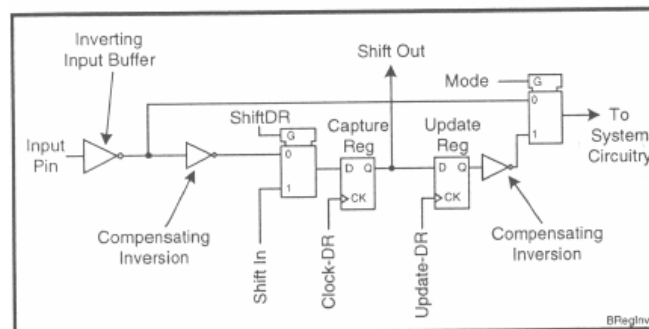
Nie wliczając komórki sterującej, komórka dwukierunkowa z rysunku 13.10 ma tą zaletę iż zajmuje tylko jedno miejsce w łańcuchu Boundary-Scan w porównaniu do dwóch dla komórki z rysunku 13.8. Podczas gdy rzeczywisty zysk w powierzchni krzemu jest niewielki, redukcja długości łańcucha jest bardzo znacząca.

Naturalną cechą komórki z rysunku 13.9 jest możliwość zapamiętywania stanu linii wejściowej niezależnie od tego co wytwarza bramka na wyjściu. To umożliwia tworzenie oprogramowania które może obserwować rozbieżność pomiędzy wytwarzanym a rzeczywistym sygnałem na wyprowadzeniu układu scalonego. Można w ten sposób stwierdzić uszkodzenie bramki wejściowej lub zwarcie wyjścia.

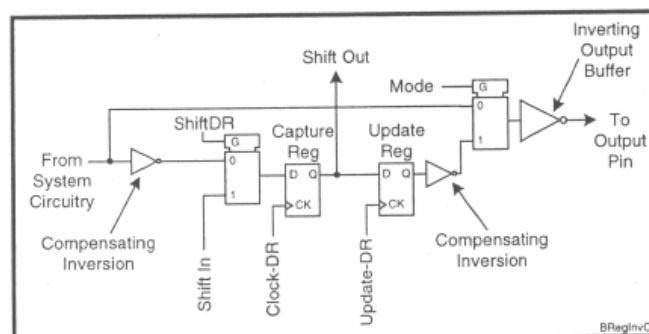
Czasami zachodzi przypadek, iż sygnał jest negowany w komórce bufora wejściowego lub wyjściowego. Równocześnie standard sztywno określa, że nie można w rejestrach przechowywać wartości zanegowanych. Wówczas należy zastosować komórki BSC kompensujące wpływ inwerterów jak to przedstawiono na rysunku 13.10.



Rys. 13.9. Dwukierunkowa komórka Boundary-Scan [7].



(a)



(b)

Rys. 13.10. Inwertery kompensujące dla komórki wejściowej (a) oraz wyjściowej (b) [7].

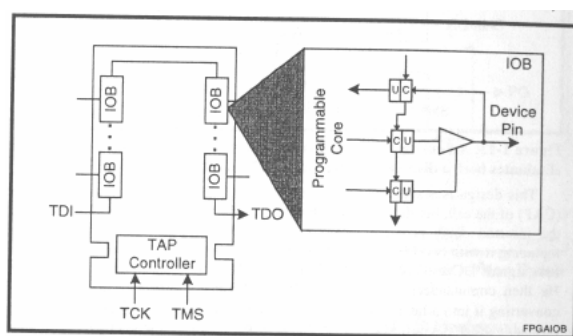
Rejestr Boundary-Scan może zawierać komórki które nie mają żadnego zastosowania. Takie komórki nazywane są komórkami wewnętrznymi. Komórki te nie są połączone z wyprowadzeniami I/O ani z wejściami Enable. Spotyka się je najczęściej w układach programowalnych gdzie dla dwukierunkowych I/O są wstawione po trzy komórki (gdyż nie znamy późniejszego przeznaczenia danego wyprowadzenia) a po zaprogramowaniu układu niezbędne są już tylko pojedyncze cele.

13.7. Układy programowalne.

Układy programowalne są „kameleonem” w świecie układów scalonych. Są to „puste strony” które mogą zostać zapisane podczas procesu programowania. Proces programowania może być powtarzany wielokrotnie i w każdym czasie. Z tego względu układy programowalne stanowią dużą niedogodność w testowaniu na poziomie płytki drukowanej. Poprzez zróżnicowaną naturę ich logika jest zmienna i trudna do testowania.

Układy programowalne, pod względem ich testowania, są dostępne w 2 gatunkach: całkowicie „puste” układy nie zawierające żadnej logiki oraz zawierające małą ilość logiki realizującą na stałe łańcuch i kontroler Boundary-Scan. Przykładami takich układów są np. XL4000 oraz XC 9500 firmy Xilinx.

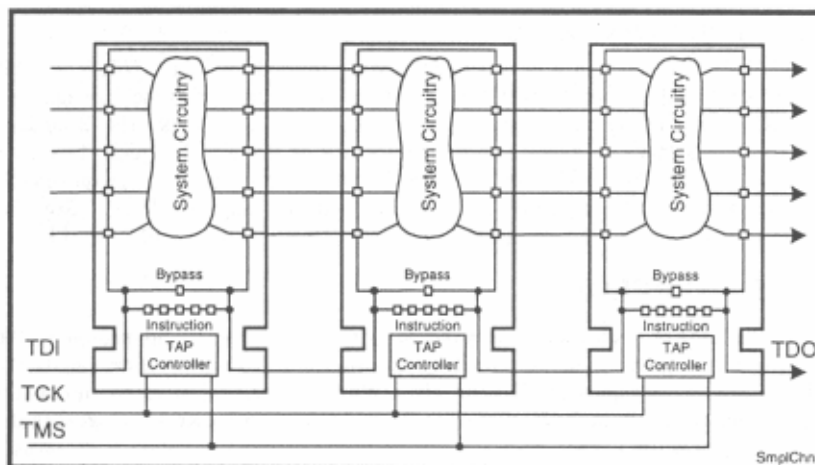
Układy „puste” zawsze mogą zostać zaprogramowane do pełnienia funkcji Boundary-Scan. W rzeczywistości można w nie wprogramować wyłącznie funkcjonalność Boundary-Scan na czas testowania. Docelowa funkcjonalność może zostać zaprogramowana w czasie po testowaniu. Oczywiście przed zaprogramowaniem układ nie jest zgodny ze standardem.



Rys. 13.14 Układ programowalny z kontrolerem Boundary-Scan wstawionym na stałe[7].

W rodzinie XL4000 oraz XC9500 Boundary-Scan jest wstawiony na stałe jak pokazano na rysunku 13.14.

13.8. Łańcuchy układów Boundary-Scan.



Rys. 13.15. Łańcuch układów Boundary-Scan [7].

Układy zawierające Boundary-Scan mogą być łączone w łańcuchy. W takim przypadku wyprowadzenia TCK, TMS oraz TRST* są łączone dla wszystkich układów razem, natomiast TDI oraz TDO są łączone szeregowo. Ważną właściwością jest to, że ponieważ sygnały sterujące wszystkich układów są ze sobą połączone, każdy z układów będzie w tym samym stanie kontrolera TAP. To oznacza, że pojedynczy graf przejść stanów wystarczy aby określić stan każdego z kontrolerów TAP wszystkich układów scalonych.

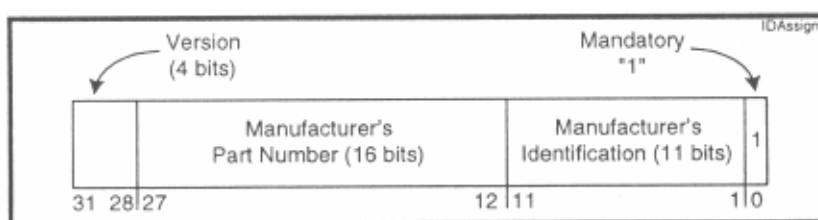
Możliwymi modyfikacjami połączeń są: kilka łańcuchów TDI-TDO w ramach tych samych sygnałów sterujących (skrócenie czasu programowania), kilka odrębnych łańcuchów z różnymi sygnałami sterującymi.

13.10 Nieinwazyjne tryby pracy.

Kontroler TAP oraz jego wyprowadzenia mogą pracować asynchronicznie i niezależnie od pozostałej logiki układu scalonego. To umożliwia wykorzystanie Boundary-Scan w sposób nie zakłócający pracy właściwego układu, płytki PCB lub systemu na którym jest umiejscowiony – jeśli wykorzystujemy nieinwazyjne tryby pracy, wyszczególnione poniżej:

BYPASS: Instrukcja ta umieszcza rejestr obejścia pomiędzy wyprowadzenia TDI a TDO. Celem instrukcji jest skrócenie ścieżki danych. Instrukcja oraz rejestr docelowy są obowiązkowe w standardzie IEEE 1148.1. Kod rozkazu zawierający same jedynki musi zostać zdekodowany jako instrukcja BYPASS. Inne, dodatkowe kody instrukcji BYPASS są również dopuszczalne. W czasie gdy instrukcja BYPASS jest aktywna, rejestr obejścia ładowany jest zerem podczas każdego przejścia przez stan CAPTURE-DR. Czynność ta powoduje inicjowanie rejestru znaną stałą wartością.

IDCODE: Instrukcja IDCODE umieszcza rejestr identyfikacji pomiędzy wyprowadzenia TDI a TDO. Instrukcja jest opcjonalna. Brak wymagań na kod rozkazu IDCODE. Rejestr identyfikacji jest ładowany równolegle stałą wartością w czasie każdego przejścia przez stan CAPTURE-DR (jeśli aktywna jest instrukcja IDCODE). Najmniej znaczący bit każdego słowa identyfikacji musi mieć wartość 1. Ten bit jest jako pierwszy wystawiany na wyprowadzenie TDO. Pozostałe bity mają znaczenie jak na rysunku 13.15.



Rys. 13.15. Przypisanie bitów rejestru identyfikacji [7].

Bity 31-28 są numerem wersji układu scalonego. Numer wersji powinien być zmieniony za każdym razem kiedy układ scalony został poprawiony. Bity 27-12 są numerem układu przypisanym przez producenta. Bity 11-1 są identyfikatorem producenta przypisanym poprzez organizację JEDEC (ang. Joint Electron Device Engineering Council). Kod „00001111111” jest zarezerwowany i nie może być użyty jako kod producenta. Można ten kod wprowadzić na TDI, wówczas pojawienie się takiego kodu na wyjściu TDO łańcucha nieznanymi układów oznacza, że został przeskanowany cały łańcuch. Instrukcja IDCODE umożliwia identyfikację zamontowanego układu poprzez port JTAG.

Zakłada się, że producenci układów na licencji powinni mieć inny nr identyfikacyjny. Należy być także ostrożnym w przypadku stosowania zamienników – mogą one mieć zaimplementowaną inną wersję IEEE1149.1 (lub wcale jej nie mieć).

Stan TEST-LOGIC-RESET ustala rejestr instrukcji wartością odpowiadającą instrukcji BYPASS lub IDCODE jeśli rejestr identyfikacji jest zaimplementowany. To umożliwia bezpośrednie przejście do skanowania łańcucha układów bez konieczności wstępnego ładowania ich instrukcjami. Skanowanie takie nazywane jest *ślepy przeszuchaniem* gdyż nie jest potrzebna znajomość zastosowanych układów jak również ich liczby i kolejności w łańcuchu. Układy posiadające rejestr identyfikacji jako pierwszą wartość wyprowadzą jedynkę a pozostałe 31 wartości jest ich identyfikatorem. Układy bez rejestru identyfikacji jako pierwszą daną wystawiają zero. Ślepe przeszuchanie może zostać użyte do rozpoznania konfiguracji systemu (również do sprawdzenia poprawności wlutowanych elementów, sprawdzenia działania łańcucha BS).

USERCODE: Instrukcja ta umieszcza ten sam rejestr identyfikacji pomiędzy TDI a TDO ale w stanie CAPTURE-DR jest on w sposób równoległy ładowany słowem zdefiniowanym przez

użytkownika. Instrukcja USERCODE jest opcjonalna ale jeśli jest zaimplementowana musi być również zaimplementowana instrukcja IDCODE. Celem tej instrukcji jest rozszerzenie instrukcji IDCODE o dodatkowe 32 bity. Możliwym zastosowaniem są np. układy programowalne gdzie każdy układ ma taki sam kod ID a funkcja układu może przez użytkownika zostać zidentyfikowana z użyciem instrukcji USERCODE.

SAMPLE: Instrukcja obowiązkowa. Kod instrukcji nie jest ustalony w standardzie. Instrukcja włącza rejestr BS pomiędzy TDI a TDO. Nie powoduje to jednak odłączenia logiki układu od jego wyprowadzeń. Odpowiada to podaniu wartości 0 na sygnał Mode w komórce BS z rysunku 13.7. W czasie przejścia przez stan CAPTURE-DR wszystkie przerzutniki Capture zapamiętują stany sygnałów do których są połączone, tj. wejścia układu scalonego lub wyjścia logiki układu przyłączone jako sygnały wyjściowe. Rejestr BS w stanie CAPTURE-DR zapamiętuje stany sygnałów wszystkich wyprowadzeń I/O układu scalonego. Dane te mogą być później odczytane w celu przeanalizowania. W zasadzie można zrealizować funkcjonalność analizatora logicznego z użyciem instrukcji SAMPLE.

PRELOAD: Instrukcja obowiązkowa. Kod instrukcji nie jest ustalony przez standard. Instrukcja włącza rejestr BS pomiędzy wyprowadzenia TDI oraz TDO. Nie powoduje odłączenia logiki układu od jego wyprowadzeń I/O. Służy do inicjalizacji przerzutników Capture ustalonym znanym stanem (poprzez szeregowe wprowadzenie w stanie SHIFT-DR) a następnie wprowadzenie tej wartości do części równoległej rejestru BS w stanie UPDATE-DR. Nie powoduje przesłania tych danych do logiki układu (sygnał Mode z rysunku 13.8 w stanie 0). (Instrukcje SAMPLE oraz PRELOAD w poprzedniej wersji standardu IEEE 1148.1 były jedną instrukcją z takim samym kodem rozkazu i były nazywane SAMPLE/PRELOAD).

Instrukcja PRELOAD nie ma wymagań co do wartości zapamiętywanej w szeregowej części rejestru BS w stanie CAPTURE-DR. To umożliwia połączenie funkcjonalności instrukcji SAMPLE z instrukcją PRELOAD w ramach jednej instrukcji.

13.10. Tryby pracy z dostępem do wyprowadzeń układu.

Instrukcje z dostępem do wyprowadzeń I/O stanowią drugą ważną grupę instrukcji. Te instrukcje charakteryzują się przełączeniem komórki wyjściowej BS w ten sposób, że na wyjście przyłączona jest zawartość rejestru. To powoduje odłączenie wejść układu od logiki układu. Bardzo ważne jest zapewnienie braku uszkodzeń rdzenia układu poprzez tak radykalną zmianę konfiguracji. Można to wykonać na przykład poprzez podanie samoistnego automatycznego RESETu w czasie instrukcji z dostępem do wyprowadzeń. Co się jednak stanie jeśli układ był w stanie RESETu a następnie tryb z dostępem do wyprowadzeń zostanie usunięty i wewnętrzny RESET zostanie dezaktywowany? Jest to poważny problem (Lobotomy problem) dla układu, płytki i systemu. Co powinna zrobić wbudowana logika układu po przebudzeniu z trybu z dostępem do wyprowadzeń? Jedną z możliwych odpowiedzi jest przyłożenie głównego resetu do całego systemu. Inną możliwością jest chwilowe wyłączenie zasilania. Jeszcze inną możliwością jest takie zaprojektowanie logiki układu, że pamięta ona fakt odłączenia od wyprowadzeń I/O a ponowne przyłączenie jest możliwe dopiero po wykonaniu głównego resetu lub wyłączeniu zasilania.

EXTEST: Instrukcja obowiązkowa. Kod instrukcji nie jest ustalony przez standard. Nie zaleca się kodu w formie same zera, chociaż we wstępnej formie standardu było to obowiązkowe (możliwość wykonania takiej instrukcji przy zwarcu montażowym prowadzące w konsekwencji do potencjalnej możliwości uszkodzenia układu). Instrukcja włącza rejestr BS pomiędzy TDI a TDO. W stanie CAPTURE-DR wszystkie wejścia układu (i wyprowadzeń wejściowych w liniach dwukierunkowych) są zapamiętywane w odpowiadających im przerzutnikach szeregowych rejestru BS. Na rysunku 13.8 sygnał Mode jest w stanie 1. Sygnały wyjściowe układu scalonego są ustalone stanami przerzutników wyjściowych rejestru BS. Z tego względu w czasie wykonywania instrukcji EXTEST możemy próbować wartości sygnałów wejściowych układu scalonego oraz ustalać sygnały wyjściowe zawartością równoległą rejestru BS. Przesuwanie rejestru BS w stanie SHIFT-DR umożliwia jednocześnie zapis nowych wartości jak również odczyt wartości „złapanych” w

stanie CAPTURE-DR. Aktualizacja nowych wartości wyjściowych jest wykonywana w stanie UPDATE-DR. Instrukcja EXTEST jest „koniem roboczym” testowania Boundary-Scan.

INTEST: Instrukcja INTEST jest nieobowiązkowa. Standard nie ustala kodu instrukcji. INTEST włącza rejestr BS pomiędzy TDI a TDO. INTEST jest instrukcją testującą logikę układu – czyli jego wnętrze. Ustala wartości sygnałów wejściowych logiki układu do wartości równoległej rejestru BS (w stanie UPDATE-DR). W części szeregowej rejestru BS zapamiętywany jest stan sygnałów wyjściowych oraz zezwalających linii dwukierunkowych (stan CAPTURE-DR). W czasie przesuwania (SHIFT-DR) można równocześnie odczytywać wartości wyjść jak i wprowadzać nowe pobudzenia na wejścia logiki badanego układu. Należy równocześnie zapewnić odpowiednie sterowanie wyjściami układu scalonego. Możliwe są dwie opcje: pierwsza - BS steruje wartościami wyjść oraz druga - wyjścia I/O są w bezpiecznym stanie zablokowanym. Niezależnie która opcja jest wybrana musi być zastosowana jednolicie do wszystkich wyprowadzeń.

Instrukcję INTEST można użyć od testowania układów scalonych w czasie gdy są one zamontowane na płycie drukowanej. Problematiczne jest jednak wydłużenie czasu testowania ze względu na szeregowe wprowadzanie/wyprowadzanie danych. Wydłużenie czasu testowania jest wprost proporcjonalne do długości łańcucha rejestru BS. Jeśli badany system jest dynamiczny jego testowanie może okazać się nie możliwe do wykonania gdyż czas wprowadzenia pobudzeń może okazać się zbyt długi.

RUNBIST: Instrukcja opcjonalna. Kod instrukcji nie ustalony przez standard. RUNBIST wykorzystuje rejestr wskazany przez użytkownika. Celem instrukcji jest dostęp do wbudowanego samotestu układu poprzez kontroler TAP. W czasie wykonywania instrukcji RUNBIST sterowanie wyprowadzeniami wyjściowymi układu scalonego jest wykonywane jak dla instrukcji INTEST t.j. po pierwsze mogą one pozostawać pod kontrolą rejestru BS lub po drugie: być wprowadzone w stan wysokiej impedancji. W pierwszym przypadku odpowiednie wartości stanów są dostarczone poprzez wcześniejsze wykonanie instrukcji PRELOAD. Obie metody umożliwiają nam eliminację potencjalnych konfliktów sygnałów z innymi elementami na płycie drukowanej.

RUNBIST jest instrukcją samoinicjującą i nie potrzebuje żadnych wstępnych ustawień ani inicjalizacji stanów początkowych. Instrukcja włącza rejestr wskazany przez użytkownika pomiędzy TDI a TDO. Może to być specjalny dedykowany rejestr albo inny dostępny rejestr jak np. obejścia czy BS. Celem rejestru jest zebranie danych odnośnie wyniku samotestu. Wynik testu musi być:

- deterministyczny, wszystkie bity muszą być zdefiniowane,
- niezmienny dla wszystkich wersji układu scalonego,
- niezależny od wszelkiej aktywności na wyprowadzeniach I/O (czasami oprócz wyprowadzenia zegara).

W rzeczywistości test jest wykonywany w stanie RUN-TEST/IDLE. Zegarowanie może zostać dostarczone poprzez TCK, zegar systemowy lub poprzez oba wyprowadzenia. Czas wykonania samotestu może być długi i trwać wiele cykli zegarowych jednak pozostanie w stanie wykonywania testu przez czas dłuższy niż niezbędny do jego wykonania nie powinno powodować zmiany wyniku testów. To „zamrożenie” wyników testu umożliwia jego wykonanie dla wszystkich układów równocześnie poprzez pozostanie w stanie RUN-TEST/IDLE przez najdłuższy wymagany czas. Wynik testu jest zapamiętywany w docelowym rejestrze w stanie CAPTURE-DR. Następnie wynik ten może zostać wyprowadzony w celu analizy.

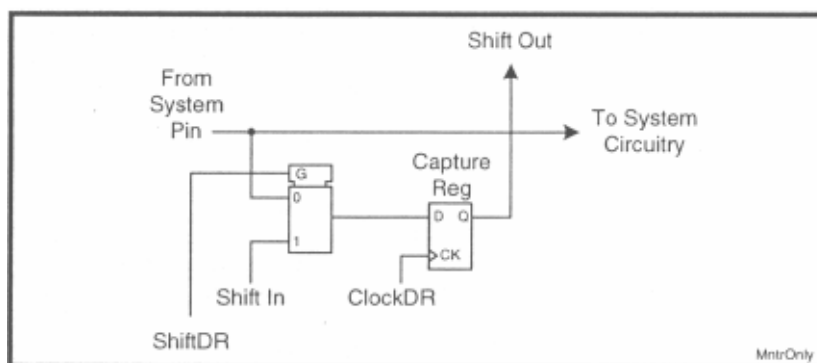
HIGHZ: Instrukcja wprowadzona do standardu w roku 1993 jako dodatek. Jest opcjonalna, kod instrukcji nie ustalony przez standard. Celem instrukcji jest poszerzenie możliwości stosowania sprzętu ATE (ang. Automatic Test Equipmemnt) złożonych systemów poprzez redukcję uszkodzeń przez przesterowanie. Poprzez załadowanie instrukcji HIGHZ powodujemy ustawienie wyprowadzeń wyjściowych i dwukierunkowych w stan wysokiej impedancji. Pomiedzy TDI a TDO włączany jest rejestr obejścia. Przejście do odłączenia wyprowadzeń następuje w stanie UPDATE-IR.

CLAMP: Instrukcja wprowadzona do standardu w roku 1993 jako dodatek. Jest opcjonalna, kod instrukcji nie ustalony przez standard. CLAMP włącza rejestr obejścia pomiędzy wyprowadzenie

TDI a TDO. Wyprowadzenia wyjściowe i dwukierunkowe są sterowane przez rejestr BS, który powinien być wcześniej zainicjowany instrukcją PRELOAD. Stany wyprowadzeń I/O stają się ważne po przejściu stanu UPDATE-IR. To umożliwia ustanowienie odpowiednich wartości sygnałów na wyjściach oraz skrócenie długości rejestru w celu szybszego ładowania danych do pozostałych układów w łańcuchu JTAG.

Wprowadzenie instrukcji CLAMP ma na celu cyfrową ochronę. Podczas testów PCB często zachodzi konieczność zapewnienia wartości „0” lub „1” na pewnych wyprowadzeniach w celu ustanowienia warunków testu.

Wyjątki ze względu na podawanie zegara. Dla wyprowadzeń które są szczególnie czułe na pogorszenie parametrów czasowych standard IEEE 1148.1 przewiduje możliwość użycia komórek z funkcjonalnością zapewniającą wyłącznie ich obserwowanie. Na rysunku 13.17 przedstawiono przykładowy schemat takiej komórki. Nie zawiera ona rejestru równoległego. Taka komórka nie umożliwia wykonania instrukcji INTEST oraz RUNBIST bo nie izoluje układu od sygnałów zewnętrznych. Standard dopuszcza jednak wyjątek, dla sygnałów zegarowych można zastosować komórkę jak na rysunku 13.17 a możliwe będzie wykonanie samotestów oraz INTEST. Niestety to komplikuje nieco wykonanie tychże testów gdyż muszą być skoordynowane z zewnętrznym zegarem systemowym.



Rys. 13.16. Komórka realizująca wyłącznie funkcję obserwacyjną [7].

13.10.1. Rozszerzalność standardu.

Bardzo ważną cechą IEEE 1148.1 jest rozszerzalność. Architektura może być rozszerzona na dwa sposoby: poprzez dodanie rejestrów oraz instrukcji użytkownika. Instrukcje użytkownika mogą być publiczne lub prywatne. Instrukcje publiczne muszą być dokładnie udokumentowane w danych układu scalonego. Instrukcje prywatne mogą być nieudokumentowane z wyjątkiem kodu instrukcji. Instrukcje użytkownika mogą odnosić się do rejestrów standardowych, części rejestrów standardowych ich dowolnych kombinacji oraz rejestrów własnych.

W ogólności standard IEEE1149.1 można uważać jako standard komunikacyjny niezbędny do implementacji nowych funkcji w układzie scalonym. Tymi funkcjami mogą być: testy na poziomie płytki, specjalne testy układu scalonego, testy hybrydowe układów analogowych i cyfrowych, wspieranie emulacji i wiele innych.

13.13. Cena oraz korzyści.

Cena:

- dodatkowe wyprowadzenia (4 lub 5),
- narzut PAD lub narzut die (PAD or die limited),
- zmniejszenie uzysku produkcyjnego,
- zwiększenie opóźnień sygnałowych,
- zwiększony czas projektowania,
- brak dyscypliny projektantów i niecałkowita zgodność układów z IEEE 1148.1.

Korzyści:

- automatyczna generacja testów,

- ponowne wykorzystywanie testów,
- standaryzacja testów,
- możliwość testowania układów których testowanie w sposób tradycyjny jest utrudnione (mały rozstaw wyprowadzeń, obudowa BGA)
- możliwość wykorzystania interfejsu do innych celów np. programowanie.

13.14. Trendy.

Do dzisiaj minęło 15 lat od wprowadzenia interfejsu JTAG. W czasie tym obserwuje się następujące trendy:

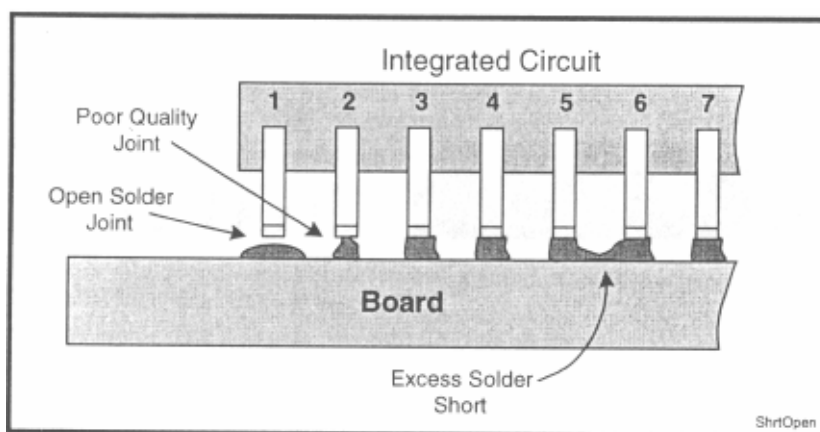
- coraz więcej układów zawiera interfejs JTAG,
- coraz większe wsparcie ze strony producentów oprogramowania EDA,
- coraz mniejszy procentowy koszt I148.1.

14. Zastosowanie interfejsu JTAG do testowania systemów cyfrowych [7].

Testowanie Boundary-Scan jest skierowane głównie do testowania układów cyfrowych, chociaż możliwe jest także testowanie układów analogowych i mieszanych. W tym rozdziale przedstawione są różne podejścia do testowania z wykorzystaniem interfejsu IEEE1149.1.

Przed szczegółową dyskusją dotyczącą testowania zdefiniujemy co będziemy testować. Testowanie Boundary-Scan jest nastawione głównie na zewnętrzne komponenty a nie wewnętrzne funkcje logiczne. Obligatoryjne rozkazy interfejsu zawierają funkcje testujące połączenia międzyukładowe. Inne opcjonalne cechy testowania Boundary-Scan, jeśli obecne, umożliwiają różne stopnie ingerencji we wnętrze logiki IC. Jako efekt uboczny otrzymujemy również informację o łańcuchu układów więc możemy je identyfikować.

Testowanie z użyciem Boundary-Scan jest dobrym rozwiązaniem dotyczącym testowania defektów lutowania układów tj. przerw i zwarc. Jeśli dany element będzie miał uszkodzone wyprowadzenie spowodowane przez ESD – również zostanie to wykryte. Jeśli element zostanie błędnie wlutowany lub obrócony to łańcuch BS będzie przerwany i można taką sytuację wykryć. Na rysunku 14.1 przedstawiono możliwe błędy lutowania.



Rys. 14.1. Widok elementu SMD przylutowanego do płytki drukowanej. Widoczne są defekty polegające na przerwie oraz zwarciu połączenia.

Boundary-Scan nie jest dobrą techniką diagnozy niektórych problemów. Na przykład, przebiecie ESD może spowodować uszkodzenie tranzystora leżącego głębiej w układzie scalonym, daleko od rejestru BS. Poza bardzo restrykcyjnym samotestem taka wada nie byłaby wykryta. Podobnie, jeśli układ ma defekt związany z parametrami np. czasem propagacji nie jest możliwe wykrycie tego przez Boundary-Scan. Jeśli połączenie lutowane jest słabej jakości (ale jest połączenie elektryczne) to oczywiście tego też Boundary-Scan nie będzie w stanie wykryć.

Głównymi zadaniami Boundary-Scan są więc testy: zwać i przerw montażowych, złych elementów, obróconych elementów, defektów stopni wyjściowych. Można więc przyjąć, że

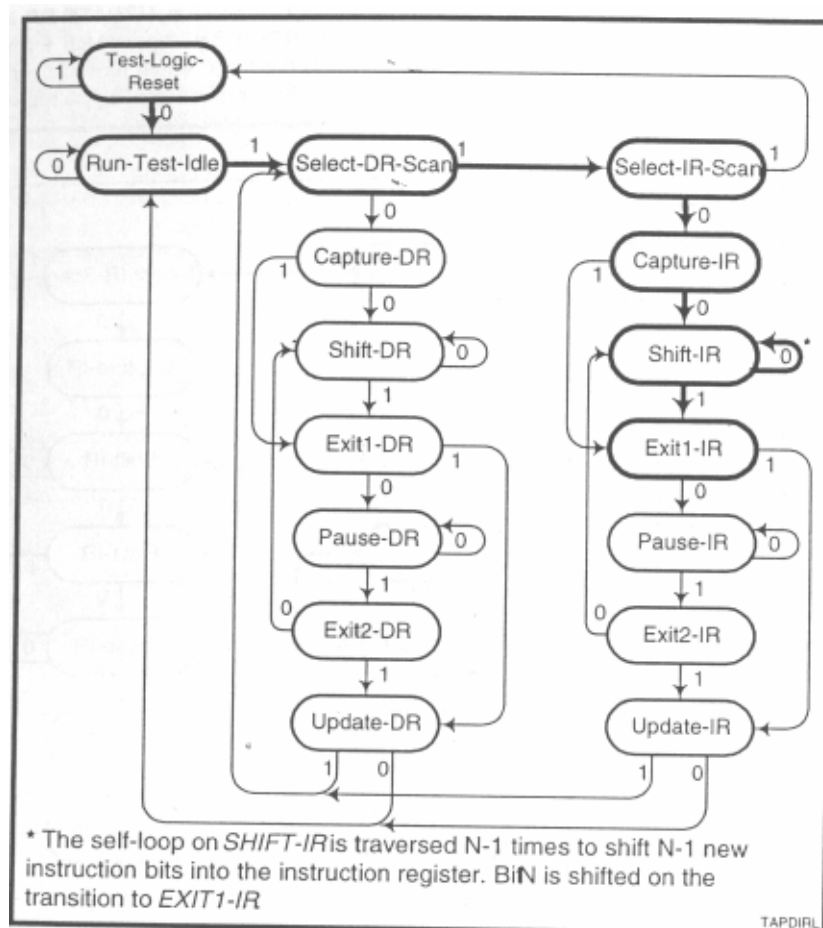
technika Boundary-Scan zakłada użycie idealnych elementów w nieidealnym środowisku montażowym w którym mogą powstać związane z tym defekty.

14.1. Podstawy testowania Boundary-Scan.

Omówienie testów wymaga wykonania szczegółowej analizy stanów kontrolera TAP oraz wykonania instrukcji BYPASS, PRELOAD, EXTEST, SAMPLE i innych opcjonalnych tj. RUNBIST, IDCODE, CLAMP, HIGHZ oraz INTEST.

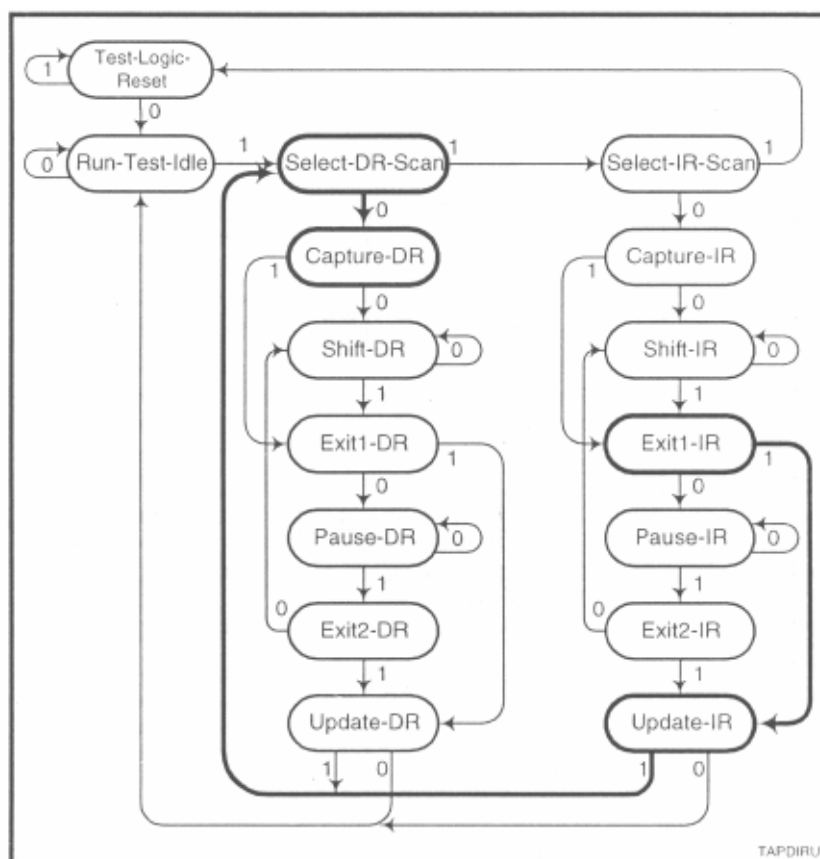
Najpierw zostanie omówione testowanie w odniesieniu do jednego układu, później dla wielu połączonych w łańcuch. W każdym z przypadków startujemy ze stanu TEST-LOGIC-RESET, ale nie możemy zakładać, że badany układ akurat w tym stanie się znajduje. W związku z tym należy uaktywnić wyprowadzenie TRST* lub na co najmniej 5 okresów TCK ustawić wyprowadzenie TMS w stan wysoki. To spowoduje przejście kontrolera TAP do stanu resetu niezależnie z którego stanu wystartował. Po resecie należy wprowadzić przewidzianą instrukcję do rejestru instrukcji. Na rysunku 14.2 pogrubiono przejścia stanów, które to zapewniają. Przejście do odpowiedniego stanu jest realizowane poprzez ustawienie odpowiedniej wartości sygnału TMS w trakcie narastającego zbocza zegara TMS. Po dojściu do stanu SHIFR-IR następuje aktywacja wyjścia TDO a instrukcja może zostać szeregowo wprowadzona przez wyprowadzenie TDI.

Maszyna stanów TAP przeszła przez stan CAPTURE-IR do stanu SHIFT-IR. Część szeregową rejestru instrukcji jest teraz włączona pomiędzy wyprowadzenia TDI oraz TDO. Zawiera ona wzór bitowy załadowany w stanie CAPTURE-IR. Dane te zostaną kolejno wyprowadzone z TDO w czasie wprowadzania nowej instrukcji do TDI. Dane wyjściowe mogą zostać przeanalizowane w celu sprawdzenia czy zawierają oczekiwaną wartość. W języku BSDL zawartość tego rejestru oznacza się poprzez atrybut INSTRUCTION_CAPTURE. Jeśli rejestr instrukcji jest o długości N bitów, pętla powrotu do stanu SHIFT-IR wykonywana jest N-1 razy. Za każdym razem należy do TDI przyłożyć kolejny bit instrukcji. N - ty bit jest przesuwany w czasie przejścia do stanu EXIT1-IR. Kolejność wprowadzania bitów jest od najmniej znaczącego.



Rys. 14.2. Wykres stanów kontrolera TAP z pogrubioną ścieżką przejścia realizującą wprowadzenie instrukcji do rejestru IR[7].

W tym momencie nowa instrukcja znajduje się w części szeregowej rejestru IR. Zostaje ona faktycznie aktywowana podczas stanu UPDATE-IR, wtedy zostaje wpisana do części równoległej rejestru IR. Należy zauważyć, że jeśli wprowadzona instrukcja zawiera rozkaz z dostępem do wyprowadzeń I/O wówczas w tym stanie nastąpi uaktywnienie dostępu do I/O.



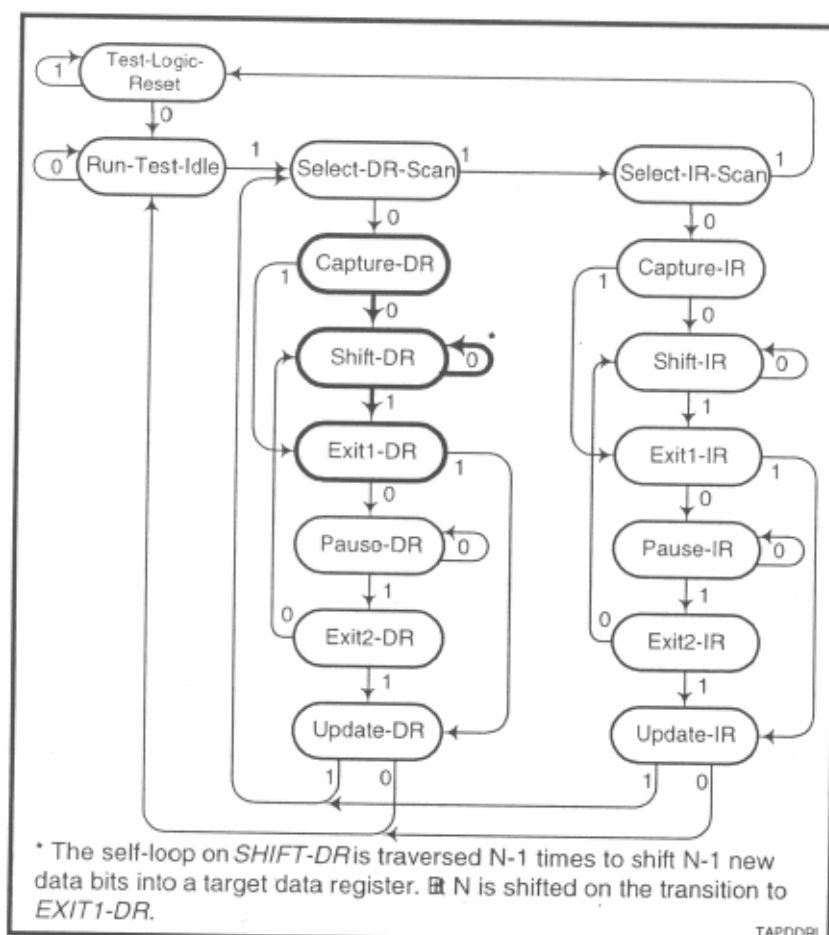
Rys. 14.3. Świeżo wprowadzona instrukcja jest aktywowana w stanie UPDATE-IR, wybrany jest nowy rejestr docelowy[7].

Na rysunku 14.3 przedstawiono kolejne typowe kroki. Przechodzimy przez stany UPDATE-IR oraz CAPTURE-DR. Część szeregową rejestru docelowego jest wpisywana w sposób równoległy w momencie przejścia ze stanu CAPTURE-DR do stanu SHIFT-DR. Dane wpisywane zależą oczywiście od typu rejestru, np. dla instrukcji BYPASS do rejestru obejścia wpisywane jest zero, dla instrukcji IDCODE do rejestru identyfikacji wpisany jest 32 bitowy DEVICE_ID.

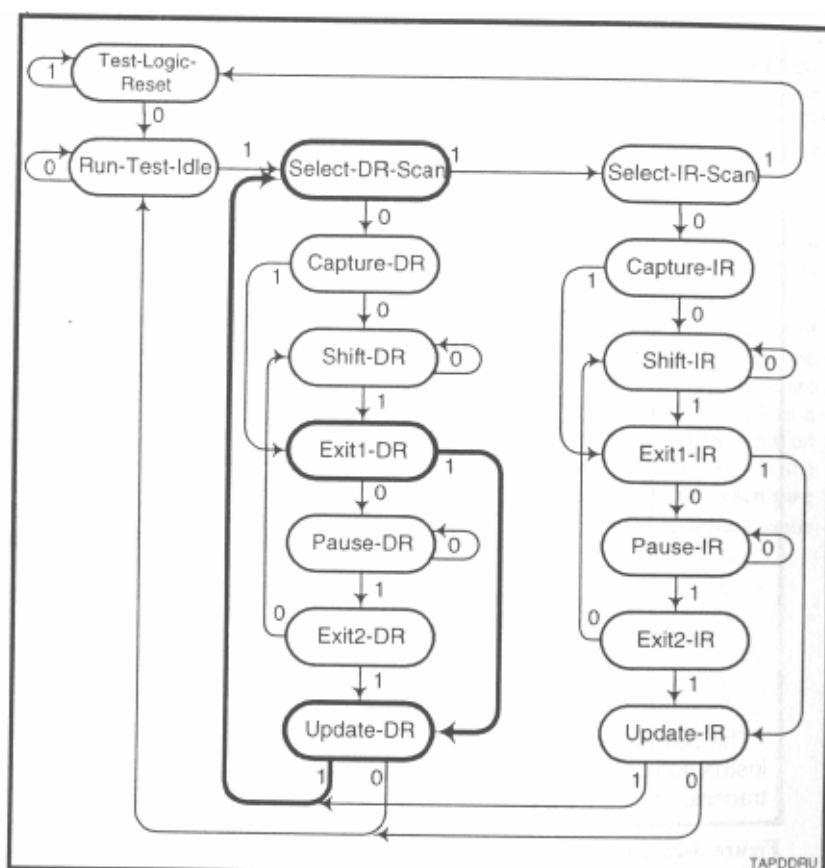
W tym momencie jesteśmy gotowi do przesuwania rejestru, jak to pokazano na rysunku 14.4. W celu wprowadzenia N bitowego słowa potrzebujemy powracać w pętli do stanu SHIFT-IR przez kolejnych N-1 razy. Ostatni, N - ty bit jest przesuwany podczas przejścia do stanu EXIT1-DR. Po zakończeniu przesuwania dane są przerzucane do części równoległej rejestru w czasie instrukcji UPDATE-DR (w czasie opadającego zbocza TCK).

W chwili obecnej przeszliśmy pełen cykl składający się z załadowania instrukcji, zapamiętania rejestru docelowego, szeregowego wyprowadzenia jego zawartości, szeregowego wprowadzenia nowej zawartości i zapamiętania jej w części równoległej rejestru docelowego. W tym momencie możemy przejść do stanu RUN-TEST/IDLE jeśli wykonujemy instrukcję używającą wiele cykli TCK, możemy załadować nową instrukcję lub możemy ponownie wykonać operacje na bieżącym rejestrze docelowym. Ta ostatnia opcja jest wykonywana najczęściej.

Za każdym razem kiedy przechodzimy przez kolumnę DATA (lewą kolumnę) kontrolera TAP a równocześnie aktywna jest instrukcja EXTEST, zapamiętujemy wartości sygnałów na wyprowadzeniach wejściowych układu (i dwukierunkowych) a następnie te dane możemy wyprowadzić szeregowo w celu przebadania. Dolna część grafu wykonuje operację zapisu nowych wartości na wyprowadzeniach wyjściowych układu scalonego (i dwukierunkowych). To daje nam główny mechanizm testujący: możemy używać wyjść układu scalonego w celu zapisu wartości logicznych w węzłach płytki drukowanej. Jeśli jedno wyjście układu zawierającego Boundary-Scan zostanie połączone z wejściem innego układu scalonego, można będzie przetestować ścieżkę połączeniową pod względem występowania ewentualnych zwarć lub przerw. Można to wykonać w sposób równoległy dla wszystkich węzłów jednocześnie. Jest to dość prosty proces.



Rys. 14.4. Ciąg stanów niezbędny do złapania danych i szeregowego ich wyprowadzenia równocześnie z szeregowym wprowadzaniem nowych danych[7].



Rys. 14.5. Zakończenie operacji przesuwania danych i zapamiętanie ich w części równoległej rejestru docelowego[7].

14.2. Prosty algorytm testowania.

Prosty algorytm testowania składa się z następujących kroków:

- Krok 1: Inicjalizacja TAP do stanu TEST-LOGIC-RESET.
- Krok 2: Załadowanie rejestru instrukcji kodem PRELOAD. To umieszcza rejestr BS pomiędzy TDI a TDO ale nie powoduje wejścia w tryb z dostępem do I/O.
- Krok 3: Wprowadzenie pierwszych pobudzeń do rejestru BS. Jest to faza wstępnego załadowania pobudzeń.
- Krok 4: Załadowanie rejestru instrukcji poleceniem EXTEST. To powoduje uaktywnienie rejestru BS i daje dostęp do wyprowadzeń I/O układu. Pierwsze pobudzenia PTV (ang. Parallel Test Vector) są wprowadzone na płytkę drukowaną.
- Krok 5: Odczyt odpowiedzi do części szeregowej rejestru BS.
- Krok 6: Wprowadzenie nowego PTV/wyprowadzenie odpowiedzi.
- Krok 7: Zapis nowych wektorów testowych do części równoległej BS.
- Krok 8: Czy został wpisany ostatni wektor testowy? Jeśli tak przejdź do kroku 9 jeśli nie wróć do kroku 5.
- Krok 9: Odczytaj ostatnią odpowiedź.
- Krok 10: Wprowadź „bezpieczny” wektor testowy podczas odczytu ostatniej odpowiedzi.
- Krok 11: Zapisz „bezpieczny” wektor.
- Krok 12: Przejdź do stanu TEST-LOGIC-RESET i zakończ testowanie.

Kroki 1-4 stanowią inicjalizację testów, kroki 5-8 dokonują testowania właściwego a w krokach 9-12 testowanie jest kończone. Dane otrzymane po procesie testowania powinny być przeanalizowane w całości w celu interpretacji wyników i przygotowania diagnozy, nie powinno się kończyć testów w momencie napotkania pierwszego błędu. Technika testowania typu „zatrzymanie na pierwszym defekcie” stosowana w testerach układowych jest nieodpowiednia dla testowania Boundary-Scan. System testujący powinien w całości wykonać gromadzenie odpowiedzi i na tej podstawie wykonać analizę uszkodzeń.

14.3. Testowanie logiki układu scalonego.

Rozkaz INTEST umożliwia testowanie logiki układu scalonego nawet jeśli jest on już zamontowany na płycie drukowanej. Gdy rejestr instrukcji TAP jest załadowany rozkazem INTEST i przechodzimy stan UPDATE-IR wyprowadzenia I/O układu scalonego są odłączane od wewnętrznych węzłów logiki. Wartości na wewnętrznych węzłach mogą być ustawiane/obserwowane poprzez rejestr BS. Aby wykonać testowanie układu scalonego należy zastosować algorytm taki jak w poprzednim rozdziale z następującymi modyfikacjami:

- Wektory testowe są wpisywane do wejść logiki układu a nie do wyjść I/O.
- Wyjścia I/O układu mogą pozostawać w stanie bezpiecznym lub być dezaktywowane poprzez dodawanie tych wartości do wektorów testowych.
- Odczyt wartości następuje na liniach wyjściowych logiki układu (i wyjściach enable).
- Stosowana jest instrukcja INTEST zamiast EXTEST.
- Wszystkie wektory testowe są przygotowane z uwzględnieniem funkcji układu a nie wyprowadzeń I/O.

W zasadzie, teoretycznie jest możliwe całkowite przebadanie układu scalonego nawet po jego przylutowaniu na płytce scalonej. Praktyka jest jednak nieco gorsza. Po pierwsze zazwyczaj testy logiki układowej są bardzo długie nawet w zwykłym testerze, ich wykonanie w formie szeregowej przez BS wydłuża taki test do wartości nieakceptowalnej. Po drugie testy mogą wymagać szybkich sygnałów testujących co w szeregowej wersji BS nie jest możliwe do osiągnięcia. I w końcu po trzecie może być problem z odpowiednim zegarowaniem sygnałów zegarowych logiki układu.

14.4. Samotest układu scalonego.

Podstawowy algorytm wykonania samotestu:

- Krok 1: Inicjalizacja TAP do stanu TEST-LOGIC-RESET.

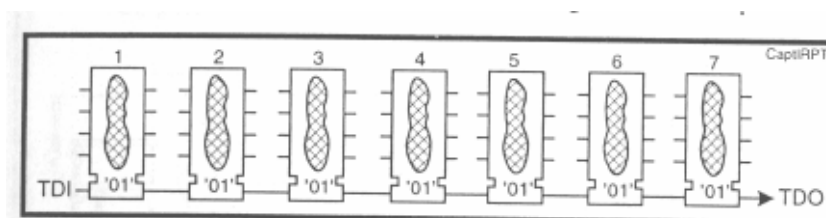
- Krok 2: Załadowanie rejestru instrukcji instrukcją PRELOAD. To umieszcza rejestr BS pomiędzy TDI a TDO ale nie powoduje wejścia w tryb z dostępem do I/O.
- Krok 3: Wprowadzenie bezpiecznego wektora do rejestru BS. Ma na celu zabezpieczenie sygnałów wyjściowych przed konfliktami z sygnałami z innych układów scalonych. Korki 2 i 3 mogą zostać pominięte jeśli w czasie wykonywania instrukcji RUNBIST wyjścia są automatycznie ustawiane w stan wysokiej impedancji.
- Krok 4: Załadowanie rejestru instrukcji poleceniem RUNBIST. To powoduje uaktywnienie rejestru wyniku.
- Krok 5: Przejście do stanu RUN-TEST/IDLE na czas niezbędny do wykonania testów.
- Krok 6: Przejście przez stan CAPTURE-DR i w ten sposób wpisanie wyniku do rejestru BIST.
- Krok 7: Odczytanie zawartości wyniku. Zapis nowych wektorów testowych do części równoległej BS.
- Krok 8: Przejście do stanu TEST-LOGIC-RESET.

14.5. Testowanie integralności łańcucha IEEE1149.1.

Zanim zaczniemy traktować wyniki testów jako ważne musimy sprawdzić poprawność funkcjonowania łańcucha Boundary-Scan. W łańcuchu tym mogą wystąpić następujące błędy:

- Element łańcucha może być uszkodzony, obrócony lub może go brakować.
- Element może mieć uszkodzone wyprowadzenia BS.
- Element łańcucha może mieć zwarcia lub przerwy montażowe.

Podstawową metodą badania spójności łańcucha jest fakt ładowania znanej wartości (przypisanej przez producenta IC + „01” do LSB) do szeregowej części rejestru instrukcji w czasie stanu CAPTURE-IR. Przez analizę zawartości słowa wyjściowego można stwierdzić spójność łańcucha. W najprostszym przypadku, jak na rysunku 14.6, gdy wszystkie elementy łańcucha mają dwubitowy rejestr instrukcji, wówczas przykłady zestawów danych odczytanych ze sprawnego i uszkodzonego łańcucha przedstawia tabela 14.1.



Rys. 14.6 Łańcuch elementów Boundary-Scan. Wszystkie układy przeszły stan CAPTURE-IR i w każdym rejestrze instrukcji znajduje się ciąg „01” [7].

W złym ciągu danych, układ IC7 najpierw trzykrotnie wypuszcza „01” a następnie czterokrotnie „11”. Można wnioskować więc, że układy IC5-7 są sprawne natomiast układy IC1-4 nie. Ten wniosek może nie być do końca prawdziwy bo może także nastąpić uszkodzenie wejścia TDI układu IC5. Zazwyczaj jednak problemy z lutowaniem są częstsze więc na początek należy sprawdzić płytkę drukowaną i poprawić ewentualne błędy lutownicze.

	Integrated Circuit						
	1 (TDI)	2	3	4	5	6	7 (TDO)
Good Data Stream	01	01	01	01	01	01	01
Bad Data Stream	11	11	11	11	01	01	01

Tabela 14.1 Przykłady danych pojawiających się na wyjściu TDO układu IC7 [7].

Ze względu na to, że przedstawione wcześniej testy rozpoczynały się od załadowania instrukcji – można ten fakt wykorzystać i równocześnie wykonać testowanie integralności łańcucha.

Niestety powyższy test integralności nie sprawdza działania wejścia TDI pierwszego układu scalonego. Można to wykonać poprzez podanie na wejście dwóch dodatkowych bitów których wzór

będzie się różnił od typowych spotykanych jako instrukcje. Przykładem może być ciąg „10”. Te dwa dodatkowe bity nazywane są bitami ochronnymi.

Testowanie integralności łańcucha można wykonać także w sposób nieco bardziej złożony. Najpierw wprowadzamy instrukcję IDCODE dla komponentów które ją mają oraz instrukcję BYPASS dla pozostałych². Następnie przechodzimy do szeregowego odczytu. To da nam dodatkową informację czy któryś z elementów nie został przylutowany w złym miejscu.

Jeden z możliwych, podstępnych błędów łańcucha może doprowadzić do błędnej oceny miejsca jego uszkodzenia. Załóżmy, że w łańcuchu jest wiele identycznych elementów co powoduje że wartości ładowane do rejestru instrukcji są identyczne. Dodatkowo załóżmy, że wyprowadzenia TDI oraz TDO układu IC4 są zwarte i występuje funkcja AND na drucie. To powoduje AND na sygnałach TDO układu IC3 i IC4. Wykonując test integralności z bitami ochronnymi „10” na wyjściu TDO zobaczymy dane jak w tabeli 14.2.

	Sentinel Bits	Integrated Circuit						
		1 (TDI)	2	3	4	5	6	7 (TDO)
Good Data Stream	10	01	01	01	01	01	01	01
Bad Data Stream	00	00	01	01	01	01	01	01

Tabela 14.2. Dane otrzymane z wyjścia TDO układu IC7 dla przypadku zwarcia TDI z TDO układu IC4 i uformowaniu AND na drucie.[7].

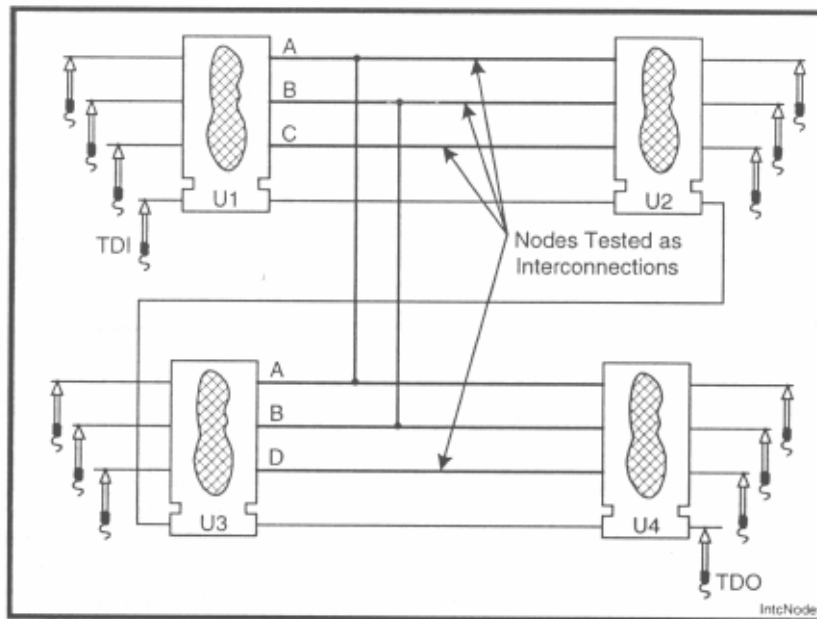
Z tabeli 14.2 wynika, że mamy problem w sąsiedztwie układu IC1 a w rzeczywistości występuje on w układzie IC4. Problem ten jest szerzej dyskutowany w literaturze, gdzie również przedstawione są metody jego unikania. Jednakże w celu jego zmniejszenia możemy stosować dwa prostsze rozwiązania: wstawienie punktów pomiarowych w układzie i w ten sposób przetestowanie klasyczną metodą bez zasilania ewentualnych zwarć oraz zmniejszenie szansy, że TDI będzie zwarte z TDO poprzez odległe usytuowanie tych wyprowadzeń układu scalonego.

14.5. Testowanie połączeń wewnętrznych.

Testowanie połączeń wewnętrznych odnosi się do testowania zwarć i przerw w ramach połączeń pomiędzy układami zawierającymi interfejs Boundary-Scan.³ Na rysunku 14.7 przedstawiono prosty przykład. Wszystkie węzły pomiędzy układami Boundary-Scan nazywane są wewnętrznymi. Węzły połączone z sondami pomiarowymi są węzłami zewnętrznymi, opis ich testowania będzie przedstawiony w dalszej części materiału. Na tym i kolejnych rysunkach założono, że na prawych krawędziach układów wszystkie wyprowadzenia są wyjściami lub dwukierunkowe a na lewych wejściami. Problem testowania połączeń wewnętrznych został przebadany długo przed wprowadzeniem układów Boundary-Scan. Testowanie połączeń wewnętrznych jest zazwyczaj wykonywane w jednym kroku jako testowanie zwarć i przerw. Tutaj oba rodzaje testów są celowo rozdzielone. Ze względu na ewentualną możliwość uszkodzeń w przypadku wystąpienia zwarć, testowanie tego typu uszkodzeń należy wykonać najpierw a dopiero po ich usunięciu powinniśmy zajmować się pozostałymi defektami.

² Uważny czytelnik może się zdziwić po co wprowadzać te instrukcje skoro w stanie TEST-LOGIC-RESET są one automatycznie wybierane. Powodem jest to, że wiele układów nie posiada rejestru IDCODE i umieszcza pojedynczy bit o wartości „0” pomiędzy TDI a TDO. W takim przypadku nie ma zmian sygnału na TDO a przez to trudno jest ocenić w którym miejscu łańcucha wystąpiła ewentualna usterka.

³ Węzły Boundary-Scan są tu zdefiniowane jako węzły zawierające co najmniej jeden punkt sterujący (wyjście) i co najmniej jeden odbiornik (wejście) które są sterowane poprzez BS. Warunki takie może spełnić przyłączenie węzła do pojedynczego wyprowadzenia dwukierunkowego. Inne węzły to: węzły analogowe, typowe węzły cyfrowe, napięcia zasilające i odniesienia oraz inne węzły które nie mogą być scharakteryzowane stanem logicznym „0” lub „1”.



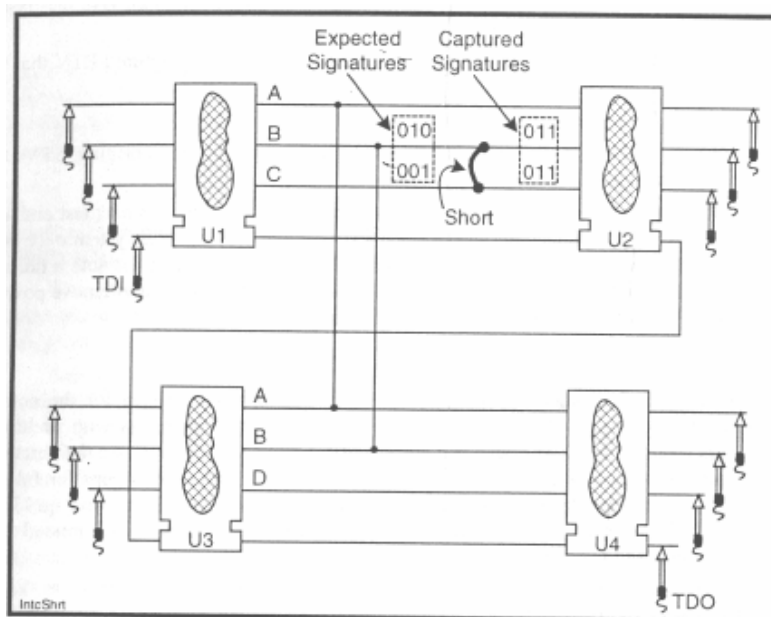
Rys. 14.6. Łańcuch układów z 4 wewnętrznymi węzłami połączeniowymi[7].

W czasie testowania zwarć nie możemy zakładać braku istnienia przerw, jednak po zakończeniu badania zwarć możemy założyć testowanie przerw przy braku zwarć.

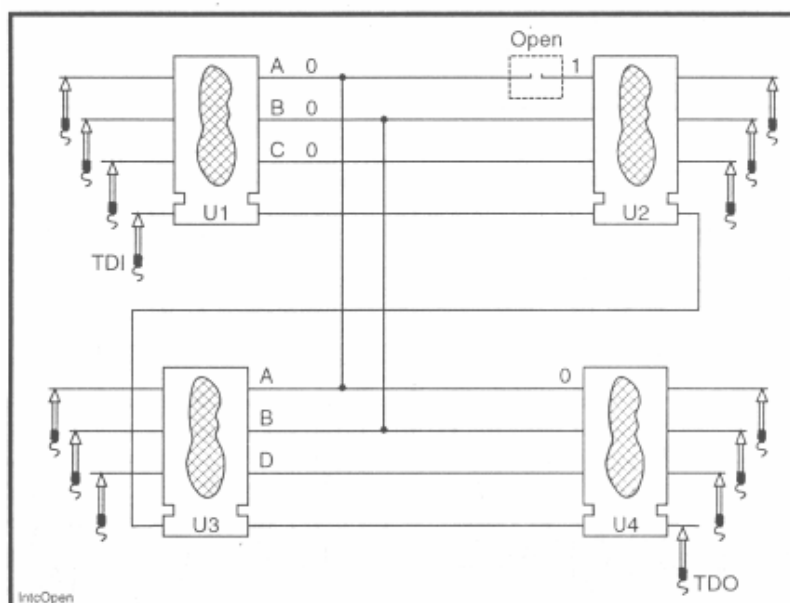
Testowanie zwarć.

Testowanie zwarć jest wykonywane z użyciem podstawowego algorytmu testu opisanego w rozdziale 14.2. Jest on używany w celu przesłania szeregowych wektorów testowych STV (ang. Sequential Test Vector) do węzłów rejestru BS. Dane odczytane w węzłach wejściowych rejestru BS nazywane są szeregowym wektorem odpowiedzi SRV (ang. Sequential Response Vector). Jeśli dany węzeł nie ma zwarć wówczas wektor STV wystawiony przez wyprowadzenie sterujące będzie taki sam jak wektor SRV odczytany przez wejście rejestru BS. W związku z powyższym należy postępować następująco:

- Krok 1: Należy przebadac listę połączeń układów oraz opis BSDL układów scalonych. Należy ponumerować połączenia Boundary-Scan oraz wszystkie dołączone wyprowadzenia układów scalonych.
- Krok 2: Dla każdego węzła należy zidentyfikować wszystkie dołączone do niego wyprowadzenia sterujące (wyjścia i dwukierunkowe wyprowadzenia IC). Należy wybrać jedno z nich i oznaczyć jako węzeł desygnowany węzła.
- Krok 3: Przypisz unikalny wektor STV do każdego z węzłów wyjściowych.
- Krok 4: Przetrasponuj STV na wektory równoległe PTV.
- Krok 5: Wykonaj test na płytce. Zapisz każdy PRV odczytany z płytki.
- Krok 6: Przetrasponuj odczytane PRV na SRV.
- Krok 7: Przeanalizuj wektory SRV i STV. Znajdź różnice pomiędzy wektorami wpisanymi i odczytanymi.

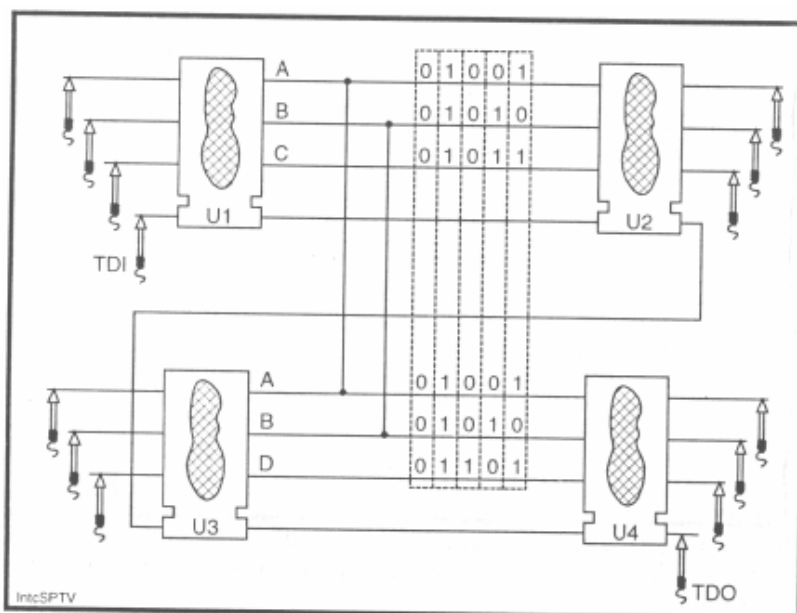


Rys. 14.7. Test połączeń wewnętrznych polegający na podaniu unikalnych wektorów testowych do każdego z węzłów. Zwarcie na rysunku łącznie z funkcją OR na drucie powoduje generację odpowiednich sygnałów zapamiętanych w rejestrze BS[7].



Rys. 14.8. Przerwa połączenia wewnętrznego powodująca brak dotarcia sygnału do jednego z wejść. Ten fakt może ułatwić diagnostykę położenia przerwy [7].

Kroki 1-4 mogą zostać wykonane jednorazowo przed testowaniem większej liczby płytek. Mogą one również zostać wykonane automatycznie o ile dysponujemy odpowiednim oprogramowaniem jak również odpowiednim opisem układów (BSDL) i płytki drukowanej. Krok 5 jest wykonywany jednokrotnie na danej płytce przez system testowy ATE (ang. Automatic Test Equipment). Kroki 6 i 7 mogą być pominięte jeśli system ATE nie wykryje błędów. W przeciwnym przypadku należy natychmiast odłączyć zasilanie testowanych układów i przeprowadzić kroki 6 i 6.



Rys. 14.9. Prosty test połączeń wewnętrznych. Kolumny stanowią testowe wektory równoległe PTV a wiersze są testowymi wektorami szeregowymi STV[7].

Dyskusja dotycząca testowania zwarć.

W kroku 2 wybierane jest jedno z wyjść sterujących na czas całego testowania zwarć. To oznacza że dla węzłów z wieloma sterownikami mogą wystąpić przerwy których nie da się wykryć. Ten rodzaj uszkodzeń będzie analizowany w dalszej części opracowania. W tej dyskusji przebadamy kilka miejsc gdzie przerwy są nierozpoznane w czasie testu zwarć.

Głównym powodem dla wybrania jednego i ustalonego sterownika jest uproszczenie i skrócenie czasu testu zwarć. Testowanie przerw na szynach danych może być relatywnie drogie licząc to w ilości użytych PTV. Przerwy są łagodniejsze dla układu i nie powodują uszkodzeń dlatego przesuwamy ich testowanie na czas gdy już wcześniej zostało wykluczone występowanie zwarć.

W kroku 3 przypisywane są unikalne wektory STV do każdego z węzłów. W literaturze można spotkać szczegółowe opracowania stosowanych rodzajów wektorów. Najprostszy i najkrótszy test używa binarnych cyfr przypisywanych sekwencyjnie jako wektory STV. Po przetransponowaniu takiego wektora na PTV daje to liczbę wektorów PTV równą $\log_2 N$, gdzie N jest liczbą węzłów poddawanych testowaniu. Ta logarytmiczna kompresja jest bardzo istotna, np. dla 1000 węzłów potrzebujemy tylko 10PTV a dla 4000 węzłów 12PTV. Długość wektora PTV jest związana z N . Np. jeśli węzeł wewnętrzny zawiera średnio 4 wyprowadzenia układów BS wówczas w łańcuchu Boundary-Scan mamy $4 \cdot N$ pinów. Oznaczmy jako A średnią liczbę wyprowadzeń przypadającą na jeden węzeł. Wówczas liczba bitów wprowadzanych szeregowo w czasie testowania wyniesie około:

$$AN \log_2 N \quad (28.1)$$

cykli. Dla 4000 węzłów daje to w przybliżeniu 192.000 cykli. Porównawczo, można przedstawić sekwencję w postaci wędrującej jedynki (lub zera, sekwencja ta eliminuje problem nakładania). W sekwencji tej przyłożona jest pojedyncza jedynka równocześnie z samymi zerami na pozostałych pozycjach. Aby przetestować każdy z węzłów trzeba więc przyłożyć N razy taki ciąg za każdym razem zmieniając położenie jedynki. To powoduje, że liczba wpisywanych wartości jest proporcjonalna do

$$AN^2 \quad (28.2)$$

Dla tej samej liczby 4000 węzłów otrzymujemy 64.000.000 cykli zegara. Sekwencja zliczająca oraz sekwencja wędrującej jedynki są dwoma ekstremalnymi przypadkami.

W kroku 4 następuje zamiana wektorów szeregowych na równoległe poprzez wybranie jako pierwszego wektora równoległego ciągu pierwszych wartości wszystkich wektorów szeregowych, jako drugiego wektora drugich wartości wszystkich wektorów szeregowych itd.

Nakładanie i zaskakiwanie.

Na podstawie badania wektorów SRV następuje diagnoza uszkodzeń. W wielu rodzinach układów logicznych przerwa jest interpretowana jako stałe odbieranie stanu logicznego „0” lub „1”. Jeśli wektor SRV zawiera te same wartości możemy podejrzewać przerwę lub zwarcie do masy lub zasilania. W niektórych rodzinach logicznych zwarte wyjścia realizują AND na drucie. Inne rodziny logiczne realizują OR na drucie. W rodzinach logicznych gdzie wyjścia mają różne siły stopni wyjściowych (np. w CMOS) nie jest możliwe ogólne przewidzenie funkcji realizowanej przez zwarcie, taki przypadek nazywamy funkcją X na drucie. Jeśli teraz dwa węzły typu AND na drucie są zwarte wówczas odebrane wektory SRV związanych z nimi komórek rejestru BS będą identyczne i będą zawierały AND bitowe wektorów testowych STV związanych z nimi wyjść rejestru BS. Wydaje się to prostą metodą tworzenia raportów uszkodzeń ale okazuje się jednak, że w zależności od doboru wektorów STV może wystąpić niesprzyjające zjawisko ***nakładania***.

Nakładanie występuje kiedy połączone błędy dwóch lub większej ilości węzłów dają taki sam wynik SRV jak dla węzła który nie ma uszkodzeń. Na przykład, jeśli węzły B i C przedstawione w tabeli 14.3 są zwarte (z funkcją AND na drucie) wówczas wynikowy SRV (0001) tych węzłów będzie identyczny jak STV (0001) węzła A. Czy to oznacza także zwarcie do węzła A? Nakładanie daje następującą właściwość: wiemy iż dwa lub więcej węzłów jest zwartych ze sobą ale prawidłowo połączony węzeł jest także podejrzany.

A	0	0	0	1
B	0	0	1	1
C	0	1	0	1
D	1	0	1	1
E	0	1	0	1

Tabela 14.3. Wektory testowe. Wiersze stanowią STV a kolumny PTV [7].

Podobnie jak poprzednio, jeśli węzły D oraz E są ze sobą zwarte wynikowy SRV (0001) będzie wskazywał na zwarcie z węzłem A. To zjawisko nazywane jest ***zaskakiwaniem*** - nie możemy określić czy w układzie występuje jedno czy dwa zwarcia oraz czy węzeł A jest zwarty w każdym z przypadków.

Nakładanie i zaskakiwanie nie blokuje możliwości wykrycia zwarcia a jedynie ogranicza możliwości diagnostyczne. Taka komplikacja może powodować utrudnienia w czasie napraw płytek.

Jeśli naszym celem jest zmniejszenie prawdopodobieństwa wystąpienia nakładania i zaskakiwania musimy zastosować dłuższy test. Poprzednio przedstawione zostały rozważania dotyczące dwóch rodzajów testów tj. binarnie zliczającego oraz wędrującej jedynki. W literaturze można znaleźć rozważania dotyczące rozwiązań pośrednich to znaczy takich które mają większą odporność na nakładanie i zaskakiwanie niż test binarny ale są krótsze (licząc w ilość wprowadzanych bitów) niż test wędrującej jedynki.

Inne problemy związane z testami zwarć:

- zabezpieczenie przed uszkodzeniami zwartych wyjść (krótki test),
- brak w rzeczywistym układzie występowania AND na drucie lub OR na drucie,
- zwarcie więcej niż dwóch wyjść daje różne kombinacje wartości sygnału w zależności od sił wyjść i aktualnego ich stanu,
- możliwość różnej interpretacji stanu logicznego sygnału o wartości pośredniej przez różne wejścia (z bramką Shmidta lub zwykłe, różne wykonania układu itp.),
- możliwość wystąpienia różnych odpowiedzi testowanego układu na dwa identycznie wykonywane testy.

Z tego względu w praktyce często stosuje się powiększone, mieszane testy jak w tabeli 14.4. Kolumny stanowią wektory pobudzeń PTV. Wektory NOTE1 oraz NOTE2 testują zwarcie do masy i zasilania przy braku konfliktów wyjść nawet jeśli występują zwarcia międzywęzłowe. Wektory NOTE4 są typem binarne zliczanie. Wektory NOTE3 są komplementarne w stosunku do prawych kolumn NOTE4. Stanowią one wektory zapewniające zmniejszenie zjawiska nakładania się.

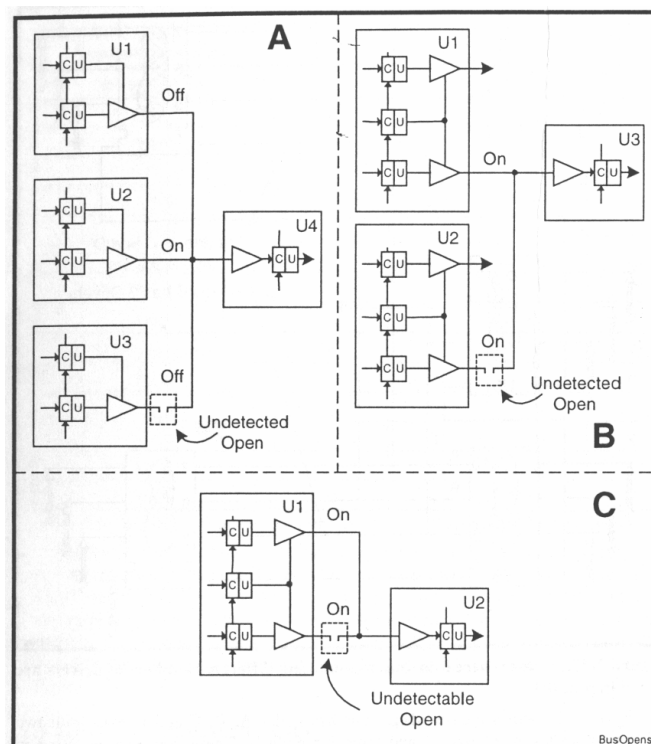
Node	Note1	Note2	Note 3		Note 4		
A	0	1	1	1	0	0	0
B	0	1	1	0	0	0	1
C	0	1	0	1	0	1	0
D	0	1	0	0	0	1	1
.	.	.	.	.	.	.	.
N	0	1	1	1	1	0	1

Tabela 14.4. Zestaw wektorów testowych testu poszerzonego, opis w tekście powyżej [7].

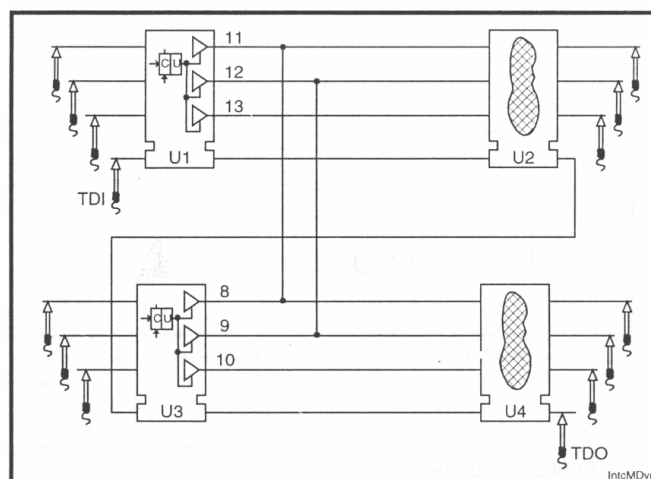
Doświadczenia praktyczne wskazują, że powyższe rozwiązanie daje dobrą rozdzielczość testowania zwarć sygnałów między wyjściami oraz do masy i zasilania, rzadko daje efekt nakładania oraz jest krótkie.

Testowanie przerw.

W ramach wcześniej wykonywanego testu zwarć zostają również wykryte niektóre przerwy. Nie wykryte przerwy obejmują przypadki węzłów z wieloma wyjściami dla których dane wyjście nie zostało uaktywnione. Przypadki takich przerw zostały przedstawione na rysunku 14.11 (A-sytuacja typowa, B - oba wyjścia są aktywowane do tych samych stanów w czasie testowania aby nie powodować konfliktów, C - dwie bramki połączone równoległe w celu zwiększenia siły sterowania).

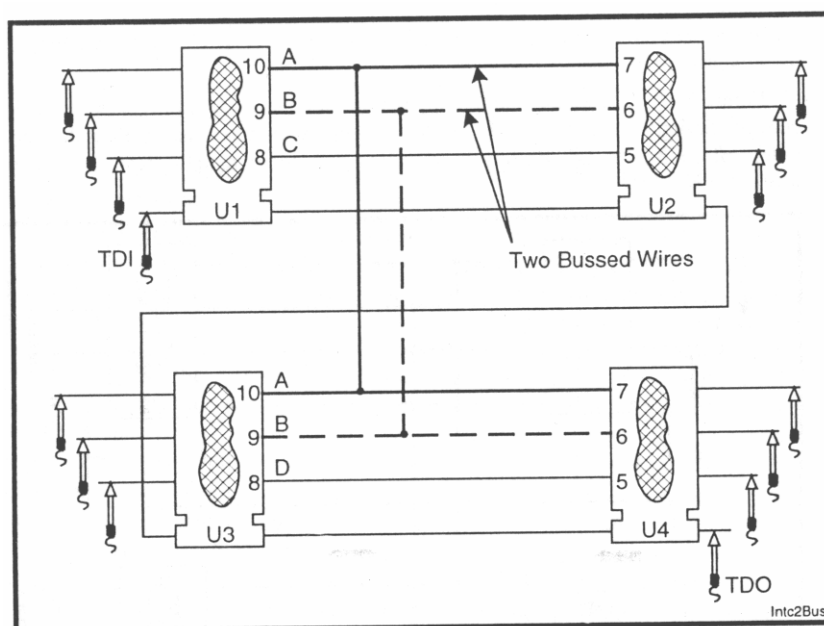


Rys. 14.10. Trzy przypadki nie wykrycia przerwy w czasie testu zwarć [7].



Rys. 14.10.1. Szczegóły przypadku z rys.14.11b, pojedyncza komórka BS steruje równocześnie kilkoma wyjściami układu scalonego co powoduje, że w celu uniknięcia konfliktów wszystkie sygnały muszą mieć tą samą wartość a więc testowanie ew. przerw jest utrudnione [7].

Jeśli mamy szynę danych z liczbą N wyjść sterujących wówczas można ją łatwo przetestować używając $2 \cdot N$ wektorów PTV (N par sygnałów „01” które włączają tylko jedno z wyjść szyny). Na rysunku 14.13 przedstawiono przykład z dwoma węzłami (nazwanymi A oraz B), każdy z nich ma dwa wyjścia sterujące. Można testować oba węzły równocześnie, przykłady pobudzeń podane są w tabeli 14.5. Na wyjściu odbiornik powinien otrzymać ciąg „0101”. Węzły C i D nie są testowane bo zostały wcześniej przetestowane podczas sprawdzania zwarc.

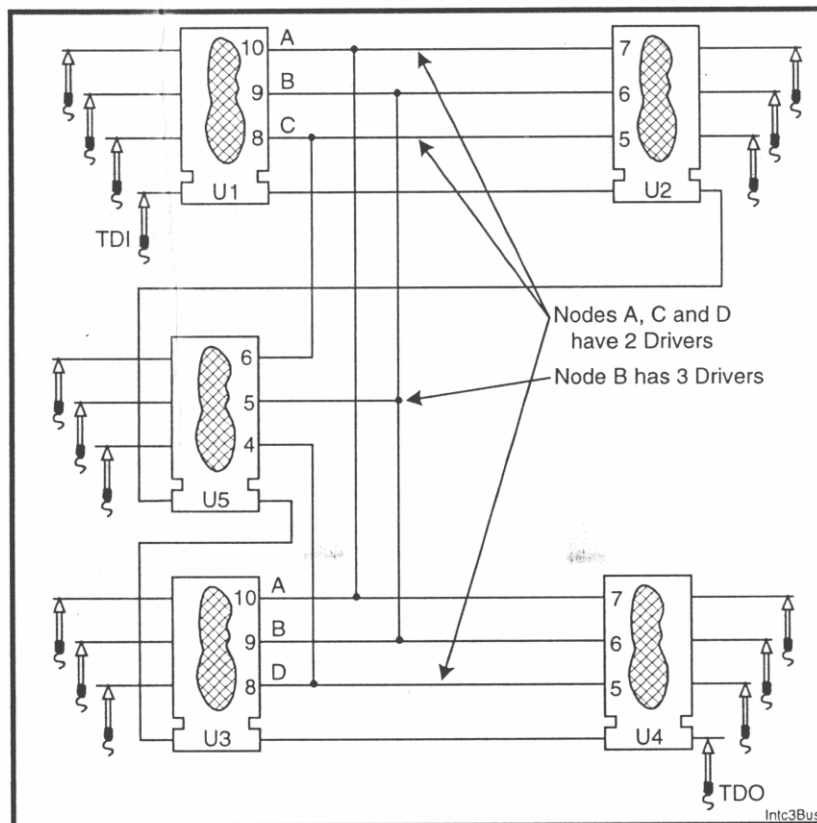


Rys. 14.13. Równoczesne testowanie dwóch węzłów szyny danych [7].

Component and Pin	Node	Bit Pattern	
U1.10 (driver)	A	01	ZZ
U3.10	A	ZZ	01
U1.9	B	01	ZZ
U3.9	B	ZZ	01
U2.6 (receiver)	B	01	01
U2.7	A	01	01
U4.6	B	01	01
U4.7	A	01	01

Tabela 14.5. Dane testujące do przypadku z rys. 14.13[7].

Na rysunku 14.14 przedstawiono przypadek wystąpienia 4 szyn sygnałowych (węzłów). Węzły A, C oraz D mają po 2 wyjścia sterujące, węzeł B ma 3 takie wyjścia. Wszystkie węzły mogą być testowane równocześnie ale jako liczbę N musimy przyjąć największą liczbę bramek wyjściowych występujących w jednym węźle czyli w naszym przypadku liczbę 3. Kiedy węzeł nie jest sterowany w danym momencie (co ma miejsce dla węzłów A, C i D) wówczas odpowiadający SRV oznaczony jest symbolem „xx” co oznacza wartość bez znaczenia.



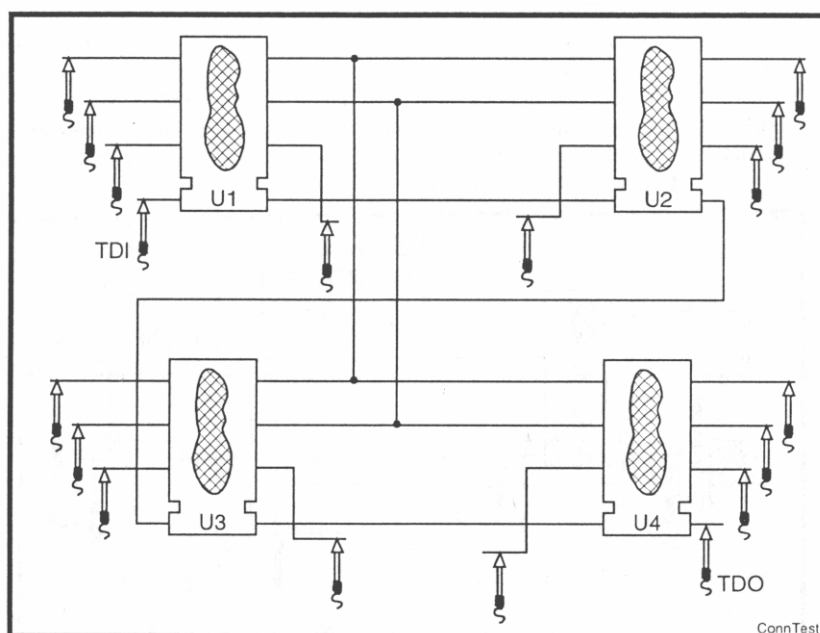
Rys. 14.14. Przypadek testowania czterech węzłów o różnych liczbach wyjść [7].

Component and Pin	Node	Bit Pattern		
U1.10 (driver)	A	01	ZZ	ZZ
U3.10	A	ZZ	01	ZZ
U1.9	B	01	ZZ	ZZ
U5.5	B	ZZ	01	ZZ
U3.9	B	ZZ	ZZ	01
U1.8	C	01	ZZ	ZZ
U5.6	C	ZZ	01	ZZ
U5.4	D	01	ZZ	ZZ
U3.8	D	ZZ	01	ZZ
U2.7 (receiver)	A	01	01	XX
U2.6	B	01	01	01
U2.5	C	01	01	XX
U4.7	A	01	01	XX
U4.6	B	01	01	01
U4.5	D	01	01	XX

Tabela 14.5. Wektory testowe wymagane podczas testowania przerw dla węzłów o różnej liczbie wyjść [7].

14.6. Testowanie połączeń zewnętrznych.

Testowanie połączeń zewnętrznych jest podobne do testowania z użyciem techniki Boundary-Scan. Testowane jest połączenie węzłów do których przypięte są sondy pomiarowe. Na rysunku 14.15 przedstawiony jest układ z wieloma punktami połączonymi z sondami, które mają zostać przetestowane. Punkty do których przyłączane są sondy mogą stanowić złącza krawędziowe lub punkty bez dojścia poprzez Boundary-Scan (np. elementów bez BS). Test połączeń zewnętrznych jest relatywnie prosty do wykonania. Jest wykonywane wyłącznie testowanie przerw gdyż zawsze przed testowaniem z włączonym zasilaniem powinno być wykonane testowanie bez zasilania polegające na badaniu ewentualnych zwarcć. Testowanie połączeń jest wykonywane identycznie jak przy użyciu techniki Boundary-Scan z tym wyjątkiem, że niektóre z sygnałów pochodzą z sond pomiarowych a niektóre z rejestrów BS. Częstość przypadkiem jest łączenie niektórych węzłów z masą lub zasilaniem. To także można sprawdzić poprzez sondy pomiarowe.

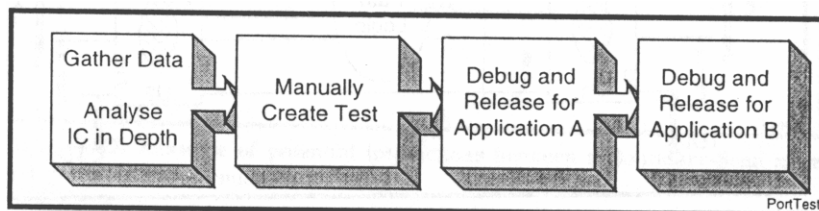


Rys. 14.15. Układ z testowaniem węzłów bez dostępu do Boundary-Scan przy użyciu sond.[7].

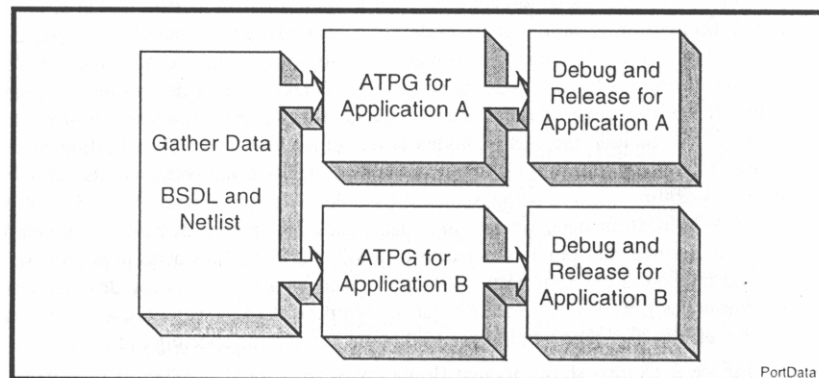
14.7. Przenoszenie testów.

Przenoszenie testów tworzonych ręcznie jest długie i bardzo wyczerpujące. Taki styl testowania stosowany był dotychczas dla układów i systemów o średniej i małej złożoności, i jest przedstawiony na rysunku 14.15. Obecnie, stosuje się programy potrafiące w sposób automatyczny wygenerować wektory testowe oraz wykonać analizę otrzymanych wektorów z testowanego systemu. Taki sposób podejścia do procesu testowania przedstawiony jest na rysunku 14.16. Program ATPG (ang. Automatic Test Program Generation) potrzebuje jako informacje wejściowe: listę połączeń płytki drukowanej oraz opisy implementacji Boundary-Scan w poszczególnych układach (zazwyczaj w języku BSLD). Zaletami stosowania programów ATPG w stosunku do ręcznego tworzenia testów jest:

- przenoszenie testów w postaci informacji o systemie zamiast informacji o testach,
- szybkie generowanie testów (sekundy zamiast miesięcy),
- równoczesna głęboka analiza ewentualnych uszkodzeń zamiast testów typu sprawne/niesprawne.



Rys. 14.15. Ręczne tworzenie testów dla dwóch podobnych aplikacji [7].



Rys. 14.15. Tworzenie testów Boundary-Scan dla podobnych aplikacji [7].

14.8. BSDL.

BSDL (ang. Boundary-Scan Description Language) jest językiem opisującym cechy zaimplementowanego w układzie interfejsu Boundary-Scan. Podstawową właściwością języka jest całkowita zgodność oraz scalenie ze standardem IEEE1149.1. W jej wyniku elementy które są w standardzie obowiązkowe nie są zawarte w języku a ich istnienie jest domyślnie przyjęte. Przykładami takich elementów mogą być: rejestr obejścia, maszyna stanów TAP. Język BSDL można zastosować w celu:

- testowania układu,
- sprawdzenia zgodności z IEEE1149.1,
- syntezy Boundary-Scan.

Składnia języka BSDL oparta jest na składni języka VHDL. Głównymi elementami opisu BSDL są:

- deklaracja ENTITY,
- parametr generic – zazwyczaj wykorzystywany jako wybór obudowy,
- opis wyprowadzeń – port (jak w VHDL),
- wyrażenie use – aktywowanie pakietów bibliotek BSDL,
- wyrażenie zgodności układu ze standardem 1990 lub 1994,
- przypisanie nr wyprowadzeń do nazw logicznych,
- grupowane wyprowadzenia (np. różnicowe),
- identyfikacja wyprowadzeń TAP,
- opis rejestru instrukcji: kody instrukcji, instrukcje nieobowiązkowe,
- opis zawartości rejestrów IDCODE oraz USERCODE,
- opis dostępnych rejestrów,
- opis komórek rejestru BS,
- opis wykonania instrukcji RUNBIST,
- opis wykonania instrukcji INTEST,
- rozszerzenia użytkownika,
- ostrzeżenia.

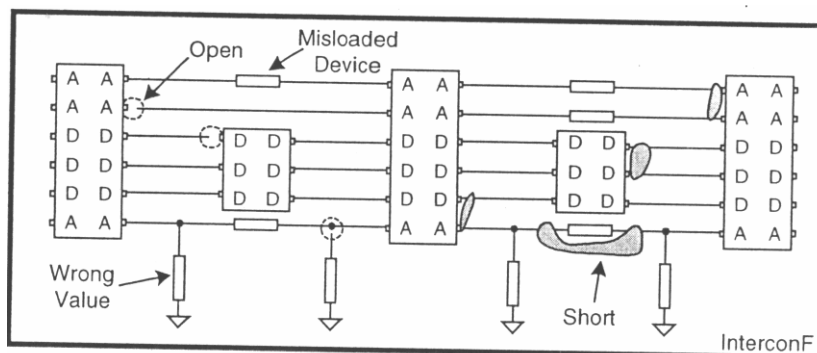
Przykład opisu BSDL układu scalonego Xilinx XC4010XL zamieszczony jest w dodatku A na końcu niniejszego opracowania.

14.9. Standard analogowy IEEE1149.4.

Standard 1148.4 zatytułowany „Mixed Signal Test Bus” jest popularnie znany jako analogowe Boundary-Scan. W rzeczywistości jednak w standardzie nie występuje zapamiętywanie i przesuwanie sygnałów analogowych w rejestrach lecz zamiast tego wprowadzone są klucze przełączające sygnały w celu łatwiejszego dostępu do pewnych punktów pomiarowych. Dostęp do tych punktów jest wykonywany poprzez elementy umieszczone w układzie scalonym zamiast poprzez sondy pomiarowe, więc postęp technologii wytwarzania układów scalonych dotyczy również implementacji elementów testowych. W standardzie 1148.1 całkowicie pominięto problem testowania wyprowadzeń analogowych, jednakże od początku lat 90-tych obserwuje się silną potrzebę integracji tego typu elementów testowych w rdzeniach układów scalonych. Po wielu burzliwych dyskusjach w międzynarodowej społeczności projektantów i producentów elementów i sprzętu elektronicznego, w 1993r przyjęto wstępne założenia standardu IEEE1149.4. Zrezygnowano jednak z wykonywania funkcji zapamiętywania i przesuwania sygnałów analogowych (tak jak to jest wykonywane dla std. cyfrowego). Przyjęto, że przede wszystkim należy udostępnić możliwość testowania zwarć, przerw, błędnych lub złych elementów w sieciach przetwarzających sygnały mieszane. Zrezygnowano również z testowania części analogowych sygnałami o bardzo dużych częstotliwościach, ograniczono pasmo pomiarowe do zakresu od napięć i prądów stałych do około 1MHz.

14.9.1. Zakres wykrywanych uszkodzeń.

Standard IEEE1149.4 został tak zaprojektowany aby był zgodny z wcześniejszym cyfrowym IEEE1149.1. Można go traktować jako pewne rozszerzenie. Kontroler TAP jest identyczny w obu rozwiązaniach. Rozszerzenie standardu umożliwia testowanie wyprowadzeń analogowych. Na rysunku 14.17 przedstawiono system zawierający kilka analogowo-cyfrowych układów scalonych łącznie z elementami pasywnymi. Przykłady możliwych uszkodzeń obejmują: przerwy, zwarcia, braki elementów, złe wartości elementów lub błędnie wstawione elementy.



Rys. 14.16. Układ analogowo-cyfrowy z pewnymi uszkodzeniami [7].

W standardzie 1148.4 rozróżnia się dwa rodzaje połączeń pomiędzy układami scalonymi:

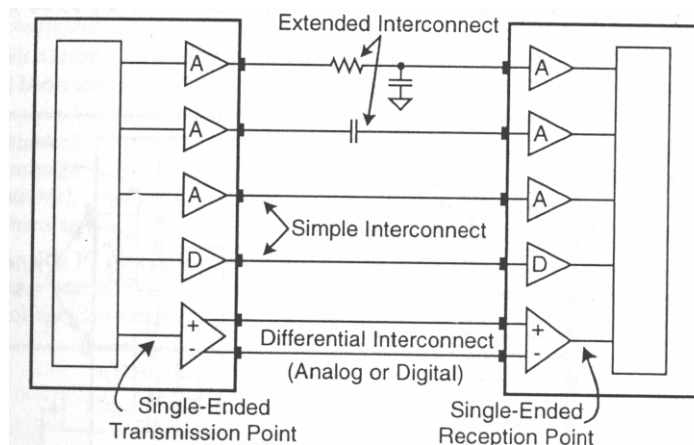
- połączenia proste, są to połączenia bezpośrednie pomiędzy wyprowadzeniami dwóch lub więcej różnych układów scalonych,
- połączenia rozszerzone, są to połączenia nie typu prostego czyli wykonywane za pośrednictwem elementów dyskretnych.

Standard umożliwia również definiowanie połączeń różnicowych (para przewodów przenosi pojedynczy sygnał). Ponieważ obserwuje się ciągły postęp technologiczny polegający na coraz większej integracji elementów wewnątrz układów scalonych, zachodzi pytanie czy warto opracowywać technikę pomiaru elementów zewnętrznych skoro mogą one niedługo zniknąć z płytek drukowanych. Niestety okazuje się, że wiele elementów, ze względu na naturę czy wartość realizowanego parametru, nie da się jeszcze długo zintegrować wewnątrz układów scalonych. Do takich elementów należą m.in.:

- dopasowanie impedancji linii; ze względu na uniwersalność niektórych układów scalonych często nie stosuje się dopasowania impedancji w buforach sygnałowych

układów typowych bo to automatycznie ogranicza zakres możliwych zastosowań danego elementu,

- duża rozpraszana moc,
- duże wartości realizowanych pojemności i indukcyjności,
- możliwość zastosowania do różnych celów w zależności od elementów zewnętrznych,
- dokładność wykonania.



Rys. 14.11. Przykłady połączeń w układzie mieszanym [7].

Celem standardu IEEE1149.4 jest możliwość wykrywania następujących uszkodzeń:

- zwarcia,
- przerwy,
- braki elementów analogowych i cyfrowych,
- błędny montaż (zamiana elementu prawidłowego na inny nieprzewidziany),
- pomiary niektórych parametrów z użyciem sygnałów stałych i o niskiej częstotliwości.

14.9.2. Wyprowadzenia cyfrowe układu scalonego.

Wyprowadzenia cyfrowe układu scalonego są traktowane identycznie jak w standardzie IEEE1149.1. Jediną różnicą jest nazwa komórki dołączonej do takiego wyprowadzenia. Teraz dołączone są one do komórki DBM (ang. Digital Boundary Module) a jego budowa jest identyczna jak wcześniej przedstawianej komórki rejestru Boundary-Scan.

14.9.3. Wyprowadzenia analogowe.

Wyprowadzenia analogowe układu scalonego są połączone przez bloki nazwane w skrócie ABM (ang. Analog Boundary Module). Komórka ABN zawiera rejestr oraz zestaw innych elementów niezbędnych do wspomagania wykonywania testów analogowych. Komórka ABM może pracować w dwóch głównych trybach:

- emulacja stylu jak dla IEEE1149.1 w celu testowania połączeń wewnętrznych oraz,
- wspomaganie podawania sygnału analogowego oraz jego pomiaru.

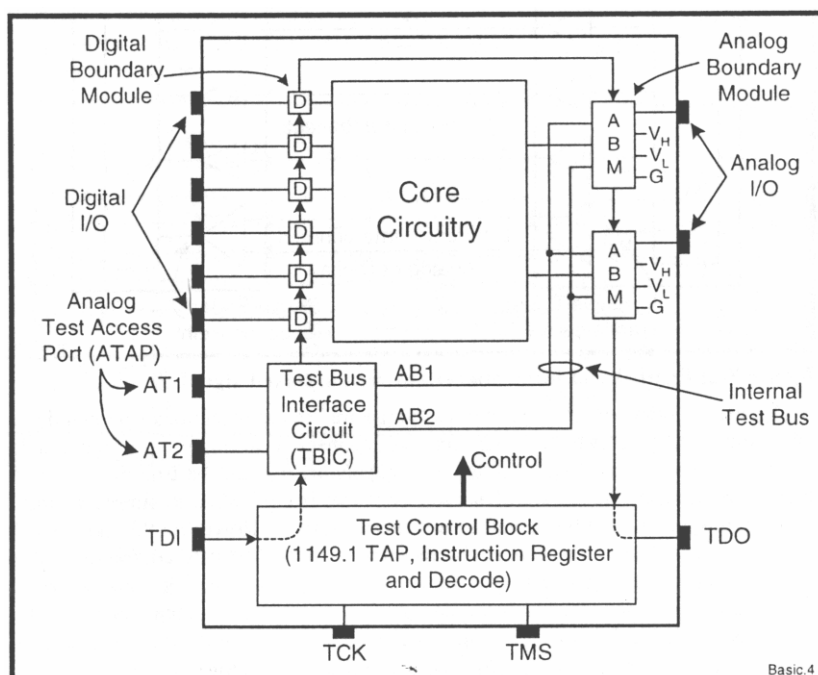
Wyprowadzenia analogowe, w odróżnieniu od cyfrowych, mogą nie mieć wyraźnie sprecyzowanego trybu pracy (kierunku przesyłania sygnału). Przykładami takich wyprowadzeń mogą być np. wyprowadzenia przyłączenia rezonatorów kwarcowych lub kondensatorów odsprężających.

14.9.4. Ogólna struktura układu zgodnego z 1148.4.

Ogólna struktura układu zawierającego strukturę testową zgodną z IEEE1149.4 przedstawiono na rysunku 14.18. W skład wchodzi następujące elementy:

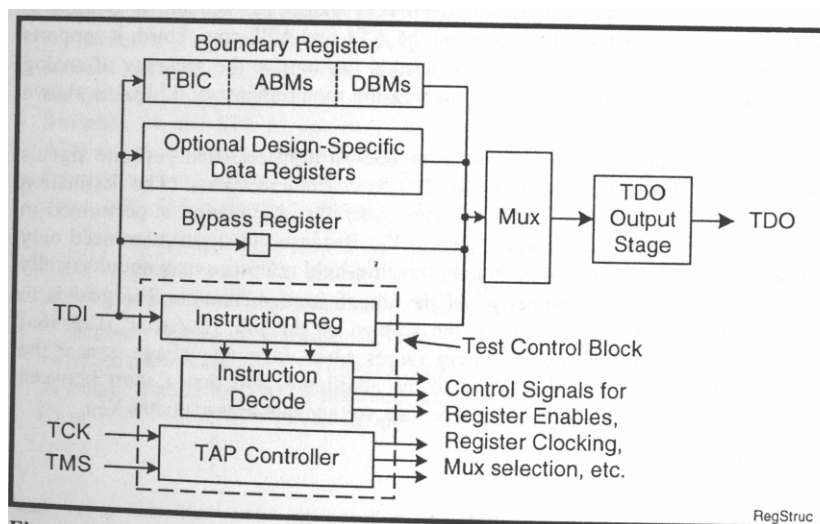
- kontroler TAP, identyczny jak dla układu cyfrowego,
- wyprowadzenia cyfrowe I/O otoczone komórkami DBM,
- wyprowadzenia AT1 oraz AT2 dostępu do zewnętrznych analogowych sygnałów testowych ATAP (ang. Analog Test Acces Port),

- układ sterujący testowymi sygnałami analogowymi TBIC (ang. Test Bus Interface Circuit),
- wyprowadzenia analogowe połączone z pośrednictwem komórek ABM.



Rys. 14.18. Ogólna architektura układu scalonego wyposażonego w interfejs zgodny z IEEE1149.4 [7].

Podobnie jak w standardzie IEEE1149.1 rejestr docelowy jest włączany pomiędzy wyprowadzenia TDI a TDO. Obowiązkowymi rejestrami są: rejestr instrukcji IR, obejścia BYPASS, oraz rejestr brzegowy zawierający obecnie komórki DBM jak i ABM.



Rys. 14.15. Szczegóły budowy rejestrów w układzie 1148.4 [7].

14.9.5. Klucze półprzewodnikowe.

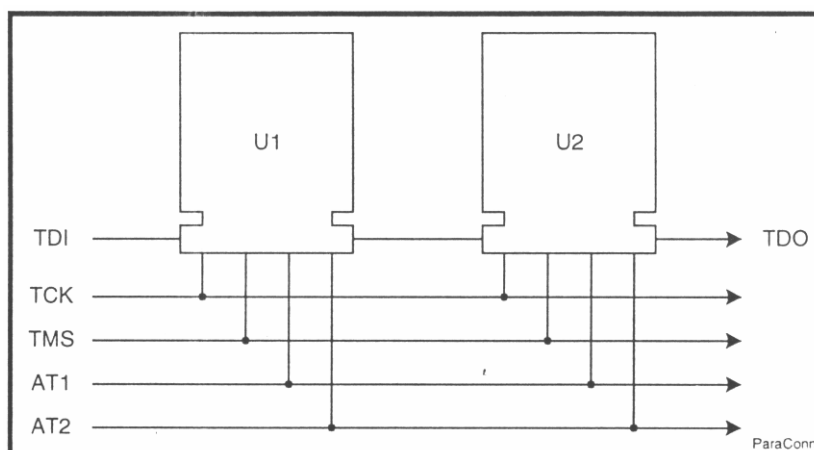
W tabeli 14.7 podano podstawowe parametry kluczy sygnałów analogowych. Klucze takie są niezbędne w celu realizacji przekazywania wybranych sygnałów z/do układu scalonego poprzez wyprowadzenia AT1 i AT2. Klucze tranzystorowe są nieliniowe, mogą mieć charakter jednokierunkowy, wprowadzają dość dużą rezystancję szeregową ale mają małe wymiary i nadają się do scalenia w docelowym testowanym układzie.

Parameter	Mechanical Relay (SMT)	CMOS Switch (0.35 μ)	Bipolar Switch (0.35 μ)
On-Resistance	10 ⁻² ohms	10 ² to 10 ³ ohms	(see discussion)
Off-Resistance	10 ¹² ohms	10 ¹² ohms	10 ¹⁰ ohms
Bidirectional?	Yes	Yes (see discussion)	No
Switching time	≥ 500 μ seconds	<1 μ second	<1 μ second
Area (approx.)	200 x 750 mils = 96.7x10 ⁶ μ m ²	20 μ m ²	100 to 5000 μ m ²

Tabela 14.6. Porównanie parametrów kluczy sygnałowych [7].

14.9.5. ATAP (ang. Analog Test Access Port).

Połączenie bloku testującego dokonywane jest poprzez zespół wyprowadzeń nazywanych ATAP (ang. Analog Test Access Port). W porównaniu do standardu cyfrowego zawarte są dwie dodatkowe linie AT1 oraz AT2. Przeznaczeniem tych linii jest podanie pobudzenia (zazwyczaj na AT1) oraz pomiar odpowiedzi (zazwyczaj poprzez AT2 ale nie jest to obowiązkowe). Typowe połączenie 2 układów przedstawione jest na rysunku 14.21. Standard umożliwia oddzielenie wyprowadzeń AT1 oraz AT2 dla wielu różnych układów. Takie połączenie jest dopuszczalne ze względu na możliwość oddziaływania jednego badanego układu na drugi, np. poprzez wprowadzane szумы.



Rys. 14. 21. Połączenia wyprowadzeń kontrolera ATAP dwóch układów zgodnych z IEEE1149.4. Wspólne połączenie wyprowadzeń AT1 oraz AT2 obu układów nie jest obowiązkowe [7].

14.9.6. Układ TBIC (ang. Test Bus Interface Circuit).

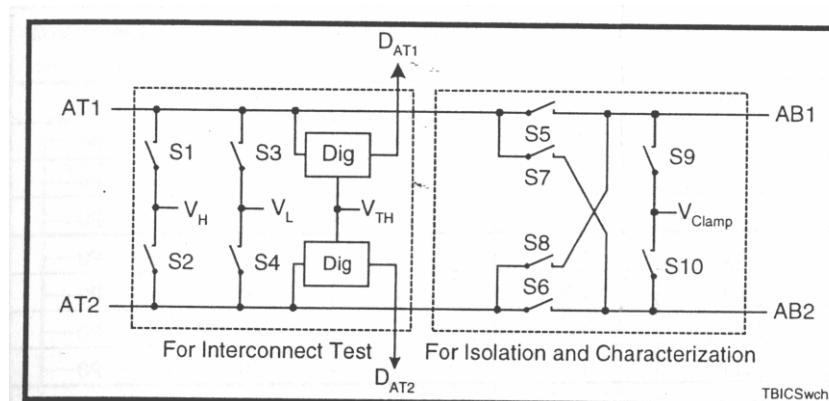
Układ TBIC można użyć w celu:

- izolacji lub połączenia wewnętrznych szyn analogowych AB1 oraz AB2 od wyprowadzeń zewnętrznych AT1 oraz AT2,
- wykonania testów połączeń wyprowadzeń AT1 oraz AT2 układu scalonego z innymi komponentami,
- wykonania pomiarów pewnych parametrów analogowych.

Klucze S1-S4 oraz komparatory Dig służą do wykonania testów połączeń wewnętrznych w stylu 1148.1 na wyprowadzeniach AT1 oraz AT2. Napięcie progowe V_{TH} jest niezbędne w celu określenia czy wartości sygnałów w mierzonych węzłach kwalifikują się do stanu niskiego czy wysokiego. Dokładność napięcia V_{TH} nie musi być wysoka. Napięcia V_H i V_L są wartościami odpowiadającymi stanom wysokiemu i niskiemu i mogą stanowić poziom zasilania i masy.

Klucze S5, S6 służą do połączenia/odizolowania węzłów wewnętrznych ABx i zewnętrznych ATx. Klucze S9 i S10 są opcjonalne i umożliwiają dołączenie wewnętrznych sygnałów do wartości zapobiegającej pojawieniu się zjawisk niepożądanych takich jak np. wzbudzenia lub efekty pasożytnicze. Klucz S9 jest zawsze w przeciwnym stanie niż S5, podobnie

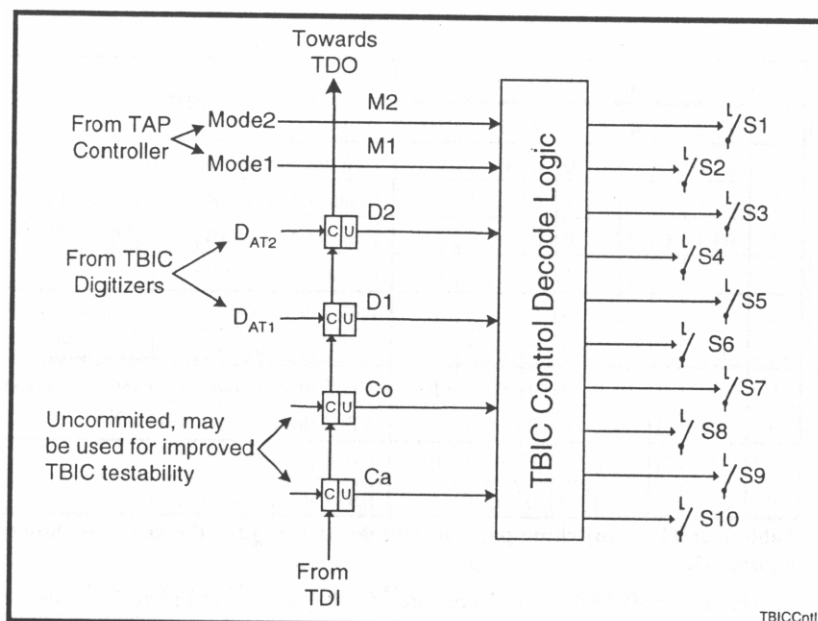
S10 jest w stanie przeciwnym niż S5. Na rysunku 14.22 jest 10 kluczy a w związku z tym są możliwe 1024 kombinacje ich włączenia. Jednakże, tylko 10 kombinacji przedstawionych w tabeli 14.8, jest dopuszczalnych.



Rys. 14.22. Struktura układu TBIC. Bloki DIG są komparatorami (digitizerami) wartości sygnału AT1 i AT2 z wartością progową V_{TH} . [7].

P#	Switch State (S1 – S10)										Function
	1	2	3	4	5	6	7	8	9	10	
0	0	0	0	0	0	0	0	0	1	1	ATn disconnect (Hi-Z), clamp ABn
1	0	0	0	0	0	1	0	0	1	0	Connect AT2 to AB2
2	0	0	0	0	1	0	0	0	0	1	Connect AT1 to AB1
3	0	0	0	0	1	1	0	0	0	0	Connect ATn to ABn
4	0	0	1	1	0	0	0	0	1	1	AT1/2 drive 00 out
5	0	1	1	0	0	0	0	0	1	1	AT1/2 drive 01 out
6	1	0	0	1	0	0	0	0	1	1	AT1/2 drive 10 out
7	1	1	0	0	0	0	0	0	1	1	AT1/2 drive 11 out
8	0	0	0	0	0	1	1	0	1	0	For characterization
9	0	0	0	0	1	0	0	1	0	1	For characterization

Tabela 14.7. Przełączanie kluczy bloku TBIC z rys. 14.22[7].



Rys. 14.23. Układ sterujący kluczami bloku TBIC[7].

Na rysunku 14.23 przedstawiony jest układ sterujący blokiem TBIC (w wersji minimalnej). Używany jest rejestr brzegowy oraz dwie linie sterujące Model1 oraz Mode2 kontrolera TAP, które

są funkcją bieżącej instrukcji załadowanej do rejestru instrukcji. Komórki rejestru brzegowego są nazwane Ca, Co, D1 oraz D2. W stanie „łapania” (Capture) komórki D1 i D2 zapamiętują wartości D_{AT1} oraz D_{AT2} z rysunku 14.22. Blok „TBIC Control Decode Logic” wytwarza sygnały sterujące kluczami analogowymi. Przypisanie sygnałów Model oraz Mode2 w zależności od bieżącej instrukcji przedstawione jest w tabeli 14.7. W tabeli 14.9 przedstawione przypisanie wzorów przełączeń wg tabeli 14.7 odpowiadającej poszczególnym stanom komórek rejestru brzegowego. Znak * oznacza stan nieprzypisany. Wynikające stąd wartości równań logicznych realizujące poszczególne klucze przedstawione są w tabeli 14.9.

TAP Instruction	Model (M1)	Mode2 (M2)
EXTEST, CLAMP, RUNBIST	1	1
PROBE, INTEST	0	1
HIGHZ	1	0
BYPASS, SAMPLE, PRELOAD, IDCODE, USERCODE	0	0

Tabela 14.7. Przypisanie wartości sygnałom Model oraz Mode2 w zależności od bieżącej instrukcji [7].

Cells Ca/Co/D1/D2	Modes 1/2 = 11 EXTEST, etc.	Modes 1/2 = 01 PROBE, etc.	Modes 1/2 = 10 HIGHZ	Modes 1/2 = 00 BYPASS, etc.
0000	P0	P0	P0	P0
0001	P1	P1	P0	P0
0010	P2	P2	P0	P0
0011	P3	P3	P0	P0
0100	P4	*	P0	P0
0101	P5	*	P0	P0
0110	P6	*	P0	P0
0111	P7	*	P0	P0
1000	P0	*	P0	P0
1001	P8	*	P0	P0
1010	P9	*	P0	P0
1011	*	*	P0	P0
1100	*	*	P0	P0
1101	*	*	P0	P0
1110	*	*	P0	P0
1111	*	*	P0	P0

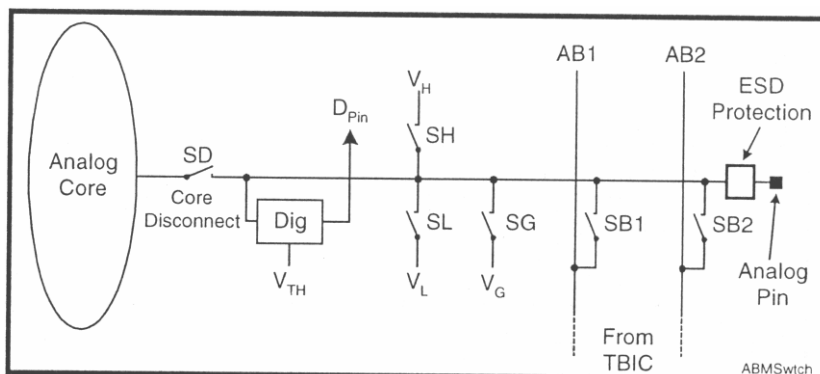
Tabela 14.8. Wybór trybu pracy bloku TBIC w zależności od zawartości komórek rejestru brzegowego [7].

TBIC Switch	Logic Equation (a trailing “*” indicates inversion)
S1	$Ca * Co * D1 * M1 * M2$
S2	$Ca * Co * D2 * M1 * M2$
S3	$Ca * Co * D1 * M1 * M2$
S4	$Ca * Co * D2 * M1 * M2$
S5	$Co * D1 * M2 * (Ca * + D2 * M1)$
S6	$Co * D2 * M2 * (Ca * + D1 * M1)$
S7	$Ca * Co * D1 * D2 * M1 * M2$
S8	$Ca * Co * D1 * D2 * M1 * M2$
S9	$S5 *$
S10	$S6 *$

Tabela 14.9. Równania logiczne sterujące kluczami S1-S10 bloku TBIC [7].

14.9.7. Komórka ABM (ang. Analog Boundary Module).

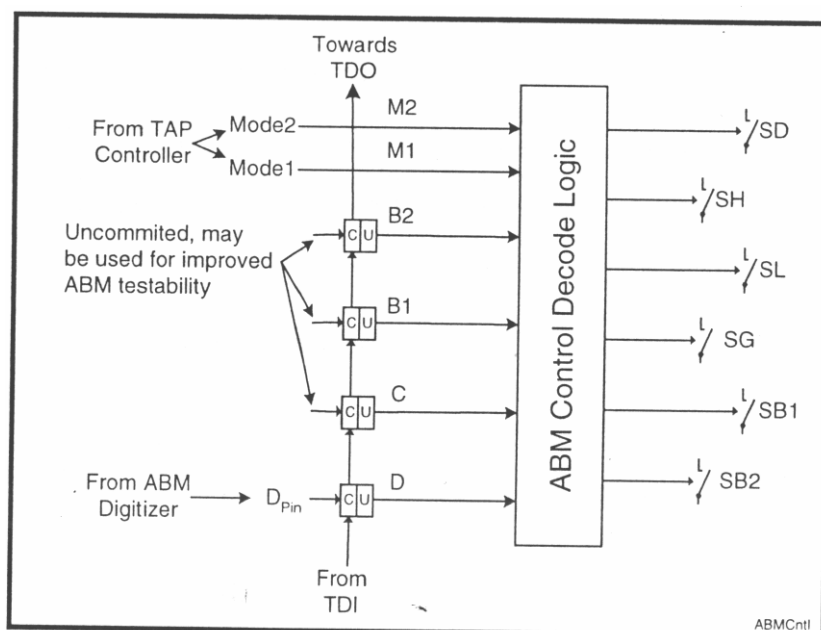
Komórka ABM przedstawiona jest na rysunku 14.24. Klucz S_D może fizycznie nie istnieć w układzie (duży prąd wyjściowy, komórka I/O i tak posiada wejście sterujące o podobnej funkcjonalności, wejście o dużej impedancji i inne powody). Ponieważ klucz może fizycznie nie być realizowany ale jego funkcjonalność występuje, nazywamy go kluczem conceptualnym. Digitizer Dig wytwarza stan logiczny (na podstawie napięcia V_{TH}) napięcia analogowego, jest to wykorzystywane do testów jak w 1148.1. Napięcia V_H i V_L są wykorzystywane do wytwarzania stanów logicznych natomiast V_G jest wykorzystywane do pomiarów elektrycznych. Sygnał V_G powinien charakteryzować się wysoką stabilnością czasową, temperaturową i niską impedancją wyjściową. Klucze SB1 oraz SB2 umożliwiają dołączenie wyprowadzenia analogowego do linii AB1 oraz AB2. Wymaga się aby linia AB1 mogła dostarczać prąd do wyprowadzenia analogowego natomiast linia AB2 mogła ten prąd pobierać. Standard 1148.4 definiuje 20 możliwych ustawień kluczy, są one przedstawione w tabeli 14.10.



Rys. 14.24. Komórka ABM dla wyprowadzenia analogowego układu scalonego [7].

P#	Switch State (0/1=open/closed)						Pin State
	SD	SH	SL	SG	SB1	SB2	
0	0	0	0	0	0	0	Completely isolated
1	0	0	0	0	0	1	Monitored by AB2
2	0	0	0	0	1	0	Connected to AB1
3	0	0	0	0	1	1	Connected to AB1, monitored by AB2
4	0	0	0	1	0	0	Connected to V_G
5	0	0	0	1	0	1	Connected to V_G , monitored by AB2
6	0	0	0	1	1	0	Connected to V_G and AB1
7	0	0	0	1	1	1	Connected to V_G & AB1, monitored by AB2
8	0	0	1	0	0	0	Connected to V_L
9	0	0	1	0	0	1	Connected to V_L , monitored by AB2
10	0	0	1	0	1	0	Connected to V_L and AB1
11	0	0	1	0	1	1	Connected to V_L & AB1, monitored by AB2
12	0	1	0	0	0	0	Connected to V_H
13	0	1	0	0	0	1	Connected to V_H , monitored by AB2
14	0	1	0	0	1	0	Connected to V_H and AB1
15	0	1	0	0	1	1	Connected to V_H & AB1, monitored by AB2
16	1	0	0	0	0	0	Connected to core, isolated from test
17	1	0	0	0	0	1	Connected to core, monitored by AB2
18	1	0	0	0	1	0	Connected to core and AB1
19	1	0	0	0	1	1	Connected to core & AB1 monitored by AB2

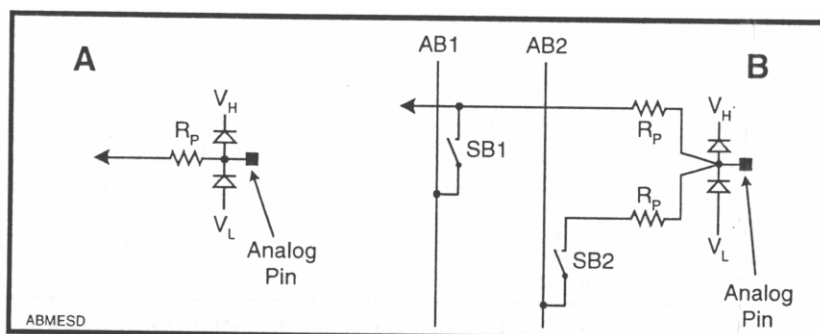
Tabela 14.10. Możliwe ustawienia kluczy bloku ABM i realizowane funkcje [7].



Rys. 14.12. Układ sterujący kluczami z rys. 14.24 [7].

Układ sterujący kluczami bloku ABM składa się z 4 rejestrów brzegowych i dwóch sygnałów sterujących Mode1 oraz Mode2 (jak dla TBIC). Podobnie jak poprzednio można wyprowadzić równania logiczne sterujące poszczególnymi kluczami.

W celu polepszenia dokładności pomiarów wykonywanych poprzez IEEE1149.4 zaleca się stosowanie zmodyfikowanego układu zabezpieczeń przed ESD jak pokazano na rysunku 14.12.



Rys. 14.12. Modyfikacja układu zabezpieczającego przed ESD. Dzięki modyfikacjom spadek napięcia na R_p nie przechodzi do linii pomiarowej AB2 [7].

14.9.8. Zestaw instrukcji 1148.4.

Zestaw instrukcji standardu 1148.4 jest powiększony w stosunku do standardu 1148.1, niektóre instrukcje są wykonywane nieco inaczej. Wprowadzona jest nowa obowiązkowa instrukcja PROBE, inaczej wykonywane są instrukcje EXTEST oraz INTTEST.

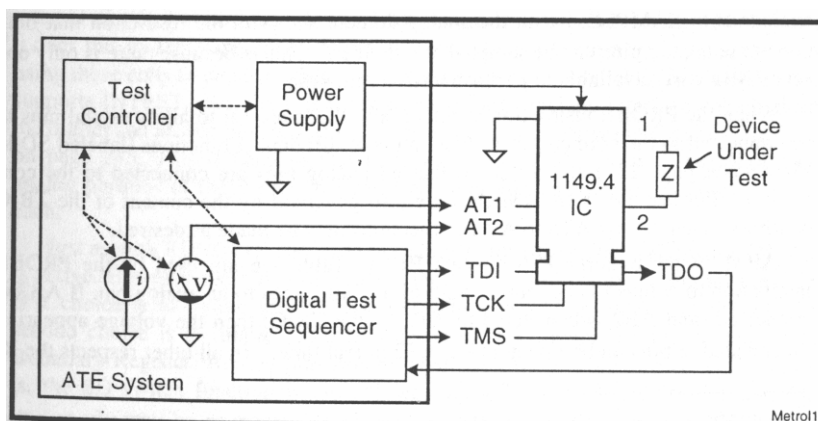
Instrukcje BYPASS, PRELOAD, IDCODE oraz USERCODE wykonywane są jak w 1148.1. Instrukcja HIGHZ wykonywana jak 1148.1 a wyprowadzenia analogowe są całkowicie izolowane. Instrukcja SAMPLE jest wykonywana jak 1148.1 z tym dodatkiem, że na wyprowadzeniach analogowych łapana jest wartość zdigitalizowana.

EXTEST: Instrukcja obowiązkowa. Na wyprowadzeniach cyfrowych układu scalonego zachowuje się identycznie jak w standardzie 1148.1. Wyprowadzenia analogowe mogą emulować wyprowadzenia cyfrowe poprzez zablokowanie wyprowadzenia bądź też podłączenie do V_H lub V_L . To umożliwia włączenie wyprowadzeń analogowych w testy połączeń. Niestety, niektóre wyprowadzenia analogowe mogą mieć połączenia rozszerzone i to może spowodować, że nie będą mogły emulować wyprowadzeń cyfrowych - np. terminacja 50omów do masy. Takie wyprowadzenie należy traktować jako uziemione. Później, podczas pomiarów analogowych

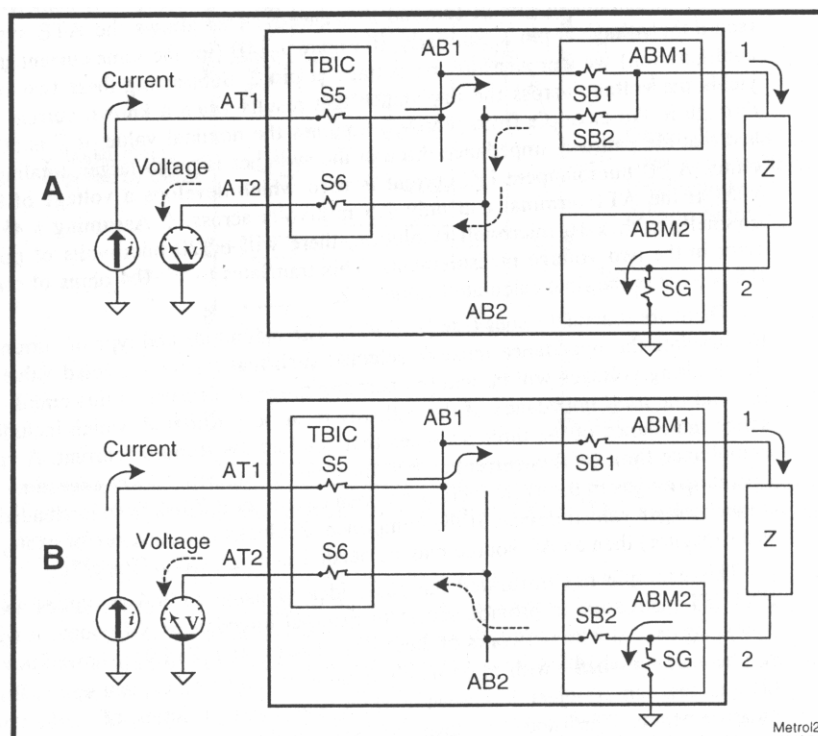
impedancji zewnętrznych można wykryć zwarcie lub przerwę, aczkolwiek sekwencyjnie z mniejszą szybkością.

W przypadku połączenia dwóch wyprowadzeń analogowych przez kondensator, węzły takie mogą być traktowane jako logicznie niezależne a ich testowanie powinno być prowadzone z użyciem dwóch niezależnych sekwencyjnych wektorów testowych STV. Takie testowanie jest możliwe ze względu na to, że każdy ABM ma właściwości dwukierunkowe umożliwiające jednocześnie wystawianie wartości logicznej jak również obserwację bieżącego stanu.

W przypadku użycia instrukcji EXTEST do wykonania pomiarów analogowych wykorzystywany w tym celu blok ABM porusza się w obszarze P1-P5 tabeli 14.10. Należy zauważyć, że w czasie gdy pewien ABM jest w trybie obserwacyjnym inne bloki ABM oraz DBM mogą służyć do ustalenia warunków pomiarowych.



Rys. 14.13. Środowisko testowe 1148.4 [7].



Rys. 14.14. Dwa pomiary (A) oraz (B) używane do ustalenia napięcia występującego na impedancji Z [7].

Rozważmy układ przedstawiony na rysunku 14.13. Środowisko testowe (ATE) używa 6 linii połączeniowych. Można za jego pomocą dokonać pomiaru impedancji Z włączonej do wyprowadzeń układu scalonego. Najpierw system ATE musi dostarczyć zasilanie do badanego układu. Następnie system ATE dostarcza wektory testowe na wyprowadzenia TAP. Najpierw badana jest integralność łańcucha 1148.4 (tak jak łańcucha 1148.1). Później przeprowadzone są testy połączeń (jak dla 1148.1). Następnie badana jest wartość impedancji Z – jak na rysunku 14.14.

W celu pomiaru impedancji Z należy:

- wprowadzić mały prąd na wyprowadzenie AT1, prąd ten płynie poprzez AT1 do TBIC dalej poprzez linię AB1 do ABM1, dalej poprzez impedancję Z do ABM2 i poprzez klucz S_G do napięcia odniesienia V_G ,
- połączyć woltomierz (i pomierzyć wartość napięcia) ATE do ABM1 i poprzez AB2 do wyprowadzenia pomiarowego AT2,
- połączyć woltomierz do ABM2 poprzez AB2/AT2,
- obliczyć impedancję jako różnicę napięć (otrzymanych poprzez pomiary w punktach poprzednich) podzieloną przez wartość znanego prądu wprowadzonego do wyprowadzenia AT1.

Przykład: Załóżmy, że mierzymy wartość w zakresie 50Ω natomiast rezystancja ścieżki pomiarowej wynosi $5k\Omega$. Załóżmy, że prąd wprowadzony do AT1 wynosi $50mA$ co daje spadek napięcia na impedancji Z o wartości $2.5mV$. Zakładając, że używamy woltomierza z 4,5 cyframi i rozdzielczością równą $10\mu V$ otrzymujemy $20\mu V$ ewentualnego błędu pomiarowego. To przekłada się na $\pm 0,4\Omega$ co daje 0,8% wartości mierzonej równej 50Ω .

Należy dobierać wartość i częstotliwość (w przypadku pomiarów kondensatorów i indukcyjności) prądu w ten sposób aby były one odpowiednie dla mierzonej, oczekiwanej impedancji Z . Wynikowe wartości napięć muszą znajdować się w zakresie dopuszczalnych wartości dla wszystkich elementów występujących w łańcuchu pomiarowym, tj. kluczy i ścieżek. Prądy dopuszczalne do pomiarów są rzędu $100-200\mu A$.

CLAMP: Instrukcja nieobowiązkowa. Na wyjścia cyfrowe i wyprowadzenia analogowe ustawia wartość wpisaną do rejestru brzegowego a jako rejestr docelowy wybiera rejestr obejścia. Służy do ustawienia warunków pomiarowych przy jednoczesnym skróceniu długości łańcucha TDI – TDO.

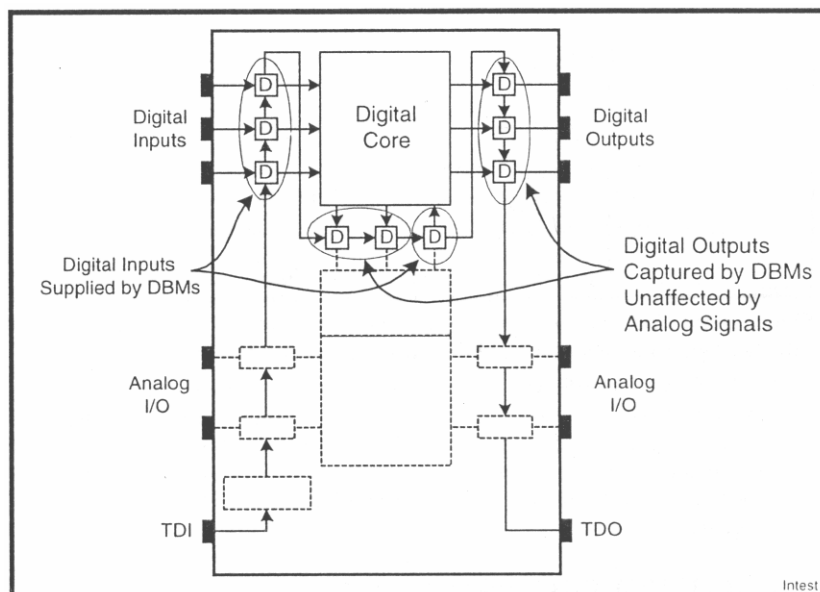
HIGHZ: Instrukcja nieobowiązkowa. Odłącza wszystkie wyjścia cyfrowe oraz wyprowadzenia analogowe a jako rejestr docelowy wybiera rejestr obejścia.

PROBE: Instrukcja obowiązkowa w 1148.4. Kod instrukcji nie jest ustalony przez standard. Jako rejestr docelowy wybiera rejestr brzegowy. Instrukcja jak SAMPLE dla 1148.1 z tą różnicą, że w danym momencie monitorowane jest tylko jedno wyprowadzenie analogowe gdyż jest dostępne tylko jedno wyprowadzenie obserwacyjne AT2. W czasie wykonywania instrukcji bloki DBM podłączają wyprowadzenia cyfrowe do logiki układu, komórki ABM włączają wyprowadzenia analogowe do rdzenia układu. Połączenia kluczy TBIC są kontrolowane przez rejestr TBIC i mogą zostać ustawione do obserwacji określonych wyprowadzeń. W czasie wykonywania instrukcji mogą być wykorzystywane polecenia P16-P19 wg tabeli 14.10.

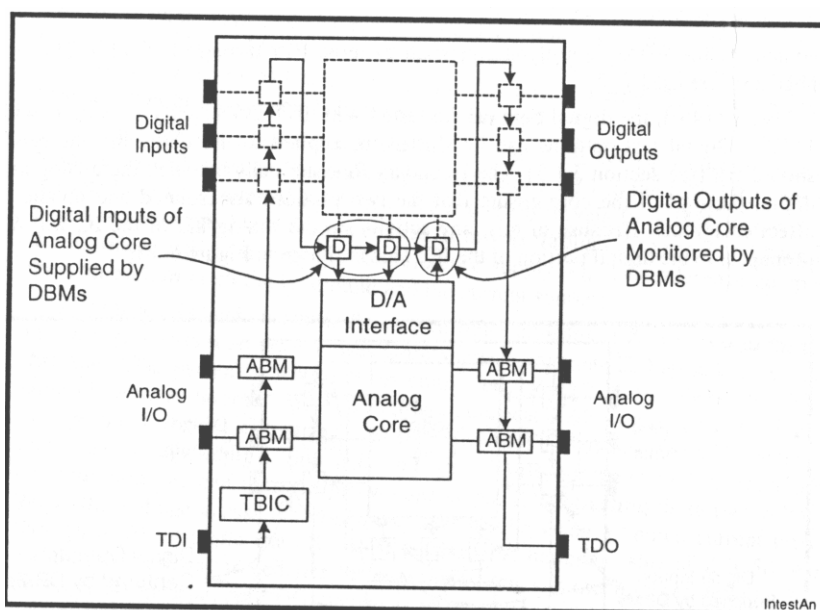
RUNBIST: Instrukcja nieobowiązkowa, kod instrukcji nie jest narzucony przez standard. Jako rejestr docelowy wybierany jest taki rejestr, który zostanie uzupełniony wynikiem samotestu po jego zakończeniu (może to być również rejestr brzegowy). Podobnie jak w 1148.1 w czasie wykonywania samotestu należy zadbać o odpowiednie wartości sygnałów na wyprowadzeniach wyjściowych i analogowych. Są dwie możliwe opcje: odłączenie lub stan bezpieczny. Wynik testu nie powinien zależeć od jakichkolwiek warunków, sygnałów czy połączeń zewnętrznych.

INTEST: Instrukcja nieobowiązkowa, kod instrukcji nie jest narzucony przez standard. Instrukcja przeznaczona do testowania struktury wewnętrznej układu scalonego w czasie gdy jest on już zamontowany na płycie drukowanej. W przypadku implementacji instrukcji, rejestr brzegowy musi zawierać komórki w interfejsie pomiędzy analogową i cyfrową częścią układu scalonego jak to pokazano na rysunku 14.28. Jest to spowodowane potrzebą obserwacji i ustawiania wyprowadzeń cyfrowych do części analogowej, które normalnie nie są dostępne jako wyprowadzenia układu scalonego. Podobnie jak w przypadku instrukcji RUNBIST, wyprowadzenia wyjściowe cyfrowe należy ustawić w stan wysokiej impedancji albo w stan bezpieczny. Wyprowadzenia analogowe powinny pozostać połączone do rdzenia układu scalonego. W czasie wykonywania instrukcji INTEST można dokonać sprawdzenia działania części analogowej układu scalonego, jak to pokazano na rysunku 14.30. Wówczas sygnały cyfrowe nie są obserwowane lecz służą do ustawiania wartości sygnałów na styku bloków analogowego i cyfrowego. Dodatkowo, niektóre wejścia mogą być pobudzane i obserwowane przez komórki ABM (tylko jedno w danym momencie

czasu). Instrukcje INTEST oraz PROBE są do siebie podobne. Różnica polega na tym, że podczas instrukcji PROBE połączenia wewnętrzne pomiędzy analogową i cyfrową częścią układu scalonego są zachowane natomiast podczas INTEST części te mogą być rozdzielone a układ analogowy jest sterowany również z komórek DBM włączonych do interfejsu analogowo-cyfrowego.



Rys. 14.28. Testowanie części cyfrowej układu mieszanego poprzez wykonanie instrukcji INTEST [7].



Rys. 14.30. Testowanie rdzenia analogowego można przeprowadzić poprzez wektory testowe na interfejsie D/A i poprzez sygnały dostarczone/odczytywane przez komórki ABM.

14.9.9. Inne cechy standardu 1148.4.

Dotychczas opisana implementacja standardu 1148.4 jest wersją minimalną. Elementami dodatkowymi mogą być:

- dodatkowy blok TBIC i wejścia AT1N, AT2N służące do obsługi sygnałów różnicowych,
- tworzenie podziału szyn AB w celu izolacji grup wyprowadzeń sygnałów analogowych.

Standard opisuje również parametry elektryczne, które muszą być spełnione w odniesieniu do części analogowej interfejsu. Niektóre z tych parametrów przedstawione są poniżej:

- impedancja ścieżki od AT1 poprzez TBIC, AB1, ABM do wyprowadzenia układu musi być mniejsza niż 10kΩ,

- impedancja AT2 – PIN mniejsza niż $10k\Omega$,
- impedancja klucza włączającego napięcie V_L, V_H lub V_G powinna być mniejsza niż $10k\Omega$ i inne ...

15. Projektowanie systemów mieszanych analogowo - cyfrowych.

Zastosowanie układów mieszanych:

- telekomunikacja,
- elektronika użytkowa,
- komputery i urządzenia peryferyjne,
- systemy multimedialne,
- systemy samochodowe,
- biomedycyna,
- robotyka i inne.

Zalety stosowania układów mieszanych:

- zredukowany rozmiar całego systemu,
- możliwe jest zwiększenie szybkości działania,
- zmniejszenie zużycia mocy,
- poprawa niezawodności,
- zmniejszenie kosztów systemu.

Problemy związane z projektowaniem układów mieszanych:

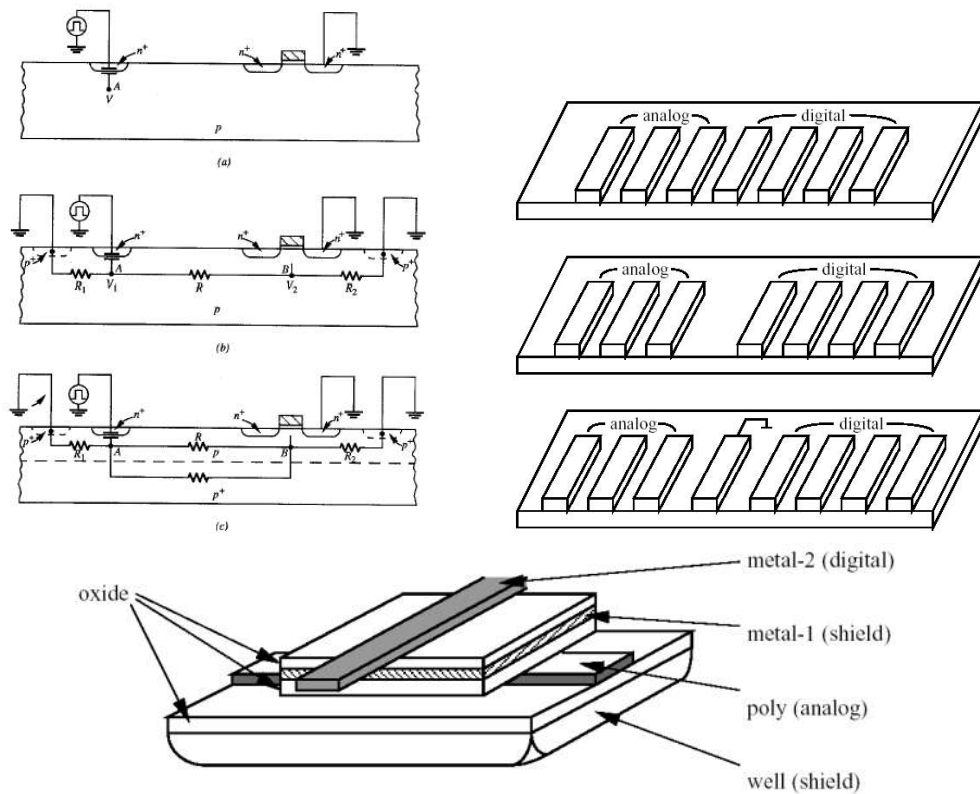
- projektowanie układów analogowych jest trudne,
- sygnały analogowe są bardzo małe (μV) w porównaniu do cyfrowych (V),
- w układzie następuje oddziaływanie układów analogowych na cyfrowe i odwrotne (znacznie bardziej niekorzystne),
- układy analogowe nie są standaryzowane,
- nie ma możliwości automatycznego projektowania układów analogowych,
- brak projektantów układów mieszanych.

Mechanizmy oraz skutki przenoszenia zakłóceń:

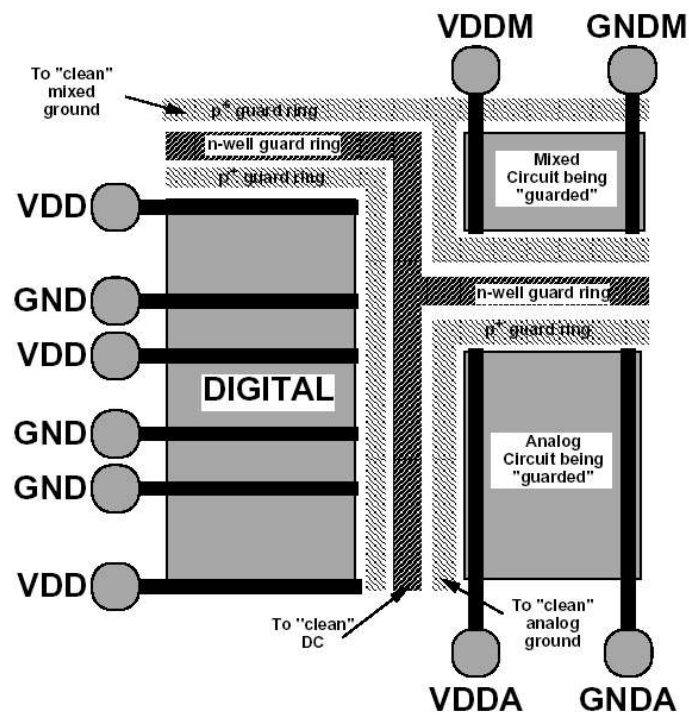
- scalone układy mieszane VLSI zawierają wiele bloków, które oddziałują na siebie,
- układy są umieszczone na wspólnym podłożu co powoduje powstanie sprzężeń podłożowych,
- korzystanie ze wspólnych wyprowadzeń zewnętrznych,
- sprzężenia indukcyjne i rezystancyjne wyprowadzeń zasilających,
- sprzężenia pojemnościowe.

Metody zmniejszenia zakłóceń w układach mieszanych:

- rozwiązania systemowe: układy hybrydowe oraz MCM (ang. multi chip modules), odpowiednie zaprojektowanie timingów sygnałów (np. próbkowanie i porównywanie sygnałów analogowych w momentach braku aktywności części cyfrowej), używanie analogowych układów różnicowych, stosowanie specjalnej logiki generującej mniejsze zakłócenia,
- ekranowanie: stosowanie pierścieni ochronnych oraz ekranowania liniami metalicznymi,

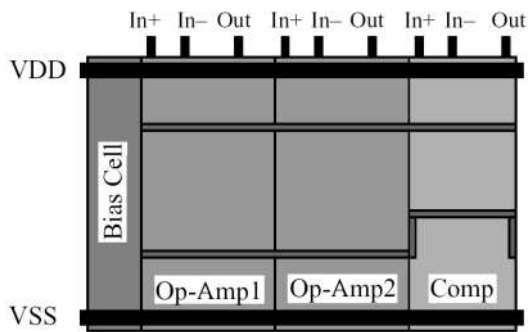


GUARD RINGS

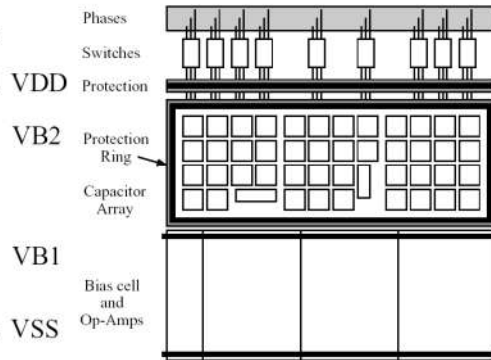


- wczesne planowanie rozmieszczenia bloków (chip floorplanning),

LAYOUT OF SC CIRCUITS

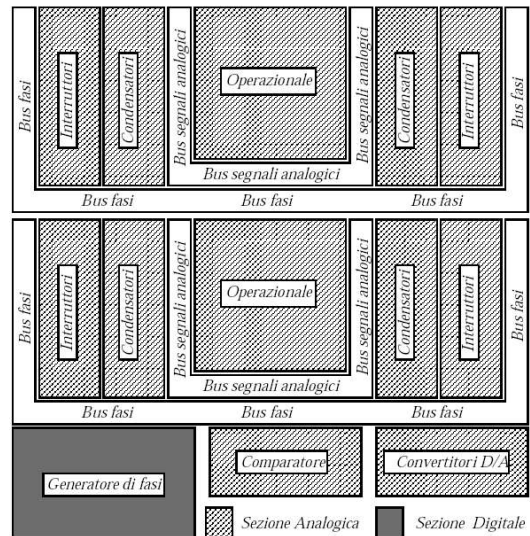
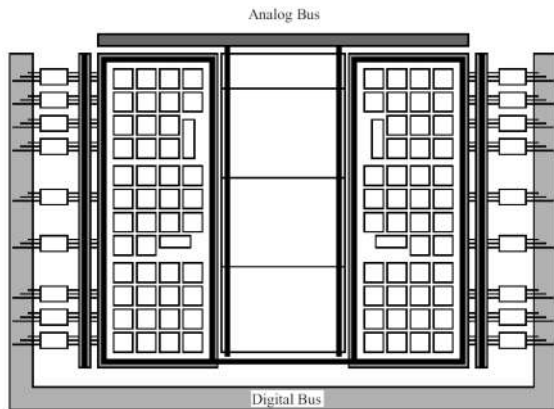


LAYOUT OF SC CIRCUITS

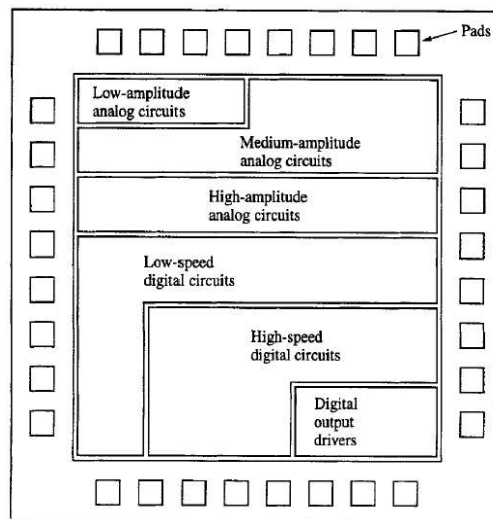


FLOORPLAN SC CIRCUITS

LAYOUT OF SC CIRCUITS



FLOORPLAN



- odpowiednie zaplanowanie wyprowadzeń układu scalonego (chip wiring),
- odpowiednie zaprojektowanie linii zasilających.

16. Tendencje rozwojowe układów ASIC.

Układy ASIC - tendencje rozwojowe:

- zmniejszanie minimalnych wymiarów geometrycznych realizowanych elementów,

- zmniejszanie grubości tlenku bramkowego a więc konieczność zmniejszania napięcia zasilającego,
- zwiększenie liczby elementów aktywnych i częstotliwości pracy więc zwiększenie zużywanej mocy w układzie scalonym,
- coraz większy udział mocy stałej w stosunku do dynamicznej (układy cyfrowe CMOS),
- stosowanie nowych materiałów jako przewodników (np. miedź) i aktywnych,
- zmiana właściwości modeli tranzystorów (np. tranzystor MOS nie ma już charakterystyki kwadratowej),
- zwiększanie znaczenia programów CAD,
- wzrost znaczenia własności intelektualnej (ang. Intellectual Property),
- coraz szersze stosowanie układów typu system on chip (SoC),
- większa integracja układów analogowych i cyfrowych,
- koszt układów programowalnych zaczyna być porównywalny do układów tradycyjnych więc ich stosowanie zaczyna być również uzasadnione ekonomicznie.

Dla częstotliwości zegara równego 10GHz okres wynosi 100ps, w tym czasie światło w próżni przebywa odległość równą 30mm. W materiałach typowych dla technologii produkcji układów CMOS światło przebywa odległość 15.2mm co jest w przybliżeniu równe szerokości dzisiejszych układów scalonych!

- Dodatek A. Opis BSDL układu scalonego Xilinx XC4010XL

```
--$ XILINX$RCSfile: xc4010xl_pc84.bsd,v $
--$ XILINX$Revision: 1.4 $
--
-- BSDL file for device XC4010XL, package PC84
-- Xilinx, Inc. $State: FINAL $ $Date: 2002-01-29 15:04:35-08 $
-- Generated by createBSDL 2.6
--
-- For technical support, contact Xilinx as follows:
-- North America 1-800-255-7778 hotline@xilinx.com
-- United Kingdom +44 870 7350 610 eurosupport@xilinx.com
-- France (33) 1 3463 0100 eurosupport@xilinx.com
-- Germany (49) 89 991 54930 eurosupport@xilinx.com
-- Japan (81) 3-3297-9163 jhotline@xilinx.com
--
-- BSDL verified to conform to 1148.1b-1994 syntax. This device has been
-- tested by the Intellitech 1148.1 Verification Lab using the Intellitech
-- Eclipse(TM) Scan Diagnostic Tool and the Intellitech RCT(TM). This
-- device has been verified to operate according to the BSDL provided,
-- and is compatible with the IEEE 1148.1 standard when the operating
-- instructions in the BSDL are followed.
-- PH: 603-868-7116 or email: scansupport@intellitech.com
--
-- This BSDL file reflects the pre-configuration JTAG behavior. To reflect
-- the post-configuration JTAG behavior (if any), edit this file as described
-- below. Many of these changes are demonstrated by commented-out template
-- lines preceeding the lines they would replace:
--
-- 1. Enable USER instructions as appropriate (see below).
-- 2. For inputs using uncontrolled paths (e.g. PGCK, SGCK), change
-- boundary cell function from 'input' to 'clock' or 'observe_only'.
-- 3. Set disable result of all pads as configured.
-- 4. Set safe state of boundary cells as necessary.
-- 5. Set safe state of INIT output to X, or as necessary (see below).
-- 5. Rename entity if necessary to avoid name collisions.
-- 6. Change INIT port direction from "in" to "inout" (see below).
-- 7. Change COMPLIANCE_PATTERNS to "(PROGRAM) (1)" (see below).
-- 8. Change INIT boundary cells from internal to controlr, output3,
-- and input, respectively (see below).
-- 9. Remove the design warning regarding keeping INIT low.
--
-- NOTE: Post-configuration JTAG is available only if the BSCAN symbol
-- is instantiated in the FPGA design.
-- NOTE: PULLUP symbols must be instantiated on the TMS and TDI pins
-- in the FPGA design to comply with IEEE Std. 1148.1-1993.

entity XC4010XL_PC84 is

generic (PHYSICAL_PIN_MAP : string := "PC84" );

port (
    CCLK: linkage bit;
    DONE: linkage bit;
    GND: linkage bit_vector (1 to 8);
-- INIT is not a compliance enable after configuration. For post-configuration
-- operation un-comment the next line and comment out the following line so
-- that INIT is of type inout.
-- INIT: inout bit;
    INIT: in bit;
    IO2: inout bit;
    IO3: inout bit;
    IO8: inout bit;
    IO9: inout bit;
    IO19: inout bit;
    IO20: inout bit;
    IO23: inout bit;
    IO24: inout bit;
    IO28: inout bit;
    IO29: inout bit;
    IO42: inout bit;
    IO49: inout bit;
    IO51: inout bit;
    IO55: inout bit;
    IO56: inout bit;
    IO63: inout bit;
    IO64: inout bit;
    IO72: inout bit;
    IO76: inout bit;
    IO77: inout bit;
```

```

IO84: inout bit;
IO85: inout bit;
IO89: inout bit;
IO97: inout bit;
IO98: inout bit;
IO105: inout bit;
IO109: inout bit;
IO110: inout bit;
IO117: inout bit;
IO118: inout bit;
IO126: inout bit;
IO127: inout bit;
IO130: inout bit;
IO131: inout bit;
IO137: inout bit;
IO138: inout bit;
IO141: inout bit;
IO152: inout bit;
IO153: inout bit;
IO158: inout bit;
IO159: inout bit;
IO162: inout bit;
IO163: inout bit;
IO168: inout bit;
IO169: inout bit;
IO179: inout bit;
IO180: inout bit;
IO183: inout bit;
IO184: inout bit;
IO190: inout bit;
IO191: inout bit;
IO194: inout bit;
IO195: inout bit;
IO205: inout bit;
IO206: inout bit;
IO211: inout bit;
IO212: inout bit;
M0: in bit;
M1: inout bit;
M2: in bit;
PROGRAM: in bit;
TCK: in bit;
TDI: in bit;
TDO: out bit;
TMS: in bit;
VDD: linkage bit_vector (1 to 8)
); --end port list

use STD_1149_1_1994.all;

attribute COMPONENT_CONFORMANCE of XC4010XL_PC84 : entity is
    "STD_1149_1_1993";

attribute PIN_MAP of XC4010XL_PC84 : entity is PHYSICAL_PIN_MAP;

constant PC84: PIN_MAP_STRING:=
    "CCLK:P73," &
    "DONE:P53," &
    "GND: (P12,P21,P31,P43,P52,P64,P76,P1)," &
    "INIT:P41," &
    "IO2:P3," &
    "IO3:P4," &
    "IO8:P5," &
    "IO9:P6," &
    "IO19:P7," &
    "IO20:P8," &
    "IO23:P9," &
    "IO24:P10," &
    "IO28:P13," &
    "IO29:P14," &
    "IO42:P18," &
    "IO49:P19," &
    "IO51:P20," &
    "IO55:P23," &
    "IO56:P24," &
    "IO63:P25," &
    "IO64:P26," &
    "IO72:P27," &
    "IO76:P28," &
    "IO77:P29," &

```

```

"IO84:P35," &
"IO85:P36," &
"IO89:P37," &
"IO97:P38," &
"IO98:P39," &
"IO105:P40," &
"IO109:P44," &
"IO110:P45," &
"IO117:P46," &
"IO118:P47," &
"IO126:P48," &
"IO127:P49," &
"IO130:P50," &
"IO131:P51," &
"IO137:P56," &
"IO138:P57," &
"IO141:P58," &
"IO152:P59," &
"IO153:P60," &
"IO158:P61," &
"IO159:P62," &
"IO162:P65," &
"IO163:P66," &
"IO168:P67," &
"IO169:P68," &
"IO179:P69," &
"IO180:P70," &
"IO183:P71," &
"IO184:P72," &
"IO190:P77," &
"IO191:P78," &
"IO194:P79," &
"IO195:P80," &
"IO205:P81," &
"IO206:P82," &
"IO211:P83," &
"IO212:P84," &
"M0:P32," &
"M1:P30," &
"M2:P34," &
"PROGRAM:P55," &
"TCK:P16," &
"TDI:P15," &
"TDO:P75," &
"TMS:P17," &
"VDD: (P2,P11,P22,P33,P42,P54,P63,P74) ";
--end pin map

attribute TAP_SCAN_IN    of TDI : signal is true;
attribute TAP_SCAN_MODE  of TMS : signal is true;
attribute TAP_SCAN_OUT   of TDO : signal is true;
attribute TAP_SCAN_CLOCK of TCK : signal is (2.0e6, LOW);
-- This is conservative. Real max is expected to be (~5MHz, BOTH).

attribute COMPLIANCE_PATTERNS of XC4010XL_PC84 : entity is
-- INIT is not a compliance enable after configuration. For post-configuration
-- operation un-comment the next line and comment out the corresponding line
-- below.
--      "(PROGRAM) (1)";
--
-- NOTE: If INIT has been high or floating since the later of power-on
--      and the last rising transition of PROGRAM, then the device may
--      be in configuration mode in which case some JTAG instructions
--      will not be available.
--      "(INIT,PROGRAM) (01)";

attribute INSTRUCTION_LENGTH of XC4010XL_PC84 : entity is 3;

attribute INSTRUCTION_OPCODE of XC4010XL_PC84 : entity is
"SAMPLE (001)," & -- Internal capture not valid until after config
"RESERVED (110)," &
"READBACK (100)," & -- Not available during configuration
"CONFIGURE (101)," & -- Not available during configuration
"USER2 (011)," & -- Not available until after configuration
"USER1 (010)," & -- Not available until after configuration
"EXTEST (000)," & -- Not available during configuration
"BYPASS (111)";

attribute INSTRUCTION_CAPTURE of XC4010XL_PC84 : entity is "X01";
-- MSB of instruction capture is low during configuration.

```

```
-- If the device is configured, and a USER instruction is implemented
-- and not private to the FPGA designer, then it should be removed
-- from INSTRUCTION_PRIVATE, and the target register should be defined
-- in REGISTER_ACCESS.
```

```
attribute INSTRUCTION_PRIVATE of XC4010XL_PC84 : entity is
    "USER1," &
    "USER2," &
    "READBACK," &
    "RESERVED," &
    "CONFIGURE";
```

```
attribute REGISTER_ACCESS of XC4010XL_PC84 : entity is
--     "<reg_name>[<length>] (USER1)," &
--     "<reg_name>[<length>] (USER2)," &
    "BYPASS (BYPASS)," &
    "BOUNDARY (SAMPLE,EXTEST)";
```

```
attribute BOUNDARY_LENGTH of XC4010XL_PC84 : entity is 488;
```

```
attribute BOUNDARY_REGISTER of XC4010XL_PC84 : entity is
-- cellnum (type, port, function, safe[, ccell, disval, disrslt])
    " 0 (BC_1, *, internal, X)," &
    " 1 (BC_1, *, internal, X)," &
    " 2 (BC_1, *, controlr, 1)," &
    " 3 (BC_1, IO190, output3, X, 2, 1, PULL1)," &
    " 4 (BC_1, IO190, input, X)," &
    " 5 (BC_1, *, controlr, 1)," &
    " 6 (BC_1, IO191, output3, X, 5, 1, PULL1)," &
    " 7 (BC_1, IO191, input, X)," &
    " 8 (BC_1, *, internal, 1)," & -- IO192.T
    " 9 (BC_1, *, internal, X)," & -- IO192.O
    " 10 (BC_1, *, internal, X)," & -- IO192.I
    " 11 (BC_1, *, internal, 1)," & -- IO193.T
    " 12 (BC_1, *, internal, X)," & -- IO193.O
    " 13 (BC_1, *, internal, X)," & -- IO193.I
    " 14 (BC_1, *, controlr, 1)," &
    " 15 (BC_1, IO194, output3, X, 14, 1, PULL1)," &
    " 16 (BC_1, IO194, input, X)," &
    " 17 (BC_1, *, controlr, 1)," &
    " 18 (BC_1, IO195, output3, X, 17, 1, PULL1)," &
    " 19 (BC_1, IO195, input, X)," &
    " 20 (BC_1, *, internal, 1)," & -- IO195.T
    " 21 (BC_1, *, internal, X)," & -- IO195.O
    " 22 (BC_1, *, internal, X)," & -- IO195.I
    " 23 (BC_1, *, internal, 1)," & -- IO196.T
    " 24 (BC_1, *, internal, X)," & -- IO196.O
    " 25 (BC_1, *, internal, X)," & -- IO196.I
    " 26 (BC_1, *, internal, 1)," & -- IO197.T
    " 27 (BC_1, *, internal, X)," & -- IO197.O
    " 28 (BC_1, *, internal, X)," & -- IO197.I
    " 29 (BC_1, *, internal, 1)," & -- IO198.T
    " 30 (BC_1, *, internal, X)," & -- IO198.O
    " 31 (BC_1, *, internal, X)," & -- IO198.I
    " 32 (BC_1, *, internal, 1)," & -- IO201.T
    " 33 (BC_1, *, internal, X)," & -- IO201.O
    " 34 (BC_1, *, internal, X)," & -- IO201.I
    " 35 (BC_1, *, internal, 1)," & -- IO202.T
    " 36 (BC_1, *, internal, X)," & -- IO202.O
    " 37 (BC_1, *, internal, X)," & -- IO202.I
    " 38 (BC_1, *, controlr, 1)," &
    " 39 (BC_1, IO205, output3, X, 38, 1, PULL1)," &
    " 40 (BC_1, IO205, input, X)," &
    " 41 (BC_1, *, controlr, 1)," &
    " 42 (BC_1, IO206, output3, X, 41, 1, PULL1)," &
    " 43 (BC_1, IO206, input, X)," &
    " 44 (BC_1, *, internal, 1)," & -- IO206.T
    " 45 (BC_1, *, internal, X)," & -- IO206.O
    " 46 (BC_1, *, internal, X)," & -- IO206.I
    " 47 (BC_1, *, internal, 1)," & -- IO207.T
    " 48 (BC_1, *, internal, X)," & -- IO207.O
    " 49 (BC_1, *, internal, X)," & -- IO207.I
    " 50 (BC_1, *, internal, 1)," & -- IO208.T
    " 51 (BC_1, *, internal, X)," & -- IO208.O
    " 52 (BC_1, *, internal, X)," & -- IO208.I
    " 53 (BC_1, *, internal, 1)," & -- IO29.T
    " 54 (BC_1, *, internal, X)," & -- IO29.O
    " 55 (BC_1, *, internal, X)," & -- IO29.I
    " 56 (BC_1, *, controlr, 1)," &
```



```

" 57 (BC_1, IO211, output3, X, 56, 1, PULL1)," &
" 58 (BC_1, IO211, input, X)," &
" 59 (BC_1, *, controlr, 1)," &
" 60 (BC_1, IO212, output3, X, 59, 1, PULL1)," &
" 61 (BC_1, IO212, input, X)," &
" 62 (BC_1, *, controlr, 1)," &
" 63 (BC_1, IO2, output3, X, 62, 1, PULL1)," &
" 64 (BC_1, IO2, input, X)," &
" 65 (BC_1, *, controlr, 1)," &
" 66 (BC_1, IO3, output3, X, 65, 1, PULL1)," &
" 67 (BC_1, IO3, input, X)," &
" 68 (BC_1, *, internal, 1)," & -- IO4.T
" 69 (BC_1, *, internal, X)," & -- IO4.O
" 70 (BC_1, *, internal, X)," & -- IO4.I
" 71 (BC_1, *, internal, 1)," & -- IO5.T
" 72 (BC_1, *, internal, X)," & -- IO5.O
" 73 (BC_1, *, internal, X)," & -- IO5.I
" 74 (BC_1, *, internal, 1)," & -- IO5.T
" 75 (BC_1, *, internal, X)," & -- IO5.O
" 76 (BC_1, *, internal, X)," & -- IO5.I
" 77 (BC_1, *, internal, 1)," & -- IO6.T
" 78 (BC_1, *, internal, X)," & -- IO6.O
" 79 (BC_1, *, internal, X)," & -- IO6.I
" 80 (BC_1, *, controlr, 1)," &
" 81 (BC_1, IO8, output3, X, 80, 1, PULL1)," &
" 82 (BC_1, IO8, input, X)," &
" 83 (BC_1, *, controlr, 1)," &
" 84 (BC_1, IO9, output3, X, 83, 1, PULL1)," &
" 85 (BC_1, IO9, input, X)," &
" 86 (BC_1, *, internal, 1)," & -- IO10.1.T
" 87 (BC_1, *, internal, X)," & -- IO10.1.O
" 88 (BC_1, *, internal, X)," & -- IO10.1.I
" 89 (BC_1, *, internal, 1)," & -- IO13.T
" 90 (BC_1, *, internal, X)," & -- IO13.O
" 91 (BC_1, *, internal, X)," & -- IO13.I
" 92 (BC_1, *, internal, 1)," & -- IO15.T
" 93 (BC_1, *, internal, X)," & -- IO15.O
" 94 (BC_1, *, internal, X)," & -- IO15.I
" 95 (BC_1, *, internal, 1)," & -- IO15.T
" 96 (BC_1, *, internal, X)," & -- IO15.O
" 97 (BC_1, *, internal, X)," & -- IO15.I
" 98 (BC_1, *, internal, 1)," & -- IO16.T
" 99 (BC_1, *, internal, X)," & -- IO16.O
" 100 (BC_1, *, internal, X)," & -- IO16.I
" 101 (BC_1, *, internal, 1)," & -- IO11.T
" 102 (BC_1, *, internal, X)," & -- IO11.O
" 103 (BC_1, *, internal, X)," & -- IO11.I
" 104 (BC_1, *, controlr, 1)," &
" 105 (BC_1, IO19, output3, X, 104, 1, PULL1)," &
" 106 (BC_1, IO19, input, X)," &
" 107 (BC_1, *, controlr, 1)," &
" 108 (BC_1, IO20, output3, X, 107, 1, PULL1)," &
" 109 (BC_1, IO20, input, X)," &
" 110 (BC_1, *, internal, 1)," & -- IO21.T
" 111 (BC_1, *, internal, X)," & -- IO21.O
" 112 (BC_1, *, internal, X)," & -- IO21.I
" 113 (BC_1, *, internal, 1)," & -- IO22.T
" 114 (BC_1, *, internal, X)," & -- IO22.O
" 115 (BC_1, *, internal, X)," & -- IO22.I
" 116 (BC_1, *, controlr, 1)," &
" 117 (BC_1, IO23, output3, X, 116, 1, PULL1)," &
" 118 (BC_1, IO23, input, X)," &
" 119 (BC_1, *, controlr, 1)," &
" 120 (BC_1, IO24, output3, X, 119, 1, PULL1)," &
" 121 (BC_1, IO24, input, X)," &
" 122 (BC_1, *, controlr, 1)," &
" 123 (BC_1, IO28, output3, X, 122, 1, PULL1)," &
" 124 (BC_1, IO28, input, X)," &
" 125 (BC_1, *, controlr, 1)," &
" 126 (BC_1, IO29, output3, X, 125, 1, PULL1)," &
" 127 (BC_1, IO29, input, X)," &
" 128 (BC_1, *, internal, 1)," & -- IO30.T
" 129 (BC_1, *, internal, X)," & -- IO30.O
" 130 (BC_1, *, internal, X)," & -- IO30.I
" 131 (BC_1, *, internal, 1)," & -- IO31.T
" 132 (BC_1, *, internal, X)," & -- IO31.O
" 133 (BC_1, *, internal, X)," & -- IO31.I
" 134 (BC_1, *, internal, X)," &
" 135 (BC_1, *, internal, X)," &
" 136 (BC_1, *, internal, X)," &

```

```

" 137 (BC_1, *, internal, X)," &
" 138 (BC_1, *, internal, X)," &
" 139 (BC_1, *, internal, X)," &
" 140 (BC_1, *, internal, 1)," & -- IO34.T
" 141 (BC_1, *, internal, X)," & -- IO34.O
" 142 (BC_1, *, internal, X)," & -- IO34.I
" 143 (BC_1, *, internal, 1)," & -- IO35.T
" 144 (BC_1, *, internal, X)," & -- IO35.O
" 145 (BC_1, *, internal, X)," & -- IO35.I
" 146 (BC_1, *, internal, 1)," & -- IO35.T
" 147 (BC_1, *, internal, X)," & -- IO35.O
" 148 (BC_1, *, internal, X)," & -- IO35.I
" 149 (BC_1, *, internal, 1)," & -- IO36.T
" 150 (BC_1, *, internal, X)," & -- IO36.O
" 151 (BC_1, *, internal, X)," & -- IO36.I
" 152 (BC_1, *, internal, 1)," & -- IO38.T
" 153 (BC_1, *, internal, X)," & -- IO38.O
" 154 (BC_1, *, internal, X)," & -- IO38.I
" 155 (BC_1, *, internal, 1)," & -- IO40.T
" 156 (BC_1, *, internal, X)," & -- IO40.O
" 157 (BC_1, *, internal, X)," & -- IO40.I
" 158 (BC_1, *, internal, X)," &
" 159 (BC_1, *, internal, X)," &
" 160 (BC_1, *, internal, X)," &
" 161 (BC_1, *, controlr, 1)," &
" 162 (BC_1, IO42, output3, X, 161, 1, PULL1)," &
" 163 (BC_1, IO42, input, X)," &
" 164 (BC_1, *, internal, 1)," & -- IO45.T
" 165 (BC_1, *, internal, X)," & -- IO45.O
" 166 (BC_1, *, internal, X)," & -- IO45.I
" 167 (BC_1, *, internal, 1)," & -- IO45.T
" 168 (BC_1, *, internal, X)," & -- IO45.O
" 169 (BC_1, *, internal, X)," & -- IO45.I
" 170 (BC_1, *, internal, 1)," & -- IO46.T
" 171 (BC_1, *, internal, X)," & -- IO46.O
" 172 (BC_1, *, internal, X)," & -- IO46.I
" 173 (BC_1, *, internal, 1)," & -- IO47.T
" 174 (BC_1, *, internal, X)," & -- IO47.O
" 175 (BC_1, *, internal, X)," & -- IO47.I
" 176 (BC_1, *, controlr, 1)," &
" 177 (BC_1, IO49, output3, X, 176, 1, PULL1)," &
" 178 (BC_1, IO49, input, X)," &
" 179 (BC_1, *, controlr, 1)," &
" 180 (BC_1, IO51, output3, X, 179, 1, PULL1)," &
" 181 (BC_1, IO51, input, X)," &
" 182 (BC_1, *, controlr, 1)," &
" 183 (BC_1, IO55, output3, X, 182, 1, PULL1)," &
" 184 (BC_1, IO55, input, X)," &
" 185 (BC_1, *, controlr, 1)," &
" 186 (BC_1, IO56, output3, X, 185, 1, PULL1)," &
" 187 (BC_1, IO56, input, X)," &
" 188 (BC_1, *, internal, 1)," & -- IO56.T
" 189 (BC_1, *, internal, X)," & -- IO56.O
" 190 (BC_1, *, internal, X)," & -- IO56.I
" 191 (BC_1, *, internal, 1)," & -- IO57.T
" 192 (BC_1, *, internal, X)," & -- IO57.O
" 193 (BC_1, *, internal, X)," & -- IO57.I
" 194 (BC_1, *, internal, 1)," & -- IO58.T
" 195 (BC_1, *, internal, X)," & -- IO58.O
" 196 (BC_1, *, internal, X)," & -- IO58.I
" 197 (BC_1, *, internal, 1)," & -- IO60.T
" 198 (BC_1, *, internal, X)," & -- IO60.O
" 199 (BC_1, *, internal, X)," & -- IO60.I
" 200 (BC_1, *, controlr, 1)," &
" 201 (BC_1, IO63, output3, X, 200, 1, PULL1)," &
" 202 (BC_1, IO63, input, X)," &
" 203 (BC_1, *, controlr, 1)," &
" 204 (BC_1, IO64, output3, X, 203, 1, PULL1)," &
" 205 (BC_1, IO64, input, X)," &
" 206 (BC_1, *, internal, 1)," & -- IO65.T
" 207 (BC_1, *, internal, X)," & -- IO65.O
" 208 (BC_1, *, internal, X)," & -- IO65.I
" 209 (BC_1, *, internal, 1)," & -- IO65.T
" 210 (BC_1, *, internal, X)," & -- IO65.O
" 211 (BC_1, *, internal, X)," & -- IO65.I
" 212 (BC_1, *, internal, 1)," & -- IO67.T
" 213 (BC_1, *, internal, X)," & -- IO67.O
" 214 (BC_1, *, internal, X)," & -- IO67.I
" 215 (BC_1, *, internal, 1)," & -- IO68.T
" 216 (BC_1, *, internal, X)," & -- IO68.O

```

```

" 217 (BC_1, *, internal, X)," & -- IO68.I
" 218 (BC_1, *, internal, 1)," & -- IO70.T
" 219 (BC_1, *, internal, X)," & -- IO70.O
" 220 (BC_1, *, internal, X)," & -- IO70.I
" 221 (BC_1, *, internal, 1)," & -- IO71.T
" 222 (BC_1, *, internal, X)," & -- IO71.O
" 223 (BC_1, *, internal, X)," & -- IO71.I
" 224 (BC_1, *, controlr, 1)," &
" 225 (BC_1, IO72, output3, X, 224, 1, PULL1)," &
" 226 (BC_1, IO72, input, X)," &
" 227 (BC_1, *, internal, 1)," & -- IO73.T
" 228 (BC_1, *, internal, X)," & -- IO73.O
" 229 (BC_1, *, internal, X)," & -- IO73.I
" 230 (BC_1, *, internal, 1)," & -- IO74.T
" 231 (BC_1, *, internal, X)," & -- IO74.O
" 232 (BC_1, *, internal, X)," & -- IO74.I
" 233 (BC_1, *, internal, 1)," & -- IO75.T
" 234 (BC_1, *, internal, X)," & -- IO75.O
" 235 (BC_1, *, internal, X)," & -- IO75.I
" 236 (BC_1, *, controlr, 1)," &
" 237 (BC_1, IO76, output3, X, 236, 1, PULL1)," &
" 238 (BC_1, IO76, input, X)," &
" 239 (BC_1, *, controlr, 1)," &
" 240 (BC_1, IO77, output3, X, 239, 1, PULL1)," &
" 241 (BC_1, IO77, input, X)," &
" 242 (BC_1, *, controlr, 1)," &
" 243 (BC_1, M1, output3, X, 242, 1, PULL1)," &
" 244 (BC_1, M1, input, X)," &
" 245 (BC_1, M0, input, X)," &
" 246 (BC_1, M2, input, X)," &
" 247 (BC_1, *, controlr, 1)," &
" 248 (BC_1, IO84, output3, X, 247, 1, PULL1)," &
" 249 (BC_1, IO84, input, X)," &
" 250 (BC_1, *, controlr, 1)," &
" 251 (BC_1, IO85, output3, X, 250, 1, PULL1)," &
" 252 (BC_1, IO85, input, X)," &
" 253 (BC_1, *, internal, 1)," & -- IO85.T
" 254 (BC_1, *, internal, X)," & -- IO85.O
" 255 (BC_1, *, internal, X)," & -- IO85.I
" 256 (BC_1, *, internal, 1)," & -- IO86.T
" 257 (BC_1, *, internal, X)," & -- IO86.O
" 258 (BC_1, *, internal, X)," & -- IO86.I
" 259 (BC_1, *, internal, 1)," & -- IO87.T
" 260 (BC_1, *, internal, X)," & -- IO87.O
" 261 (BC_1, *, internal, X)," & -- IO87.I
" 262 (BC_1, *, controlr, 1)," &
" 263 (BC_1, IO89, output3, X, 262, 1, PULL1)," &
" 264 (BC_1, IO89, input, X)," &
" 265 (BC_1, *, internal, 1)," & -- IO90.T
" 266 (BC_1, *, internal, X)," & -- IO90.O
" 267 (BC_1, *, internal, X)," & -- IO90.I
" 268 (BC_1, *, internal, 1)," & -- IO91.T
" 269 (BC_1, *, internal, X)," & -- IO91.O
" 270 (BC_1, *, internal, X)," & -- IO91.I
" 271 (BC_1, *, internal, 1)," & -- IO92.T
" 272 (BC_1, *, internal, X)," & -- IO92.O
" 273 (BC_1, *, internal, X)," & -- IO92.I
" 274 (BC_1, *, internal, 1)," & -- IO93.T
" 275 (BC_1, *, internal, X)," & -- IO93.O
" 276 (BC_1, *, internal, X)," & -- IO93.I
" 277 (BC_1, *, internal, 1)," & -- IO95.T
" 278 (BC_1, *, internal, X)," & -- IO95.O
" 279 (BC_1, *, internal, X)," & -- IO95.I
" 280 (BC_1, *, internal, 1)," & -- IO95.T
" 281 (BC_1, *, internal, X)," & -- IO95.O
" 282 (BC_1, *, internal, X)," & -- IO95.I
" 283 (BC_1, *, controlr, 1)," &
" 284 (BC_1, IO97, output3, X, 283, 1, PULL1)," &
" 285 (BC_1, IO97, input, X)," &
" 286 (BC_1, *, controlr, 1)," &
" 287 (BC_1, IO98, output3, X, 286, 1, PULL1)," &
" 288 (BC_1, IO98, input, X)," &
" 289 (BC_1, *, internal, 1)," & -- IO101.T
" 290 (BC_1, *, internal, X)," & -- IO101.O
" 291 (BC_1, *, internal, X)," & -- IO101.I
" 292 (BC_1, *, internal, 1)," & -- IO102.T
" 293 (BC_1, *, internal, X)," & -- IO102.O
" 294 (BC_1, *, internal, X)," & -- IO102.I
" 295 (BC_1, *, internal, 1)," & -- IO103.T
" 296 (BC_1, *, internal, X)," & -- IO103.O

```

```

" 297 (BC_1, *, internal, X)," & -- IO103.I
" 298 (BC_1, *, internal, 1)," & -- IO104.T
" 299 (BC_1, *, internal, X)," & -- IO104.O
" 300 (BC_1, *, internal, X)," & -- IO104.I
" 301 (BC_1, *, controlr, 1)," &
" 302 (BC_1, IO105, output3, X, 301, 1, PULL1)," &
" 303 (BC_1, IO105, input, X)," &
-- INIT is not a compliance enable after configuration. For post-configuration
-- operation un-comment the next line and comment out the following line.
-- Repeat for registers 304 through 305.
-- " 304 (BC_1, *, controlr, 1)," &
-- " 304 (BC_1, *, internal, 1)," &
-- " 305 (BC_1, INIT, output3, X, 304, 1, PULL1)," &
-- " 305 (BC_1, *, internal, 0)," &
-- " 306 (BC_1, INIT, input, X)," &
-- " 306 (BC_1, *, internal, X)," &
" 307 (BC_1, *, controlr, 1)," &
" 308 (BC_1, IO109, output3, X, 307, 1, PULL1)," &
" 309 (BC_1, IO109, input, X)," &
" 310 (BC_1, *, controlr, 1)," &
" 311 (BC_1, IO110, output3, X, 310, 1, PULL1)," &
" 312 (BC_1, IO110, input, X)," &
" 313 (BC_1, *, internal, 1)," & -- IO110.T
" 314 (BC_1, *, internal, X)," & -- IO110.O
" 315 (BC_1, *, internal, X)," & -- IO110.I
" 316 (BC_1, *, internal, 1)," & -- IO110.1.T
" 317 (BC_1, *, internal, X)," & -- IO110.1.O
" 318 (BC_1, *, internal, X)," & -- IO110.1.I
" 319 (BC_1, *, internal, 1)," & -- IO113.T
" 320 (BC_1, *, internal, X)," & -- IO113.O
" 321 (BC_1, *, internal, X)," & -- IO113.I
" 322 (BC_1, *, internal, 1)," & -- IO114.T
" 323 (BC_1, *, internal, X)," & -- IO114.O
" 324 (BC_1, *, internal, X)," & -- IO114.I
" 325 (BC_1, *, controlr, 1)," &
" 326 (BC_1, IO117, output3, X, 325, 1, PULL1)," &
" 327 (BC_1, IO117, input, X)," &
" 328 (BC_1, *, controlr, 1)," &
" 329 (BC_1, IO118, output3, X, 328, 1, PULL1)," &
" 330 (BC_1, IO118, input, X)," &
" 331 (BC_1, *, internal, 1)," & -- IO118.T
" 332 (BC_1, *, internal, X)," & -- IO118.O
" 333 (BC_1, *, internal, X)," & -- IO118.I
" 334 (BC_1, *, internal, 1)," & -- IO115.T
" 335 (BC_1, *, internal, X)," & -- IO115.O
" 336 (BC_1, *, internal, X)," & -- IO115.I
" 337 (BC_1, *, internal, 1)," & -- IO122.T
" 338 (BC_1, *, internal, X)," & -- IO122.O
" 339 (BC_1, *, internal, X)," & -- IO122.I
" 340 (BC_1, *, internal, 1)," & -- IO123.T
" 341 (BC_1, *, internal, X)," & -- IO123.O
" 342 (BC_1, *, internal, X)," & -- IO123.I
" 343 (BC_1, *, internal, 1)," & -- IO124.T
" 344 (BC_1, *, internal, X)," & -- IO124.O
" 345 (BC_1, *, internal, X)," & -- IO124.I
" 346 (BC_1, *, internal, 1)," & -- IO112.T
" 347 (BC_1, *, internal, X)," & -- IO112.O
" 348 (BC_1, *, internal, X)," & -- IO112.I
" 349 (BC_1, *, controlr, 1)," &
" 350 (BC_1, IO126, output3, X, 349, 1, PULL1)," &
" 351 (BC_1, IO126, input, X)," &
" 352 (BC_1, *, controlr, 1)," &
" 353 (BC_1, IO127, output3, X, 352, 1, PULL1)," &
" 354 (BC_1, IO127, input, X)," &
" 355 (BC_1, *, internal, 1)," & -- IO114.T
" 356 (BC_1, *, internal, X)," & -- IO114.O
" 357 (BC_1, *, internal, X)," & -- IO114.I
" 358 (BC_1, *, internal, 1)," & -- IO128.T
" 359 (BC_1, *, internal, X)," & -- IO128.O
" 360 (BC_1, *, internal, X)," & -- IO128.I
" 361 (BC_1, *, controlr, 1)," &
" 362 (BC_1, IO130, output3, X, 361, 1, PULL1)," &
" 363 (BC_1, IO130, input, X)," &
" 364 (BC_1, *, controlr, 1)," &
" 365 (BC_1, IO131, output3, X, 364, 1, PULL1)," &
" 366 (BC_1, IO131, input, X)," &
" 367 (BC_1, *, controlr, 1)," &
" 368 (BC_1, IO137, output3, X, 367, 1, PULL1)," &
" 369 (BC_1, IO137, input, X)," &
" 370 (BC_1, *, controlr, 1)," &

```

```

" 371 (BC_1, IO138, output3, X, 370, 1, PULL1)," &
" 372 (BC_1, IO138, input, X)," &
" 373 (BC_1, *, internal, 1)," & -- IO138.T
" 374 (BC_1, *, internal, X)," & -- IO138.O
" 375 (BC_1, *, internal, X)," & -- IO138.I
" 376 (BC_1, *, internal, 1)," & -- IO140.T
" 377 (BC_1, *, internal, X)," & -- IO140.O
" 378 (BC_1, *, internal, X)," & -- IO140.I
" 379 (BC_1, *, controlr, 1)," &
" 380 (BC_1, IO141, output3, X, 379, 1, PULL1)," &
" 381 (BC_1, IO141, input, X)," &
" 382 (BC_1, *, internal, 1)," & -- IO142.T
" 383 (BC_1, *, internal, X)," & -- IO142.O
" 384 (BC_1, *, internal, X)," & -- IO142.I
" 385 (BC_1, *, internal, 1)," & -- IO143.T
" 386 (BC_1, *, internal, X)," & -- IO143.O
" 387 (BC_1, *, internal, X)," & -- IO143.I
" 388 (BC_1, *, internal, 1)," & -- IO144.T
" 389 (BC_1, *, internal, X)," & -- IO144.O
" 390 (BC_1, *, internal, X)," & -- IO144.I
" 391 (BC_1, *, internal, 1)," & -- IO145.T
" 392 (BC_1, *, internal, X)," & -- IO145.O
" 393 (BC_1, *, internal, X)," & -- IO145.I
" 394 (BC_1, *, internal, 1)," & -- IO145.T
" 395 (BC_1, *, internal, X)," & -- IO145.O
" 396 (BC_1, *, internal, X)," & -- IO145.I
" 397 (BC_1, *, internal, 1)," & -- IO147.T
" 398 (BC_1, *, internal, X)," & -- IO147.O
" 399 (BC_1, *, internal, X)," & -- IO147.I
" 400 (BC_1, *, internal, 1)," & -- IO148.T
" 401 (BC_1, *, internal, X)," & -- IO148.O
" 402 (BC_1, *, internal, X)," & -- IO148.I
" 403 (BC_1, *, controlr, 1)," &
" 404 (BC_1, IO152, output3, X, 403, 1, PULL1)," &
" 405 (BC_1, IO152, input, X)," &
" 406 (BC_1, *, controlr, 1)," &
" 407 (BC_1, IO153, output3, X, 406, 1, PULL1)," &
" 408 (BC_1, IO153, input, X)," &
" 409 (BC_1, *, internal, 1)," & -- IO154.T
" 410 (BC_1, *, internal, X)," & -- IO154.O
" 411 (BC_1, *, internal, X)," & -- IO154.I
" 412 (BC_1, *, internal, 1)," & -- IO155.T
" 413 (BC_1, *, internal, X)," & -- IO155.O
" 414 (BC_1, *, internal, X)," & -- IO155.I
" 415 (BC_1, *, internal, 1)," & -- IO155.T
" 416 (BC_1, *, internal, X)," & -- IO155.O
" 417 (BC_1, *, internal, X)," & -- IO155.I
" 418 (BC_1, *, internal, 1)," & -- IO156.T
" 419 (BC_1, *, internal, X)," & -- IO156.O
" 420 (BC_1, *, internal, X)," & -- IO156.I
" 421 (BC_1, *, controlr, 1)," &
" 422 (BC_1, IO158, output3, X, 421, 1, PULL1)," &
" 423 (BC_1, IO158, input, X)," &
" 424 (BC_1, *, controlr, 1)," &
" 425 (BC_1, IO159, output3, X, 424, 1, PULL1)," &
" 426 (BC_1, IO159, input, X)," &
" 427 (BC_1, *, controlr, 1)," &
" 428 (BC_1, IO162, output3, X, 427, 1, PULL1)," &
" 429 (BC_1, IO162, input, X)," &
" 430 (BC_1, *, controlr, 1)," &
" 431 (BC_1, IO163, output3, X, 430, 1, PULL1)," &
" 432 (BC_1, IO163, input, X)," &
" 433 (BC_1, *, internal, 1)," & -- IO164.T
" 434 (BC_1, *, internal, X)," & -- IO164.O
" 435 (BC_1, *, internal, X)," & -- IO164.I
" 436 (BC_1, *, internal, 1)," & -- IO165.T
" 437 (BC_1, *, internal, X)," & -- IO165.O
" 438 (BC_1, *, internal, X)," & -- IO165.I
" 439 (BC_1, *, internal, 1)," & -- IO165.T
" 440 (BC_1, *, internal, X)," & -- IO165.O
" 441 (BC_1, *, internal, X)," & -- IO165.I
" 442 (BC_1, *, internal, 1)," & -- IO166.T
" 443 (BC_1, *, internal, X)," & -- IO166.O
" 444 (BC_1, *, internal, X)," & -- IO166.I
" 445 (BC_1, *, controlr, 1)," &
" 446 (BC_1, IO168, output3, X, 445, 1, PULL1)," &
" 447 (BC_1, IO168, input, X)," &
" 448 (BC_1, *, controlr, 1)," &
" 449 (BC_1, IO169, output3, X, 448, 1, PULL1)," &
" 450 (BC_1, IO169, input, X)," &

```

```

" 451 (BC_1, *, internal, 1)," & -- IO172.T
" 452 (BC_1, *, internal, X)," & -- IO172.O
" 453 (BC_1, *, internal, X)," & -- IO172.I
" 454 (BC_1, *, internal, 1)," & -- IO173.T
" 455 (BC_1, *, internal, X)," & -- IO173.O
" 456 (BC_1, *, internal, X)," & -- IO173.I
" 457 (BC_1, *, internal, 1)," & -- IO175.T
" 458 (BC_1, *, internal, X)," & -- IO175.O
" 459 (BC_1, *, internal, X)," & -- IO175.I
" 460 (BC_1, *, internal, 1)," & -- IO175.T
" 461 (BC_1, *, internal, X)," & -- IO175.O
" 462 (BC_1, *, internal, X)," & -- IO175.I
" 463 (BC_1, *, internal, 1)," & -- IO176.T
" 464 (BC_1, *, internal, X)," & -- IO176.O
" 465 (BC_1, *, internal, X)," & -- IO176.I
" 466 (BC_1, *, internal, 1)," & -- IO177.T
" 467 (BC_1, *, internal, X)," & -- IO177.O
" 468 (BC_1, *, internal, X)," & -- IO177.I
" 469 (BC_1, *, controlr, 1)," &
" 470 (BC_1, IO179, output3, X, 469, 1, PULL1)," &
" 471 (BC_1, IO179, input, X)," &
" 472 (BC_1, *, controlr, 1)," &
" 473 (BC_1, IO180, output3, X, 472, 1, PULL1)," &
" 474 (BC_1, IO180, input, X)," &
" 475 (BC_1, *, internal, 1)," & -- IO181.T
" 476 (BC_1, *, internal, X)," & -- IO181.O
" 477 (BC_1, *, internal, X)," & -- IO181.I
" 478 (BC_1, *, internal, 1)," & -- IO182.T
" 479 (BC_1, *, internal, X)," & -- IO182.O
" 480 (BC_1, *, internal, X)," & -- IO182.I
" 481 (BC_1, *, controlr, 1)," &
" 482 (BC_1, IO183, output3, X, 481, 1, PULL1)," &
" 483 (BC_1, IO183, input, X)," &
" 484 (BC_1, *, controlr, 1)," &
" 485 (BC_1, IO184, output3, X, 484, 1, PULL1)," &
" 486 (BC_1, IO184, input, X)," &
" 487 (BC_1, *, internal, X);"
--end boundary register

attribute DESIGN_WARNING of XC4010XL_PC84 : entity is
"CCLK and DONE are not represented in BOUNDARY_REGISTER." &
"This BSDL file must be modified by the FPGA designer in order to" &
"reflect post-configuration behavior (if any)." &
"If INIT has been high or floating since power-on or the last" &
"rising edge of PROGRAM, then the device may be in" &
"configuration mode in which case this file is not valid." &
"The output and tristate capture values are not valid until after" &
"the device is configured." &
"The fast output mux (where used) is not captured properly." &
"The tristate control is not captured properly when GTS is activated." &
"Some pins have both controlled and uncontrolled input paths.";

end XC4010XL_PC84;

```