

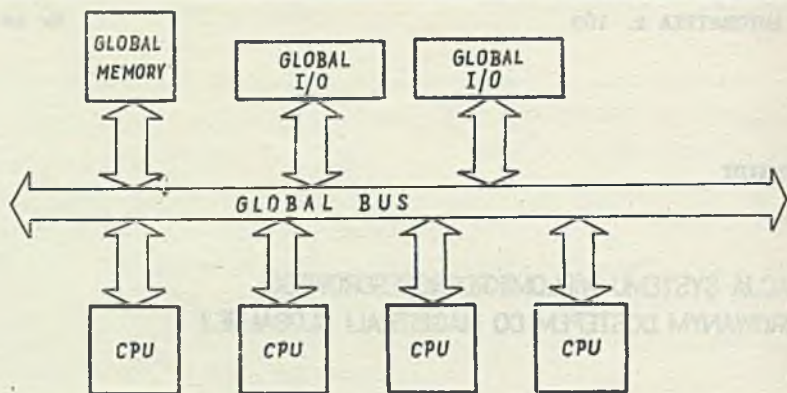
Adam SCHMIDT

REALIZACJA SYSTEMU WIELOMIKROPROCESOROWEGO Z BUFOROWANYM DOSTĘPEM DO MAGISTRALI GLOBALNEJ

Streszczenie. W pracy omówiono konstrukcję prostego, jednomagistralowego systemu wielomikroprocesorowego, opartego o mikroprocesory Intel 8085. Poszczególne procesory wyposażone są w pamięci lokalne ROM i RAM i mają dostęp do pamięci globalnej RAM. W związku z tym, że czas dostępu do pamięci globalnej jest mniejszy niż czas cyklu READ/WRITE mikroprocesora, zastosowano układ zatraskiwania w rejestrze danej odczytanej z pamięci globalnej. Dostęp do magistrali globalnej synchronizowany jest przez układ arbitrażu, którym jest wielowójściowy przerzutnik RS. W razie konieczności oczekiwania mikroprocesor wprowadzany jest w stan WAIT. Do synchronizacji przesyłania informacji między procesorami wykorzystany został układ przerwań. Informacje przekazywane są przez pamięć globalną. W obecnej chwili system zorganizowany jest na zasadzie MASTER/SLAVE, to jest zadania zlecane są przez jeden procesor innym. Skonstruowano układ badania efektywności systemu, umożliwiający zliczanie np. liczby dostępu do pamięci globalnej, czasu wykonywania programu (w taktach zegarowych) i cykli WAIT. Omówiono wyniki eksperymentów.

1. WSTĘP

Poszukiwania w dziedzinie architektury wieloprocessorowej są jednym z najbardziej rozwijanych kierunków badań współczesnej informatyki. Przedstawiono już olbrzymią liczbę propozycji ciągle powstają nowe (przykładowe artykuły przeglądowe: [3], [16], [16]). Jednym z rozwiązań, które osiągnęło już dojrzałość konstrukcyjną jest architektura jednomagistralowa (rys. 1). Przyspieszony jej rozwój został spowodowany pojawieniem się w latach siedemdziesiątych mikroprocesorów jednoukładowych - z tego okresu pochodzą pierwsze konstrukcje prostych (to jest opartych na niedrogich układach mikroprocesorowych) systemów, jak np. [4], [9] i próby ich opisu (np. [11], [12], [13]).



Rys. 1. Ogólny schemat jednomagistralowego systemu wieloprocessorowego
 Fig. 1. General block diagram of a multiprocessor system with one bus and one global memory

Podstawową cechą systemów magistralowych jest ograniczona możliwość rozbudowy (dołączania procesorów), ze względu na prędkość transmisji na magistrali. Z tego powodu stosowane procesory mają stosunkowo dużą jednostkową moc obliczeniową, co od razu predystynuje je do rozwiązywania problemów obliczeniowych o tzw. dużej ziarnistości [20].

Ogólnie stosowaną przez producentów filozofią jest przystosowanie wykorzystanej w komputerze magistrali do pracy wieloprocessorowej i pozostawienie użytkownikowi decyzji o ewentualnym rozszerzeniu posiadanego systemu o dalsze procesory - takie podejście zastosowali na przykład projektanci najczęściej używanych magistrali: MULTIBUS [17] i VMEbus [2], [5].

Architekturę wieloprocessorową wykorzystuje się z kilku powodów. Podstawowym jest potrzeba przyspieszenia pracy całego systemu, jak np. w komputerach Hewlett-Packard 9000, Balance 8000 czy SUPRENUM. W przypadku HP 9000 [1], [15] wykorzystano magistralę z multipleksowaniem adresów i danych oraz potokowym odczytem z pamięci. Można do niej dołączyć do 7 pakietów CPU lub procesorów wejścia/wyjścia i 15 pakietów pamięci. W komputerze Balance 8000 firmy Sequent [14] do magistrali można przyłączyć do 12 pakietów CPU, opartych na mikroprocesorze NS 32000. Każdy pakiet CPU dysponuje pamięcią CACHE o pojemności 8 KB i 32-bitowym buforem zapisu. Zastosowano, podobnie jak w HP 9000, potokowy odczyt z pamięci globalnej. Eksperymentalny komputer SUPRENUM [20] jest przykładem obejścia ograniczenia liczby procesorów. Zastosowano w nim łączenie jedną magistralą grupy: 16 procesorów opartych na układzie MC 68020, jednego procesora wektorowego, procesora komunikacyjnego i 8 MB pamięci. Każda grupa może łączyć się z 4 innymi, tworząc system o nieograniczonych rozmiarach.

Drugim powodem stosowania architektury wieloprocessorowej jest potrzeba zwiększenia niezawodności systemu, jak np. w komputerze ATTEMPTO [6]. Wadą systemu jednomagistralowego jest w takim przypadku brak odporności na uszkodzenia magistrali i układu arbitrażu, której to wady nie mają systemy oparte na innych topologiach, jak np. DIRMU [19].

Ostatnim powodem jest chęć połączenia w jeden scalony system pomiarowo-kontrolny kilku niezależnych komputerów sterujących procesami przemysłowymi. Jest to jakby odwrotne podejście do przetwarzania wieloprocessorowego - kilka procesów obliczeniowych scala się w jeden, aby osiągnąć kompleksowe sterowanie. Architektura komputera pozostaje jednak dokładnie taka sama, mimo innego rozłożenia akcentów.

Ogólnie rzecz biorąc nadal decydującą zaletą architektury jednomagistralowej jest jej prostota i możliwość oparcia konstrukcji na gotowych elementach. Pojawienie się układów przystosowanych do innego sposobu łączenia (jak np. transputery [7]) może jednak zmienić sytuację.

Aby zbadać wpływ poszczególnych elementów systemu wielomikroprocesorowego na całkowitą efektywność należy skonstruować jego możliwie prostą wersję (prostota wymuszana jest też przez dostępność materiałów) i zaimplementować na niej niektóre algorytmy przetwarzania współbieżnego. W niniejszym artykule przedstawiono właśnie taki system, w którym zastosowano układ zwiększający przepustowość magistrali. Wyniki badań nad uzyskanym wzrostem efektywności zostaną przedstawione osobno.

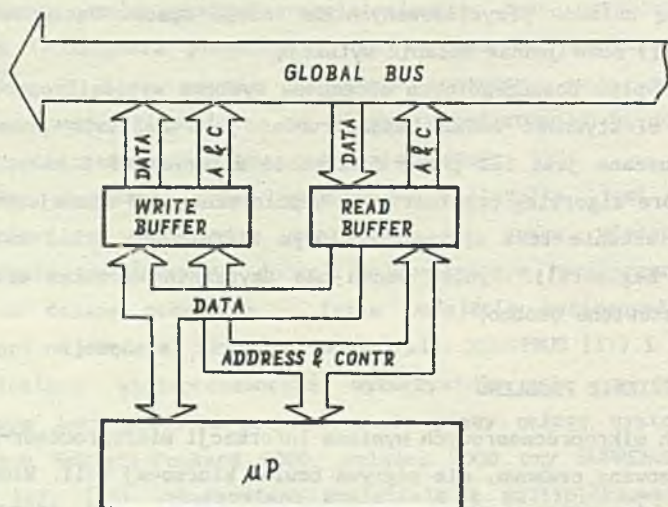
2. PRZEDSTAWIENIE PROBLEMU

W systemach mikroprocesorowych wymiana informacji mikroprocesor-pamięć nie jest optymalizowana czasowo, nie odgrywa bowiem kluczowej roli. Widoczne jest przede wszystkim niedopasowanie w czasie trwania cyklu odczytu/zapisu. Dawniej układy mikroprocesorowe były szybsze od pamięci - aby zsynchronizować pracę przewidziano mechanizm wstawiania taktów WAIT dla przedłużania tego cyklu. W chwili obecnej starsze wersje mikroprocesorów (ośmiobitowe) są wyraźnie wolniejsze od układów pamięci scalonych. Naturalnie w systemach jednoprocessorowych ta różnica nie prowadzi do żadnych zakłóceń, gdyż praca odbywa się z szybkością dyktowaną przez mikroprocesor. Inaczej jest w systemach wielomikroprocesorowych. Tutaj nie ma właściwie powodu, aby cykl wymiany magistralowej odbywał się zgodnie z taktmem procesora, jeżeli cykl ten można skrócić do wartości, na jaką pozwala pamięć. Odwrotną sytuację można zaobserwować w dużych komputerach wieloprocessorowych, opartych na procesorach 32-bitowych wykorzystujących duże pamięci o skomplikowanej organizacji,

Jak wspomniany Balance 8000 [14]. Tam procesor ponownie jest szybszy i oczekiwanie na reakcję pamięci niepotrzebnie spowolniłoby jego pracę.

W celu zapobieżenia trudnościom w dopasowaniu prędkości stosuje się buforowany zapis i odczyt (rys. 2).

W przypadku gdy pamięć jest szybsza od procesora, stosuje się tylko jednobajtowy bufor odczytu. Przy odczycie z pamięci procesor wysyła na magistralę adres i informacje sterujące, i po czasie równym czasowi dostępu do pamięci zapisuje dane w buforze, po czym zwalnia magistralę. Dane podtrzymywane są w buforze tak długo, jak wymaga tego długość cyklu procesora, a w tym czasie na magistrali mogą się odbywać kolejne cykle wymiany. Wykorzystanie układu pamięć globalna-magistrala jest efektywniejsze. Takie podejście zastosowano w przedstawionym systemie wieloprocesorowym.



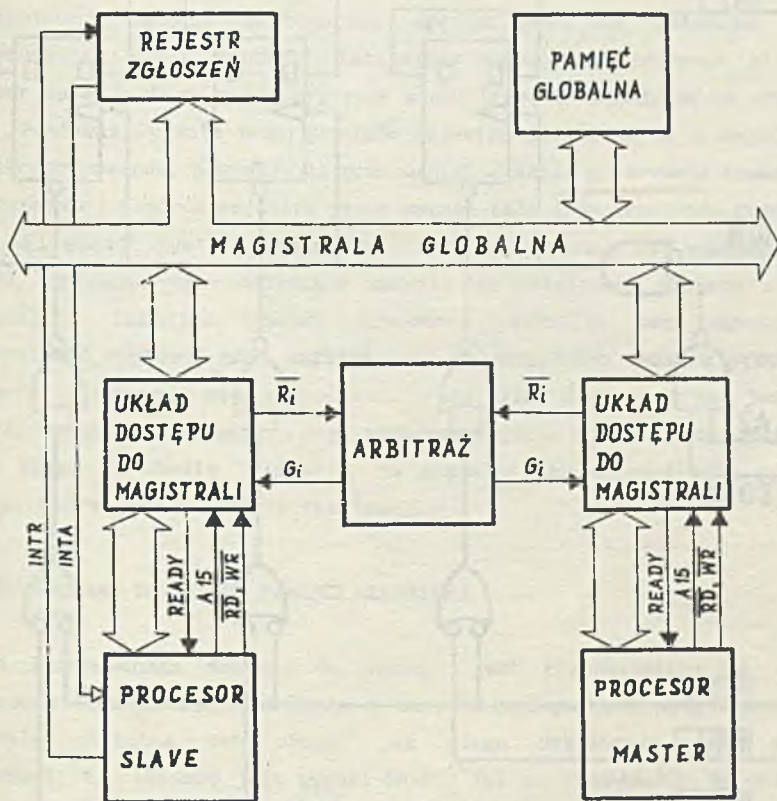
Rys. 2. Schemat buforowanego dostępu do magistrali

Fig. 2. The idea of a buffered bus access

3. OGÓLNY OPIS PREZENTOWANEGO SYSTEMU

Przedstawiany system (rys. 3) zbudowany jest w oparciu o mikroprocesor Intel 8085 i szybkie pamięci statyczne 8K * 8bit, o czasie dostępu około 100 ns. Poszczególne procesory wyposażone są w pamięci lokalne, których zawartość kontrolowana jest programowo. Programista musi napisać odpowiednie procedury transmisji i w odpowiedniej chwili inicjować przesyłanie bloków informacji. Istnieją już, jak wspomniano ([14]), systemy z automatycznym

przepisywaniem bloków informacji między pamięcią globalną a lokalną (traktowaną jako CACHE MEMORY), ale wymagają one automatycznej aktualizacji zawartości pamięci globalnej po każdym zapisie do CACHE, co związane jest z dużym narzutem sprzętowym. Poza tym konstruowany system, jako eksperymentalny, musiał zapewniać możliwość prostego kontrolowania przesyłów magistralowych.

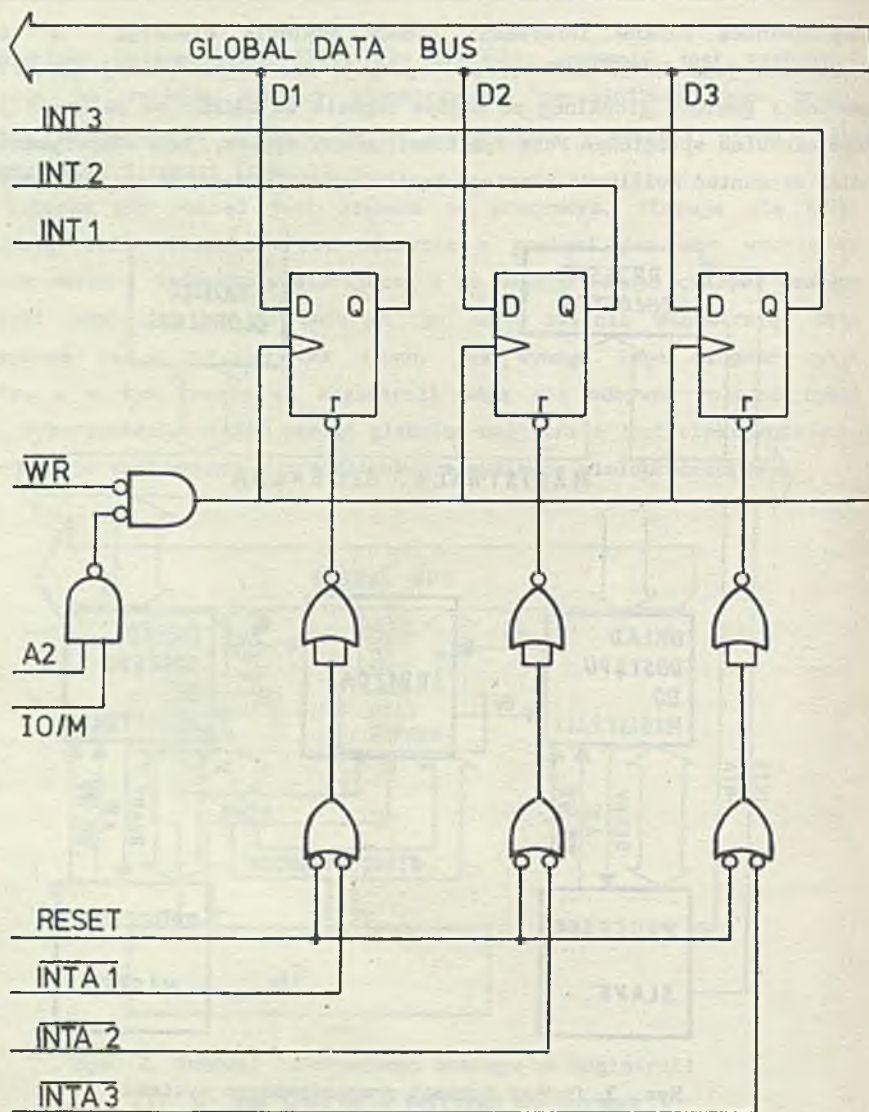


Rys. 3. Ogólny schemat prezentowanego systemu

Fig. 3. Block diagram of the system being discussed.

"Rejestr zgłoszeń" - interprocessor communication register (shown in fig. 4), "pamięć globalna" - global memory, "magistrala globalna" - global bus, "układ dostępu do magistrali" - buffered bus access unit, "arbitraż" - bus controller, "procesor" - processor

Pamięć lokalna zajmuje mniej znaczące 32kB przestrzeni adresowej, pozostaje 32kB to pamięć globalna. Umożliwia to proste kwalifikowanie cyklu pamięci jako lokalnego lub globalnego przez badanie wartości bitu adresowego A15.



Rys. 4. Rejestr zgłoszeń przerwań synchronizujących wymianę informacji

Fig. 4. Interprocessor communications register, which latches interrupts used for interprocessor communication

Przy odwołaniu do pamięci globalnej procesor generuje sygnał REQUEST₁ (generacja tego sygnału zostanie omówiona później), który kierowany jest do układu arbitrażu. Po przyznaniu magistrali układ ten informuje procesor o tym fakcie sygnałem GRANT₁. Jako arbiter zastosowany został wielowejściowy

przerzutnik RS, mający tę zaletę, że jest szybszy od scalonych układów arbitrażowych, produkowanych na przykład przez firmę Intel. Można go zaliczyć do układów arbitrażu równoległego priorytetowego, w którym priorytety są wynikiem opóźnień w bramkach przerzutnika. W przypadku gdy magistrala jest zajęta, mikroprocesor wstrzymuje pracę, bowiem na jego wejście READY podany jest iloczyn $REQUEST_i \cdot \overline{GRANT_i}$.

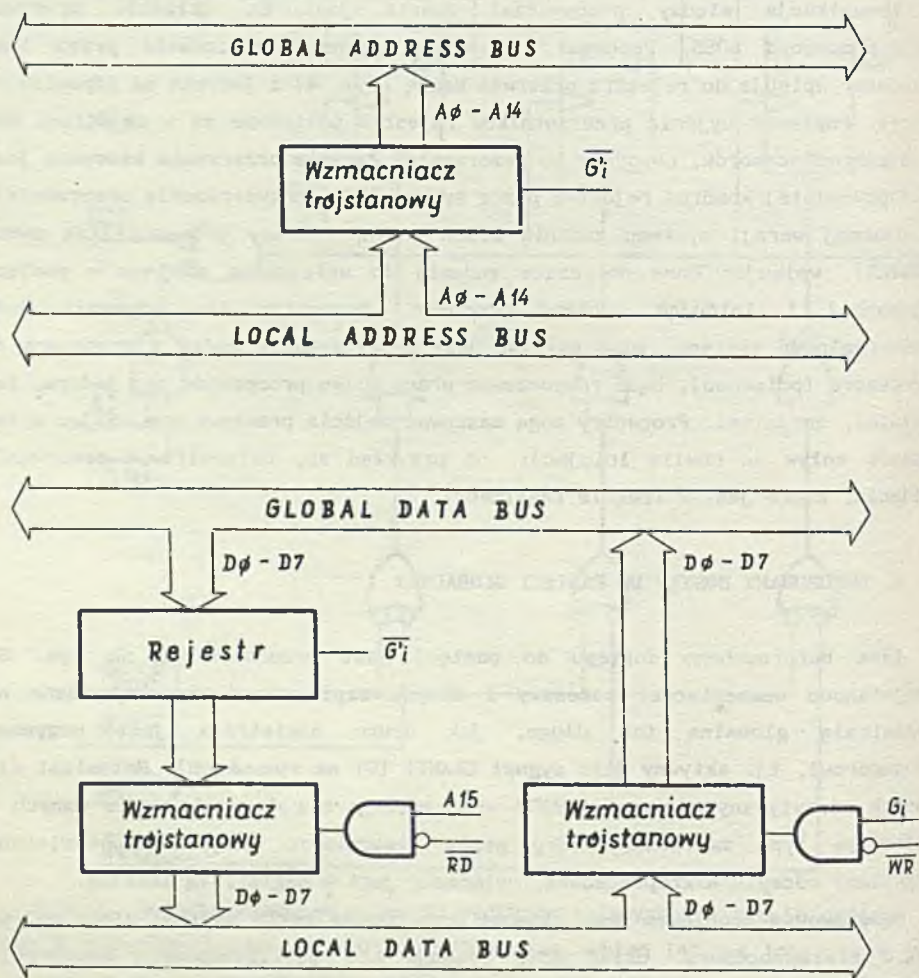
Komunikacja między procesorami oparta jest na układzie przerwań mikroprocesora 8085. Procesor inicjujący wykonywanie zadania przez inny procesor wpisuje do rejestru przerwań maskę (rys. 4) z jedyneką na odpowiednim bicie. Ponieważ wyjścia przerzutników rejestru połączone są z wejściami RST 6.5 mikroprocesorów, powoduje to przerwanie. Żądanie przerwania kasowane jest w odpowiedniej komórce rejestru przez sygnał INTA (potwierdzenie przerwania). W obecnej wersji systemu zadania zleca jeden, wybrany procesor (tak zwany MASTER), wpisując dane dotyczące zadania do ustalonego miejsca w pamięci globalnej i inicjując wybrany procesor. Mechanizm ten zapewnia dużą uniwersalność systemu, gdyż możliwe jest przekazywanie zadań z procesora do procesora (potokowo), bądź równoczesna praca kilku procesorów nad jednym, lub różnymi, zadaniami. Procesory mogą maskować wejścia przerywające, mając w ten sposób wpływ na chwilę inicjacji, na przykład aby uniemożliwić przerwanie zadania, które jest w trakcie realizacji.

4. BUFOROWANY DOSTĘP DO PAMIĘCI GLOBALNEJ

Idea buforowanego dostępu do pamięci jest przedstawiona na rys. 5. Trójstanowe wzmacniacze: adresowy i danych zapisywanych wysyłają dane na magistralę globalną tak długo, jak długo magistrala jest przynana procesorowi, tj. aktywny jest sygnał $GRANT_i$ (G_i na rysunkach). Natomiast dla danych odczytywanych sygnał $GRANT_i$ wykorzystuje się do zapisywania danych w rejestrze typu zatrask, który przez wzmacniacze trójstanowe otwierane sygnałami odczytu mikroprocesora, połączony jest z magistralą lokalną.

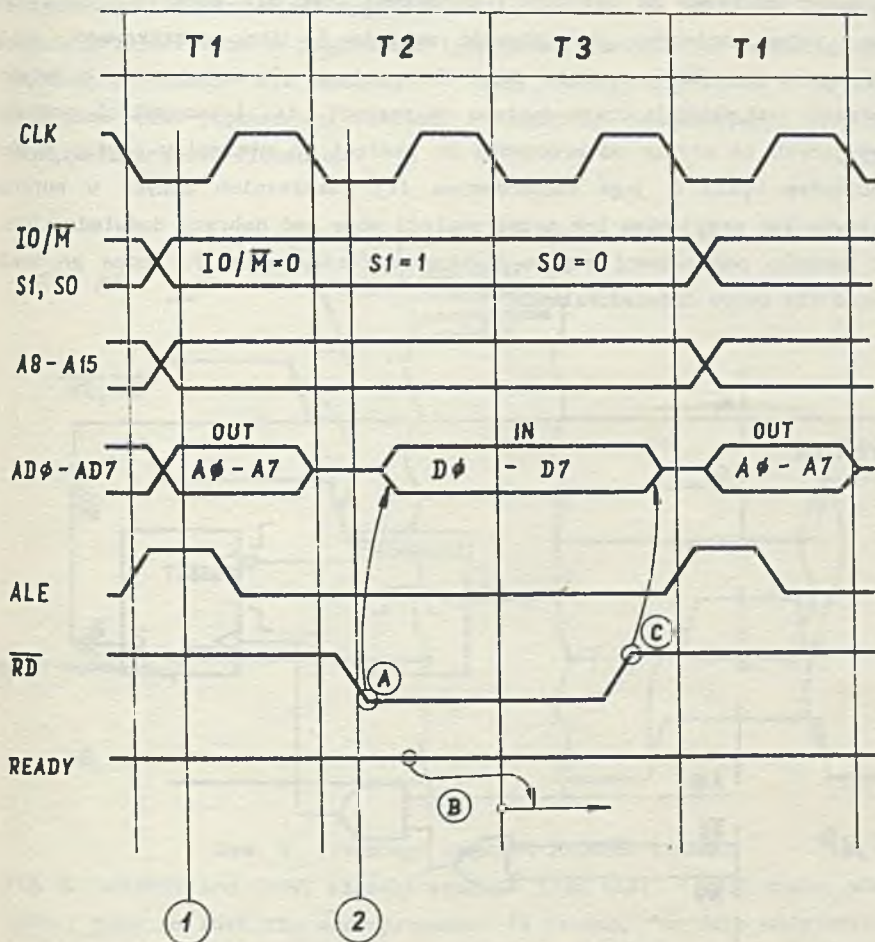
Podstawowym problemem jest czasowe dopasowanie cyklu pamięci globalnej do cyklu mikroprocesora. Układ 8085 cechuje się multipleksowaną magistralą danych i adresową, w związku z czym cykl dzieli się na czas potrzebny na wysłanie na tej magistrali mniej znaczącej części adresu i potrzebny na zapis/odczyt danej. Z drugiej strony rozpoczęcie generacji sygnału $REQUEST_i$ powinno być podjęte odpowiednio wcześniej przed oczekiwaną chwilą pojawienia się sygnału $GRANT_i$, ponieważ sama generacja sygnału $REQUEST_i$ trwa pewien czas, a trzeba też wziąć pod uwagę opóźnienie arbitrażu i układu generacji

sygnału READY, podawanego na wejście o tej samej nazwie w mikroprocesorze. Istnieje tu sprzeczność. Zbyt wczesna generacja sygnału REQUEST1 spowoduje, że dane na magistrali nie zdążą się ustabilizować przed zapamiętaniem ich w pamięci, zbyt późna - że układy logiczne nie zdążą wygenerować na czas sygnału READY, i mikroprocesor niepotrzebnie wprowadzi dodatkowy cykl WAIT. Możliwe położenia chwili generacji sygnału REQUEST pokazuje rys. 6.



Rys. 5. Układ buforowanego odczytu z magistrali globalnej

Fig. 5. Buffer registers, which latch byte read from the global bus and hold it until microprocessor completes READ cycle, so that the bus can be released without waiting for the microprocessor. "Rejestr" - register, "wzmacniacz trójstanowy" - tri-state bufer

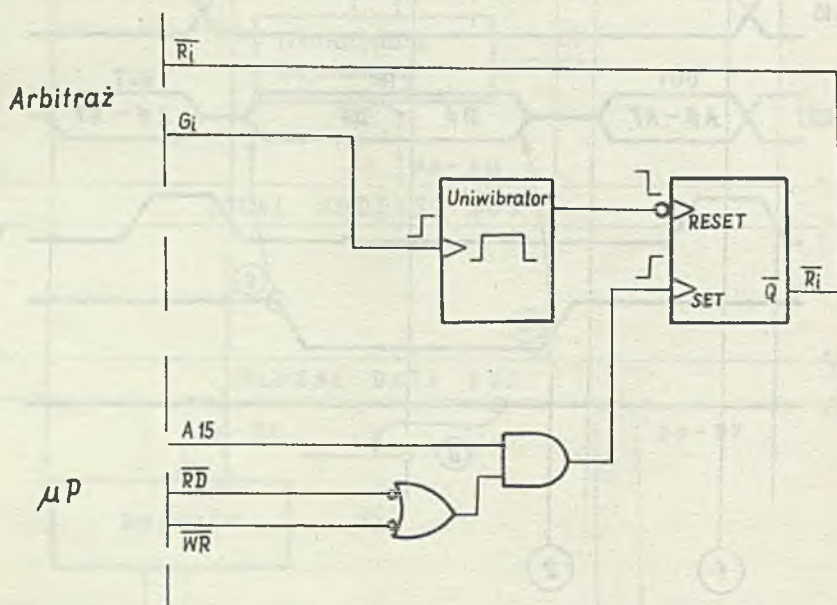


Rys. 6. Możliwe położenia w chwili rozpoczęcia generowania sygnału REQUEST na tle przebiegów mikroprocesora 8085 (według [10])

Fig. 6. Timings of READ/WRITE cycles with possible moments of a start of generating a REQUEST signal

Na podstawie przedstawionych przebiegów można ustalić położenia cyklu magistrali globalnej. Pierwszą możliwością jest rozpoczęcie cyklu magistrali w chwili 1, wtedy bowiem znane są już wszystkie informacje o cyklu maszynowym, można więc odczytać dane z pamięci (dla cyklu odczytu), i po czasie t_A zapamiętać je w rejestrze buforowym. To postępowanie nie sprawdza się jednak dla cyklu zapisu, bowiem dane wyjściowe pojawiają się na magistrali AD dopiero w chwili 2. Ponieważ korzystne jest wykonanie jednego układu

sterującego dostępem do magistrali globalnej, tak dla odczytu, jak i dla zapisu, zatem konieczne jest obranie momentu t_2 jako początkowego, wtedy bowiem znane są już wszystkie dane dla pamięci. Ponieważ czas zajętości magistrali jest funkcją czasu dostępu do pamięci t_A i wszystkich opóźnień występujących na drodze od procesora do pamięci to minimalny odstęp między rozpoczęciem cyklu a jego zakończeniem (t.j. zapisaniem danych w buforze wejściowym lub przyjęciem ich przez pamięć) może być dobrany doświadczalnie, przez badanie poprawności zapisu i odczytu (test pamięci). Można go także zmieniać dla celów doświadczalnych.

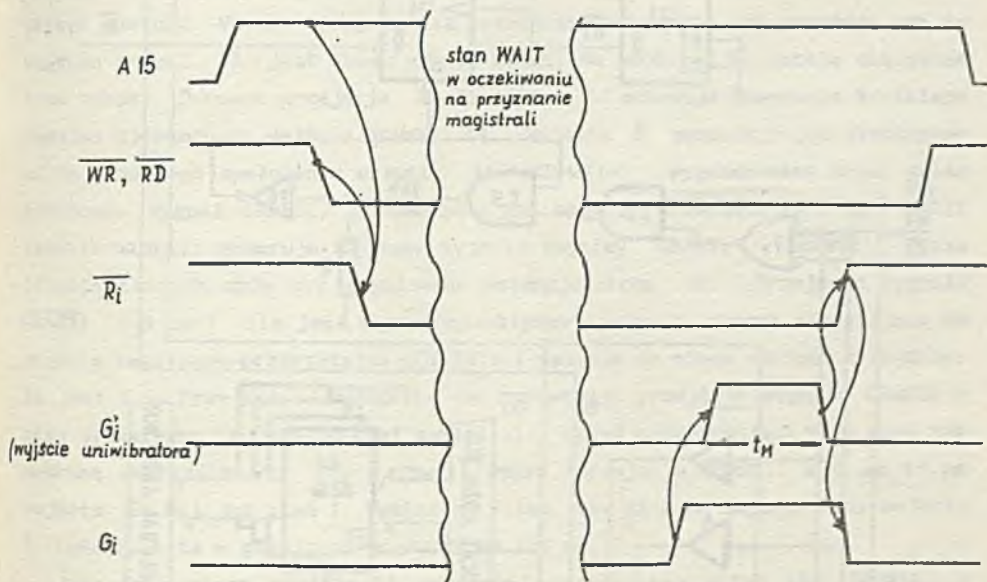


Rys. 7. Generacja sygnału REQUEST

Fig. 7. REQUEST signal forming circuit. "Arbitraż" - bus control unit, "uniwibrator" - monostable multivibrator

Ogólna koncepcja układu generacji sygnału REQUEST jest wyjaśniona na rys. 7, a przebiegi sygnałów przy dostępie do pamięci globalnej przedstawia rys. 8. Czas dostępu do magistrali globalnej może być wyznaczony przez sygnał GRANT i wyjście uniwibratora GRANT', przy czym GRANT' jest nieco krótszy (narastanie - o czas zadziałania uniwibratora, opadanie - o czas zadziałania przerzutnika generującego sygnał $\overline{REQUEST}$ i czas arbitrażu). Ponieważ zwolnienie magistrali wymaga pewnego czasu, do sterowania wzmacniaczem

trójstanowymi wykorzystano sygnał $GRANT_i'$. Wyklucza to możliwość wystąpienia stanów przejściowych na magistrali (gdy jeden procesor już rozpoczął cykl, a poprzednik jeszcze nie skończył). Stany takie, aczkolwiek praktycznie niegroźne dla poprawności zapisu i odczytu, powodowałyby niepotrzebne obciążenie prądowe elementów.



Rys. 8. Przebiegi sygnałów REQUEST i GRANT

Fig. 8. REQUEST and GRANT signals timing. "Stan WAIT" - WAIT state, when the bus is busy so that the microprocessor is frozen, "wyjście uniwibratora" - monostable multivibrator output

Dla magistralii adresowej można zastosować pojedynczy wzmacniacz trójstanowy, otwierany sygnałem $GRANT_i'=1$. Dla magistrali danych konieczne jest rozpatrzenie dwóch przypadków. Przy wysyłaniu danych na magistralę globalną taki sam wzmacniacz trójstanowy otwierany jest iloczynem sygnałów $GRANT_i' \cdot \overline{WR}$. Przy odczycie danych niezbędne jest użycie rejestru sterowanego sygnałem $GRANT_i'$ do zapamiętywania danych. Jednocześnie spełnienie warunku $A_{15} \cdot \overline{RD} = 1$ otwiera wzmacniacz trójstanowy, co powoduje wysłanie danych na magistralę lokalną. Po upływie czasu t_w sygnał $GRANT_i'=0$ zapisuje w rejestrze dane, które mogą być teraz odczytywane przez mikroprocesor dowolnie długo. Kwalifikowanie odczytu dodatkowo sygnałem A_{15} zapobiega wprowadzeniu na magistralę lokalną informacji z rejestru przy komunikacji z pamięcią lokalną.

Realizacja praktyczna układu buforowanego dostępu pokazana jest na rys. 9. Jego działanie jest następujące. Po spełnieniu warunku $A15 \cdot (RD + WR) = 1$ na wyjściu bramki T2 pojawia się stan 0, a na wyjściu bramki T3 stan 1, czyli $V_{OH} = 3.5 \dots 4.2V$. Stan ten przenosi się przez kondensator układu różniczkującego i bramkę T4 na wejście \bar{S} przerzutnika, ustawiając $\overline{REQUEST1} = 0$. Ładujący się od $U_c = V_{OL}$ do $V_c = V_{OL}$ kondensator po ok. 30 ns osiąga wartość $V_c = V_p$ (progową dla przełączenia bramki), co powoduje, że na wejściu bramki T4 jest znowu sygnał niski. Na wejściu \bar{S} ustala się zatem stan wysoki. Ponowne przejście T2 z 0 na 1 powoduje powstanie krótkiego impulsu ujemnego na wejściu bramki T4. Wejście \bar{S} pozostaje już nieaktywne aż do ponownego spełnienia warunku $A15 \cdot (RD + WR) = 1$. Wygenerowany przez układ arbitrażu sygnał GRANT1, wprowadzony na wejście przerzutnika UCY 74121 (uniwibratora), generuje na jego wyjściu impulsy GRANT1' i $\overline{GRANT1}'$, czas trwania których może być regulowany potencjometrem Rt. Przejście sygnału $\overline{GRANT1}$ z 0 na 1 (to jest w stan nieaktywny) generuje zbocze narastające na wejściu zegarowym przerzutnika UCY 7474 i wpisuje do niego wartość z wejścia, to jest 0. Przejście $\overline{REQUEST1}$ na 1 powoduje przejście sygnału GRANT1 w stan nieaktywny. Kończy to cykl magistrali. Układ różniczkujący musi mieć tak dobraną stałą czasową, aby w chwili wpisu (przejście $\overline{GRANT1}'$ z 0 na 1) na wejściu \bar{S} był już stan 1. Wystarczy zatem, aby długość impulsu 0 na wejściu \bar{S} była zawarta w granicach od 30 ns do 100 ns.

Jako wzmacniacz magistrali adresowej zastosowano układ UCY 74S416, z wejściem $\overline{DIEN} = 0$. Sygnał $\overline{GRANT1}$ otwiera go na magistralę globalną. Linie adresowe przyłączone są do wejść DI_n. Analogicznie połączone są wejścia wzmacniacza wyjściowego danych, ale w tym przypadku wejście \overline{DIEN} połączone jest z \overline{WR} , a \overline{CS} z $\overline{GRANT1}'$. Zapewnia to w cyklu zapisu transmisję w kierunku zasobów globalnych. W cyklu odczytu układ transmituje dane z magistrali globalnej na wyjścia DO_n, które jednak nie są do niczego przyłączone. Jako układ odczytu z magistrali globalnej zastosowano układ 8212 w trybie 0 (MODE 0), łącząc DS2 z A15, DS1 z \overline{RD} i STB z GRANT1'.

Inny problem występuje przy łączeniu z magistralą globalną sygnałów sterujących \overline{RD} , \overline{WR} i IO/ \overline{M} . Należy tu zastosować np. bramki SN 74125, bowiem wzmacniacze trójstanowe UCY74S416 produkcji polskiej wytwarzają przy przejściu z trzeciego stanu do stanu HIGH stosunkowo szeroki (kilkanaście nanosekund) impuls zerowy, który mógłby być odczytany przez pamięć jako aktywny. Sygnał RESET OUT zeruje wstępnie sygnał REQUEST1.

Tablica 1

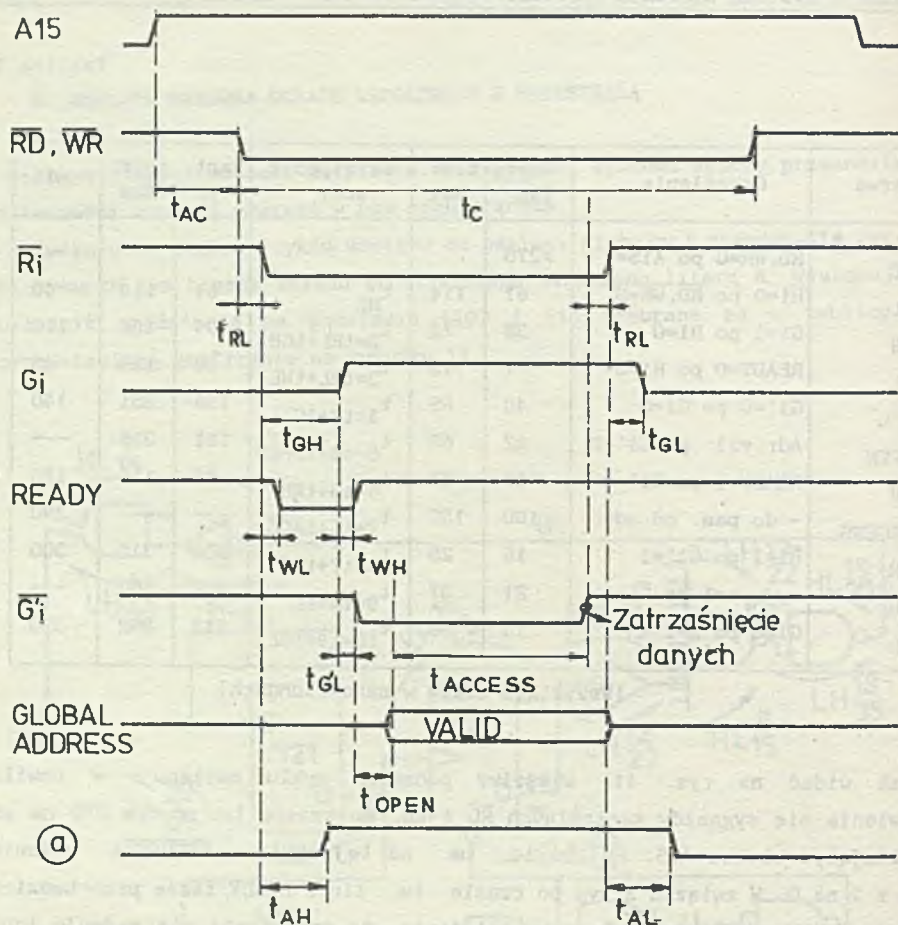
Nazwa	Określenie	Wart.		Określenie sumy	Wart.		Pomiar
		typ	teor. max		typ	teor. max	
t_{AC}	$\overline{RD}, \overline{WR}=0$ po $A15=1$	>270					---
t_{RL}	$\overline{Ri}=0$ po $\overline{RD}, \overline{WR}=0$	61	114	t_{RL}	61	114	40
t_{GH}	$Gi=1$ po $Ri=0$	38	72	$t_2=t_{RL}+t_{GH}$	99	186	110
t_{WL}	$READY=0$ po $Ri=0$	7	15	$t_3=t_{RL}+t_{WL}$	68	129	45
$t_{G'L}$	$\overline{Gi}'=0$ po $\overline{Gi}=0$	40	65	$t_4=t_2+t_{G'L}$	136	251	140
t_{OPEN}	Adr. val. po $Gi'=0$	42	65	$t_5=t_4+t_{OPEN}$	181	316	---
t_{WH}	$READY=1$ po $Gi'=1$	19	37	$t_6=t_3+t_{WH}$	87	116	120
t_{ACCESS}	- do pam. od adr.	100	150	$t_7=t_5+t_{ACCESS}$	---	---	290
t_{RH}	$\overline{Ri}=1$ po $\overline{Gi}'=1$	16	25	$t_8=t_7+t_{RH}$	306	315	300
t_{AH}	wyj. $a=1$ po $\overline{Ri}=1$	21	37	$t_9=t_8+AL$	327	352	---
t_{GL}	$Gi=0$ po $a=1$	15	30	$t_{10}=t_9+t_{GL}$	342	382	330

(Wszystkie czasy w nanosekundach)

Jak widać na rys. 11, właściwy początek cyklu następuje w chwili pojawienia się sygnałów kontrolnych \overline{RD} i \overline{WR} , po czasie t_{AC} równym 270 ns po narastającym zboczku $A15$. Po czasie t_{RL} od tej chwili $\overline{REQUESTi}$ zmienia stan z 1 na 0. W związku z tym po czasie t_{WL} linia $READY$ także przechodzi z 0 na 1. Arbitraż (oczywiście przy założeniu, że magistrali nie zajmuje inny procesor) trwa przez czas t_{GH} , i po tym czasie na wejściu procesora pojawia się sygnał \overline{GRANTi} . Powoduje on po upływie czasu t_{WH} na wejściu $READY$ stan wysoki i otwarcie, po t_{GL} , wzmacniaczy (stan 0 na \overline{CS}). Adres ważny pojawi się na magistrali po czasie równym t_{OPEN} . Zatrzaśnięcie danych nastąpi po czasie t_{ACCESS} , kiedy uniwibrator zmieni stan \overline{GRANTi} na wysoki.

6. UKŁAD BADANIA EFEKTYWNOŚCI

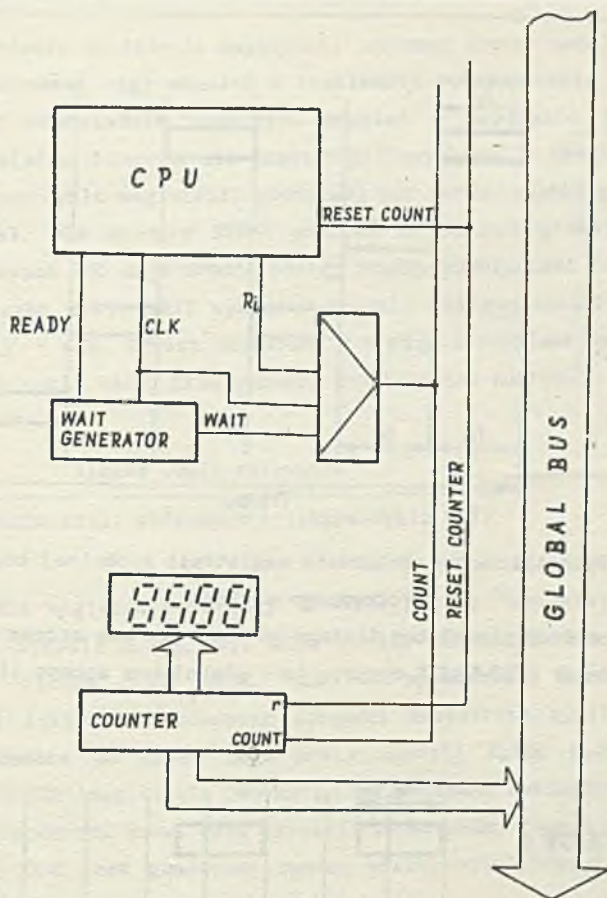
Aby określić efektywność systemu należy znaleźć zależność czasu wykonywania programu od liczby procesorów i długości cyklu magistralowego. W tym celu należy zmierzyć liczbę taktów zegarowych (CLK), liczbę odwołań do pamięci globalnej ($\overline{REQUESTi}$) i cykli WAIT, które świadczą o tym, że dany procesor musi czekać na zwolnienie magistrali. Jest to zadanie układu badania efektywności przedstawionego na rys. 12.



Rys. 11. Przebiegi sygnałów w układzie dostępu do magistrali

Fig. 11. Buffered bus access unit signals timings, with delays shown in fig.10

Na każdym pakiecie znajduje się układ generacji sygnału WAIT (sygnał ten nie jest wyprowadzony w mikroprocesorze 8085) i układ wyboru sygnału, który ma być zliczany (oparty o multiplexer i rejestr). Obecnie może być to jeden z wymienionych powyżej, ale ogólnie rzecz biorąc można przyłączyć do układu (wejścia multiplexera) dowolny sygnał z pakietu procesora. Wybrane impulsy kierowane są na globalną linię (przez bramkę OC) i zliczane przez specjalny licznik. Istnieje możliwość zerowania tego licznika przez dowolny procesor. Liczba zliczonych impulsów może być odczytana na wyświetlaczu. Zawartość licznika jest także dostępna pod jednym z adresów wejścia.



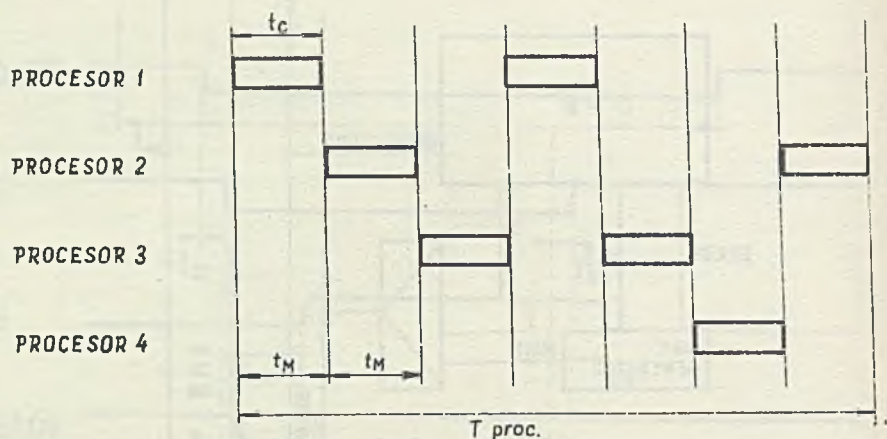
Rys. 12. Układ badania efektywności wykorzystania magistrali globalnej
Fig. 12. Bus utilization measurement device

7. WPŁYW BUFORA NA EFEKTYWNOŚĆ

Zysk czasowy wynikający z przyspieszenia cykli magistralowych przez zastosowanie bufora wejściowego można intuicyjnie przedstawić na wykresach czasowych z rys. 13 i 14.

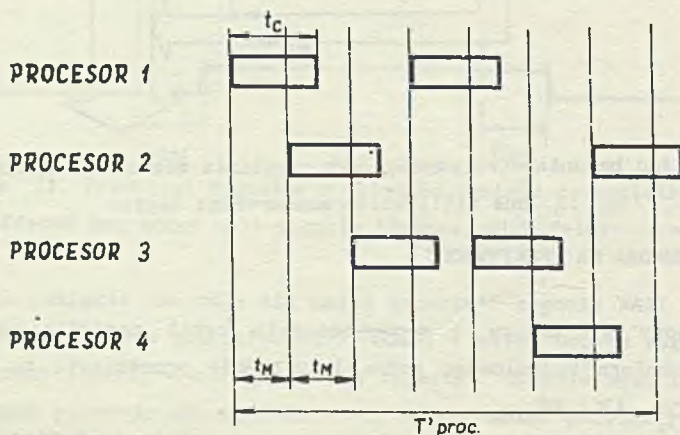
Rys. 13 obrazuje sytuację, gdy odczyt z pamięci globalnej nie jest buforowany. Czas cyklu magistralowego t_M jest wtedy dokładnie równy czasowi cyklu mikroprocesora t_C . W takiej sytuacji całkowity czas przetwarzania t_{Proc} , równy sumie wszystkich t_M , jest jednocześnie równy sumie wszystkich t_C . W drugim przypadku, dla odczytu buforowanego (rys. 14), jest $t_M > t_C$, i moglibyśmy się spodziewać, że:

$$t'_{Proc} = \frac{t_C}{t_M} t_{Proc}$$



Rys. 13. Wyidealizowane obciążenie magistrali globalnej bez układu buforowanego dostępu

Fig. 13. Idealized global bus timings without the bus access unit. t_c - procesor READ/WRITE cycle, t_M - global bus access time



Rys. 14. Wyidealizowane obciążenie magistrali z układem buforowanego dostępu

Fig. 14. Idealized global bus timings with the bus access unit

Niedoskonałość tego rozumowania widoczna jest jednak na pierwszy rzut oka. Zakłada ono, że magistrala jest zajęta przez cały czas, i w dodatku kolejny cykl dostępu rozpoczyna się już w chwili zakończenia poprzedniego. W rzeczywistości założenia te nie są spełnione i wzrost efektywności będzie mniejszy.

W celu zbadania obciążenia magistrali napisany został specjalny program, w którym zasymulowano ciąg odwołań o rozkładzie poissonowskim (przypadkowym). Ich częstość odpowiadała częstości odwołań w programie typu transmisji blokowej. Pomiarów dokonano dla magistrali 'szybkiej' - $t_k=220$ ns i 'wolnej' $t_k=640$ ns (czas cyklu magistrali globalnej bez układu szybkiego dostępu byłby równy 400 ns). Dla średnio 32768 odwołań do pamięci globalnej na 2861000 taktów zegarowych (co daje średni odstęp między odwołaniami równy 87,3 taktu) zmierzona liczba cykli WAIT wynosiła średnio dla magistrali 'wolnej' 1523, dla 'szybkiej' - 628. Wzrost szybkości 2.9 raza spowodował zatem 2.42-krotny spadek liczby cykli WAIT. Efektywność, zdefiniowana wzorem:

$$\eta = \frac{\text{liczba cykli aktywnych}}{\text{liczba cykli aktywnych} + \text{liczba cykli WAIT}}$$

równa była dla magistrali 'wolnej' 0.99946, a dla 'szybkiej' 0.99978. Zysk efektywności wynosił 0.86%, był więc bardzo niewielki, mimo tak znacznego zmniejszenia liczby cykli WAIT. Jest to spowodowane bardzo małą liczbą procesorów i faktem, że stosunek długości wszystkich cykli magistralowych jednego procesora do czasu jego pracy wynosił tylko 0.0076 (magistrala 'szybka') i 0.022 (magistrala 'wolna'). Jak wskazują analogiczne badania [9], dla trzech procesorów nawet przy wspomnianym stosunku równym 0.25 efektywność wynosi około 0.8, zaś gwałtowny spadek efektywności obserwuje się dopiero powyżej 5 procesorów.

8. PODSUMOWANIE

Przedstawiony komputer stał się narzędziem do obserwacji procesów zachodzących na magistrali systemu wieloprocessorowego. Pozwolił doświadczać stwierdzić, że wyposażenie procesorów w układ buforowanego dostępu do magistrali globalnej zwiększa efektywność systemu, przy czym należy się spodziewać, że w miarę wzrostu liczby procesorów wpływ ten będzie coraz większy. W chwili obecnej prowadzone są prace nad rozbudową systemu (wyposażenie go w większą liczbę procesorów, pamięć masową i urządzenia wejścia/wyjścia), a także dalsze badania nad efektywnością, przy użyciu różnych programów współbieżnych.

LITERATURA

- [1] J.W.Beyers, E.R.Zeller, S.D.Seecombe: 'VLSI Technology Packs 32-Bit Computer System into a Small Package', HP Journal, August 1983.
- [2] R.Brause 'Prozessoren tauschen Nachrichten ueber Dual - Ported - RAMs aus', VMEbus, April 1988.
- [3] B.Buzbee 'Parallel Processing Makes Tough Demands', Computer Design, August 15, 1985.
- [4] O.Caprani et al: 'Microprocessors Connected to a Common Memory', Proceedings of Euromicro Symposium, Amsterdam 1977.
- [5] A.Czarnecka: 'Metastabilność systemów VME i MULTIBUS II', Informatyka 2-3, 1986.
- [6] M. Dal Cin et al. 'ATTEMPTO: An Experimental Fault Tolerant Multiprocessor System', Microprocessing and Microprogramming 20 (1987).
- [7] W. Hahn et al. 'A Multi-Transputer-Net As a Hardware Simulation Environment', Proceedings of Euromicro Conference, Zurich 1988.
- [8] H.J.Hindin: 'Parallel Architectures Put Fifth-Generation-Machines on Right Track', Electronic Design, May 3, 1984.
- [9] S.Hoener, W.Roehder: 'Efficiency of a Multi - Microprocessor System with Shared Busses', Proceedings of Euromicro Symposium, Amsterdam 1977.
- [10] Intel Corporation, katalogi.
- [11] Kwok-Tung Fung, H.C.Torng 'On the Analysis of Memory Conflicts And Bus Contentions in a Multiple-Microprocessor System', IEEE Transactions on Computers, January 1979.
- [12] M.Aymone Marsan, M.Gerla: 'Markov Models for Multiple Bus Multiprocessor Systems' IEEE Transactions on Computers, March 1982.
- [13] M.Aymone Marsan, G.Balbo, G.Conte 'Comparative Performance Analysis of Single Bus Multiprocessor Architectures', IEEE Transactions on Computers, December 1982.
- [14] W. Mayberry, G. Efland: 'Cache Boosts Mutiprocessor Performance', Computer Design, Nov. 1984, str. 133.
- [15] B.D.Osecky, D.D. Georg, R.J.Bury: 'The Design of a General - Purpose Multiple Processor System', HP Journal, March 1984.
- [16] C.Patton 'Parallel Architectures Put Fifth-Generation Computers on the Right Track', Electronic Design, May 3, 1984.
- [17] Proposed Microcomputer System Bus Standard (P796 BUS), IEEE Computer Society Subcommittee, October 1980.

- [18] W.Sasal: 'Układy scalone serii UCY 64/UCY 74. Parametry i zastosowania', WKŁ, Warszawa 1985.
- [19] E. Schmidt: 'Pyramide bevorzugt', elektronikpraxis, Janura 1986.
- [20] U.Trottenberg, K.Solchenbach: 'Parallele Algorithmen und ihre Abbildungen auf parallele Rechnerarchitekturen', Informationstechnik 30(1988)2.

Recenzent: Prof.dr hab.inż. Jan Piecha

Wpłynęło do Redakcji 20.06.1988 r.

РЕАЛИЗАЦИЯ МНОГОПРОЦЕССОРНОЙ СИСТЕМЫ
С БУФЕРНЫМ ДОСТУПОМ К ГЛОБАЛЬНОЙ
МАГИСТРАЛЬНОЙ ЛИНИИ

Р е з ю м е .

В работе рассмотрена конструкция простой одномагистральной многопроцессорной системы, опирающаяся на микропроцессорах Intel 8085. Отдельные процессоры оснащены локальной памятью ПЗУ и ОЗУ и имеют доступ к глобальной памяти ОЗУ. В связи с тем, что время доступа к глобальной памяти меньше, чем время цикла READ/WRITE микропроцессора, применена схема фиксации в регистре данной считанной с глобальной памяти. Доступ к глобальной магистрали синхронизирован арбитражной системой, являющейся многоходовым RC-тиртрером. В случае необходимости ожидания микропроцессор вводится в состояние WAIT. Для синхронизации передачи информации между процессорами использовано прерывающее устройство. Информация передаётся через глобальную память. В данном моменте система организована по принципу MASTER/SLAVE - задания предписываются одним процессором другому. Сконструировано схему исследования эффективности системы, дающая возможность считывания, числа доступов к глобальной памяти, времени выполнения программы (в генераторных тактах) и циклов WAIT. Описаны результаты экспериментов.

SINGLE BUS MULTIMICROPROCESSOR SYSTEM WITH A BUFFERED BUS ACCESS

S u m m a r y

This paper describes a simple, single bus multi - microprocessor system, based on Intel 8085 microprocessors, with one global memory RAM and local memories ROM and RAM for each processor. Because a global memory access time is shorter than the READ/WRITE cycle of the microprocessor, latch register has been applied to store a read byte. so that the microprocessor can read it even after the bus is released. A bus access of processors (which send REQUEST signal before accessing the bus) is synchronized by a bus control unit (multiple input RS flip-flop). A GRANT signal enables bus access. When a bus conflict occurs (no GRANT signal), READY signal is disabled and the microprocessor is halted until the bus is available. Interprocessor communication is synchronized by interrupts. Messages from one processor to the other one are passed through the global memory. The system is organised in a MASTER/SLAVE manner, so that one processor (MASTER) allocates tasks for all other processors (SLAVES). A measurement device was constructed to examine bus utilization by adding up signals like global bus requests, the program execution time or WAIT cycles (when the microprocessor is halted). Some results of experiments are given.