

Halina NOWAK
Piotr STRÓŻYNA

REALIZACJA STEROWNIKA TWARDYCH DYSKÓW Z WYKORZYSTANIEM MIKROPROCESORÓW SEGMENTOWYCH

Streszczenie. W artykule zostało przedstawione rozwiązanie sterownika twardego dysków dla systemu ComPAN-8. Na wstępie krótko scharakteryzowano pamięć dyskową MERA-9450 i przeanalizowano możliwości wykorzystania różnych typów mikroprocesorów do realizacji sterownika. W dalszej części zaprezentowano szczegółowo rozwiązanie wykorzystujące bipolarne mikroprocesory segmentowe Intel 3000. Na zakończenie zwrócono uwagę na możliwości modyfikacji i zwiększenia funkcji spełnianych przez sterownik.

1. Wprowadzenie

Wzrastająca liczba produkowanych na świecie mikroprocesorów powoduje rozszerzenie dziedzin ich zastosowań. Rosną również wymagania stawiane systemom mikrokomputerowym, dotyczące ich możliwości programowych. Systemy operacyjne, translatory, biblioteki i programy użytkowników wymagają pamięci zewnętrznych o dużej pojemności, tzn. pamięci dyskowych.

W Polsce produkowana jest obecnie pamięć dyskowa MERA-9450. Problemy związane z podłączeniem tej pamięci do systemu mikrokomputerowego ComPAN-8 [2] stanowią treść niniejszego artykułu.

2. Założenia

MERA-9450 jest kasetową pamięcią dyskową o pojemności 5 lub 10 bajtów (o pojedynczej lub podwójnej liczbie ścieżek) [1]. Jednostka pamięci jest wyposażona w dysk stały oraz wymienną kasetę. Zbiór linii interfejsu oraz podstawowe dane techniczne zawierają tablice 1.1 i 1.2.

Zadaniem sterownika jest zapewnienie dwukierunkowej transmisji bloków danych pomiędzy mikrokomputerem ComPAN-8 i jedną lub większą liczbą jednostek pamięci. Transmisja bloku danych między ComPAN-em a sterownikiem powinna przebiegać z maksymalną szybkością, tzn. w trybie DMA. Długość przesyłanych bloków, określona przez system operacyjny CP/M, powinna wynosić 128 bajtów ewentualnie wielokrotność (256 lub 512). Sterownik powinien umożliwiać kontrolę poprawności transmisji, ochronę przed zapisem oraz sformatowanie dysków.

Tablica 1.1

Sygnały interfejsu pamięci

Nazwa sygnału	Znaczenie
UNIT SELECT 1 UNIT SELECT 4	Wybór jednostki pamięci
TA 0 : TA 8	adres ścieżki
STROBE	strob adresu ścieżki
RESTORE	powrót do ścieżki zerowej
HEAD SELECT	wybór głowicy
DISC SELECT	wybór dysku
WRITE DATA	linia zapisu zegara i danych
WRITE GATE	bramka zapisu
ERASE GATE	bramka kasowania
READ GATE	bramka odczytu
WRITE PROTECT	ochrona zapisu
WRITE PROTECT STATUS	zapis chroniony
DENSITY	podwójna liczba ścieżek
ON CYLINDER	głowica nad żądaną ścieżką
READ DATA	linia odczytu danych
READ CLOCK	linia odczytu zegara
INDEX	znacznik początku ścieżki
SECTOR	znacznik początku sektora
SECTOR ADD 0 : SECTOR ADD 4	adres sektora
SEEK ERROR	błąd szukania
FAULT	błędny stan pamięci
UNIT READY	gotowość pamięci
ATTENTION 1 : ATTENTION 4	koniec operacji szukania lub błąd szuka- nia
ADD ACCEPTED	adres przyjęty
ADD INVALID	adres nie przyjęty

Tablica 1.2

Podstawowe dane techniczne pamięci

Sredni czas dostępu do ścieżki	35 ms
System zapisu	podwójna częstotliwość
Częstotliwość transmisji	2.5 MHz
Liczba ścieżek/cylinder	4
Liczba cylindrów	200 lub 400
Liczba sektorów	32,24,16,12
Liczba jednostek pamięci w systemie	max. 4
Liczba bitów/ścieżkę	62.500
Liczba bitów/pamięć	50.000.000 lub 100.000.000
Prędkość obrotowa dysku	2400 obr/min

2. Koncepcje rozwiązań

Można w zasadzie wyróżnić 2 rozwiązania:

- wykorzystujące elementy MSI i SSI,
- wykorzystujące mikroprocesor.

Wady pierwszego z nich są oczywiste:

a) duża liczba elementów (około 250) i z nią związane:

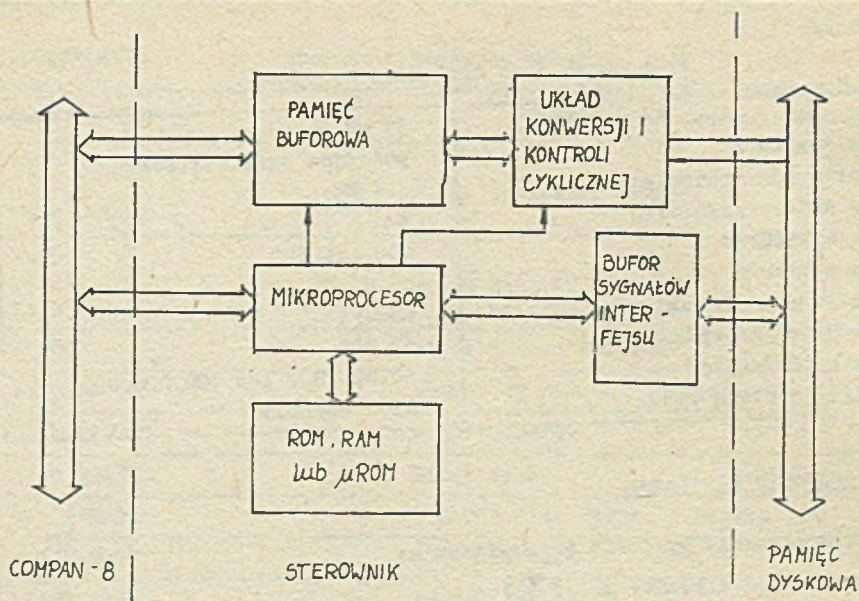
- wysoki koszt,
- duży pobór mocy,
- konieczność realizacji sterownika na 3 lub 4 pakietach,
- niska niezawodność,
- kłopotliwe uruchamianie,

b) sztywność konstrukcji, utrudniająca modyfikacje związane np. ze zmianami wprowadzonymi przez producenta pamięci.

Przykład rozwiązania tego typu zawarty jest w [5]. Na rozwiązanie drugie ma wpływ typ zastosowanego mikroprocesora. Obecnie dostępne są na rynku krajowym mikroprocesory 8-bitowe produkcji polskiej i NRD (odpowiedniki Intel 8080 i Z-80) oraz bipolarne mikroprocesory segmentowe produkcji ZSRR (odpowiedniki serii Intel 3000).

W strukturze sterownika niezależnej jeszcze od typu mikroprocesora można wyróżnić następujące bloki (rys. 2.1):

- pamięć buforową, przechowującą transmitowane bloki danych,
- układy konwersji szeregowo-równoległej i równoległo-szeregowej oraz generacji i sprawdzania słów kontroli cyklicznej,
- bufony sygnałów interfejsu pamięci dyskowej,
- mikroprocesor z pamięciami ROM i RAM lub μ ROM.



Rys. 2.1. Ogólne struktura sterownika

Fig. 2.1. General structure of the controller

orowanie przesyłanych bloków stało się konieczne ze względu na wymagany s transmisji 1 bajtu do/z pamięci dyskowej ($3,2 \mu\text{s}$) i niezależność syg-ów zegarowych ComPAN-a (sterownika) i pamięci.

Komunikacja sterownika z mikrokomputerem została zatem rozdzielona na tory:

- or danych, wykorzystujący jeden z kanałów DMA mikrokomputera,
- or sterowania/statusu, stanowiący dla mikrokomputera 8-bitowy port.

la zapewnienia dużej wiarygodności danych niezbędne jest zastosowanie icznej kontroli poprawności (CRC). Wymagana szybkość transmisji unie- iwia wykonanie konwersji szeregowo-równoległej i równoległo-szerego- oraz generowanie i sprawdzanie słów kontrolnych przez mikroprocesor wno 8-bitowy (ze względu na czas realizacji rozkazu), jak i segmentowy wględu na ograniczone możliwości przetwarzania danych) ani też wyko- tanie do tego celu innych specjalizowanych elementów LSI, np. Z-80 realizującego konwersję i kontrolę cykliczną dla szybkości transmis- do 800 KB/s. Tak więc do spełnienia tych funkcji należy wykorzystać enty MSI i SSI.

przypadku wykorzystania mikroprocesorów 8-bitowych, np. Z-80, o prze- nym czasie wykonywania instrukcji powyżej $1,6 \mu\text{s}$, do transmisji danych z ComPAN-em a pamięcią buforową należałoby zastosować trudno dostępny ent Z-80 DMA (8257 nie umożliwia transmisji między dwiema pamięciami) licznikowy układ generacji adresu pamięci buforowej (wówczas można

byłoby wykorzystać jeden z istniejących kanałów DMA mikrokomputera). Podobnie w torze transmisji pamięć buforowa - dysk należałoby wykorzystać element 8257 lub licznikowy układ adresowania. Zadaniem mikroprocesora byłoby więc zainicjowanie obu transmisji oraz sterowanie pozostałymi liniami interfejsu pamięci.

Układ taki zawierałby około 60 elementów, głównie LSI i MSI. Przyspieszenie transmisji, wynikające z redukcji czasów inicjowań, można byłoby osiągnąć np. przez odczyt kilku kolejnych sektorów lub też przez osobne buforowanie zapisywanych i odczytywanych bloków, co spowodowałoby jednak znaczny wzrost liczby elementów.

Wykorzystanie bipolarnych mikroprocesorów segmentowych umożliwiłoby uproszczenie układu transmisji danych. Czas wykonania mikroinstrukcji rzędu 200 ns oraz funkcje procesora 3002 pozwalają na generację adresu pamięci buforowej podczas transmisji między pamięcią a układem konwersji lub pamięcią COMPAN-a, co umożliwia z kolei wykorzystanie kanału DMA mikrokomputera.

Funkcje spełnione przez sterownik nie wymagają skomplikowanego przetwarzania danych, zatem zestaw operacji procesorów 3002 wydaje się wystarczający dla takiego zastosowania. Za wykorzystaniem mikroprocesorów segmentowych przemawiają więc: szybkość, niski koszt, niezawodność i dobre dostosowanie możliwości mikroprocesora w sensie listy instrukcji do realizacji tego zadania.

Wadą natomiast jest brak narzędzi wspomagających tworzenie oprogramowania, a więc żmudny proces pisania mikroprogramu i rozmieszczania go w pamięci.

Biorąc pod uwagę zalety bipolarnych mikroprocesorów oraz przewidziany rozmiar mikroprogramu (poniżej 500 mikroinstrukcji), korzystniejsze wydaje się zastosowanie mikroprocesorów segmentowych.

3. Elementarne operacje dyskowe

Wymiana informacji między sterownikiem a mikrokomputerem odbywa się dwoma torami: torom danych przekazywane są w obu kierunkach bloki danych o żądanej długości, torom sterowania przesyłane są do mikrokomputera słowa statusu oraz do sterownika słowa sterujące, inicjujące wykonanie określonych operacji. Przyjęte założenia spowodowały wyodrębnienie 11 elementarnych operacji dyskowych realizowanych przez sterownik:

- | | |
|------------------|---------------------------------|
| 1) RESTORE | powrót do ścieżki zerowej, |
| 2) SEEK | szukanie ścieżki (cylindra), |
| 3) READ ADDRESS | odczyt pola adresowego sektora, |
| 4) WRITE ADDRESS | zapis pola adresowego sektora, |
| 5) READ DATA | odczyt pola danych sektora, |

- | | |
|---------------------|---|
| 6) WRITE DATA | zapis pola danych sektora, |
| 7) SELECT | wybór jednostki, dysku i głowicy, |
| 8) STATUS | odczyt słów statusu sterownika i dysku, |
| 9) MODE | ustalenie trybu transmisji, |
| 10) SET-W, PROTECT | ochrona zapisu, |
| 11) CLEAR-W-PROTECT | kasowanie ochrony zapisu. |

Zainicjowanie ich wymaga wysłania od 1 do 5 słów sterujących-kodów operacji i ich parametrów (rys.3.1).

Operacje SELECT, RESTORE, SEEK, SET-W-PROTECT i CLEAR-W-PROTECT bezpośrednio ustalają stan odpowiednich linii interfejsu pamięci dyskowej. Wykonanie operacji RESTORE i SEEK zakończone jest wysłaniem przez sterownik do mikrokomputera słowa statusu pamięci dyskowej, zawierającego informację o stanie wybranych linii interfejsu pamięci (por. tabl. 1.1).

Operacja MODE USTALA tryb transmisji:

- | | |
|--------------------------------|---|
| -L ₁ L ₀ | - długość bloku (128, 256, 512 bajtów), |
| -RDCRC | - przerwanie operacji odczytu danych w przypadku błędu CRC pola adresu, |
| -RDAF | - przerwanie operacji odczytu danych w przypadku błędu pola adresu (tzn. niezgodności z fizycznym numerem sektora, ścieżki, głowicy lub jednostki), |
| -WRCRC | - przerwanie operacji zapisu danych w przypadku błędu CRC pola adresu, |
| -WRAP | - przerwanie operacji zapisu danych w przypadku błędu pola adresu (tzn. niezgodności z fizycznym numerem sektora, ścieżki, głowicy lub jednostki), |
| -CHECK | - przerwanie operacji zapisu pola adresu w przypadku błędu pola adresu. |

W wyniku wykonania operacji STATUS sterownik wysyła (torem sterowania) w 1-6 kolejnych bajtach pełną informację o stanie pamięci dyskowej i sterownika. Bajt 2 zawiera informację o błędach, które mogą wystąpić w czasie wykonywania operacji zapisu lub odczytu:

- | | |
|-----------|--|
| -SECT ERR | - nieznanie sektora, |
| -SYNC ERR | - brak synchronizacji (ścieżka niesformatowana), |
| -CRC ADD | - błąd CRC pola adresu, |
| -CRC DAT | - błąd CRC pola danych, |
| -ADD ERR | - niezgodność pola adresu z numerem sektora, ścieżki, głowicy, dysku lub jednostki, |
| -ADD INV | - adres ścieżki nieprawidłowy (odpowiada linii ADD INVALID pamięci dyskowej), |
| -FAIL | - operacja przerwana z powodu wystąpienia błędów SECT ERR, SYNC ERR, ADD INV oraz CRC ADD lub ADD ERR przy określonym trybie transmisji. |

SELECT

KOD				X	X	X	X
NUMER JEDNOSTKI	X ⁵⁾	X ⁵⁾	X ⁵⁾	NUMER DYSKU	NUMER GŁOWICY	X	

RESTORE

KOD				X	X	X	X
-----	--	--	--	---	---	---	---

status pamięci

0	0	0	ON CYLINDER	SEEK ERROR	FAULT	UNIT READY	0
---	---	---	-------------	------------	-------	------------	---

SEEK

KOD				X	X	X	NUMER ŚCIEŻKI
-----	--	--	--	---	---	---	---------------

NUMER ŚCIEŻKI							
---------------	--	--	--	--	--	--	--

status pamięci

0	0	0	ON CYLINDER	SEEK ERROR	FAULT	UNIT READY	0
---	---	---	-------------	------------	-------	------------	---

SET - W - PROTECT

KOD				X	X	X	X
-----	--	--	--	---	---	---	---

CLEAR - W - PROTECT

KOD				X	X	X	X
-----	--	--	--	---	---	---	---

MODE

KOD				X	X	X	X
-----	--	--	--	---	---	---	---

L ₁	L ₀	X	RDCRC	RDAF	WRCRC	WRAF	CHECK
----------------	----------------	---	-------	------	-------	------	-------

Rys. 3.1. Formaty operacji dyskowych

Fig. 3.1. Formats of disc operations

STATUS

	KOD			ILOŚĆ ZADANYCH BAJTÓW STATUSU				
status pamięci	0	0	0	ON CYLINDER	SEEK ERROR	FAULT	UNIT READY	WRITE PROTECT
status sterownika	1	SECT ERR	SYNC ERR	CRC ADD	CRC DAT	ADD ERR	ADD INV	FAIL
tryb	L _i	L _o	0	RDCRC	RDAF	WRRCRC	WRRAF	CHECK
	NUMER JEDNOSTKI	0 ⁵⁾	0 ³⁾	0 ⁵⁾	NUMER DYSKU	NUMER GŁOWICY	NUMER ŚCIEŻKI	
NUMER ŚCIEŻKI								
0	0	0	NUMER SEKTORA					

READ DATA

KOD							
X	X	X	NUMER SEKTORA				

status sterownika - operacja zakończona	1	0	0	CRC 1) ADD	CRC DAT	ADD 2) ERR	0	0
operacja przerwana	status pamięci	0	0	0	ON CYLINDER	0	FAULT	UNIT READY
		status sterownika	1	SECT ERR	SYNC ERR	CRC 3) ADD	0	ADD 4) ERR
								0
								1

WRITE DATA

KOD			X	X	X	X
-----	--	--	---	---	---	---

X	X	X	NUMER SEKTORA			
---	---	---	---------------	--	--	--

status sterownika
-operacja zakończona

1	0	0	CRC 1) ADD	0	ADD 2) ERR	0	0
---	---	---	---------------	---	---------------	---	---

operacja
przerwana

status
pamięci
status
sterownika

0	0	0	ON CYLINDER	0	FAULT	UNIT READY	0
---	---	---	----------------	---	-------	---------------	---

1	SECT ERR	SYNC ERR	CRC 3) ADD	0	ADD 4) ERR	0	1
---	-------------	-------------	---------------	---	---------------	---	---

READ ADDRESS

KOD			X	X	X	X
-----	--	--	---	---	---	---

X	X	X	NUMER SEKTORA			
---	---	---	---------------	--	--	--

status sterownika
operacja zakończona

1	0	0	CRC ADD	0	ADD ERR	0	0
---	---	---	------------	---	------------	---	---

pole adresu

SŁOWO STATUSU SEKTORA							
-----------------------	--	--	--	--	--	--	--

NUMER JEDNOSTKI	0 ⁵⁾	0 ⁵⁾	0 ⁵⁾	NUMER DYSKU	NUMER ŚLONICY	NUMER ŚCIEŻKI
--------------------	-----------------	-----------------	-----------------	----------------	------------------	------------------

NUMER ŚCIEŻKI							
---------------	--	--	--	--	--	--	--

0	0	0	NUMER SEKTORA			
---	---	---	---------------	--	--	--

operacja
przerwana

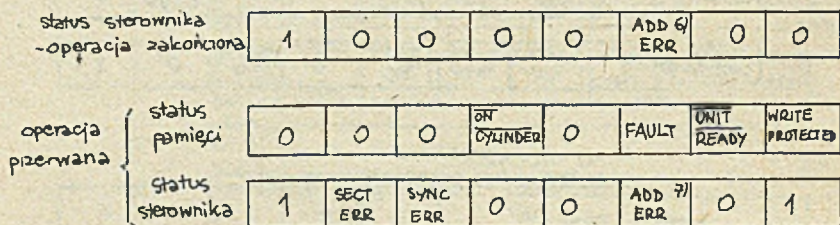
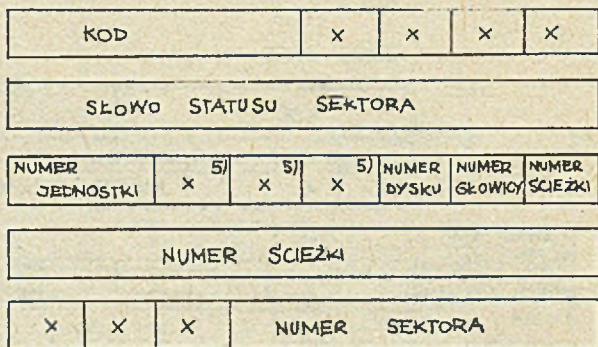
status
pamięci
status
sterownika

0	0	0	ON CYLINDER	0	FAULT	UNIT READY	0
---	---	---	----------------	---	-------	---------------	---

1	SECT ERR	SYNC ERR	0	0	0	0	1
---	-------------	-------------	---	---	---	---	---

Rys. 3.1. od.

WRITE ADDRESS



Rys. 3.1. cd.

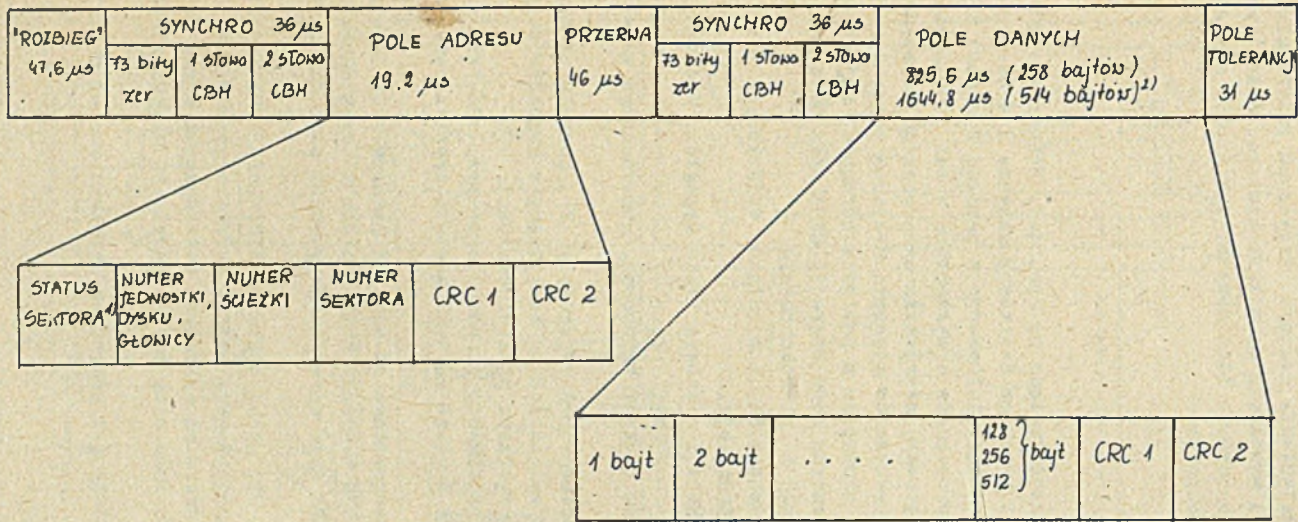
Format sektora, przedstawiony na rys. 3.2, rozszerzono w stosunku do zaleceń zawartych w [1] o dwa słowa synchronizacji, poprzedzające pola adresu i danych, co zwiększyło wiarygodność odczytywanej informacji. Parametry operacji zapisu i odczytu z dysku (tzn. numery jednostki, dysku, głowicy i ścieżki) powinny być zgodne z parametrami ostatnio wykonanych operacji SELECT i SEEK.

Podczas operacji zapisu danych (WRITE-DATA) wysłanie słów sterujących poprzedza transmisję torem danych bloku danych do pamięci buforowej. Po odzyskaniu wybranego sektora oraz odczytaniu i sprawdzeniu jego pola adresu dane przepisywane są na dysk. Operację kończy wysłanie do mikrokomputera słowa statusu sterownika lub pamięci dyskowej.

Podczas operacji odczytu danych (READ DATA) po odzyskaniu wybranego sektora i kontroli jego pola adresu dane przepisywane są z sektora do pamięci buforowej, a następnie po wysłaniu słowa statusu transmitowane są torem danych do mikrokomputera.

W przypadku wystąpienia błędu pola adresu obie powyższe operacje jak również operacja WRITE ADDRESS mogą zostać przerwane (por. operacja MODE).

Operacje WRITE ADDRESS i READ ADDRESS wykonywane są w czasie formatowania i testowania dysku. Zawartość pola adresu w przypadku bezbłędnego zakończenia operacji READ ADDRESS jest również przesyłana torem sterowania.



- 1) dla użytku programisty
- 2) dla dysku 12 - sektorowego

Ryb. 3.2. Format sektora
Fig. 3.2. Format of a sector

Bezwzględne przerwanie operacji zapisu lub odczytu następuje w przypadku błędu pamięci dyskowej, naruszenia ochrony zapisu (dla operacji zapisu), braku synchronizacji lub nieznaalezienia sektora. Przyczyna przerwania określona jest w słowie statusu wysykanym do mikrokomputera. Jeżeli przerwana zostanie operacja odczytu zarówno blok danych, jak i pole adresu nie są transmitowane do mikrokomputera.

4. Struktura sterownika

Struktura sterownika jest przedstawiona na rys. 4.1. Sekwenter 3001 wraz z pamięcią mikroprogramu i procesor 8-bitowy, złożony z 4 elementów 3002, stanowią system, przetwarzający informacje z pamięci dyskowej i z mikrokomputera. Wykorzystanie procesorów i sekwentera serii 3000 narzuca pewien podział magistrali wewnętrznej sterownika na poszczególne szyny.

Połączone magistrale D i I procesorów stanowią 8-bitową szynę danych, przesyłanych między rejestrami procesora a pamięcią buforową, rejestrem równoległo-szeregowym RRS, rejestrem kodu operacji RKOE oraz buforami wyjściowymi dla interfejsów dysku i mikrokomputera.

Magistralą M przesyłane są dane z RRS i z buforów wejściowych interfejsów dysku i mikrokomputera do procesora.

50-bitową szynę mikroinstrukcji I stanowią linie sygnałów, omówionych szczegółowo w [7].

Szynę A stanowią linie adresu pamięci buforowej, generowanego przez procesor.

Układ modyfikacji adresu, szczegółowo omówiony w [7], pozwala na określenie adresu mikroinstrukcji w zależności od wewnętrznych sygnałów sterownika oraz wybranych sygnałów interfejsu pamięci dyskowej.

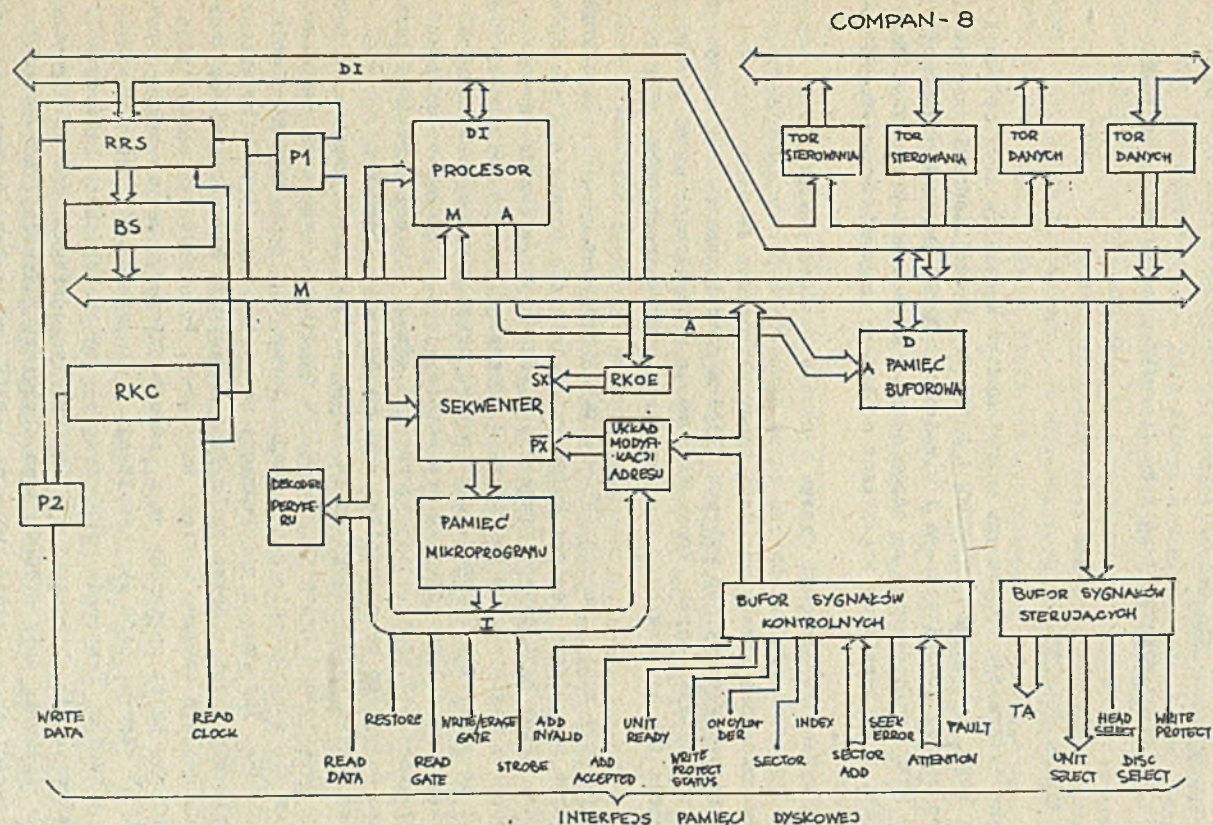
Konwersja danych z postaci szeregowej na równoległą przy odczycie i z równoległej na szeregową przy zapisie na dysk została zrealizowana w rejestrze przesuwym RRS.

Wyjścia równoległe rejestru odseparowano od szyny M buforem trójstanowym BS. Układ kontroli cyklicznej RKC generuje przy zapisie na dysk słowa CRC zgodnie ze standardem CCIITT oraz sprawdza poprawność słów CRC odczytanych z dysku.

Rejestr RRS, przełącznice P1 i P2, układ kontroli RKC, bufor BS, rejestry sygnałów sterujących i kontrolnych dysku, pamięć buforowa oraz bufor torów sterowania i danych sterowane są sygnałami generowanymi przez dekodery strobów peryferii na podstawie pola adresu peryferii mikroinstrukcji.

Część rejestrów roboczych RO-R9 i T procesora wykorzystano do przechowania:

- kodu operacji dyskowej (CODE),
- numeru jednostki, dysku i głowicy (HEAD),



Rys. 4.1. Sterownik - schemat blokowy

Fig. 4.1. Controller - block diagram

- numeru ścieżki (CYLINDER),
- numeru sektora (SECTOR),
- adresu pamięci buforowej (BUFADR),
- słowa statusu sterownika zawierającego flagi błędów (STATUS),
- trybu transmisji (MODE).

Pozostałe rejestry wykorzystano jako liczniki i rejestry pomocnicze.

5. Przepływ informacji

Wymianę informacji rozpoczyna mikrokomputer wysyłając kody operacji SELECT i MODE, a następnie RESTORE i SEEK. Kod każdej operacji zapisywany jest w rejestrze CODE procesora i w rejestrze RKOE. Parametry operacji SELECT i SEEK zapamiętywane są odpowiednio w rejestrach procesora: HEAD i CYLINDER, a następnie są przesyłane do bufora sygnałów sterujących pamięci dyskowej.

Jeżeli wykonanie operacji wymaga wysłania statusu pamięci, stan odpowiednich linii interfejsu poprzez bufor sygnałów kontrolnych i jeden z rejestrów procesora przesyłany jest torem sterowania do mikrokomputera.

Po ustaleniu głowicy nad wybraną ścieżką mikrokomputer może rozpocząć operacje transmisji danych. Torem sterowania przekazywane są do procesora kod operacji i numer sektora, zapamiętywane w rejestrach CODE, RKOE i SECTOR.

Przy operacji zapisu danych (WRITE, DATA) procesor ustala w rejestrze BUFADR adres początkowy pamięci buforowej i inicjuje transmisję danych w kanale DMA. Każdy bajt danych przesyłany jest z rejestru wejściowego toru danych poprzez magistralę M do rejestru procesora, a następnie przez magistralę DI do pamięci buforowej, adresowanej przez zawartość rejestru BUFADR, zwiększaną o 1 po kolejnym przesłaniu (jeżeli transmitowany jest blok o długości 512 bajtów, najstarszy bit adresu pamięci określony jest przez stan odrębnego przerzutnika, sterowanego mikroprogramem).

Po przepisaniu całego bloku linie adresu sektora porównywane są w procesorze z zawartością rejestru SECTOR. Po napotkaniu wybranego sektora rozpoczyna się odczytywanie szeregowej informacji z pola SYNCHRO i pola adresu sektora. Informacja przetworzona na postać równoległą w rejestrze RRS przepisywana jest poprzez bufor BS i magistralę M do procesora, a następnie do wolnego obszaru pamięci buforowej. Po odczytaniu całego pola adresu kolejne bajty pobierane są z pamięci buforowej do procesora i porównywane ze wzorami słów synchronizacji i zawartościami rejestrów HEAD, CYLINDER i SECTOR. Następnie rozpoczyna się zapis kolejnych bitów zer i słów synchronizacji, utworzonych w procesorze i przepisanych do rejestru RRS, a po zapisaniu pola SYNCHRO dane z pamięci buforowej, adresowanej przez uaktualnioną zawartość rejestru BUFADR, przesyłane są poprzez rejestr procesora do rejestru RRS i zapisywane w polu danych sektora. Równocześnie

w układzie RKC tworzone są dwa słowa CRC, zapisywane na końcu pola danych. Wysłanie z rejestru STATUS procesora do mikrokomputera słowa statusu sterownika kończy operację WRITE DATA.

Operacja READ DATA rozpoczyna się poszukiwaniem wybranego sektora, odczytaniem i sprawdzeniem jego pola adresu w taki sposób, jak w operacji WRITE DATA. Następnie odczytywane jest kolejne pole SYNCHRO i pole danych. Kolejne bajty danych przepisywane są z rejestru RRS poprzez procesor do pamięci buforowej. Po odczytaniu całego pola danych w rejestrze BUFADR ustalony jest adres początkowy i procesor inicjuje transmisję w torze danych. Po transmisji całego bloku w torze sterowania wysyłany jest status sterownika.

Przebieg informacji podczas wykonywania operacji READ ADDRESS i WRITE ADDRESS jest zbliżony do wyżej omówionego, z tą różnicą, że zamiast bloku danych między sterownikiem a mikrokomputerem transmitowane są w torze sterowania cztery bajty pola adresu.

6. Podsumowanie

Przyjęte rozwiązanie sterownika wymaga wykorzystania oprócz sekwentera, procesorów i pamięci mikroprogramu 47 elementów SSI i MSI i mieści się na pojedynczym pakiecie systemu COMPAN-8.

Układy sterownika taktowane są autonomicznym zegarem o częstotliwości 2.5 MHz. Operacja przesłania bajtu między pamięcią buforową a rejestrami toru danych oraz testowanie końca bloku wymaga 4 mikroinstrukcji (1.6 μ s), zatem szybkość transmisji danych między COMPAN-em a sterownikiem jest ograniczona szybkością kanału DMA mikrokomputera.

Zastosowanie kontroli cyklicznej umożliwiło wykrycie 99.9985% błędów oraz wszystkich błędów, jeżeli ich liczba jest nieparzysta [5].

Przyjęty zestaw operacji elementarnych pozwala na współpracę z 4 jednostkami pamięci o interfejsach MERA-SYSTEM lub SM EMC. Jednostki produkowane obecnie posiadają 2 talerze, jednak mikroprogram i format operacji pozwalają na współpracę z pamięciami 11-talerzowymi (także o podwójnej liczbie ścieżek), które mają być w niedalekiej przyszłości produkowane przez Fabrykę Mierników i Komputerów "ERA". Modyfikacja układowa polega na umieszczeniu dodatkowego bufora dla linii adresu ścieżki TAB i 3 linii wyboru dysku.

Mikroprogramowa realizacja operacji dyskowych pozwala na elastyczne zwiększenie i modyfikację funkcji sterownika poprzez wymianę pamięci mikroprogramu. Możliwa np. byłaby realizacja transmisji wielosektorowej (zastosowano pamięć buforową o pojemności 1K) zależnie od wymagań systemu operacyjnego. Modyfikując mikroprogram można również zapisać konwersję adresów logicznych na fizyczne.

Celowe także wydaje się powiększenie zestawu elementarnych operacji o operacje samotestowania sterownika i dysku, określania reakcji sterownika

na różne typy błędów (np. wielokrotna próba zapisu/odczytu w przypadku błędów CRC). Ostatnie modyfikacje wymagałyby najprawdopodobniej zwiększenia pamięci mikroprogramu, ponieważ obecnie jest już wykorzystanych około 75% komórek [7]. Metody zwiększania pojemności pamięci mikroprogramu zostały przedstawione w [3].

Istotnym ułatwieniem na etapie uruchamiania jest zastąpienie pamięci stałej mikroprogramu przez szybkie pamięci EPROM 2732A (o czasie dostępu około 200 ns), co zostało wykorzystane przez autorów w trakcie uruchamiania modelu sterownika twardych dysków w systemie ComPAN-8.

Prosty interfejs sterownika, "widzianego" przez mikrokomputer jako porty: programowy i DMA, pozwala na jego implementację także w innych systemach mikrokomputerowych.

LITERATURA

- [1] Dokumentacja techniczno-ruchowa pamięci dyskowej MERA-9450.
- [2] Dokumentacja techniczno-ruchowa systemu RTDS-8. Zakład Systemów Automatyki Kompleksowej PAN, Gliwice 1983.
- [3] Series 3000 bipolar microcomputer system - Intel Corporation 1976.
- [4] Rzymkowski K.: Mikroprocesory, ROJNTBEEA, Warszawa 1979.
- [5] Dokumentacja techniczno-ruchowa modułu pamięci zewnętrznych MPZ-400. Centrum Naukowo-Produkcyjne Technik Komputerowych i Pomiarów, Warszawa 1979.
- [6] Cruce A.C., Alexander S.A.: Building a Hard Disc Interface for an S-100 Bus System - BYTE, March 1983.
- [7] Nowak H., Stróżyńska P.: Mikroprogramowanie segmentowych mikroprocesorów Intel 3000 na przykładzie sterownika twardych dysków. Zeszyty Naukowe Politechniki Śląskiej s. Informatyka, z. 8, Gliwice 1986.

Recenzent: dr inż. Jerzy Jaworowski

Wpłynęło do Redakcji: 14.06.1985.

РЕАЛИЗАЦИЯ КОНТРОЛЛЕРА ДИСКОВОЙ ПАМЯТИ ИСПОЛЬЗУЮЩАЯ СЕГМЕНТНЫЕ МИКРОПРОЦЕССОРЫ

Резюме

В статье представлено конструктивное решение контроллера дисковой памяти Мера 9450 для системы ComPAN - 8.

В одной части дано краткое описание дисковой памяти и проведен анализ возможности использования различных типов микропроцессоров для реализации контроллера. Затем детально представлено решение, использующее сегментные

микроспроцессоры. В завершительной части статьи обращено внимание на модификацию и расширение функции контроллера.

A HARD DISC CONTROLLER BASED ON BIT-SLICED MICROPROCESSORS

S u m m a r y

The paper presents the solution of a hard disc controller for the system ComPAN-8. First short characteristics of the disc memory Mera 9450 is described and possibilities of applying different types of microprocessors to construct the controller are analysed. A detailed solution using bit-sliced microprocessors is presented. Finally the further modifications and development of the controller functions are discussed.