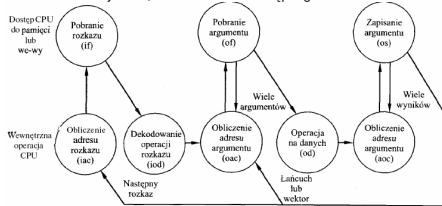


Arytmetyka zmiennopozycyjna

X=xs*B^x
 Y=ys*B^y
 X*Y=(ys*xs)*B^(x+y)
 X:Y=(xs:ys)*B^(x-y)
 X+Y=[xs*B^x(x-y)+ys] *B^y
 X-Y=[xs*B^x(x-y)-ys] *B^y

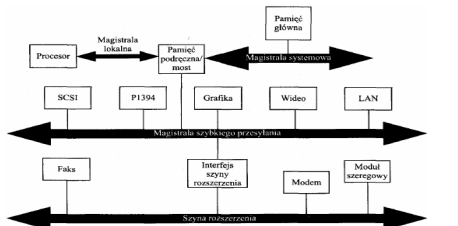
ROZKAZY: *Transfer danych: MOVE- przeniesienie slosa/bloku ze zrodla do celu; STORE- przeniesienie slosa z procesora do pamieci; LOAD- z pamieci do procesora; EXCHANGE- zamiana zawartosci; CLEAR- przeniesienie 000 do celu; SET- przeniesienie 111 do celu; PUSH- slosa na wiezcholek stosu; POP- podniesienie; *Arytmetyczne: ADD- obliczenie sumy; SUBTRACT- roznicy; MULTIPLY- iloczynu; DIVIDE-ilorazu; ABSOLUTE-wartebezwzględna; NEGATE- INCREMENT-++; DECREMENT--1; *Logiczne: AND, OR, NOT, XOR; TEST- badanie warunku; COMPARE; SET CONTROL VARIABLE- ustala elementy sterowania; SHIFT- przesunięcie bitów prawo/lewo; ROTATE 1011-> 1101; *Przeniesienie sterowania: JUMP- przeniesienie bezwarunkowe; JUMP CONDITIONAL- zbadanie warunku i załadowanie adresu zgodnie z warunkiem; JUMP TO SUBROUTINE- umieszczenie informacji kontrolnej bieżącego programu, skok do określonego adresu; RETURN; SKIP- powiększenie licznika rozkazów; HALT; WAIT- oczekiwanie na spełnienie warunku; INPUT- przeniesienie informacji z pamięci głównej do rejestrów procesora; OUTPUT; Start INPUT/OUTPUT - sterowanie modułem I/O; **Elementy rozkazu:** *kod operacji; *odniesienie do argumentów, *odniesienie do wyników; * odniesienie do następnego rozkazu



RISUNEK 9.1. Graf stanów cyklu rozkazu

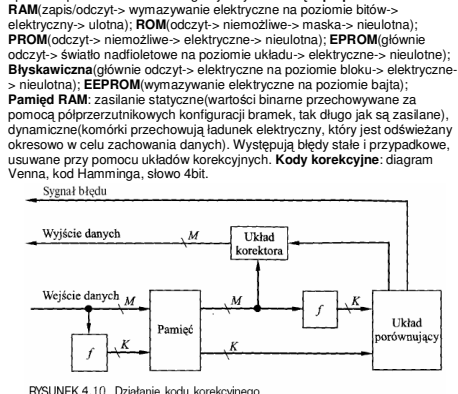
Stos: uporządkowany zestaw elementów. **PUSH**-dodanie, **POP**-usunięcie. **Adresowanie:** Pola adresowe rozkazu SA małe. Tryby adresowania: *natchmiasłowe(kod argumentu); argument obecny w rozkazie, ograniczony wielkością pola adresowego, ***pośrednie**(kod-adres-> pamięć(argument)); ograniczona przestrzeń adresowa; ***pośrednie**(kod-adres-> pamięć(adres-> argument)); ***rejestrów** (kod | IR -> arg) małe pola adresowe; rejestrów pośrednie, z przesunięciem, stosowe. **Obliczanie wyrażenia:** a+b := ab+ a(b'c):=abc+; (a+b)'c:=ab+c' Reguły operacji: jeżeli element jest zmienna umieszczamy go na stosie, jeżeli operatorem, to pobieramy dwa elementy ze stosu, wykonujemy operację, i odkładamy na stos wynik. **Elementy w RISC:** ograniczony i prosty zbiór rozkazów; duża liczba rejestrów roboczych lub zastosowanie kompilatorów do optymalizacji wykorzystania rejestrów; akcent na optymalizację kodu rozkazu.

Architektura von Neumanna: Dane i rozkazy przechowywane w tej samej pamięci umożliwiającej zapis i odczyt; Zawartość tej pamięci może być adresowana przez wskazanie miejsca; Wykonywanie rozkazów następuje w sposób szeregowy rozkaz po rozkazie. MBR- rejestr adresowy pamięci, MBR- rejestr buforowy pamięci, I/O AR, I/O BR. **Działanie komputera:** START-> Pobranie rozkazu-> Wykonanie rozkazu->(STOP lub pobranie). **Model komputera:** rozkazy są 16bit: 0-3bit kod, 4-15bit adres. Liczba całkowita 16bit: 1bit znak, 15bit liczba. **Rejestry procesora:** PC- licznik rozkazu, IR- rejestr rozkazu, AC- akumulator. 0001- laduj AC z pamięci, 0010- zapisz AC w pamięci, 0101- dodaj zawartość pamięci do AC. **Cykl rozkazowy:** pobranie-> wykonanie (z np. ry). **Podstawy potrzebne trzech cykli rozkazowych. A=A+B -> (x)** dostęp CPU do pamięci lub I/O (2,3,8,2) wewnętrzna operacja CPU: 1,2,4,6,7,9. Schemat: 1.Obliczenie adresu rozkazu; 2.Pobranie rozkazu; 3.Dekodowanie operacji rozkazu; 4.Obliczenie adresu argumentu; 5.Pobranie argumentu; 6.Operacja na danych; 7.Obliczanie adresu argumentu; 8.Zapis argumentu; 9.Następny rozkaz. **Przerwanie:** Moduły I/O oraz pamięć mogą przerywać przetwarzanie danych przez procesor. **Klasy przerwań:** 1)Programowe-generowane przez warunek wykonania rozkazu, 2)Zegarowe-generowane przez wewnętrzny zegar komputera, 3)Przerwania I/O-generowane przez sterownik I/O. 4)Uszkodzenia sprzętu-błąd parzystości, defekt zasilania. Schemat przerwania: Start -> Pobierz rozkaz-> Wykonaj-> Sprawdź czy jest przerwanie->(Pobierz rozkaz lub Obsługa przerwania-> Pobierz rozkaz). Sekwencyjne przetwarzanie przerwań, przerwania zagnieźdzone, priorytet przerwania: przerwanie o wyższym priorytecie przerywa program obsługi przerwaniami o niższym. **STRUKTURA POŁĄCZEN:** zbiór ścieżek łączących moduły; **musi umożliwiać przesłanie danych:** z pamięci do CPU i na odwrót, z I/O do CPU i na odwrót, z I/O do pamięci (DMA). **Połączenia magistralowe:** 50-100 linii(magistrala systemowa). Każda linia- 1 funkcja. Schemat: *CPU+ *PAMIĘD+... *PAMIĘD+ *I/O+... *I/O+ Sterowanie. Adres, Dane. Linie danych-szyjna danych składa się z 8, 16 lub 32 linii, z których każda przenosi 1bit informacji. Linie adresowe- szerokość szyjni adresowej determinuje maksymalną pojemność pamięci, adresowane są także urządzenia I/O. Linie sterowania- zapis, odczyt z pamięci; zapis, odczyt do urządzenia I/O; potwierdzenie przerwania; zapotrzebowanie na magistralę(bus request); rezygnacja z magistrali; przywracanie- wszystkie moduły do stanu początkowego. **Działanie magistrali:** uzyskał dostęp do magistrali-> przekazał zapotrzebowanie na dane do modułu 2-> przekazał dane do modułu 2.



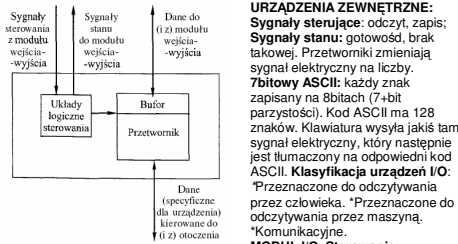
RISUNEK 3.18. Przykładowe konfiguracje magistralowe: stralka; (b) architektura o dużej wydajności

*Rodzaje linii magistrali: specjalistyczne, multipleksowane (do wielu celów, czasowe). *Metody arbitrażu: centralny, rozproszony. *Koordynacja czasowa: (asynchroniczna). *Magistrala PCI-szerokokopasowa magistrala niezależna od procesora, może funkcjonować jako magistrala „miedzypiętrowa” lub peryferyjna. **PAMIĘĆ:** **Położenie:** procesor, pamięć wewnętrzna, zewnętrzna. **Pojemność:** rozmiar slosa, liczba slosów. **Jednostka transferowa:** slosa, blok. **Sposób dostępu:** sekwencyjny, bezpośredni, swobodny, skojarzeniowy. **Wydajność:** czas dostępu, szybkość transferu. **Rodzaj fizyczny:** półprzewodnikowa, magnetyczna. **Własności fizyczne:** nie/uloina, niewymazywalna. **Rodzaj dostępu do pamięci:** *sekwencyjny: rekordy pamięci, dostęp możliwy w określonej sekwencji liniowej (np. pamięć taśmowa). ***bezpośredni:** dostęp do najbliższego odczenia, sekwencyjne poszukiwanie lokalizacji docelowej(np. pamięć dyskowa). ***swobodny:** każda lokalizacja ma unikalny adres, czas dostępu jednokowy dla wszystkich adresów(np. pamięć główna, półprzewodnikowa). ***skojarzeniowy:** w pewnym sensie dostęp swobodny, który umożliwił badanie zgodności wybranych bitów wewnątrz slosa, jednocześnie dla wielu slosów(np. pamięć podręczna). **Parametry miary wydajności:** czas dostępu, czas cyklu pamięci, szybkość przesyłania. **Hierarchia pamięci:** należy zauważyć, że mniejszy czas dostępu, to większy koszt jednego bitu; większa pojemność to mniejszy koszt jednego bitu i dłuższy czas dostępu. **Piramida,** od góry: rejestry, pamięć podręczna, główna, dyskowa podręczna, dyskowa, pamięci taśmowe i optyczne. Pamięć główna dzieli się na dwa rodzaje: półprzewodnikowa i oparta na pierścieniach ferromagnetycznych. **Pamięci półprzewodnikowe:** **RAM**(zapis/odczyt-> wymazywanie elektryczne na poziomie bitów-> elektryczny-> uloina), **ROM**(odczyt-> niemożliwe-> maska-> nieuloina); **EPROM**(odczyt-> niemożliwe-> elektryczne-> nieuloina); **EEPROM**(głównie odczyt-> światło nadfioletowe na poziomie układu-> elektryczne-> nieuloina); **Bliskowizna**(głównie odczyt-> elektryczne na poziomie bloku-> elektryczne-> nieuloina); **EEPROM**(wymazywanie elektryczne na poziomie bajta); **Pamięć RAM:** zasilanie statyczne(wartości binarne przechowywane za pomocą półprzewodnikowych konfiguracji bramek, tak długo jak są zasilane), dynamiczne(komórki przechowują ładunek elektryczny, który jest odfieżywany okresowo w celu zachowania danych). Występują błędy stałe i przypadkowe, usuwane przy pomocy układów korekcyjnych. **Kody korekcyjne:** diagram Venna, kod Hamminga, slos 4bit.



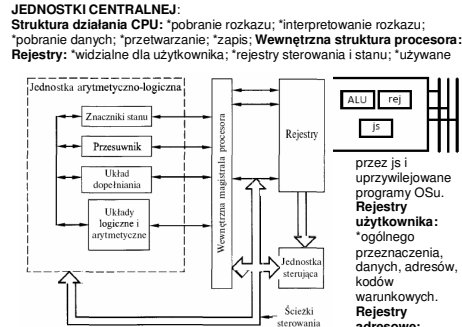
RISUNEK 4.10. Działanie kodu korekcyjnego

Pamięć podręczna: Schemat (CPU-<-> Podręczna-><-> Główna). **Odczyt z pamięci podręcznej:** Schemat (START-> otrzymanie adresu RA z CPU-> Czy blok zawierający RA jest w podręcznej?(TAK-> Pobranie slosa do CPU-> WYKONANIE; NIE-> Dostęp do głównej aby znaleźć ten blok-> Przdzielenie wiersza podręcznej blokowi z RA z głównej-> Ładowanie bloku-> Pobranie slosa do CPU-> WYKONANIE). Ze względu na zjawisko lokalności odniesienia jest bardzo prawdopodobne, że przyszłe odniesienie będzie dotyczyło innych slosów zawartych w tym bloku. Parametry pamięci podręcznych: *pojemności: optymalna 1K-512K slosów. **Rodzaj odziorowywania:** bezpośrednie, skojarzeniowe, sekwycjno-skojarzeniowe. **Algorytm wymian:** najmniej ostatnio używany, pierwszy wszedł-pierwszy wychodzi, najrzadziej używany, swobodny. **Pamięci zewnętrzne:** dysk magnetyczny- to okrągła płyta metalowa/plastikowa pokryta ferromagnetykiem; zapis/odczyt możliwy dzięki cewce przewodzącej-główny; podczas zapisu/odczytu obraca się dysk, nie glowica. Na każdej ścieżce 10-100 sektorów, tyle samo bitów; gęstość bitów/calki wzrasta dla ścieżek wewnętrznych. 1 sektor fizyczny=1 blok danych. **Pamięć RAID:** redundancyjna tablica niezależnych dysków, RAID składa się z 6 poziomów. Wspólne cechy: *RAID- zespoły fizyczne istniejących napędów dyskowych widzianych przez system jako jeden napęd logiczny; *dane są rozproszone pomiędzy fizyczny dyskami; *nadmiarowa przestrzeń dyskowa jest wykorzystana do przechowywania informacji o parzystości. **Poziomy RAID:** ***RAID 0-** dane są rozproszone. Gdy dwa różne żądania I/O dotyczą dwóch różnych bloków danych, jest duże prawdopodobieństwo, że znajdują się one na różnych dyskach, mogą być przetwarzane równolegle. ***RAID 1-** redundancja osiągnięta przez duplikowanie wszystkich danych; każdy pasek logiczny jest umieszczony na dwóch różnych dyskach. Korzyści: wykorzystywany jest do odczytu ten dysk, który ma krótszy czas poszukiwania. Wady: dwukrotnie mniejsza pojemność. ***RAID 2-** metoda równoległego dostępu; wszystkie dyski uczestniczą w realizacji pojedynczego żądania; napędy są tak zsynchronizowane, że każda glowica znajduje się w tej samej pozycji na każdym z dysku; kod korekcji błędów jest obliczany na podstawie bitów na każdym dysku danych(kod Hamminga); bity kodu są przechowywane na osobnych dyskach. ***RAID 3-** tylko jeden dysk redundancyjny niezależnie od wielkości matrycy dysku; obliczany bit parzystości znajduje się na tej samej pozycji dla wszystkich dysków. ***RAID 4-** dostęp niezależny, każdy dysk działa niezależnie, wejścia obsługiwane równolegle, pakowanie danych, ***RAID 5-** duża częstość żądań, duża intensywność odczytów, wyszukiwanie danych. **Pamięci optyczne:** płyta CD. Dane umieszczane są na tych dyskach w segmentach o tych samych rozmiarach. Dane odczytywane są ze stałą prędkością liniową wynikającą ze zmiennej szybkości kątovej. Ścieżki danych ułożone w spiralę, stała prędkość liniowa 1,2 m/s.



RISUNEK 6.2. Urządzenie zewnętrzne

transferem danych z urządzenia peryferyjnego do procesora: Procesor żąda sprawdzenia stanu urządzenia; Moduł I/O udziela odpowiedzi o stanie; Jeśli urządzenie jest gotowe, to procesor zgłasza zapotrzebowanie na przeniesienie danych posługując się rozkazem do modułu I/O; Moduł otrzymuje blok danych z urządzenia zewnętrznego; Dane przenoszone są do procesora. **Struktura modułu I/O:** **Metody wykonywania operacji I/O:** *procesor wykonuje program sterujący bezpośrednio operacją I/O; *operacja I/O sterowana przerwaniem (wykonywanie innych rozkazów); *bezpośredni dostęp do pamięci(DMA). **Bezpośredni dostęp do pamięci(DMA):** Rozkaz odczytu zawiera informacji: *odczyt/zapis; *adres urządzenia I/O; *adres początkowej komórki pamięci; *liczba slosów do odczytu/zapisu; *moduł DMA przenosi blok danych bezpośrednio do/z pamięci; *gdy proces zakończony, DMA wysyła sygnał przerwania do procesora; *DMA przyjmuje kontrolę nad magistralą, gdy procesor jej nie potrzebuje lub DMA potrzebuje chwilowego zawieszenia operacji procesora. **Interfejs zewnętrzne:** *szeregowy *równoległy. **STRUKTURA I DZIAŁANIE**



przez js i uprzywilejowane programy OSu. **Rejestry użytkownika:** *ogólne przeznaczenia, danych, adresów, kodów warunkowych. **Rejestry adresowe:** wskaźnik segmentu, rejestry indeksowe, wskaźnik stosu(wieżchołek). **Rejestry kodów warunkowych:** informacja o tym, czy wynik dodawania jest +, -, używane gdy mamy rozgałęzienie. **Rejestry sterowania i stanu:** licznik programu-adres rozkazu przewidzianego do pobrania; *rejestr rozkazu(IR), zawiera ostatnio pobrany rozkaz; *rejestry adresowe pamięci(MBR), zawiera adres pamięci z/do którego mamy odczyt/ładowanie; *rejestr- slos adresu programu(PSW) - zawiera następujące pola: znak-bit znaku ostatniej operacji arytmetycznej, zero-ustawiamy gdy wynik jest 0, bit przeniesienia, równość, pełnienie, zezwolenie/blokowanie.