informa

OŚRODEK BADAWCZO-ROZWOJOWY INFORMATYKI

ZASTOSOWANIE KODOWANIA DO KONTROLI BŁĘDÓW W TRANSMISJI DANYCH ORAZ SPRZĘCIE

Europejski Program Badawczy Diebolda

42

Warszawa 1973

OŚRODEK BADAWCZO-ROZWOJOWY INFORMATYKI

ZASTOSOWANIE KODOWANIA DO KONTROLI BŁĘDÓW W TRANSMISJI DANYCH ORAZ SPRZĘCIE

Europejski Program Badawczy Diebolda

Wyłącznie do użytku na terenie PRL

42

Warszawa 1973

Tytuł oryginału: ERROR CONTROL CODING IN COMMUNICATIONS
AND HARDWARE
Document No E-76, August 1970

Tłumaczenie: K. Mazanek
Opiniodawca: M. Frydrychewicz
Redakcja: A. Idźkiewicz

Komitet Redakcyjny:

Mieczysław Gula, Franciszek Haratym, Andrzej Idźkiewicz, Stanisław Nelken /zastępca przewodniczącego/, Janina Jerzykowska /sekretarz/, Jerzy Kisielnicki, Krzysztof Skulski, Zdzisław Zapolski /przewodniczący/

Wydawca:

Działowy Ośrodek Informacji, Warszawa, ul. Marszałkowska 104/122

OBRI. Warszawa 1973 r. Nakład 850 + 162 egz. Objętość; ark. wydawn. 3. Ark.druk. 7. Papier offsetowy kl.III.80 g 61 x 86

Druk DOI/P. Zam nr 155/73 R-5 Cena zł 92.-

SPIS TREŚCI

I. DOTYCHCZASOWE OSIĄGNIĘCIA

A. WPROWADZENIE

Urządzenia użytkowane przy przetwarzaniu danych nie są niezawodne. Przyczynami zawodności są błędy powstałe w okresie projektowania, niedoskonałość podzespołów, szumy występujące w kanałach komunikacyjnych itp. czynniki. Niekiedy nawet przepalenie bezpiecznika może spowodować całkowite unieruchomienie urządzeń. Przyczyną większości zaburzeń w ich pracy są czynniki przejściowe: niekontaktowanie połączeń i styków, zwarcia, chwilowe szumy w kanałach, których istnienie nie przerywa pracy całego systemu.

Zakłócenia przejściowe występują zazwyczaj niezmiernie rzadko i mogą nie być przyczyną istotnych zniekształceń. Zniekształcenia nie są jednak dopuszczalne w systemach zbierania i transmitowania danych z odległych terminali w rozległych systemach komputerowych, w zastosowaniach militarnych i kosmicznych - czyli tam, gdzie wymagana jest bezbłędność pracy systemu w czasie długotrwałej eksploatacji.

Drogą do uniknięcia niektórych ze wspomianych zaburzeń jest usprawnienie pracy sprzętu, jego żywotności i wydajności; większość badań obecnie prowadzonych zdąża właśnie w tym kierunku. Jeśli chodzi jednak o badania w zakresie kontroli błędów, to rozwija się praktyczne zastosowanie teorii redundancji, tzn. rozwoju techniki dublowania, pozwalającej na prawidłową pracę urządzeń, mimo chwilowych zakłóceń.

Niniejsze opracowanie zajmuje się głównie technikami nadmiarowymi z uwzględnieniem zagadnień kodowania i technik nadmiarowych wypróbowanych i sprawdzonych w transmisji, przechowywaniu i przetwarzaniu danych w procesorze. Ponadto, dla czytelników, którzy z tymi zagadnieniami nie mieli do czynienia, podano skrócone opisy najbardziej znanych technik nadmiarowych.

B. TECHNIKI NADMIAROWE

Stosowanie technik nadmiarowych wymaga użycia dodatkowego wyposażenia lub dodatkowych danych. Techniki te mogą być wykorzystane przy poszczególnych urządzeniach, obwodach lub w całym systemie, przy czym mogą one wykorzystywać podobną lub odmienną strukturę niż system. Ich rola może sprowadzać się do dublowania obwodów logicznych lub do specyficznego przetwarzania danych.

Bez względu na sposób stosowania, zadaniem technik nadmiarowych jest wykrywanie błędów w pracy systemu i stworzenie możliwości korekty jak największej ilości błędów. Uzyskuje się to przez powtórzenie cyklu pracy urządzeń wstrzymanych chwilowymi lub trwałymi zakłóceniami, przez samo-naprawę lub zmianę konfiguracji urządzeń, przez duplikowania funkcji niektórych urządzeń i przez wewnętrzną korektę błędów.

1. Logika decydowania większościowego

Pierwsze prace z zakresu teorii nadmiaru przeprowadził John von Neumann. Jego system zwany logiką większościową zawiera 2n +1 podobnych lub różnych układów logicznych wykonujących identyczne operacje przetwarzania.

Układ decyzyjny oddziaływuje na wszystkie sygnały wyjściowe układów logicznych, ustalając optymalną ze względu na prawdopodobieństwo wystąpienia błędów, postać sygnałów wyjściowych. Istnieje wiele metod decydowania większościowego różniących się kryteriami szacowania.

Na przykład system głosowania większościowego przyjmuje za poprawny sygnał identyczny w większości układów logicznych. Poprawna decyzja będzie podjęta zawsze, gdy co najmniej n + 1 układów logicznych pracuje bezbłędnie, przy założeniu poprawnego działania układu głosowania.

Dla kontroli poprawnosci procesu głosowania można przewidzieć wielokrotne powtarzanie procesu podejmowania decyzji - równocześnie lub sekwencyjnie.

Ponieważ poszczególne układy logiczne różnią się od siebie, różny jest również ich stopień niezawodności. Zatem działanie jednych układów może być mniej zawodne od innych. Sugeruje to inne podejście do zagadnienia, polegające na przypisywaniu wagi głosowi każdego układu, zależnie od stopnia niezawodności tego układu.

Istnieje możliwość oceny niezawodności poszczególnych układów przez porównanie sygnałów wychodzących z każdego układu logicznego z sygnałami odebranymi z elementu decyzyjnego. Ustalić można wówczas optymalną procedurę głosowania, w której stopień ważności poszczególnych głosów jest odpowiednio liczony dla każdego układu. Systemowi takiemu - zwanemu przystosowanym systemem głosowania - poświęca się obecnie wiele uwagi.

2. Spleciona logika nadmiaru

Całkowicie odmienną technikę nadmiarową wprowadził J.E. Tryon. W jego systemie wszystkie wielkości transmitowane są czterema przewodami, a wszystkie układy logiczne zastąpione są przez układy z podwójnymi wejściami. Wyjścia są rozdzielane i wprowadzane do dwóch kolejnych układów. W rezultacie takiej techniki uzyskujemy złożony układ, korygujący wszystkie pojedyncze błędy przez wytłumienie wyjścia z wadliwie pracującego układu.

Uogólnienia tego systemu zwane splecioną logiką nadmiaru zyskują ostatnio na znaczeniu. Wspólną zasadą tych systemów jest korekta błędów prowadzona w tych samych układach, w których przeprowadza się operacje logiczne.

3. Kodowanie

Dwie podstawowe techniki nadmiarowe: logika decydowania większościowego i spleciona logika nadmiaru opierają się na powielaniu części lub całego systemu. Stosuje się je dlatego głównie do sprawdzania operacji wewnętrznych procesora. Nie nadają się one jednak do zabezpieczenia bezbłędnego funkcjonowania kanałów transmisji danych oraz końcówek wejścia/wyjścia. Ich wspólną wadą jest konieczność stosowania dodatkowego wyposażenia, którego ilość gwałtownie wzrasta z wymaganym stopniem zabezpieczenia systemu przed błędami. Dodatkowe wyposażenie nie tylko samo z kolei nie jest niezawodne, ale jeszcze zwiększa koszty systemu.

Całkowicie odmienne możliwości stwarza zastosowanie teorii Shannon'a przesyłania informacji w obecności szumów. Praktyczne możliwości stosowania tej teorii można zaobserwować w tzw. twierdzeniu kodowania, które mówi, że jeśli informacja jest prawidłowo przygotowana, może być transmitowana kanałem komunikacyjnym z dowolnie dużą niezawodnością, jedynie pod warunkiem, że szybkość transmisji nie przekroczy pewnej wartości maksymalnej, określonej przepustowością kanału. Wysoką bezbłędność transmisji uzyskuje się przez stosowanie kodowania, które wprowadza nadmiar nie do systemu transmisji, lecz do danych, które mają być przekazywane. Dane są wstępnie przekształcane do postaci nadmiarowej w urządzeniu zwanym urządzeniem kodującym. W czasie transmitowania kanałem, dane mogą być zniekształcone wskutek obecności szumów, jednakże nadmiar jest tak dobrany, że odbierane dane mogą być z wysokim stopniem prawdopodobieństwa bezbłędnie odtworzone przez wyjściowy dekoder. Dodatkowe wyposażenie potrzebne do kodowania i dekodowania jest na ogół skromne; następuje jednak wydłużenie czasu transmisji.

Pozwalając kosztem wydłużenia czasu transmisji osiągnąć znaczne korzyści, kodowanie jest naturalnym sposobem unikania błędów, powstających w takich systemach jak: systemy rezerwacji miejsc lotniczych, z ich znaczną ilością terminali i linii transmisyjnych. Kodowanie ma również wiele zalet w porównaniu z innymi technikami nadmiarowymi przy zapewnianiu prawidłowości działania jednostek arytmetycznych, logicznych i sterowania komputera.

Ponieważ kryteria niezawodności pracy zespołów komputera i kryteria bezbłędności pracy kanału są na ogół różne, kody optymalne i najlepiej dające się zastosować do komputera nie zawsze muszą być optymalne dla kanału. Pewne zasadnicze różnice wynikają stąd, że błędy powstające w obwodach dają inny efekt niż błędy spowodowane szumami w kanale komunikacyjnym, a także z tego względu, że kodery i dekodery również nie są niezawodne. Poza tym stosowane kody muszą być zgodne z normalnymi kodami, które stosuje się w komputerze.

Większość badań związanych z opracowaniem optymalnego i łatwego do zastosowania kodu była związana w przeszłości z aspektami transmisji w sieciach komputerowych. W tej dziedzinie osiągnięto wiele doskonałych wyników.

C. CEL I ZAKRES OPRACOWANIA

W niniejszym opracowaniu omówiono:

- najważniejsze techniki kodowania stosowane dla zabezpieczenia przed błędami,
- w jaki sposób techniki te wpływają na zdolność systemu do funkcjonowania przy pojawieniu się błędów,
- zakres możliwości stosowania i koszty,
- ostatnie osiągnięcia w tej dziedzinie, łącznie z zaleceniami odnośnie stosowanych technik w systemach obecnie pracujących i systemach w przyszłości.

W opracowaniu położono nacisk na te zagadnienia przetwarzania danych, w których stosowanie kodowania może mieć istotny wpływ na szybkość i bezbłędność, a więc kontrolę błędów w teletransmisji, operowaniu danymi oraz przechowywaniu danych.

Opracowanie omawia ogólne podstawy technik kodowania stosowanych w przetwarzaniu danych. Może ono pomóc w oszacowaniu możliwości korekty błędów przez hardware oraz być podstawą modernizacji dotychczasowego wyposażenia. Wreszcie, ma ono pomóc czytelnikowi w precyzowaniu wymagań w zakresie kontroli korekcji błędów przy zakupie nowych systemów.

D. PODSTAWOWE WNIOSKI

1. Kodowanie jest sprawną metodą wykrywania błędów w przekazywaniu, przechowywaniu danych i przetwarzaniu danych w PAO. Ma ono przewagę nad innymi technikami nadmiarowymi w systemach o wymaganej wysokiej bezbłędności.

2. Kody mogą być wprowadzone do protekcji przed błędami pojedynczymi, wielokrotnymi lub występującymi seryjnie. Najprostszymi w zastosowaniu są kody korygujące pojedyncze błędy.

3. Zastosowanie kodów do korekcji pojedynczych błędów wymaga nieznacznych nakładów inwestycyjnych i nieznacznie wydłuża czas pracy komputera.

4. Z punktu widzenia wdrażania i wydajności najlepsze są kody: dla transmisji danych - kod Hamminga i kod BCH, dla przetwarzania danych w procesorze - kody resztowe, dla przechowywania danych - kody z dwuwymiarową kontrolą parzystości.

5. Kodowanie w zasadniczym stopniu obniża ilość błędów w czasie transmisji danych, przy nieznacznym zwiększeniu czasu przetwarzania i złożoności wyposażenia.

II. TERMINOLOGIA I EFEKTYWNOŚĆ KODOWANIA

A. DEFINICJE

Kodowanie może być efektywnym środkiem wykrywania błędów. Może rywalizować z innymi technikami nadmiarowymi, jeśli procedury kodowania i dekodowania wymagają jedynie niewielkiego rozbudowania hardware i jeśli efektem jest znaczne zmniejszenie przestojów systemu. W celu osiągnięcia takiego zmniejszenia ważnym jest, aby sposoby kodowania koncentrowały się na korekcie tych błędów, które występują najczęściej. Ponieważ, na przykład błędy powstające w arytmometrze procesora różnią się diametralnie od tych, które powstają przy transmisji danych przez łącza telefoniczne, oba rodzaje błędów wymagają odmiennego kodowania. Wybór kodu zależy od tego, w jakim podsystemie będzie on użyty.

Należy więc wyodrębnić poszczególne operacje w systemie przetwarzania danych, określić typy błędów, które mogą wystąpić podczas tych operacji i przyporządkować im odpowiednie techniki kodowania.

1. Operacje w systemie przetwarzania danych

W systemie przetwarzania danych możemy wyodrębnić następujące operacje:

a. Transmisja danych
Należą tu wszystkie operacje polegające na przesyłaniu danych między dwoma punktami systemu. Szczególnymi, ale najczęściej występującymi przypadkami, są transmisja danych między odległymi terminalami a centralnie umieszczonym komputerem oraz transmisja między częściami składowymi sieci komputerowych.

b. Przetwarzanie danych w procesorze
Do operacji tych zalicza się wszystkie standardowe procedury arytmetyczne, procedury przesuwania, obiegu, arytmetyki uzupełnieniowej oraz procedury sterowania danymi.

c. Przechowywanie danych
Zalicza się tu przechowywanie danych w pamięci taśmowej, bębnowej, dyskowej i rdzeniowej oraz w rejestrze przesuwnym. Operacje wczytywania i odczytywania zapisów mogą następować szeregowo lub równolegle.

2. Charakterystyka błędów

Błędy, które powstają w systemach przetwarzania danych można sklasyfikować następująco:

- pojedyncze błędy w bloku cyfr binarnych o stałej długości, występujące niezależnie od siebie ilościowo; przyczyną tych błędów są krótkie, chwilowe zakłócenia w pracy hardware i szumy w liniach transmisyjnych,

- błędy wielokrotne, tego samego pochodzenia co wyżej,
- błędy występujące seriami; są one wynikiem chwilowej wadliwej pracy urządzeń, obserwowanej najczęściej w urządzeniach pamięci i liniach transmisji,
- błędy asymetryczne, których występowanie jest związane ze stałymi zakłóceniami w obwodach logicznych i urządzeniach pamięci.

3. Systemy kodowania

Systemy kodowania klasyfikuje się zgodnie z ich własnościami matematycznymi:

a. Kody liniowe

W tym systemie blok cyfr binarnych zwiększony jest o pewną liczbę cyfr kontrolnych. Ich postać jest funkcją cyfr informacyjnych, zaś związek ten jest określony tzw. macierzą testów parzystości. Przy prawidłowym doborze liczby cyfr kontrolnych oraz przez określenie ich zależności od cyfr informacyjnych możliwe jest użycie kodów liniowych dla zabezpieczenia przed błędami w transmisji danych.

b. Kody resztowe

Kody resztowe służą do zabezpieczenia przed błędami procesu przetwarzania danych w procesorze. Blok binarnych znaków, który ma być przetwarzany mnożony jest przez określoną liczbę, a później po przetworzeniu dzielony przez tę samą liczbę. Jeżeli reszta powstała w wyniku tego dzielenia różni się od zera znaczy to, że w czasie operacji zaistniał błąd. Przy prawidłowo dobranym mnożniku możliwa jest korekta wszystkich pojedynczych błędów we wszystkich operacjach przetwarzania w PAO.

c. Kody asymetryczne

Pewne typy urządzeń, takie jak pamięci na rdzeniach magnetycznych, diody i niektóre tranzystory, są miejscem powstawania błędów asymetrycznych. Wadliwie pracujący rdzeń powoduje zamianę wprowadzonej do pamięci jedynki na zero - rzadko kiedy odwrotnie. Kody przeznaczone dla zabezpieczenia przed omawianym typem błędów nazywane są asymetrycznymi lub kodami korekty jedynek. Stosuje się je w transmisji, przetwarzaniu i przechowywaniu danych.

d. Dwuwymiarowe kody z kontrolą parzystości

W wielu miejscach w systemie przetwarzania danych, dane występują w formie dwuwymiarowej tablicy znaków binarnych /na przykład na taśmie magnetycznej będącej wejściem lub w pamięci rdzeniowej/. Ochronę danych w tej formie umożliwiają dwuwymiarowe kody z kontrolą parzystosci. Są to najczęściej kody liniowe i stosuje się je zwykle w transmitowaniu i przechowywaniu danych.

Zestawienie 1 stanowi wyszczególnienie typów kodów, ich zastosowanie i najważniejsze elementy charakterystyczne. Opis matematyczny każdego z kodów znajduje się w dodatku B.

B. PORÓWNANIE EFEKTYWNOŚCI I KOSZTÓW

Zastosowanie kodowania zarówno do wykrywania, jak i korekty błędów wymaga dodatkowego wyposażenia w sprzęt i zwiększenia czasu przetwarzania ze względu na operacje kodowania i dekodowania oraz przygotowanie cyfr kontrolnych. Aby kodowanie było efektywne, wzrost stopnia niezawodności działania urządzeń oraz redukcja czasu potrzebnego do usuwania usterek muszą być tak duże, by zrównoważyć lub przewyższyć dodatkowy koszt hardware'u i jego większą złożoność. Przy określonych kosztach stopień niezawodności osiągnięty przy zastosowaniu kodowania musi być co najmniej równoważny niezawodności, którą można osiągnąć przy użyciu logiki większościowej i splecionej logiki nadmiaru.

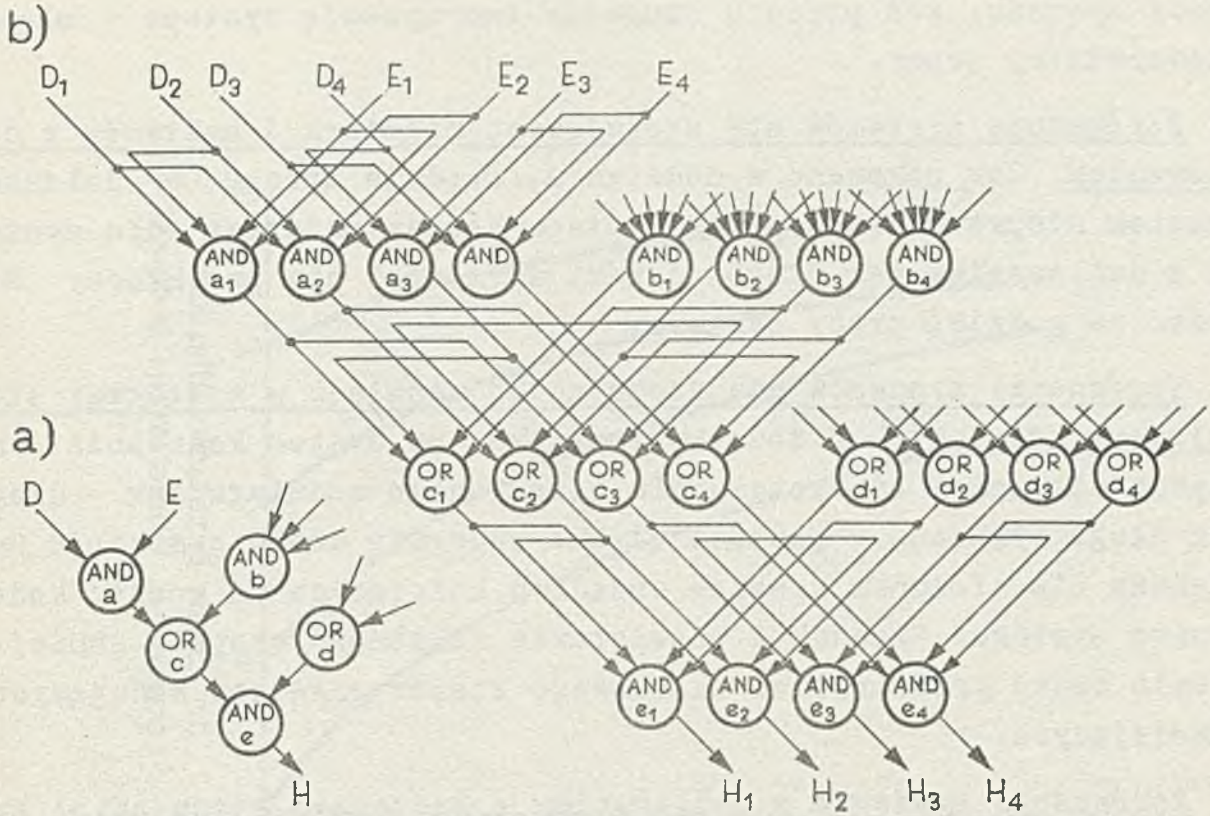
W praktyce, najszerzej stosowaną formą nadmiaru jest zwykłe dublowanie co najmniej niektórych urządzeń systemu. Interesującym jest zatem porównanie efektywności tej metody z efektywnością uzyskiwaną przy zastosowaniu kodowania.

Oznaczmy przez T średni czas przestoju w systemie nie stosującym nadmiaru, wyrażony w godzinach przypadających na godzinę

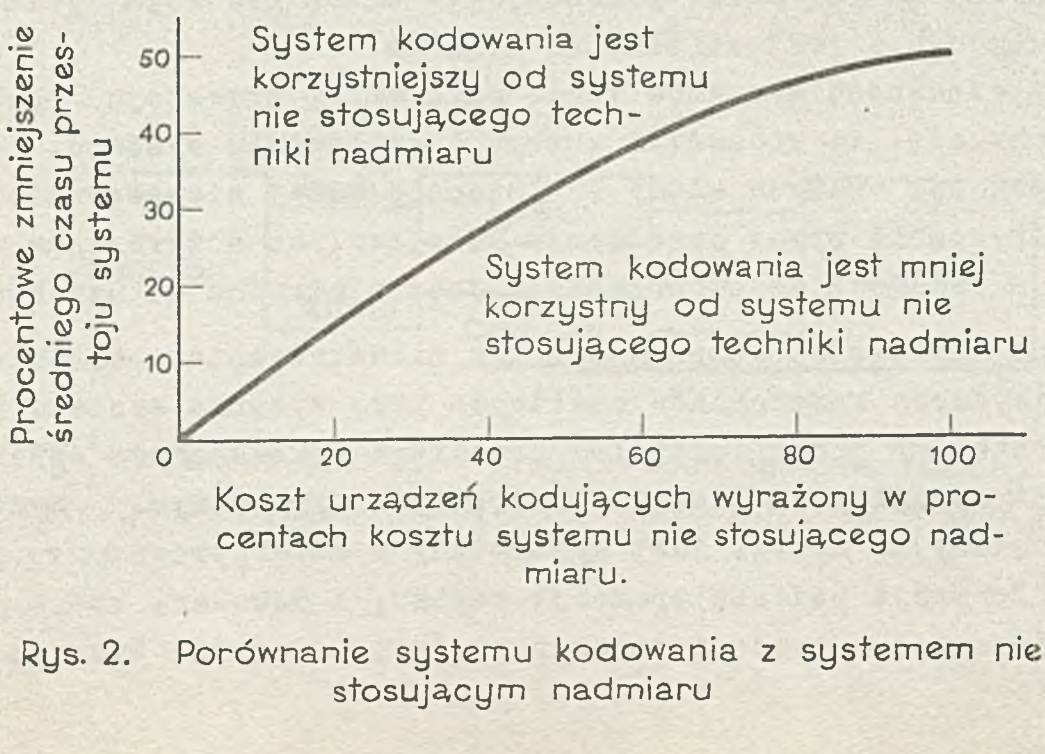
Zestawienie 1

Rodzaje kodów i ich własności

Rodzaj kodu	Zastosowanie	Nadmiar	Funkcja kodu	Kodowanie	Dekodowanie	Systemy, w których został zastosowany kod
Kod liniowy	Transmisja danych	Niewielki dla pojedynczych błędów	Wykrywanie i korekcja pojedynczych i wielokrotnych błędów	Proste	Wymaga wielu urządzeń dekodujących	IBM 2820
Kod resztowy	Przetwarzanie danych w procesorze	Niewielki	Wykrywanie i korekcja pojedynczych i wielokrotnych błędów	Proste	Proste	UNIVAC III
Kod asymetryczny	Transmisja, przetwarzanie danych w procesorze, przechowywanie danych	Mniejszy niż dla kodów liniowych	Wykrywanie i korekcja pojedynczych błędów tego samego rodzaju	Proste	Proste	IBM 650 IBM 7070
Dwuwymiarowy kod z kontrolą parzystości	Transmisja danych, przechowywanie danych	Niewielki	Koryguje dowolną liczbę błędów tego samego rodzaju w wierszu tablicy	Bardziej złożone	Bardziej złożone	Szeroko stosowane w przeszłości i obecnie



Rys. 1. Technika nadmiarowa Tryon'a :
a) - obwód bez nadmiaru,
b) - obwód z zastosowaną techniką nadmiarową Tryon'a



Rys. 2. Porównanie systemu kodowania z systemem nie stosującym nadmiaru

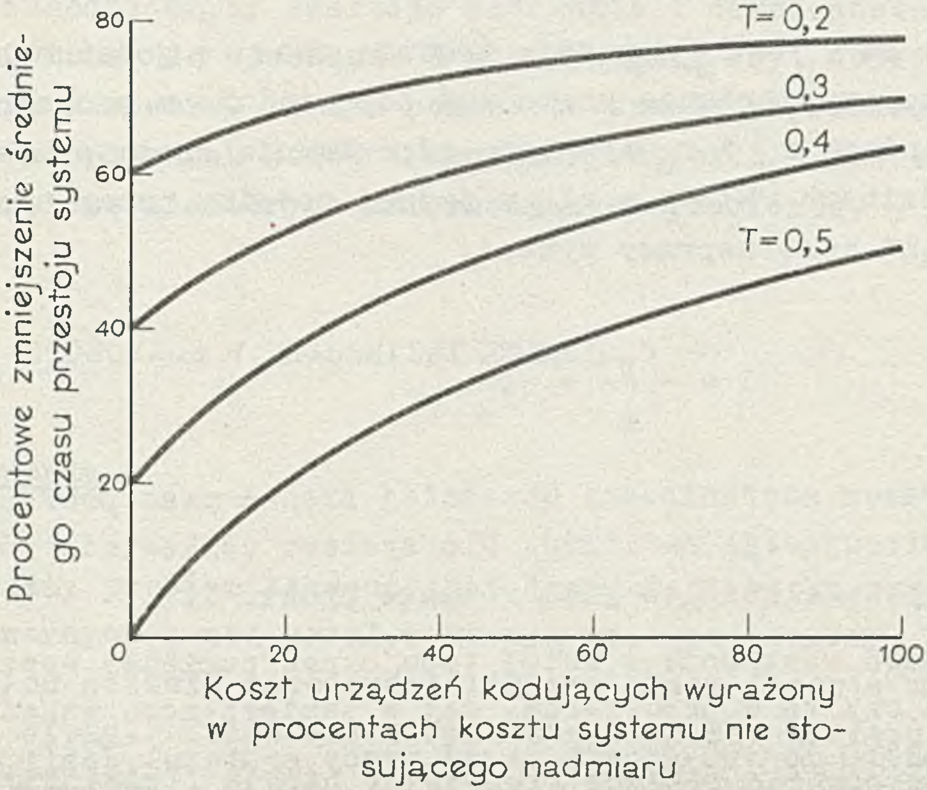
pracy systemu, zaś przez C oznaczmy amortyzację systemu w dol/godz. rzeczywistej pracy.

Porównanie systemów nie stosujących nadmiaru i systemów z dublowaniem. Jak pokazano w dodatku A, średnia strata w dolarach wskutek nieprawidłowej pracy systemu będzie mniejsza dla systemu z dublowaniem tak długo, dopóki wartość T nie przekroczy 0,5 godz. na godzinę pracy systemu.

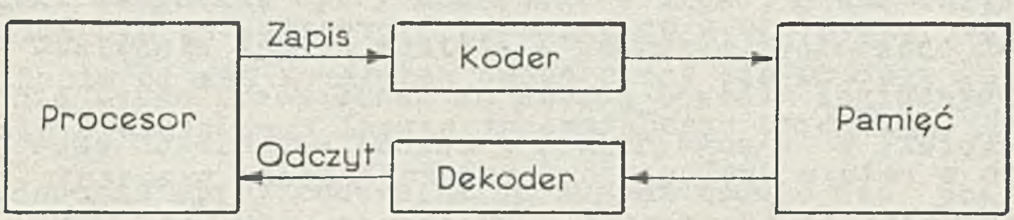
Porównanie systemów nie stosujących nadmiaru z systemami stosującymi kodowanie. Z dodatku A wynika, że system kodowania ma lepsze własności od systemu nie stosującego nadmiaru tak długo, jak długo wyrażona w postaci ułamka redukcja czasu przestoju jest większa niż stosunek kosztów urządzeń kodujących do kosztu kodowanego systemu. Rysunek 2 przedstawia zależność stopnia zmniejszenia czasu przestoju od ułamkowego kosztu urządzeń kodujących i dekodujących.

Porównanie systemów z dublowaniem z systemami stosującymi kodowanie. Na rysunku 3 przedstawiono wymaganą redukcję czasu przestoju jako funkcję kosztów kodowania i stałe wartości T czasu przestoju w systemie nie stosującym nadmiaru. Dla każdej krzywej na wykresie, system stosujący kodowanie jest sprawniejszy i ekonomiczniejszy powyżej krzywej. Należy zwrócić uwagę, że dla systemów o bardzo małym czasie przestoju wymagana redukcja jest duża i w praktyce nie jest zależna od kosztów.
Ponieważ większość systemów takie małe czasy przestoju posiada, wydawałoby się, że kodowanie ustępuje dublowaniu systemu. Jednakże przypadkowe rzadkie błędy w systemach dużej niezawodności są łatwo korygowane przez urządzenia kodujące, co sugeruje, że wymagana duża redukcja czasu przestoju jest nietrudna do uzyskania.

Główne możliwości kompromisu. Dla zilustrowania ilościowego najważniejszych kompromisów możliwych przy wyborze systemu nadmiaru, rozważmy prosty przykład procesora wykonującego operacje zapisów i odczytów na pamięci o dostępie wyrywkowym. System, przedstawiony na rys.4, jest wyposażony w koder, przez który przepływa informacja podczas operacji zapisu, i dekoder, który przyjmuje informacje z pamięci podczas operacji odczytu. Funkcją ko-



Rys. 3. Porównanie systemu z dublowaniem z systemem kodowania.



Rys. 4. Kontrola błędów przy pomocy systemu kodowania

dera jest dodanie nadmiarowych cyfr kontrolnych do bitów adresowych i bloków, które mają być przechowane w pamięci. Przy operacji czytania dekoder przeprowadza badanie błędów i przekazuje zdekodowany blok do procesora.

Załóżmy, że f_p i f_m wyrażają średnie czasy w godzinach pomiędzy momentami wystąpienia uszkodzeń odpowiednio w procesorze i w pamięci, i niech r_p i r_m wyrażają odpowiednie średnie czasy naprawy w godzinach. Wtedy w ciągu jednej godziny pracy systemu całkowity średni czas naprawy wynosi:

$$T = \frac{r_p}{f_p} + \frac{r_m}{f_m}$$

Jest to zarazem zdefiniowany wcześniej średni czas przestoju systemu nie stosującego nadmiaru. Dla systemu całkowicie równolegle nadmiarowego, wielkość ta jest równa po prostu T^2.

W systemach rozważanego tutaj typu, okres pomiędzy wystąpieniem uszkodzenia czy to w procesorze, czy w pamięci, może wahać się w granicach od 10 do 100 godzin czasu pracy systemu. Jeśli przyjmiemy 50 godzin jako wartość średnią i 5 godzin jako czas usunięcia uszkodzenia, to średni czas przestoju w systemie nie stosującym nadmiaru wyniesie ok.0,2 godziny na każdą godzinę pracy systemu. Jeśli przyjmiemy ponadto, że koszt wyposażenia kodującego wynosi około 10 procent kosztu całego systemu to, jak wynika z rys.2, system stosujący kodowanie musi dać w wyniku zmniejszenie czasu przestoju o co najmniej 9 procent, by być korzystniejszym od systemu nie stosującego nadmiaru. Jak to widać na rysunku 3, aby system tu przedstawiony wypadł korzystnie w stosunku do systemu z pełnym dublowaniem, redukcja czasu przestoju musi wynosić co najmniej 65 procent. Czy dany system kodowania spełni te warunki, zależeć będzie od rodzaju błędów, które są przyczyną zawodności pracy systemu.

III. WPROWADZANIE KODÓW I SKUTECZNOŚĆ ICH DZIAŁANIA

Aby kodowanie było efektywnym narzędziem kontroli błędów, jest sprawą zasadniczą, by operacje kodowania i dekodowania były proste, szybkie, łatwe do wdrożenia i nie wymagały zbyt dużej ilości dodatkowego hardware. W tym rozdziale omawiamy problemy wdrażania każdego z opisanych kodów i wyciągamy wnioski o skuteczności poprzez analizę niektórych interesujących problemów.

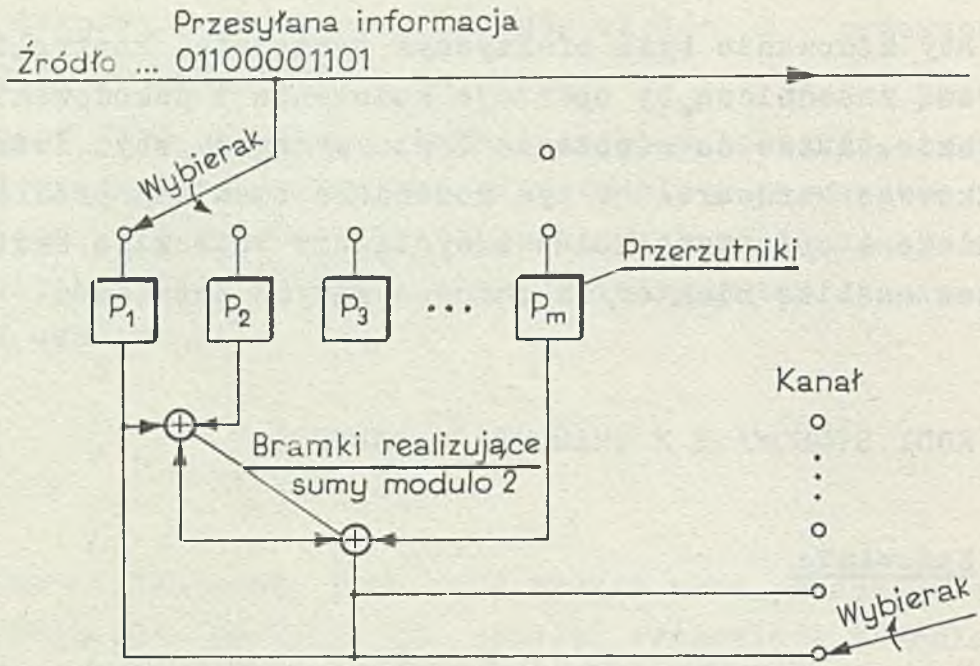
A. KODY STOSOWANE W TRANSMISJI DANYCH

1. Kodowanie

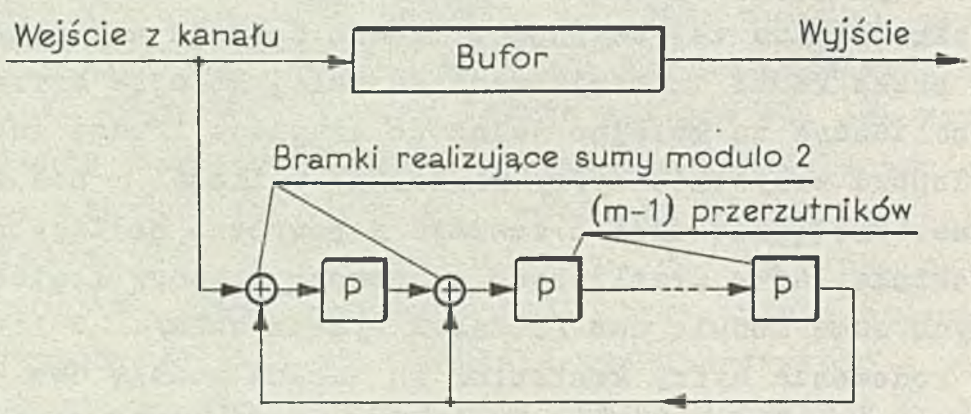
Rozważmy problem sekwencyjnej transmisji bloku zawierającego k cyfr binarnych przez kanał, w którym występują szumy. Przed transmisją blok zostaje zakodowany liniowo przez dodanie m cyfr kontrolnych do bloku. Kodowanie odbywa się w sposób następujący: na początku transmisji źródło informacji zostaje podłączone do kanału i następuje równoczesne przesłanie przez kanał k cyfr informacyjnych. Cyfry te zapamiętywane są również w szeregu k przerzutników. Po zakończeniu tej czynności źródło zostaje odłączone od kanału, zaś przez kanał transmitowane są dalej kolejne cyfry kontrolne, które obliczane są kolejno jedna po drugiej. Proces ten trwa tak długo, dopóki wszystkie cyfry kontrolne w ilości m nie zostaną przesłane. Następnie źródło zostaje z powrotem podłączone do kanału. Kodowanie łatwo zrealizować za pomocą obwodów logicznych realizujących sumę modulo dwa /różnica symetryczna/.x/ W liniowym systemie kodowania cyfry kontrolne są sumami modulo dwa poszczególnych cyfr w przesłanym bloku. Rysunek 5 przedstawia schemat obwodu kodującego.

---

x/ ang. "exclusive-or" /przyp.red./



Rys. 5. Obwód kodujący dla kodu liniowego



Rys. 6. Obwód dekodujący dla kodu Hamminga

Czas potrzebny do zakodowania i transmisji cyfr kontrolnych jest równy większej z dwóch następujących wielkości: m-tej wielokrotności czasu pracy układów logicznych realizujących sumę modulo dwa, lub m-tej wielokrotności czasu transmisji jednej cyfry. Dodatkowe wyposażenie układu stanowi k przerzutników i co najwyżej m /k-1/ bramek realizujących różnicę symetryczną. Tak więc, zarówno czas kodowania, jak i złożoność układu kodowania rosną liniowo z wielkością nadmiaru. Jest to podstawowa zaleta kodów liniowych.

2. Dekodowanie

Dekodowanie kodów liniowych nie jest takie proste jak omówione wyżej kodowanie. Ogólnie dekoder musi zapamiętać co najmniej 2^{n-k} cyfr informacyjnych i wykonywać dużą ilość obliczeń po odebraniu każdego bloku. Czas wymagany do wykonania tych czynności i wielkość pamięci potrzebnej czyni kody liniowe mało przydatnymi. Nie mają takich ograniczeń specjalne kody liniowe, takie jak kody Hamminga czy BCH. Ich przydatność jest większa i są bardziej efektywne.

a. Kody Hamminga

Kody Hamminga są klasą kodów liniowych korygujących pojedyncze błędy powstające w trasmisji danych. Jeśli liczba cyfr kontrolnych wynosi m, to istnieje kod Hamminga, który ma 2^m-1-m cyfr informacyjnych. Dekodowanie takiego kodu odbywa się w dekoderze, którego uproszczony schemat logiczny przedstawiono na rysunku 6.

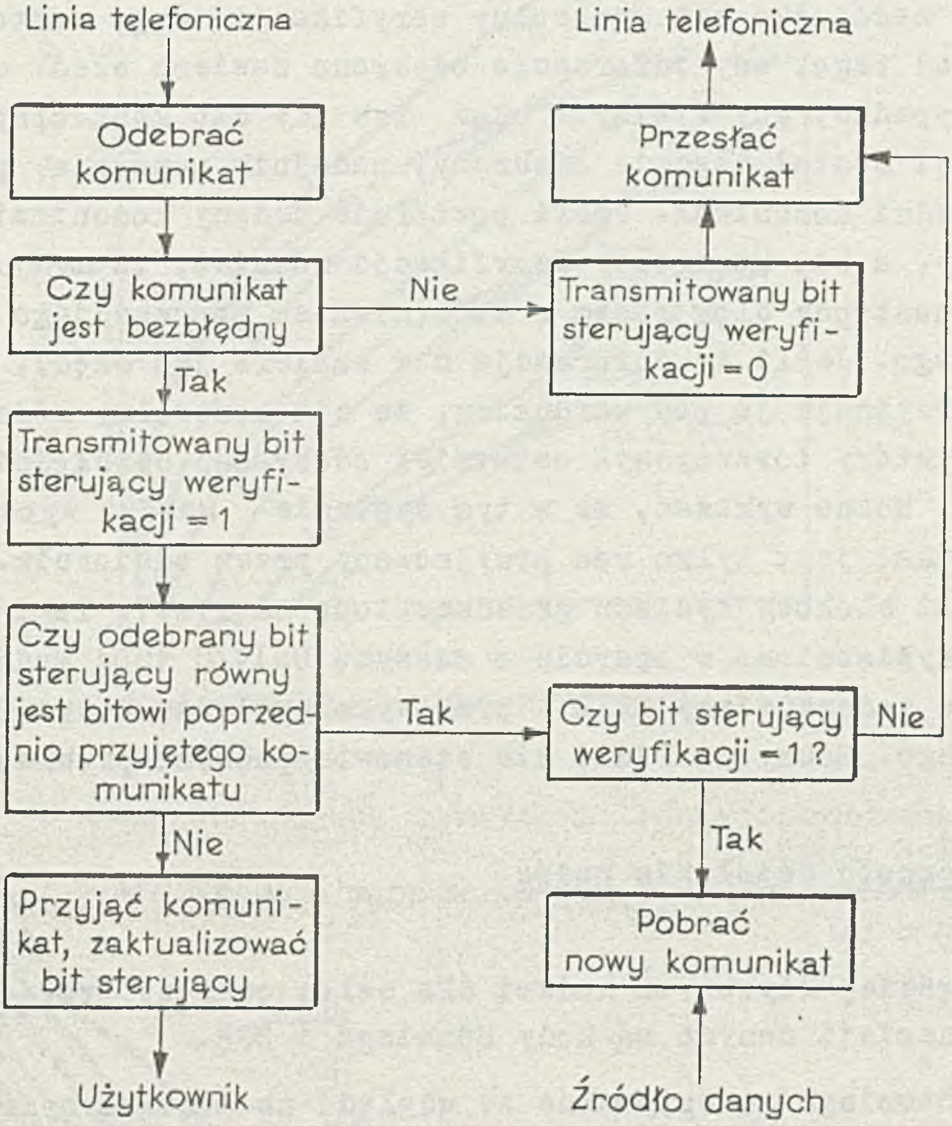
Odebrany ciąg bitów jest wprowadzany równocześnie do buforu i do rejestru przesuwnego zaopatrzonego w bramki i przerzutniki. Następnie bity są czytane z bufora i automatycznie przesuwają się bity wynikowe w rejestrze przesuwnym. Jeśli wszystkie bity w rejestrze przesuwnym są zerowe, znak odebrany był bezbłędnie.

Jeśli któryś jest jedynką /jeden lub więcej/, wystąpił pojedynczy błąd. Jeśli w danym momencie w rejestrze na pierwszej pozycji jest jedynka a pozostałe pozycje są zerowe, to ten bit który miał właśnie opuścić bufor należy skorygować.

b. Kody BCH
Dekodowanie kodów binarnych BCH jest znacznie bardziej skomplikowane niż dekodowanie kodów Hamminga.
W ogólnym przypadku dekoder zawiera: bufor, pewną liczbę przesuwnych rejestrów cyklicznych i centralny procesor, który lokalizuje pozycje, gdzie wystąpiły błędy. Można stwierdzić, że stopień złożoności dekodera zależy liniowo od stopnia nadmiaru i dekodowanie może odbywać się z szybkością powyżej miliona bitów/sek.

c. Kody cykliczne korekty błędów występujących seriami
Czas trwania zakłócenia w czasie transmisji danych jest często wystarczająco długi, by spowodować powstanie większej ilości błędów. Rozważmy teraz sposoby dekodowania kodów liniowych umożliwiających zabezpieczenie transmisji przed błędami tego typu. Proces dekodowania przebiega następująco: cały blok zostaje wczytany równocześnie do buforu i rejestru przesuwnego, w którym bity przesuwają się przy każdej wczytywanej cyfrze. Zawartość buforu jest następnie odczytywana cyfra po cyfrze, przy równoczesnym przesuwaniu rejestru po każdym znaku i wprowadzeniu zera. Zawartość rejestru po każdym przesunięciu wskazuje na obecność błędu lub jego brak.

d. Wykrywanie błędów i retransmisja
Dotychczas rozważaliśmy jedynie transmisję w jednym kierunku, całkowicie ignorując możliwość retransmisji bloków, które zawierają błąd. Taka retransmisja jest bardzo wskazana, jeśli zakłócenia mają charakter przejściowy. Ponieważ większość zakłóceń występujących w transmisji danych ma właśnie charakter przemijający należy przebadać tę możliwość - tym bardziej, że czysta detekcja błędów jest nawet łatwiejsza do wprowadzenia niż korekta błędów pojedynczych.
Opisany tutaj system składa się z dwukierunkowej linii transmisyjnej z możliwością transmisji niejednoczesnej /halfduplex/. Wykorzystywane jest standardowe łącze telefoniczne. Szybkość transmisji wynosi 2.000 bitów/sek. Istnieje źródło komunika -



Rys. 7. Schemat blokowy systemu wykrywania błędów i retransmisji

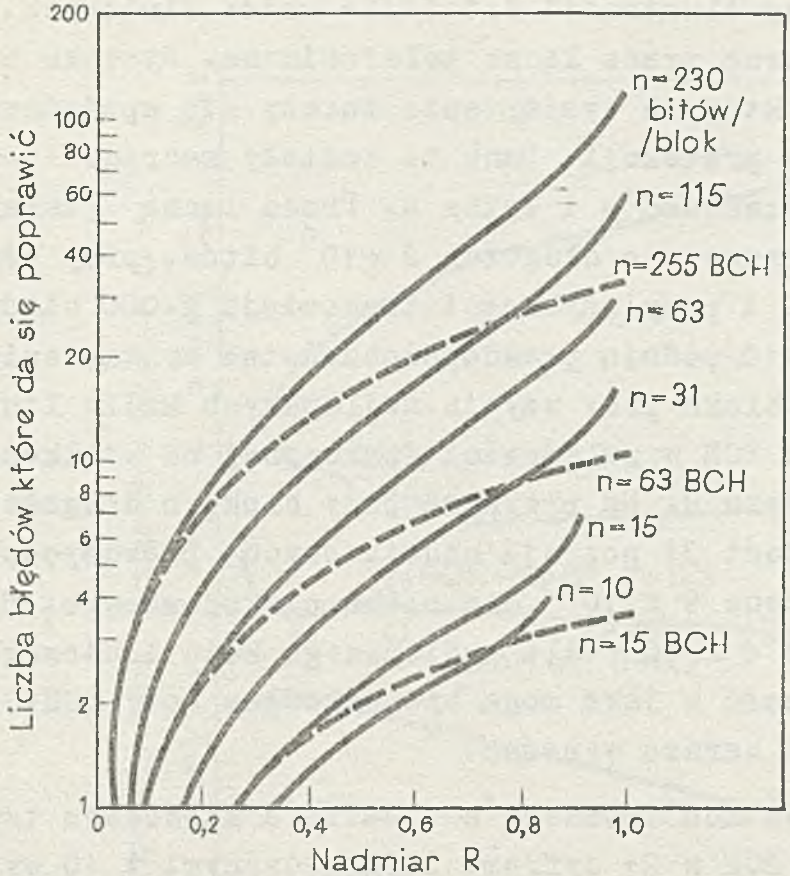
tu z urządzeniem kodującym i odbiornik z niemal idealną zdolnością wykrywania błędów. Do każdego bloku informacji przesyłanego ze źródła do odbiornika dołączony zostaje tzw. bit kontrolny. Odbiornik dekoduje odebraną informację i przesyła do nadajnika bit kontrolny weryfikacji. Jego wartość zależy od tego, czy informacja odebrana zawiera błąd czy nie. W przypadku, gdy wystąpił błąd lub gdy bit kontrolny weryfikacji został błędnie odebrany, nadajnik ponownie przesyła poprzedni komunikat. Jeśli powtórnie nadany komunikat jest poprawny, a bit kontrolny weryfikacji również, to nadajnik przesyła następny blok razem z dopełnieniem poprzedniego bitu kontrolnego. Jeśli ta informacja nie zawiera już błędu, odbiornik przyjmuje ją pod warunkiem, że bit kontrolny różni się od bitu, który towarzyszył ostatniej odebranej bezbłędnie informacji. Można wykazać, że w tym systemie każdy wyodrębniony komunikat jest tylko raz przyjmowany przez odbiornik. Schemat blokowy systemu przedstawiono na rys.7. Taki system urzeczywistniono w oparciu o maszynę UNIVAC 1004 Model A,przy użyciu modemu firmy BELL i przy wykorzystaniu łącza telefonicznego. Szumy na linii nie stanowią podobno problemu.

3. Skuteczność działania kodów

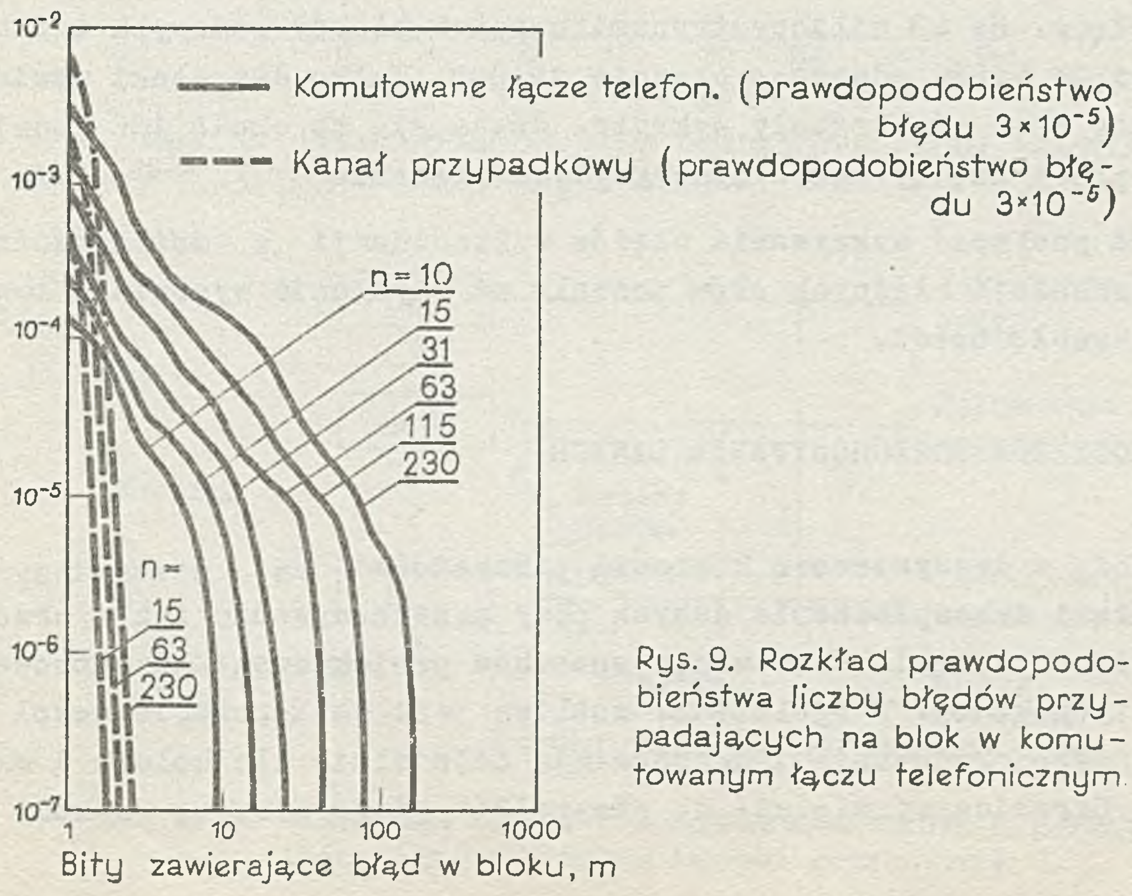
Najczęściej używanymi kodami dla celów ochrony przed błędami w transmisji danych są kody Hamminga i BCH.

Kody Hamminga są optymalne ze względu na minimum cyfr kontrolnych niezbędnych dla celów korekty pojedynczych błędów. Ich wadą jest jednakże to, że mogą poprawiać tylko pojedyncze błędy. Zdolność korekcyjna najlepszych kodów liniowych z kodami BCH w zależności funkcyjnej od stopnia nadmiaru R = m/n przedstawiono na rys.8.

Kody optymalne są często niezwykle trudne do wdrożenia ze względu na braki w strukturze. Należy jednakże zauważyć, że kody BCH są stosunkowo łatwe do realizacji, a przy tym ich działanie jest zbliżone do rezultatów osiąganych przy użyciu kodów liniowych.



Rys. 8. Zdolność korekty błędów przez kody liniowe



Rys. 9. Rozkład prawdopodobieństwa liczby błędów przypadających na blok w komutowanym łączu telefonicznym

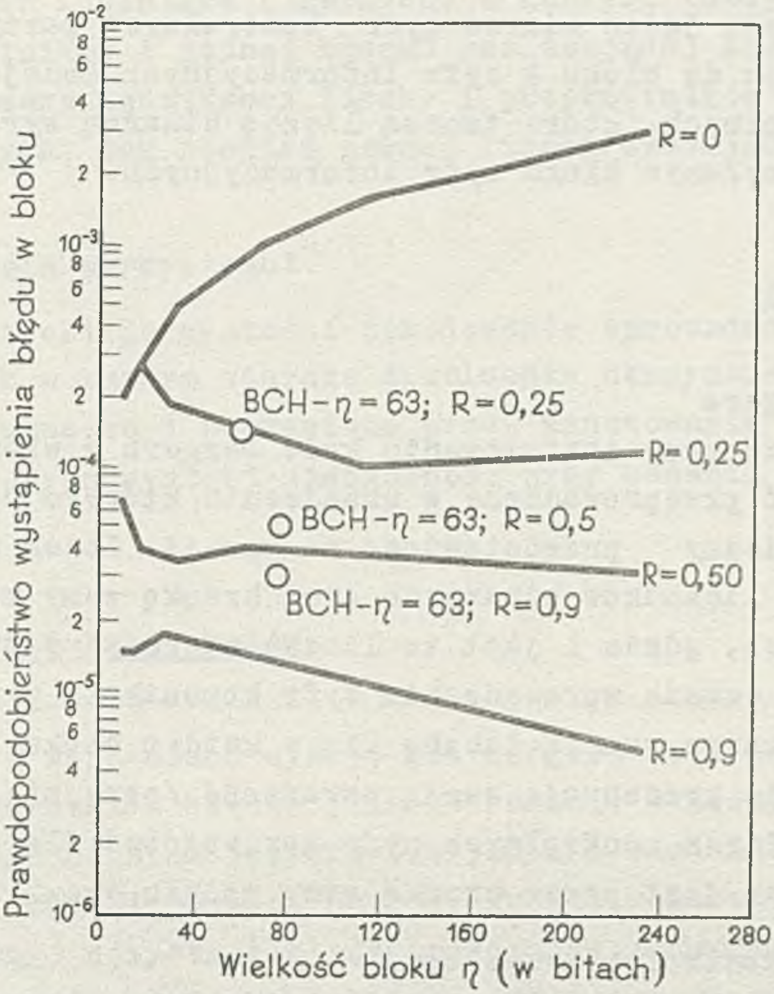
Celem dalszej ilustracji działania kodów liniowych, rozważmy transmisję binarną przez łącze telefoniczne. Rysunek 9 podaje liczbę błędów, których wystąpienia należy się spodziewać przy pracy łącza bez protekcji. Dane te zostały zebrane doświadczalnie przez Townsend'a i Watts'a. Przez łącza telefoniczne przesyłano informacje o długości 2×10^9 bitów, przy użyciu modemu firmy BELL i przy szybkości transmisji 2.000 bitów/sek. Wykresy na rys.10 podają prawdopodobieństwo wystąpienia błędu w zdekodowanym bloku przy użyciu najlepszych kodów liniowych i wybranych kodów BCH w zależności funkcyjnej od wielkości bloku i stopnia nadmiaru R. Na przykład przy bloku o długości 63 bitów, w którym jest 31 pozycji nadmiarowych, prawdopodobieństwo błędu będzie równe 9×10^{-3} dla bloku niekodowanego; $5{,}5 \times 10^{-5}$ dla kodu BCH; i 4×10^{-5} dla najlepszego kodu liniowego. Jeśli uwzględnić łatwość z jaką mogą być wdrożone kody BCH, ich przewaga tutaj jest bardzo wyraźna.

W doświadczeniach Townsend'a i Watts'a stosowano tzw. pięciodystansowy kod BCH z 21 cyframi informacyjnymi i 10 cyframi kontrolnymi do wykrywania serii zawierających, nie więcej niż cztery błędy. Na 63 miliony transmitowanych bloków kodowych o długości 31 bitów odebrano błędnie 29.000. Tylko dwa bloki zawierające błąd nie zostały wykryte. Równa się to około 130 godzinom pracy między dwoma niewykrytymi błędami.

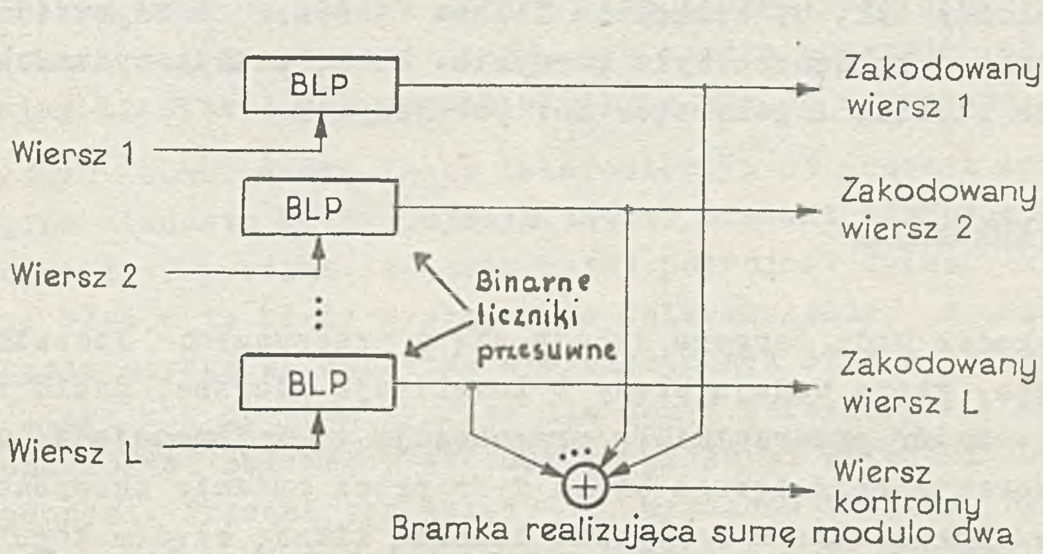
Ta zdolność wykrywania błędów w kombinacji z możliwością retransmisji błędnych słów pozwala na uzyskanie wysokiego stopnia bezbłędności.

B. KODY DLA PRZECHOWYWANIA DANYCH

Kody z dwuwymiarową kontrolą parzystości są zasadniczymi środkami zabezpieczenia danych przy przechowywaniu ich w urządzeniach pamięci. Jest wiele sposobów projektowania i stosowania tych kodów. W ogólności, możliwe jest zastosowanie dowolnych metod kodowania i dekodowania oddzielnie dla kolumn i wierszy. Ograniczymy się się do przypadków, gdzie wiersze zakodo -



Rys. 10. Prawdopodobieństwo wystąpienia błędu w łączu telefonicznym



Rys. 11. Koder dla kodu z dwuwymianową kontrolą parzystości. (BLP- binarne liczniki przesuwne).

wane są w kodzie Bergera lub w kodzie z badaniem parzystości i gdzie dodawany jest jeden wiersz cyfr kontrolnych parzystości. W kodzie Bergera do bloku k cyfr informacyjnych dodaje się $m=1+/\log_2 k/$ cyfr kontrolnych, które tworzą liczbę binarną wyrażającą liczbę zer w przesyłanym bloku cyfr informacyjnych.

1. Kodowanie

a. Kod Bergera

Kodowanie przy zastosowaniu kodu Bergera w wierszach tablicy może być przeprowadzone w urządzeniu, którego uproszczony schemat logiczny przedstawiono na rys.11. Koder zawiera l przesuwnych liczników binarnych oraz bramkę sumy modulo dwa o/l+1/ wejściach, gdzie l jest to liczba wierszy w dwuwymiarowym kodzie. W czasie wprowadzania cyfr komunikatu przesuwne liczniki binarne sumują liczbę zer w każdym bloku komunikatu, a następnie przesuwają swoją zawartość /seryjnie/ o następne m cyfr. Wiersz kontrolnych cyfr parzystości dla całego bloku formowany jest przez bramkę sumy modulo dwa.

b. Prosta kontrola parzystości

Kodowanie przy zastosowaniu prostej kontroli parzystości polega na dodaniu pojedynczej cyfry do każdego wiersza i każdej kolumny tak, by całkowita liczba jedynek w odpowiednich wierszach i kolumnach była parzysta. Do kodowania wystarczy jedynie licznik i generator zer lub jedynek.

2. Dekodowanie

Dekoder kodu Bergera składa się z przesuwnych liczników binarnych, które badają błędy i lokalizują wiersze, gdzie wykryto błędy. Zbiór przerzutników przechowuje tę informację i umożliwia korektę opóźnionego bloku cyfr przez badanie parzystości kolumny. Dekoder może korygować dowolną liczbę błędów tego samego rodzaju, tzn. samych błędów - zer lub samych błędów - jedynek w pojedynczym wierszu. Na zasadniczy koszt sprzętu składają się: koszt przesuwnych liczników binarnych, z których każdy stanowi zbiór m przerzutników i jednej bramki realizującej sumę modulo dwa. Dekoder wymaga dodatkowej liczby l przerzutników i l rejestrów buforowych, jak również pewnej liczby dwuwejściowych bramek AND i OR.

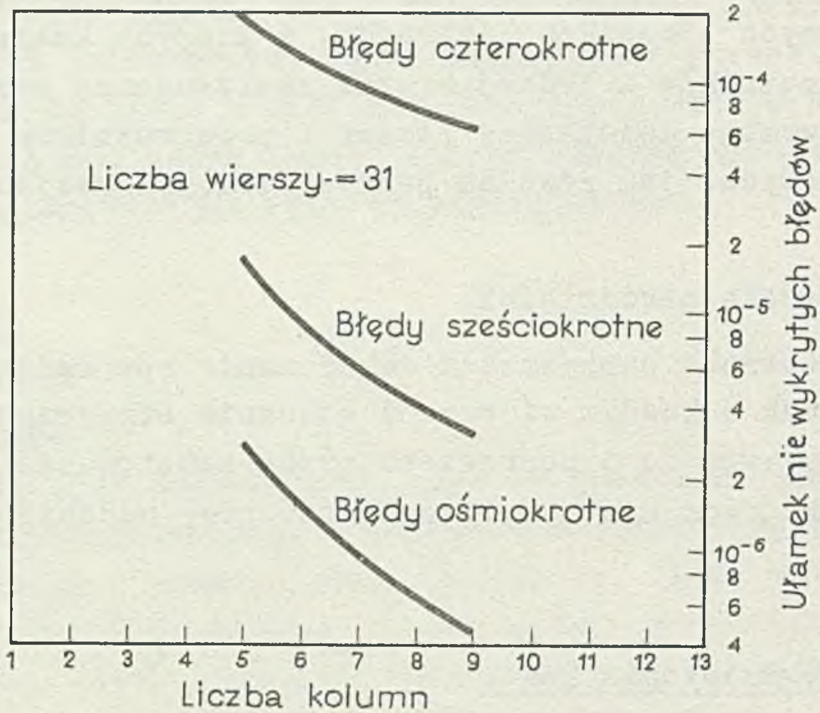
b. Prosta kontrola parzystości

Dla prostej kontroli parzystości dekodowanie sprowadza się do liczenia jedynek w każdym wierszu i kolumnie otrzymanej tablicy. Błędy wykrywane są i poprawiane przez zanotowanie wierszy i kolumn, dla których wystąpi niezgodność przy badaniu parzystości.

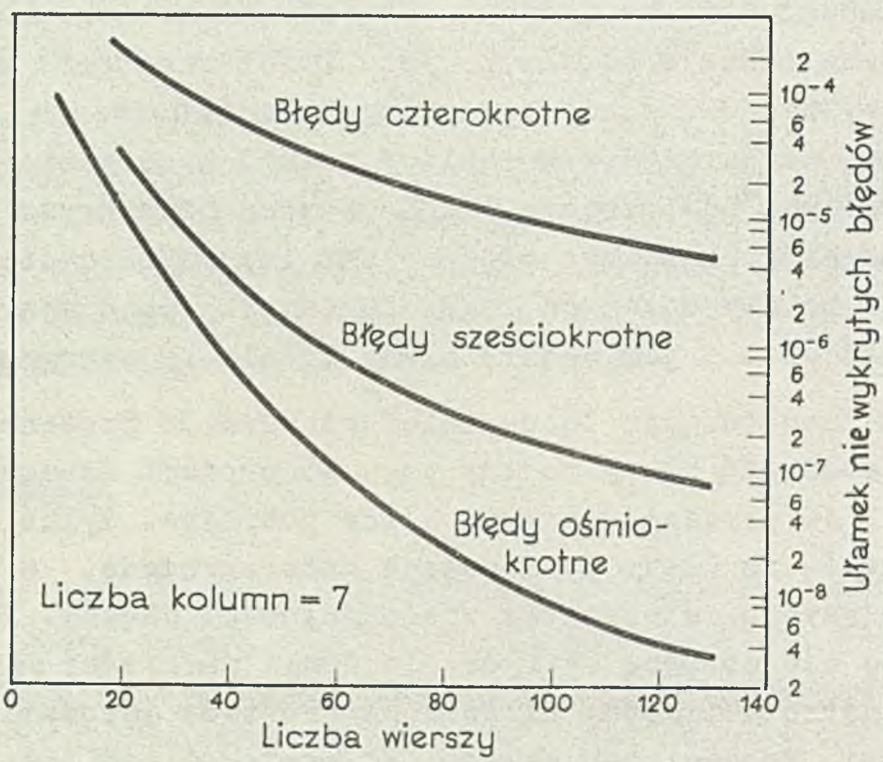
3. Skuteczność działania kodów

Jak już o tym wspomniano wyżej, kod Bergera może przeprowadzać korektę wszystkich błędów jednego rodzaju w jednym wierszu: zamienionego zera na jedynkę lub jedynki zamienionej w zero, ale nie oba rodzaje błędów na raz. Prosta kontrola parzystości może wykryć wszystkie błędy pojedyncze, podwójne, potrójne i wszystkie serie błędów w tablicy, jeśli sumaryczna liczba błędów jest liczbą nieparzystą. Mogą zostać niewykryte natomiast serie, w których liczba błędów jest liczbą parzystą. Skuteczność kodu zależy zatem od tego, jak często powstają błędy w parzystej liczbie i jak często błędy te nie są wykrywane.

W typowym komutowanym łączu telefonicznym 35 procent wszystkich błędów stanowią błędy pojedyncze, 15 procent stanowią błędy podwójne i 10 procent stanowią błędy potrójne. Tylko około 5 procent błędów to błędy występujące czterokrotnie, a procent błędów spada szybko ze wzrostem wielokrotności błędów. Procent błędów, które nie zostaną wykryte dla danej parzystej wielokrotności może zostać obliczony stosunkowo prostymi metodami kombinatorycznymi. Procent ten zależy od wymiarów tablicy. Rysunki 12 i 13 przedstawiają odpowiednio wyniki dla 31 wierszy i liczby kolumn do 13 oraz wyniki dla 7 kolumn i liczby wierszy do 140.



Rys. 12. Ułamek niewykrywalnych błędów w seriach parzystych



Rys. 13. Ułamek niewykrywalnych błędów w seriach parzystych

Dla przykładu weźmy kod tablicowy z 7 kolumnami i 41 blokami informacji. Tylko jeden błąd nie zostanie wykryty na każde 10.000 czterokrotnych błędów, jeden błąd na każde 50.000 sześciokrotnych błędów i jeden błąd na każdy milion ośmiokrotnych błędów. Widzimy zatem, że metoda ta daje duże efekty w wykrywaniu błędów.

C. KODY DLA PRZETWARZANIA DANYCH W PAO

Do zabezpieczenia przed błędami podczas operacji wykonywanych w arytmometrze i bloku sterującym komputera najlepiej nadają się kody resztowe. Wykrywanie błędów może odbywać się w ten sposób, że operandy są mnożone przez trzy, zaś po otrzymaniu wyniku operacji sprawdza się czy jest on podzielony przez trzy.

1. Kodowanie

Urządzenie kodujące dla kodu wykrywającego błędy jest prostym obwodem mnożenia przez 3. Mnożenie przez 3 polega na przesunięciu danej liczby o 1 pozycję w lewo /tzn. mnożenie przez 2/ i dodaniu danej liczby do tego wyniku.

W przypadku, gdy pożądane jest nie tylko wykrycie, lecz również korekta pojedynczego błędu, operacja kodowania jest bardziej złożona, gdyż wówczas należy przyjąć inny mnożnik. Dla większości mnożników obowiązują procedury standardowe. Dla pewnych wartości mnożnika, takich jak: 13, 19 lub 37 mnożenie może być wykonane w dwóch krokach pełnego dodawania przy użyciu elementów opóźniających.

2. Dekodowanie

Dekodowanie może odbywać się przy użyciu standardowych operacji arytmetycznych liczenia reszty po podzieleniu przez mnożnik. Gdy reszta jest zerem to nie ma błędu. W wyniku błędu w r-tej pozycji powstanie reszta $\pm 2^r$, zredukowana modulo mnożnik. Z tego można obliczyć umiejscowienie błędu. Algorytmy badania błędów opisane powyżej stosuje się do zabezpieczenia poszczególnych operacji arytmetycznych: dodawania, odejmowania, mnożenia lub dzielenia. Kodowanie polegające na badaniu reszty może również być stosowane do innych operacji. Dla przykładu kody te są efektywne w badaniu błędów podczas operacji przesunięcia, uzupełnienia i operacji cyklicznych, które zwykle wykonywane są w arytmometrze.

3. <u>Skuteczność działania kodów</u>

Kody resztowe są zdolne do korekty dowolnych, pojedynczych błędów, które mogą wystąpić podczas którejkolwiek z operacji arytmetycznych i sterowania w systemie komputerowym.

IV. NAJNOWSZE OSIĄGNIĘCIA W DZIEDZINIE KODOWANIA

A. KODY DLA WYKRYWANIA BŁĘDÓW WYŻSZYCH RZĘDÓW

W poprzednich rozdziałach uwaga była skoncentrowana na projektowaniu, wdrażaniu i skuteczności kodów, które mogą wykrywać lub przeprowadzać korektę błędów podczas przesyłania danych, przy przechowywaniu danych i podczas przetwarzania danych w PAO. Skoncentrowanie się na błędach występujących pojedynczo umożliwiało nam projektowanie stosunkowo prostych obwodów kodujących i dekodujących o możliwych do przyjęcia czasach liczenia. Ponieważ błędy występują podczas większości operacji, we względnie długich odstępach czasu, schematy te reprezentują rozwiązania zadowalające. Jednakże istnieją składniki systemu przetwarzania, dla których częstotliwość występowania błędów jest znacznie większa niż możliwość ich poprawienia przez kody korekty pojedynczych błędów. Typowym przykładem jest transmi -

sja danych przez łącze telefoniczne. W takich przypadkach wymagany jest nadmiar, który łącznie ze skomplikowaną strukturą kodów bardzo szybko zwiększa złożoność standardowego wyposażenia kodującego i dekodującego do granicy niepraktyczności. Wyjątkiem są tu kody BCH i kody z dwuwymiarową kontrolą parzystości, których działanie omawialiśmy w poprzednim rozdziale.

Najnowsze badania dotyczą kodów i metod dekodowania, których wymagania odnośnie urządzeń są konkurencyjne dla innych systemów nadmiarowych. Pośród wielu kodów, nad którymi prowadzi się intensywne badania, należy wyróżnić kody rekurencyjne, czyli konwolucyjne.

Kodowanie rekurencyjne charakteryzuje się tym, że może być dokonywane bez konieczności dzielenia ciągu informacji elementarnych na duże bloki, jak to miało miejsce we wszystkich dotychczas omawianych kodach. Zamiast tego są one dzielone na małe bloki. Po przyjęciu takiego bloku koder, dodaje kontrolne cyfry binarne, które są funkcją nie tylko cyfr w danym bloku, lecz także cyfr z bloków poprzednio transmitowanych. Zaletą takiego schematu jest fakt zależności cyfr kontrolnych w pewnym zakresie od zdarzeń przeszłości.

Bardzo mało jest znanych kodów rekurencyjnych do korekty błędów przypadkowych, pomimo, że skuteczność działania najlepszych z nich jest dobrze sprawdzona. Ogólnie, najlepsze kody rekurencyjne przewyższają najlepsze kody liniowe, przy tej samej szybkości przesyłania danych. Istnieją jednakże rozwiązania konstrukcyjne opracowane specjalnie dla korekty błędów występujących seriami.

Do najlepszych algorytmów dla dekodowania kodów rekurencyjnych należą dekodowanie sekwencyjne i dekodowanie progowe. Dekodowanie progowe może być również użyte dla innych kodów, lecz daje zdecydowanie gorsze wyniki od liniowych metod dekodowania. Jednakże dekodery progowe są łatwiejsze do wdrożenia dla kodów, które można uczynić statystycznie niezależnymi.

W przeciwieństwie do metod liniowych, które są całkowicie określone strukturą kodu, dekodowanie sekwencyjne da się zastosować do większości typów kodów, niezależnie od ich struktur.

Algorytm podejmuje próbną decyzję odnośnie poprawności każdej otrzymanej cyfry, a potem przystępuje do dekodowania kolejnych cyfr na bazie decyzji poprzednich. Gdy okazuje się to niemożliwe, dekodowanie podejmowane jest od nowa po ustaleniu poprzednich decyzji. Ilość obliczeń wymagana do dekodowania dowolnej danej cyfry jest zmienną losową, co sprawia, że czasem potrzebny jest wyjątkowo długi czas do obliczeń dla pojedynczej cyfry binarnej. Chociaż prawdopodobieństwo takiego zdarzenia jest małe, jednak ogranicza to pełną użyteczność tej metody.

B. WIELKI STOPIEŃ INTEGRACJI /LSI/x/

W ostatnich latach powszechną uwagę zwraca zastosowanie do kontroli błędów urządzeń o wielkim stopniu integracji. Podstawową ideą jest tu zastosowanie regularnej struktury układów logicznych LSI w celu uproszczenia testowania błędów i ich lokalizacji i zapewnienie w pewnym stopniu "opanowania" uszkodzeń poprzez nadmiarowość modułową. W takich systemach jak komórkowe sieci układów logicznych, wyizolowane komórki zawierające uszkodzenie mogą być omijane za pomocą podprogramów lub systemów wykrywania uszkodzeń w połączeniu z przełączaniem logicznym eliminujących uszkodzone komórki.
Te cechy, łącznie z większą niezawodnością systemów LSI czynią je systemami o potencjalnym znaczeniu w zakresie kontroli błędów.

C. WIELOWARTOŚCIOWE SYSTEMY LOGICZNE

W transmisji danych przez kanały z szumem, szybkość rzetelnej transmisji jest ograniczona przepustowością kanału. Jednakże, kanał jest funkcją liczby poziomów użytych w transmisji.

---

x/ LSI - Large-Scale Integration.

Często szybkość rzetelnej transmisji może być znacznie popra - wiona w stosunku do operacji binarnej przez użycie wielowartościowych systemów logicznych.

Bieżące badania związane są z dwoma zasadniczymi aspekta - mi: jeden z nich to metody wdrażania układów logicznych przy użyciu standardowej technologii elementów składowych; drugi to efektywność kodowania w zastosowaniu do kontroli błędów w transmisji danych i podczas przetwarzania danych w procesorze.

DODATEK A

MATEMATYCZNA ANALIZA KOSZTÓW I EFEKTYWNOŚCI

Pomimo dodatkowych urządzeń hardware i czasu, które są konieczne do jego wdrożenia, kodowanie może konkurować zarówno z systemem nie stosującym nadmiaru, jak i z innymi systemami nadmiarowymi. W przedstawionych niżej porównaniach pokazujemy co należy uczynić, by systemy stosujące dublowanie i systemy kodowania były bardziej efektywne, z punktu widzenia kosztów niż systemy nie stosujące nadmiaru. Dalej wykażemy, jakie wymagania należy postawić przed systemem kodowania, aby był korzystniejszy od systemu stosującego dublowanie.

Załóżmy, że T będzie średnim czasem przestoju układu nie stosującego nadmiaru wyrażonym ułamkiem w godz/godz operacji i niech koszt urządzenia amortyzuje się z szybkością C dol/godz rzeczywistego czasu operacyjnego. Wtedy średnie straty wynikające z błędnego działania wyrażone w dol/godz czasu operacyjnego są:

$$L_o = CT$$

Dla systemu całkowicie zdublowanego średni czas przestoju na godzinę wynosi T, a ponieważ koszt takiego układu jest podwójny, całkowite straty przypadające na godz czasu pracy wynoszą:

$$L_o = 2\,CT$$

Zakodowanie systemu bez nadmiaru umożliwia redukcję średniego czasu przestoju o ułamek α, ponieważ system może pracować pomimo zawodnego działania, na skutek poprawiania błędów przez schemat kodujący. Jednakże urządzenia kodujące mogą również ulegać awarii i wymagać naprawy, co w rezultacie zwiększa całkowity czas przestoju o ułamek β. Redukcja czasu przestoju wyraża się zatem różnicą $\alpha - \beta$. Zakładając, że koszt urzą - dzeń kodujących wyraża się ułamkiem γ kosztu całego systemu, średnie straty w dol/godz. wynoszą dla systemu kodowanego:

$$L_C = /1+\gamma/C \left\{ /1-\alpha/T + \beta T \right\}$$

Wiemy, że sposób zabezpieczania przed błędami jest tym lepszy, im średnie straty czasu pracy układu są mniejsze. Zatem system z dublowaniem jest lepszy od systemu nie stosującego nadmiaru, jeśli L_D jest mniejsze od L_o. Rozwiązanie tej nierówności prowadzi do warunku:

$$T < 0{,}5$$

który określa potrzebny średni czas przestoju. System kodowany jest lepszy od systemu nie stosującego nadmiaru, jeżeli L_C jest mniejsze od L_o, stąd wynika zależność:

$$\alpha - \beta > \frac{\gamma}{1+\gamma}$$

która mówi, że redukcja czasu przestoju uzyskana dzięki kodo - waniu wyrażona ułamkiem musi być większa od stosunku kosztu u- rządzeń kodujących do kosztu kodowanego systemu.

Do najmniejszych granicznych wartości czasu przestoju do - datkowy czas przetwarzania potrzebny na przeprowadzenie czyn - ności kodowania i dekodowania staje się znaczący i musi być do- dany do czasu przestoju w systemie nie stosującym nadmiaru. Jeśli wartość tego czasu wyrazimy w postaci ułamka δ, to cał- kowita strata przypadająca na godzinę wynosi:

$$L_C = /1+\gamma/C \left\{ /1-\alpha/ \, /1+\delta/T + \beta /1+\delta/T \right\}$$

i musi być spełniona nierówność:

$$\alpha - \beta > 1 - \frac{1}{/1+\gamma/ \, /1+\delta/}$$

aby można było powiedzieć, że system stanowi ulepszenie w stosunku do poprzedniego.

System z kodowaniem jest lepszy od systemu z dublowaniem, jeśli L_C jest mniejsze niż L_D. Po przekształceniu odpowiednich równań otrzymamy:

$$\alpha - \beta > 1 - \frac{2T}{1 + \delta}$$

DODATEK B

MATEMATYCZNY OPIS KODÓW

W przeszłości, badania nad praktyką i teorią kodowania były skoncentrowane na czterech głównych klasach kodów: kodach liniowych lub badania parzystości, używanych przeważnie do kontroli błędów w urządzeniach pamiętających i podczas transmisji przez łącza; kodach resztowych używanych głównie do badania poprawności operacji arytmetycznych; kodach asymetrycznych przeznaczonych dla systemów, w których przeważnie występują błędy jednego rodzaju; i kodach z dwuwymiarową kontrolą parzystości, używanych do kontroli błędów w urządzeniach pamięciowych i tablicach liczb. Każdy kod ma sobie właściwe zalety i problemy związane z zastosowaniem. Poniżej zostaną podane matematyczne opisy tych czterech wymienionych klas kodów z ich najważniejszymi własnościami.

1. <u>Kody liniowe</u>

Rozważmy blok k cyfr binarnych, które mają być transmitowane poprzez łącze danych do innej części sieci komputerowej. Ze względu na obecność szumów w liniach transmisyjnych, zawodności podzespołów i innych przyczyn, otrzymany blok będzie na ogół różnić się od bloku transmitowanego w jednej lub kilku pozycjach. By uczynić system transmisji odpornym na te błędy losowe, blok k cyfr zostaje przed transmisją zamieniony na blok m > k cyfr. Odpowiedni wybór cyfr kontrolnych umożliwi wówczas odtworzenie bloku oryginalnego z prawdopodobieństwem,które jest funkcją wielkości nadmiaru i jego powiązania z oryginalnym blokiem.

W liniowym systemie kodowania cyfry kontrolne stanowią sumy liniowe modulo dwa pewnych cyfr w bloku oryginalnym. Dla przykładu załóżmy, że transmitowany blok składa się z czterech cyfr binarnych /k=4/, i że mają być dołączone trzy cyfry kontrolne /n-k=3/. Cyfry te mogłyby być wybrane jak następuje:

$$R_1 = C_1 + C_2 + C_3$$

$$R_2 = C_1 + C_2 + C_4$$

$$R_3 = C_1 + C_3 + C_4$$

gdzie R_i jest i-tą binarną cyfrą kontrolną, C_i jest i-tą binarną cyfrą w oryginalnym bloku i dodawanie jest dodawaniem modulo dwa. Tak więc konkretny blok

0110

byłby zakodowany następująco:

0 1 1 0 0 1 1

Przypuśćmy teraz, że podczas transmisji bloku nadmiarowego, druga cyfra zostanie zmieniona z 1 w 0, tak że otrzymany blok będzie miał postać

0 0 1 0 0 1 1

Dekoder po stronie odbiorczej, będąc uczulony na funkcjonalną zależność cyfr kontrolnych od cyfr komunikatu oryginalnego , zbada tę zależność dla otrzymanego bloku i odkryje, że ostatnie trzy cyfry powinny mieć układ 1 0 1. Ponieważ to się nie zgadza z otrzymanym blokiem, wnioskuje na tej podstawie, że został popełniony błąd. Celem jego lokalizacji dekoder może założyć wystąpienie błędu w pozycjach: pierwszej, drugiej, trzeciej i czwartej, licząc za każdym razem cyfry kontrolne. Dla pierwszej cyfry założony przez niego blok transmitowany miałby postać:

1 0 1 0 0 1 1

co jest niemożliwe, gdyż binarne cyfry kontrolne nie spełniają swych definicyjnych równań. Podobne wnioski dotyczą cyfr binarnych trzeciej i czwartej, i jedynie założenie błędu w drugiej pozycji da potwierdzenie poprawności cyfr kontrolnych. W ten sposób dekoder stwierdza, że został popełniony błąd w drugiej pozycji. Oryginalny blok zostaje poprawiony przez dokonanie odpowiedniej zmiany, a cyfry kontrolne zostają usunięte.

Dla uogólnienia tego schematu zakładamy 2^k różnych bloków komunikatów, składających się każdy z k cyfr binarnych. Do każdego z tych bloków dołączony jest szereg n-k cyfr kontrolnych. Ten zbiór formuje kod /n,k/. Jeśli wszystkie cyfry kontrolne są liniowymi sumami modulo dwa cyfr komunikatu, to kod formuje grupę. Można zatem wybrać podzbiór k z tych bloków, jako generatorów grupy. Układając je w formie macierzy mającej k wierszy i n kolumn tworzymy tzw. opis macierzy generatorów kodu. W zrozumieniu pomoże nam następujący przykład: zakładamy, że blok oryginalnego komunikatu zawiera cztery cyfry i niech trzy cyfry kontrolne zostaną ustalone przez poprzednio podaną relację. Wtedy kod składa się z szesnastu bloków 7 cyfrowych:

0 0 0 0	0 0 0
0 0 0 1	0 1 1
0 0 1 0	1 0 1
0 0 1 1	1 1 0
0 1 0 0	1 1 0

binarne cyfry komunikatu	binarne cyfry kontrolne
0 1 0 1	1 0 1
0 1 1 0	0 1 1
0 1 1 1	0 0 0
1 0 0 0	1 1 1
1 0 0 1	1 0 0
1 0 1 0	0 1 0
1 0 1 1	0 0 1
1 1 0 0	0 0 1
1 1 0 1	0 1 0
1 1 1 0	1 0 0
1 1 1 1	1 1 1

a macierz

$$G = \begin{bmatrix} 1&0&0&0&0&1&1 \\ 0&1&0&0&1&1&0 \\ 0&0&1&0&1&0&1 \\ 0&0&0&1&0&1&1 \end{bmatrix}$$

jest generatorem w tym sensie, że cały kod może być generowany przez wzięcie wszystkich możliwych, cyfra po cyfrze, sum modulo dwa bloków w macierzy G.

Alternatywny opis kodu można podać przy pomocy tzw. macierzy kontroli parzystości H. Ogólna forma tej macierzy przedstawia się następująco:

$$H = \left[P_{n-kxk} \;\middle|\; I_{n-kxn-k} \right]$$

gdzie $I_{n-kxn-k}$ jest macierzą jednostkową, która ma n-k wierszy i tę samą liczbę kolumn, pierwsza pozycja i-tego rzędu i j- tej kolumny P jest równa jeden jeśli i-ta cyfra kontrolna jest funkcją j-tej cyfry komunikatu, i zerem w przeciwnym wypadku. W kodzie zilustrowanym wcześniej druga cyfra kontrolna R_2, dla przykładu, jest funkcją cyfr pierwszej, drugiej i czwartej komunikatu. Tak więc drugi rząd P zawiera jedynkę w wymienionych ko -

lumnach i zero w trzeciej kolumnie. Pełna macierz kontroli parzystości dla tego kodu podana jest poniżej.

$$H = \left[\begin{array}{cccc|ccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

W ogólności otrzymujemy 2^k bloki zakodowane przez dodanie do każdego z 2^k możliwych bloków komunikatu n-k cyfr kontrolnych, zgodnie z regułą określoną wyżej podaną macierzą kontroli parzystości.

Oznaczmy odległość między dwoma blokami kodu literą d jako liczbę pozycji, w których bloki się różnią i załóżmy, że w procesie transmisji powstaje w bloku e błędów. Jeśli będzie d>e, to otrzymany blok będzie różnił się od dowolnego bloku kodu i dekoder będzie mógł odkryć, że wystąpił błąd z samego tego faktu. Jeśli d>2e, to otrzymany blok będzie się różnił od bloku transmitowanego w e pozycjach, lecz od każdego innego bloku w co najmniej e+1 pozycjach. Błędy e mogą być poprawione przez zamianę otrzymanego bloku z blokiem kodu, który różni się od niego najmniejszą liczbą pozycji.

W poprzednim przykładzie kod ma minimalną odległość równą 3, to jest każde dwa bloki różnią się co najmniej w trzech pozycjach. Kod może zatem wykryć wszystkie podwójne błędy lub poprawić wszystkie pojedyncze błędy.

Ogólna procedura może być zdefiniowana za pomocą tablicy dekodującej. Na tej tablicy umieszczamy w pierwszym wierszu bloki kodowe. Blok otrzymywany umieszczany jest poniżej bloku kodowego, jeśli ma być w niego zdekodowany. W ten sposób, każdy otrzymany blok ukazuje się pod jednym i tylko jednym blokiem kodowym. Prawdopodobieństwo, że błąd nie zostanie poprawiony, będzie najmniejsze z możliwych, gdy blok otrzymany dekodowany jest do bloku kodowego, który różni się od niego najmniejszą liczbą pozycji. Transmisja jednego bloku kodu wymaga przysyłania k bitów wiadomości. Zatem, szybkość komunikacji wynosi k/n bitów na transmitowaną cyfrę.

Podstawowe twierdzenie teorii informacji Shannona mówi, że istnieją kody, które umożliwiają transmisję zasadniczo wolną od błędów pod warunkiem, że ta szybkość transmisji jest mniejsza od teoretycznego ograniczenia, jakie stanowi przepustowość kanału, i że długość bloków kodu może być dowolnie duża. Z tego ostatniego wymagania wynika, że dla zadowalająco dużych szybkości liczba bloków w kodzie jest ogromna i staje się niemożliwe sporządzenie tablicy kodowania lub dekodowania. Na tym polega zaleta kodów liniowych. Ich właściwa struktura matematyczna umożliwia określenie reguł zapisu kodu za pomocą generatora jakim jest macierz generująca kod lub macierzy kontroli parzystości oraz dekodowanie kodu przy pomocy prostych metod liniowych zamiast wyczerpujących metod porównywania.

Wdrażanie operatorów kodujących i dekodujących dla kodów liniowych zależy bardzo od typu struktury, jaką te kody posiadają. Z tej przyczyny została opracowana pewna liczba specjalnych kodów liniowych mających pewne pożądane własności, zaś poniżej w opracowaniu będą omówione najważniejsze z nich.

a. Kody Hamminga

Binarne kody Hamminga stanowią klasę kodów liniowych korygujących pewne błędy, dającą się opisać w sposób najprostszy przy pomocy macierzy kontroli parzystości. Jeśli przez m oznaczymy liczbę cyfr kontrolnych w kodowanym bloku to istnieje kod o długości $2^m - 1$ z liczbą $2^m - 1 - m$ cyfr informacyjnych dla każdej całkowitej liczby m. Macierz testów parzystości ma m wiersze i $2^m - 1$ kolumn. Kolumny stanowią wszystkie możliwe m-krotności zer i jedynek z wyjątkiem kolumny zawierającej same zera, i są ustawione w kolejności rosnącej. Przykładem takiego kodu jest zilustrowany w poprzednim punkcie kod, pod warunkiem, że kolumny będą właściwie ułożone. Liczba cyfr kontrolnych wynosi $m = 3$, a ponieważ długość bloku wynosi $n = 7$, to liczba pozycji informacyjnych jest równa $k = 4$.

Korekta pojedynczego błędu dla kodu Hamminga ma przebieg następujący: zakładamy, że słowo kodowe u jest transmitowane i podczas transmisji wystąpi pojedynczy błąd w i-tej pozycji. Zatem otrzymane słowo ma postać u + e, gdzie e jest słowem kodowym zawierającym jedynkę w i-tej pozycji, zaś w pozostałych pozycjach zera. Dekoder oblicza funkcję:

$$/u + e/H^T$$

która, na podstawie definicji H jest równa

$$eH^T$$

Ten ostatni rezultat jest równy i-tej kolumnie H.
Jeśli ta kolumna zostanie przedstawiona jako liczba dziesiętna, to zidentyfikuje położenie błędu i zostanie dokonana odpowiednia korekcyjna zmiana błędnej pozycji.
Dla ponownej ilustracji rozważmy znowu poprzedni przykład. Po odpowiedniej zmianie macierz kontroli parzystości ma postać:

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Załóżmy, że jest transmitowany blok kodowy 0 1 1 1 1 0 0 i piąty symbol otrzymany jest błędny. Wtedy e = 0 0 0 0 1 0 0 i

$$/u+e/H^T = uH^T + eH^T$$

$$= 0111100 \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} + 0000100 \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}$$

Ostatni wynik jest równoważny liczbie dziesiętnej 5, określającej położenie błędnej pozycji. Jest nią piąta cyfra.

Kody Hamminga są uogólnieniem dobrze znanych kodów parzystości. Ich główną zaletą jest łatwość kodowania i dekodowania. Są to oczywiście kody korekty pojedynczych błędów i jako takie, nie mogą być stosowane w systemach, w których błędy występują parami lub w większej liczbie.

b. Kody cykliczne

Kod nazywa się cyklicznym, jeśli każdy blok w kodzie, otrzymany przez przesunięcie cykliczne cyfr o jedną pozycję w prawo, również wchodzi w skład kodu. Jeśli kod ma długość n i ma m cyfr kontrolnych, może być całkowicie określony przez wielomian generujący $g/x/$ m-tego stopnia, który jest dzielnikiem wielomianu X^n-1 w algebrze dwuelementowej. Kody cykliczne mogą być określone również za pomocą macierzy generującej G złożonej z wierszy $g/x/$, $xg/x/$, $x^2g/x/ \ldots x^{k-1}g/x/$ lub macierzy kontroli parzystości H składającej się z wierszy $h/x/$, $xh/x/, \ldots x^{m-1}h/x/$, gdzie $x^n-1 = g/x/\ h/x/$. Tu blok kodu jest zbiorem współczynników odpowiadającego mu wielomianu modulo x^n-1.

Dla przykładu załóżmy n = 7. Ponieważ wielomian x^7-1 może być przedstawiony w postaci iloczynu

$$x^7 - 1 = /1-x/\ /1+x+x^3/\ /1+x^2+x^3/$$

zatem wielomian

$$g/x/ = 1 + x^2 + x^3$$

generuje kod cykliczny o długości n = 7 z trzema nadmiarowymi cyframi kontrolnymi i czteroma cyframi zawierającymi informację.
Blok kodowy odpowiadający wielomianowi $g/x/$ jest równy

$$1\ 0\ 1\ 1\ 0\ 0\ 0$$

zaś całkowita macierz generująca ma postać

$$G = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Macierz kontroli parzystości określona wielomianem $h/x/=/1-x/\ /1+x+x^3/$ jest równa:

$$H = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Należy zauważyć, że różni się ona jedynie zmianą położenia kolumny /permutacją/ kodu Hamminga przedstawionego wcześniej.

Wszystkie kody Hamminga są kodami cyklicznymi. Najlepszymi ze znanych kodów cyklicznych są kody należące do klasy BCH /kody Bose-Chandhuri-Hocquenghema/. Są one opisywane za pomocą pierwiastków wielomianu generującego g/x/. Ich teoria wykracza poza ramy niniejszego opracowania i kody te nie będą omawiane tu szerzej. Ogólnie, dla dowolnych dwóch liczb całkowitych m i t, istnieje binarny kod BCH o długości bloku $n = 2^m-1$, który przeprowadza korektę wszystkich kombinacji t lub mniejszej liczby błędów i wymaga co najwyżej mt cyfr kontrolnych. Chociaż teoria kodów BCH jest skomplikowana, to ich struktura jest w zasadzie prosta. Jest to przyczyną szerokiego zainteresowania tymi kodami. Zakodowanie wymaga jedynie rejestru przesuwnego z pętlą liniowego sprzężenia zwrotnego, zaś złożoność sprzętu dekodującego zależy jedynie liniowo od długości bloku. Te dwa czynniki powodują, że kody BCH są najbardziej obiecującą klasą kodów korekty wielokrotnych błędów.

c. Kody korekty zniekształceń seryjnych

W wielu rodzajach układów, błędy występują seriami. Jednak ze względu na specyficzną strukturę tego rodzaju błędów dobrze zaprojektowane kody doskonale sobie z nimi radzą. Większość tych tzw. kodów korekty błędów seryjnych projektuje się tak, aby korygowały wszystkie pojedyncze serie zniekształceń w bloku kodu. Wśród nich są kody binarne Fire´a generowane przez wielomian

$$g/x/ = p/x/ \cdot /x^c - 1/$$

gdzie p/x/ jest wielomianem nieprzywiedlnym stopnia l, którego pierwiastki mają rangę e, i gdzie e nie jest podzielne przez e. Długość n jest najmniejszą wspólną wielokrotnością e i c, liczba cyfr binarnych jest równa c + l, zaś liczba cyfr informacji n-c-l.

Kody te mogą korygować wszystkie serie błędów o długości nie większej niż b i jednocześnie wykrywać dowolne serie zniekształceń o długości d> b lub krótsze, jeśli c≥ b+d-1 i l≥ b. Jako przykład rozważmy binarny kod Fire´a generowany przez wielomian

$$g/x/ = /1+x^2 + x^7/ \; /x^{16} - 1/$$

W tym kodzie, l = 7, c = 16, zaś ranga pierwiastków wielomianu $1 + x^2 + x^7$ jest równa 127.x/

Ponieważ liczba 16 nie dzieli się przez 127, długość kodu wynosi n = 16 x 127 = 2032 i ma on 7 + 16 = 23 cyfry kontrolne. Kod może korygować wszystkie serie o długości 10 cyfr binarnych lub mniej.

Są jeszcze inne kody korygujące serie błędów, które wymagają mniejszego nadmiaru niż kody Fire´a, przy tym samym stopniu protekcji. Niektóre z nich mogą służyć do korekty specyficznych typów serii zniekształceń, na przykład takich, które zawierają parzystą lub nieparzystą liczbę błędów. Wszystkie te kody mogą być wdrożone przy użyciu tej samej zasadniczej techniki.

d. Kody o małej gęstości

Istnieje problem optymalizacji kodów korygujących i wykrywających serie błędów, polegający na tym, aby spośród kodów o tej samej zdolności wykrywania i korygowania, znaleźć kody z minimalną liczbą pozycji kontrolnych. Jednakże kod optymalny z punktu widzenia liczby cyfr kontrolnych nie zawsze jest najbardziej

---

x/ Celem wyjaśnienia przykładu podajemy tu pojęcie: Rangą pierwiastków wielomianu g/x/ nazywamy najmniejszą liczbę całkowitą e taką, że wielomian $1 + x^e$ jest podzielny przez g/x/. Przy znajdowaniu rangi wielomianu jest pomocne następujące twierdzenie:

Wielomian $1 + x^{2^k - 1}$ jest podzielny przez:

/1/ każdy wielomian nieprzywiedlny stopnia 1

/2/ każdy wielomian nieprzywiedlny stopnia k

/3/ każdy wielomian nieprzywiedlny stopnia r będącego dzielnikiem liczby k, a między innymi i przez wielomian 1+x. /Przyp. tłum./.

odpowiednim z punktu widzenia złożoności urządzeń kodująco-dekodujących. W sytuacji, gdy ten drugi czynnik jest brany pod uwagę, pierwszorzędne znaczenie ma klasa tzw. kodów o małej gęstości testów. Kody tej klasy umożliwiają budowę prostszych urządzeń do kodowania i dekodowania. Osiągane to jest jednak kosztem nieco większej redundancji niż minimalna dla tego typu błędów. Poniżej podano opis kodu o małej gęstości testów przy pomocy macierzy kontroli parzystości H.

Każdy z m wierszy H zawiera co najwyżej μ jedynek i każda z n kolumn jest różna od pozostałych i zawiera co najmniej jedną jedynkę. Liczba cyfr zawierających informację k jest największą z możliwych dla danych m i μ przy minimalnej odległości między blokami kodu. Przy tych ograniczeniach wśród kolumn macierzy H muszą występować wszystkie m kolumn zawierających pojedyncze jedynki, $/2^m/$ kolumn zawierających dwie jedynki itd. , dalsze zawierać będą zwiększające się ilości jedynek, aż do takiej liczby, jaka jest możliwa bez przekroczenia maksymalnej liczby jedynek przypadających na wiersz. Pierwsze m kolumn ma razem m jedynek, a każda z ostatnich k kolumn musi zawierać co najmniej dwie jedynki. Tak więc całkowita liczba jedynek w H wynosi co najmniej m + 2k. Musi być przy tym spełniona nierówność $m + 2k \leqslant m\mu$, jeśli pierwsze ograniczenie ma zostać nie naruszone. W ten sposób maksymalna, dopuszczalna liczba cyfr binarnych informacji jest określona wzorem:

$$k = \frac{m/\mu - 1/}{2}$$

w znacznej większości przypadków ta graniczna liczba może rzeczywiście być osiągnięta.

Dla przykładu załóżmy $\mu = 3$. Wówczas liczba cyfr stanowiących informację nie może przekroczyć liczby cyfr kontrolnych. Dla $m \geqslant 3$ granica może być osiągnięta w sposób taki, jak to przedstawiono poniżej na przykładzie macierzy kontroli parzystości dla m = k = 5.

$$H = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Poniżej przedstawiono najmniejszą wartość k, dla której granica ta może być osiągnięta i niezbędny nadmiar dla $\mu = 2, 3, 4$ i 5.

	$\mu = 2$	$\mu = 3$	$\mu = 4$	$\mu = 5$
Liczba cyfr binarnych zawierających informację	1	3	5	9
Liczba cyfr kontrolnych /m/	2k	k	$\frac{2k+2}{3}$	$\frac{k+1}{2}$

I. Parametry kodu o małej gęstości

2. Kody resztowe

Kody resztowe są kodami, które mogą być użyte w celu wykrywania i korekty pojedynczych błędów, jakie mogą wystąpić w procesie przetwarzania danych w arytmometrze procesora. Są bardziej użyteczne dla przetwarzania równoległego niż szeregowego. Zasady kodowania dla kodów resztowych są proste. Załóżmy, że dwie liczby i, j mają być przetwarzane w arytmometrze. Przed przetwarzaniem obie liczby mnożone są przez stałą 3. Otrzymuje się odpowiednio wyniki pośrednie 3i i 3j. Po dodaniu tych dwóch liczb wynik końcowy będzie mieć postać $3/i \pm j/$. Będzie on podzielny przez 3 w przypadku gdy nie wystąpi błąd. Załóżmy teraz, że pojedynczy błąd powstanie w l-tej pozycji liczby $3/i \pm j/$ przedstawionej binarnie. Ponieważ 2^l jest niepodzielne przez 3, test "podzielności przez 3" wykaże błąd. Podobny test wykaże także pojedyncze błędy operandów. Inne operacje arytmetyczne również mogą być kontrolowane przy użyciu tej metody.

Jeśli do przedstawienia liczby dziesiętnej i w postaci binarnej potrzeba k cyfr binarnych, to do zakodowania liczby i wymagane będzie co najwyżej k + 2 cyfr binarnych. Nadmiar m tego kodu jest zatem nie większy niż 2 cyfry binarne.

Ten sam kod z innymi mnożnikami może być efektywnie wykorzystany do korekty pojedynczych błędów. Poniższa tabela podaje zestawienie niektórych mnożników, łącznie z maksymalną dopuszczalną liczbą k binarnych cyfr reprezentujących liczby dziesiętne i odpowiadającym im nadmiarem m.

MNOŻNIK	13	19	23	29	37	47	53	59	61	67	71	79	83	101
INFORMACJA /k/	2	4	6	9	12	17	20	23	24	26	28	32	34	42
NADMIAR /m/	4	5	5	5	6	6	6	6	6	7	7	7	7	8

II. Mnożniki kodów resztowych dla korekty pojedynczych błędów

Korekta błędów w dodawaniu i odejmowaniu odbywa się przez podzielenie przetworzonej liczby przez mnożnik. Jeśli resztą jest zero, oznacza to, że nie został popełniony błąd. Pojedynczy błąd na r-tej pozycji zapisu binarnego powoduje powstanie reszty przystającej do $\pm 2^r$ modulo użytego mnożnika. Obliczenie 2^r i dodanie tej liczby do sumy daje poprawny wynik.

Jako przykład rozważmy dodawanie dwóch liczb dziesiętnych 380 i 150. Ponieważ suma tych liczb wyraża się w postaci binarnej 9 bitami mnożnik dla korekty pojedynczego błędu musi być równy 29. Tak więc liczby 11020 i 4350 w postaci dziesiętnej lub 10101100001100 i 01000011111110 w postaci binarnej będą stanowić postać zakodowaną. Jak widać, liczba cyfr nadmiaru wynosi m = 5. Załóżmy teraz, że wystąpi pojedynczy błąd na trzeciej pozycji od lewej strony w sumie przedstawionej binarnie. W wyniku otrzymujemy liczbę o 2048 mniejszą od sumy tych liczb. Będzie to liczba 13322. Natomiast poprawny wynik wynosi 15370. Dzieląc rezultat przez 29 otrzymujemy resztę 11, która modulo 29 przystaje do 2048. Dodając tę liczbę do sumy otrzymujemy poprawny wynik.

Często obwody przetwarzające mają tę własność, że może wystąpić w nich tylko jeden rodzaj błędów na przykład jedynki mogą być zamienione przez zera, lecz nie ma sytuacji przeciwnej tzn. zera nie mogą zostać zamienione na jedynki. W takich przypadkach istnieje możliwość zmniejszenia nadmiaru,gdyż zbiór wszystkich możliwych błędów jest mniejszy, wszystkie bowiem będą tego samego znaku. Przy pozostawieniu tych samych mnożników tabela I może być zmodyfikowana i dostosowana do tej sytuacji, przez przyjęcie maksymalnej liczby cyfr zawierających informację /A-1/:2, gdzie liczba A jest mnożnikiem.

3. Kody asymetryczne

Pewne podzespoły składające się z takich elementów jak rdzenie magnetyczne, diody, niektóre tranzystory mają tę własność, że występujące błędy, są w swej naturze niesymetryczne. Na przykład w rdzeniach magnetycznych zapamiętane jedynki zamieniane zostają na zera. Jedynie wyjątkowo nastąpić może sytuacja odwrotna. Kody umożliwiające detekcję i korektę tego typu błędów są znane pod nazwą kodów asymetrycznych lub kodów korekty jedynek. W ogólności, dowolne kody liniowe mogą być użyte do korekty asymetrycznych błędów. Ze względu na specyficzną naturę tych błędów możliwa jest oszczędność przy budowie sprzętu kodującego i dekodującego oraz zmniejszenie koniecznego nadmiaru. Poniżej omówione zostaną niektóre kody przeznaczone do korekty asymetrycznych błędów.

a. Kody Bergera

Optymalnym kodem asymetrycznym ze względu na najmniejszą liczbę cyfr nadmiaru jest tzw. kod Bergera. Kod ten ma k cyfr informacyjnych powiększonych o $m = 1 + /\log_2 k/$ cyfr nadmiaru. Te ostatnie stanowią binarne przedstawione liczby zer w bloku przesyłanej informacji.Kod może wykrywać dowolną liczbę błędów jedynek jak również dowolną liczbę błędów-zer,pod warunkiem,że błędy - jedynki i błędy - zera nie występują jednocześnie w tym samym bloku. Wykrywanie błędów następuje w ten sposób, że każde zgubienie jedynki w przesyłanej informacji zwiększa liczbę zer, a strata jedynki w cyfrach nadmiarowych zmienia liczbę, którą one reprezentują. Tak więc, każdy rodzaj błędów stwarza niezgodność między cyframi informacji i cyframi nadmiarowymi.

b. Kody o stałej wadze

Cechą charakterystyczną tej klasy kodów jest stała liczba jedynek w każdym słowie kodu. Najbardziej charakterystyczną cechą jest nierozdzielność słów - nie jest mianowicie możliwe podzielenie bloku kodu na cyfry informacyjne i cyfry kontrolne.

Oczywiście wykrywany będzie każdy błąd, który powstał na skutek zamiany jedynki na zero lub zera na jedynkę, przy wykluczeniu możliwości wystąpienia obu tych przypadków jednocześnie. Błąd nie zostanie wykryty w przypadku, gdy liczba zmian jedynek na zera jest równa ilości zmian zer na jedynki. Istnieje nieskończenie wiele kodów stałej wagi. Jeśli długość danego kodu wynosi n, zaś liczba jedynek wynosi m, to kod ma

$$\frac{n!}{m!\,/n-m/!}$$

bloków kodowych.

Przykładem kodu o stałej wadze jest tzw. kod "dwa z pięciu " przedstawiony poniżej. Inne kody używane szeroko to kody "cztery z ośmiu" oraz "dwa z siedmiu". Ten ostatni znalazł zastosowanie w arytmetyce wielu komputerów.

0 1 1 0 0
1 1 0 0 0
1 0 1 0 0
1 0 0 1 0
0 1 0 1 0
0 0 1 1 0
1 0 0 0 1
0 1 0 0 1
0 0 1 0 1
0 0 0 1 1

III. Kod o wadze "dwa z pięciu"

Najbardziej efektywnymi kodami o stałej wadze są kody "[n/2] z n" zawierające n/2 jedynek w każdym bloku kodowym. Są one czasem używane w powiązaniu z pamięcią, celem zabezpieczenia przed błędami obwodów dostępu.

4. Kody z dwuwymiarową kontrolą parzystości

W wielu typach pamięci na przykład na taśmach magnetycznych i w pamięci rdzeniowej, dane przechowywane są w formie dwuwymiarowej tablicy cyfr binarnych. Zabezpieczenie przed błędami może być efektywnie dokonane przez zastosowanie kodów z dwuwymiarową kontrolą parzystości. Takimi kodami mogą być kody liniowe na przykład Hamminga lub BCH, zastosowane zarówno w wierszach, jak i kolumnach tablicy cyfr binarnych. Jeśli d_r jest minimalną odległością ciągu kodowego wiersza i d_c jest minimalną odległością ciągu kodowego kolumny, to minimalna odległość między poszczególnymi tablicami ciągu kodów dwuwymiarowych będzie równa $D = d_r \cdot d_c$. Możliwe jest zatem wykrycie wszystkich kombinacji błędów w ilości D-1 lub mniej względnie korekty wszystkich kombinacji /D-1/:2 błędów w tablicy.

Najprostszym przykładem kodu z dwuwymiarową kontrolą parzystości może być kod stosujący pojedynczą kontrolę parzystości zarówno w wierszach, jak i kolumnach tablicy. Ponieważ daje to kod o minimalnej odległości dwa, tak w wierszach jak i w kolumnach, kod tablicowy ma zatem minimalną odległość cztery i może wykrywać potrójne błędy lub przeprowadzać korektę pojedynczych błędów i wykrywać jednocześnie podwójne błędy.

Kody z dwuwymiarową kontrolą parzystości znajdują głównie zastosowanie w zapisie na taśmie magnetycznej. Informację zapisuje się na taśmie, na kilku ścieżkach równolegle do kierunku ruchu, dzieląc ją na bloki przy pomocy przerw międzyblokowych, jednocześnie na wszystkich ścieżkach. Tak więc każdy blok składa się z prostokątnej tablicy, w której kolumny są ścieżkami, wierszami zaś są grupy cyfr przesuwające się jednocześnie pod głowicami zapisu odczytu. Kodowanie i dekodowanie odbywa się równolegle wzdłuż wierszy i szeregowo wzdłuż kolumn.

Doświadczenie wykazuje, że błędy na taśmie magnetycznej występują zazwyczaj seriami. W wielu przypadkach zatem, zastosowanie kodów korekty błędów seryjnych jest bardziej celowe niż użycie kodów Hamminga czy BCH.

Kody z dwuwymiarową kontrolą parzystości są również szeroko stosowane w transmisji danych. W jednej z wersji, pojedynczy bit parzystości dodawany jest do każdej informacji składającej się z siedmiu bitów. Na końcu każdego bloku informacji transmitowany jest znak parzystości dla każdej kolumny. W odbiorniku cecha parzystości obliczana jest dla każdego pojedynczego bloku. Jeśli rachunek będzie zgodny, odbiornik przesyła o tym informację do nadajnika, zaś nadajnik przesyła wówczas następny blok informacji. Jeśli nie ma odpowiedzi lub otrzymywany sygnał jest niewłaściwy, nadajnik powtarza /retransmituje/ dany blok informacji. W przypadku, gdy kilka retransmisji zawiedzie, operator jest o tym powiadomiony automatycznie.

BIBLIOGRAFIA

1. Armstrong, D.B. "A General Method of Applying Error Corection to Synchronous Digital Systems" Bell Systems Technical Journal, Vol.40, March 1961.

2. Franco, A.G. and L.J.Saporta. "Performance of Random Error Correcting Codes on the Switched Telephone Network." IEEE Transactions on Communication Technology, Vol.COM-15, No.6, December 1967.

3. Hamming, R.W. "Error Detecting and Error Correcting Codes."Bell Systems Technical Journal, Vol.29, 1950.

4. Kautz, W.H. "Codes and Coding Circuitry for Automatic Error Correction Within Digital Systems" In Redundancy Techniques for Computing Systems, W.C. Mann, ed. New York: Spartan Books,1962.

5. Knox-Seith, J.K. "A Redundancy Technique for Improving the Reliability of Digital Systems" Solid State Electronics Laboratory, Stanford University, December 1963./Technical Report No.4816-1/

6. Lynch, W.C. "Reliable Full-Duplex File Transmission Over Half-Duplex Telephone Lines". Communications of the ACM, Vol. 11,No 6, June 1968.

7. Roa, R.N. "Use of Error Checking Codes on Memory Words for Improved Reliability". IEEE Transactions on Reliability. Vol.R-17, June 1968.

8. Townsend, R.L. and R.N. Watts. "Effectiveness of Error Control in Data Communication over the Switched Telephone Network".Bell Systems Technical Journal, Vol.43, November 1964.

9. Tryon, J.G. "Redundant Logic Circuitry". U.S. Patent No.2942193, June 21, 1960.

10. von Neumann, J. "Probabilistic Logic and the Synthesis of Reliable Organisms from Unreliable Components". In Automata Studies by Claude E.Shannon and J.McCarthy. Vol.34, Annals of Mathematics Studies. Princeton: Princeton University,1956.

Cena zł 92.-