

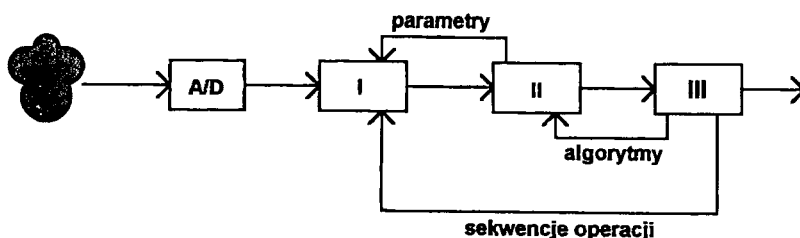
ROZPOZNAWANIE KSZTAŁTÓW I RUCHU OBIEKTÓW DLA POTRZEB STEROWANIA URZĄDZEŃ W CZASIE RZECZYWISTYM Z WYKORZYSTANIEM DEDYKOWANYCH ARCHITEKTUR SYSTEMU WIELOPROCESOROWEGO I POTOKOWEJ OBRÓBKİ WSTĘPNEJ DANYCH OBRAZOWYCH

dr inż. Kazimierz Wiatr
Instytut Elektroniki AGH

Projektu badawczy nr 8 S505 008 07 pod powyższym tytułem jest realizowany w Instytucie Elektroniki AGH w Krakowie pod kierownictwem autora niniejszego opracowania od dnia 15 lipca 1994 roku. Przedmiotem projektu jest opracowanie i realizacja sprzętowa systemu wieloprocesorowego zbudowanego w oparciu o magistralę VME przeznaczoną do przetwarzania obrazów w czasie rzeczywistym celem rozpoznawania kształtów i ruchu obiektów. Poniżej przedstawiono wybrane (ze względu na szczupłość miejsca) aspekty realizowanego projektu.

1. Poziomy analizy obrazów

W algorytmach analizy obrazów można wyróżnić kilka poziomów obróbki i przetwarzania obrazu. Najczęściej przyjmuje się trzy poziomy (rys. 1). Najniższy poziom analizy obrazów (I), nazywany często obróbką sygnału video, ma na celu eliminację zakłóceń, wydobywanie obiektu z tła, detekcję krawędzi, ustalanie poziomów szarości obiektu na podstawie histogramu, liczenie histogramu itp. Średni poziom analizy obrazów (II) dokonuje rozpoznania kształtu obiektu i wyróżnienia cech charakterystycznych tego kształtu. Najwyższy poziom (III) to analiza okna wizyjnego w sensie analizy ruchu obiektu, sterowania obiektem, zadawanie parametrów do obróbki i analizy obrazów na pozostałe poziomy itp.



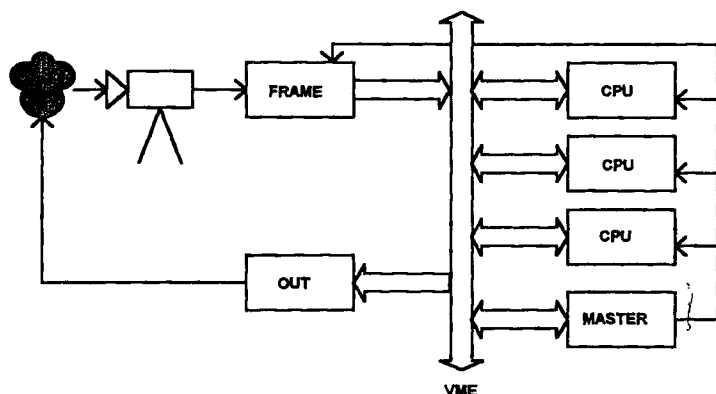
Rys. 1. Poziomy analizy obrazów.

Dla podniesienia możliwości szybkich obliczeń stosujemy systemy wieloprocesorowe. Skuteczność efektywnego wykorzystania struktury wieloprocesorowej związana jest z

optymalnym przydziałem zadań obliczeniowych na poszczególne jednostki-procesory i odpowiedni przepływ danych. Istotnym staje się także problem synchronizacji pracy poszczególnych procesorów. Wymagania te powodują konieczność stosowania dedykowanych architektur systemów wieloprocessorowych oraz stosowanie procesorów specjalizowanych.

2. Architektura systemu

Architektura projektowanego systemu wieloprocessorowego składa się z wielu modułów jednostek centralnych dołączonych do wspólnej magistrali VME (rys. 2).



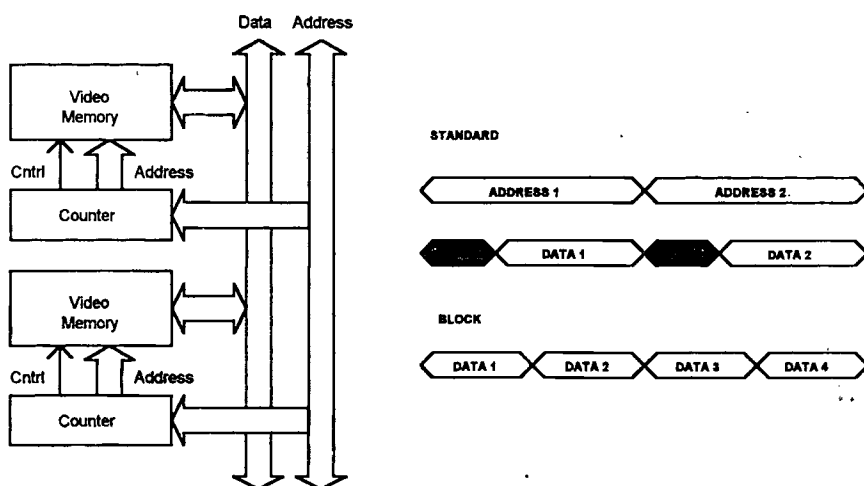
Rys. 2. Architektura systemu wieloprocessorowego do analizy obrazów.

Celem projektu jest realizacja systemu modułowego o dużych mocach obliczeniowych, pozwalających na dokonywanie nawet skomplikowanych obliczeń na danych obrazowych w czasie rzeczywistym. Systemy tego typu są obecnie mało znane od strony naukowej, gdyż powstające zagraniczne opracowania chronione są tajemnicą wojskową lub firmową. Zamiarem autorów jest zaprojektowanie i zbudowanie systemu, a następnie zbadanie jego rzeczywistych parametrów i stworzenie naukowych przesłanek do wytwarzania w Polsce takich systemów.

Do wstępnej obróbki danych obrazowych (na rys. 2 blok FRAME) autorzy zamierzają wykorzystać procesory specjalizowane pracujące jako potok obróbczy na danych pochodzących z kamery wizyjnej. Wstępne prace badawcze na temat takiej architektury zostały przeprowadzone i wyniki uznać należy za obiecujące. Dalszy rozwój architektury potokowej będzie dotyczył opracowania procesorów potokowych do detekcji ruchu oraz opracowania koncepcji rekonfigurowalnych w trakcie pracy systemu procesorów hardware'owych zrealizowanych w technologii PLD-FPGA.

Średni i wysoki poziom analizy obrazów mają obsługiwać moduły jednostek centralnych zbudowanych w oparciu o mikroprocesor MC68040 firmy MOTOROLA (z własną pamięć RAM oraz ROM i własnymi urządzeniami wejścia-wyjścia) wyposażone w mechanizm szybkiej blokowej transmisji danych po magistrali VME (na rys. 2 bloki CPU). Na

karcie CPU będzie zainstalowany profesjonalny kontroler magistrali VME pozwalający na dokonywanie transmisji blokowej danych magistralą danych o szerokości słowa do 32 bitów (a nawet do 64 bitów wówczas gdy transmisja blokowa danych odbywa się po magistrali danych i adresów). Bardzo ważnym zagadnieniem jest problem obciążenia magistrali VME ruchem danych wizyjnych na poszczególnych etapach obróbki i przetwarzania. Transmisja blokowa słów 64 bitowych znacznie zmniejsza czas transmisji danych obrazowych: 256 kbajtów - ok. $32k \times 100ns = 3,2 \text{ ms}$ (dla transmisji standardowej ok. $256k \times 200 \text{ ns} = 51,2 \text{ ms}$). Pozwala to na korzystanie z danej magistrali VME wielu jednostek centralnych wykonujących szereg szczegółowych zadań obróbki i analizy obrazu.



Rys. 3. Struktura transmisji blokowej.

Podział zadań pomiędzy procesory będzie zależny od rodzaju realizowanego algorytmu, rodzaju obiektu itp. Główny ciężar nadzoru pracy jednostek CPU będzie spoczywał na jednostce MASTER (rys. 2). Jako moduł jednostki centralnej wykorzystano pakiet SYS68K/CPU-30 firmy FORCE. Jest to jednopłytowy komputer zbudowany w oparciu o mikroprocesor Motorola 68030, posiadającym możliwość dołączenia do 16 Mbajtów pamięci RAM. Jest on wyposażony w 32-bitowy kontroler DMA pozwalający na szybką transmisję danych zarówno do lokalnej pamięci DRAM jak i pamięci umieszczonej na magistrali VMEbus. Obsługa pamięci masowej realizowana jest przez kontroler SCSI oraz pojedynczy kontroler dysków miękkich. Oba kontrolery są dołączone do 32-bitowego kontrolera DMA. Karta posiada cztery całkowicie niezależne wieloprotokołowe kanały szeregowo. Moduł posiada możliwość dołączenia do 4 Mbajtów pamięci EPROM, 32 kbajtów pamięci SRAM i zegara czasu rzeczywistego. System VMEPROM ("PDOS Real Time Kernel") jest standardowo dołączany do płyty. Na płycie jest zainstalowane osiem przerwań mechanizmu "mailbox".

Konfiguracja modułu MASTER jest następująca: CPU: Motorola 68030, FPCP: Motorola 68882, 32-bitowy szybki kontroler DMA obsługujący pamięć RAM płyty oraz transmisję na VMEbus, pamięć RAM od 4 do 16 Mbajtów z kontrolą parzystości (możliwość pracy w "68030 cache burst file mode"), system przesyłania informacji FMB, cztery szeregowo

kanaly I/O (jeden kanał w standardzie RS232, pozostałe mogą pracować jako RS232, RS422 lub RS485), 8-bitowy interface równoległy ("Centronics") z "handshake", cztery systemowe pamięci EPROM (DIL 28 lub DIL32) wykorzystujące 32-bitową magistralę danych, pojedynczy "Boot EPROM" do inicjalizacji układów I/O i układu "Gate Array", 32 kbajty SRAM z podtrzymaniem baterijnym (DIL28 lub DIL32), "Real Time Clock" z podtrzymaniem baterijnym, interface SCSI połączony z kontrolerem DMA na płycie, interface do dysków miękkich (3", 3.5" lub 5.25"), dwa 24-bitowe timery z 5-bitowym dzielnikiem wstępnym (prescaler), jeden 8-bitowy timer, przerwania na poziomie programowym dla lokalnego CPU od wszystkich lokalnych I/O, BERR działający pod kontrolą programową różnych liczników (lokalnych lub z magistrali VME), pełny 32-bitowy interface "VME master/slave" transmisja danych w modach: a. A32, A24, A16: D8, D16, D32 - Master, b. A32: D8, D16, D32 - Slave, c. UAT i "Read-Modify Write", arbiter DMA "single level", licznik "Time Out" dla pamięci lokalnej i z magistrali VME.

Moduł MASTER będzie pracował pod nadzorem systemu operacyjnego czasu rzeczywistego OS-9, którego mechanizmy synchronizacji pracy poszczególnych zadań zostaną krótko przedstawione w rozdziale 3.

Kolejnym elementem tworzonej architektury wieloprocessorowej do obróbki i przetwarzania obrazów jest moduł do szybkich obliczeń oparty o układ scalonego procesora sygnałowego DSP56001 firmy Motorola. Moduł taki zbudowany z czterech procesorów sygnałowych, trójportowej pamięci dynamicznej firmy Micron oraz szeregu multiplexerów pozwoli na realizację on-line niektórych obliczeń dotyczących średniego poziomu analizy obrazów.

3. Synchronizacja pracy procesorów

Całość systemu będzie pracowała pod nadzorem systemu operacyjnego czasu rzeczywistego OS-9. Ważnym zagadnieniem związanym ze współpracą wielu procesorów (CPU) jest synchronizacja ich pracy. Możliwości systemu operacyjnego OS-9 w tym zakresie pozwalają na różne sposoby komunikacji pomiędzy modułami i związanych z tym możliwości kształtowania różnych architektur programów użytkowych.

Jądro systemu operacyjnego OS-9 pozwala na jednoczesne wykonywanie w czasie rzeczywistym kilku niezależnych programów dzięki możliwościom wielozadaniowości i komunikacji pomiędzy procesami (programami). Wszystkie programy w systemie OS-9 uruchamiane są jako procesy, a każdy z nich może mieć dostęp do pożądanых zasobów systemu poprzez odpowiednie wywołania żądań obsługi. Praca z podziałem czasu umożliwia procesom wspólne korzystanie z procesora. OS-9 wykorzystuje do tego celu priorytetowy program szeregujący (ang. *round robin*), który przydziela czas procesora odpowiednio dla każdego procesu. Praca CPU jest przerywana przez zegar czasu rzeczywistego co pewien czas (ang. *tick*), określony początkowo jako 10 ms. Po każdym takim impulsie przerywającym OS-9 zawieszają wykonywanie bieżącego programu i rozpoczynają wykonywanie następnego. Długość trwania czasu nieprzerwanego wykonywania programu jest więc zależna od sprzętu. Aby ją zmienić użytkownik musi zmodyfikować sterownik zegara i dokonać ponownej inicjalizacji systemu. Najdłuższy okres czasu, przez który proces zajmuje jednostkę centralną, zanim ją nie przeliczy kolejki aktywnych procesów, nazwana zostanie kwantem czasu (ang. *time slice*). Początkowo jest on równy dwu okresom impulsów przerywających, czyli wynosi 20 ms. Długość trwania tego czasu może być zmieniona np. przez modyfikację jednej ze zmiennych systemowych. System operacyjny OS-9 umożliwia także szybką odpowiedź na przerwanie dla aplikacji o szczególnie krytycznych wymaganiach czasowych. Dokonywane jest

to przez anulowanie wykonywania zadania (ang. *pre-empting*) w momencie, gdy proces o wyższym priorytecie stanie się procesem aktywnym. Wówczas proces o niższym priorytecie traci pozostałość swojego kwantu czasu i zostaje z powrotem umieszczony w kolejce procesów aktywnych.

Dla efektywnego wykorzystania możliwości kształtowania architektur software'u OS-9 dostarcza wielu mechanizmów synchronizacji procesów i komunikacji pomiędzy nimi. Są nimi sygnały (ang. *signals*), alarmy (ang. *alarms*), zdarzenia (ang. *events*), potoki (ang. *pipes*) i moduły danych (ang. *data moduls*). Sygnały używane są do synchronizacji współbieżnych procesów. Alarmy wysyłają sygnały lub inicjują wykonanie podprogramów w określonym czasie. Zdarzenia mogą być wykorzystane do synchronizacji dostępu do wspólnych zasobów systemu dla procesów współbieżnych. Potoki służą do przesyłania danych pomiędzy procesami. Moduły danych natomiast umożliwiają transfer danych oraz wspólne korzystanie z danych w nich zawartych przez różne procesy.

Powyższe mechanizmy uruchomiane są przez użytkownika jako wywołania systemowe. Za ich obsługę odpowiedzialne jest jądro systemu. W systemie OS-9 wyróżnia się dwa główne rodzaje wywołań systemowych: wywołania funkcji wejścia/wyjścia, takich jak odczyt lub zapis, oraz wywołania funkcji systemowych, jak np.: zarządzanie pamięcią, inicjalizacja systemu, tworzenie nowych procesów i zarządzanie ich wykonywaniem czy też obsługa przerwań i stanów wyjątkowych. Gdy pojawia się wywołanie systemowe, jądro systemu sprawdza rodzaj wywołania i bezpośrednio wykonuje jedynie wywołania funkcji systemowych. Wywołania dotyczące funkcji wejścia/wyjścia kierowane są do odpowiednich modułów zarządzania plikami (ang. *file managers*). Dodatkowo wywołania funkcji systemowych dzielą się na dwa typy: wywołania z poziomu użytkownika (ang. *user state*) i wywołania z poziomu systemu (ang. *system state*). Jedną z głównych cech tych ostatnich jest fakt, że nie występuje podczas ich wykonywania przełączanie zadań - nie są więc one przerywane przez inne procesy, poza obsługą przerwań.

4. Tryb "MESSAGE BROADCAST"

Praca wieloprocesorowych systemów mikroprocesorowych do analizy obrazów wymaga przesyłania dużej ilości danych do poszczególnych modułów składowych systemu (modułów hardware'owych). Zadaniem wysokosprawnej transmisji w trybie "MESSAGE BROADCAST" jest transmisja pomiędzy modułami hardware'owymi zrealizowanymi w standardzie VME.

W dotychczasowych klasycznych rozwiązaniach transmisji po magistrali VME w przekazie uczestniczyły dwa podmioty: nadawca i odbiorca. Ze względu na parametr szybkości przetwarzania korzystnie byłoby zwiększyć liczbę podmiotów uczestniczących w transmisji - w szczególności aby w transmisji uczestniczyło kilku odbiorców. Przesyłanie informacji do kilku odbiorców jednocześnie jest istotne nie tylko ze względu na parametr szybkości przetwarzania, ale także daje możliwość precyzyjnej synchronizacji poprzez jednoczesne otrzymanie informacji przez kilka modułów.

Najnowsza wersja standardu magistrali VME - VMEbus rev. D [16] opublikowana w 1990 roku przez VITA (międzynarodowa organizacja producentów i handlowców sprzętem standardu VME), nie dopuszcza przesyłania danych do kilku odbiorców jednocześnie. Brak takiego trybu przesyłania danych stanowi niewątpliwie wadę tego rozpowszechnionego i uznanego standardu. Różne firmy (np. FORCE COMPUTERS) korzystając z niewykorzystanych możliwości dolnego złącza VME wprowadziły już swoje własne mechanizmy transferu danych do kilku odbiorców jednocześnie, ale ilość przekazywanej w ten sposób informacji jest zwykle bardzo ograniczona (maksymalnie kilka bajtów). Takie tryby

transferu danych należy zakwalifikować raczej do mechanizmów typu MESSAGE PASSING i są to tryby wystarczająco skuteczne w wielu aplikacjach. Niestety, wiele innych aplikacji wymaga przesyłania znacznie większych porcji danych do wielu odbiorców równocześnie.

W ramach podjętych prac opracowano standard, który wykorzystując mechanizmy wprowadzonego w VMEbus rev.D transferu blokowego, wzbogacił go o możliwość przesyłania danych do wielu odbiorców równocześnie.

System równoczesnego przesyłania danych w trybie blokowym (zwany dalej systemem RTB - równoczesny transfer blokowy) do wielu odbiorców po magistrali VME tworzą:

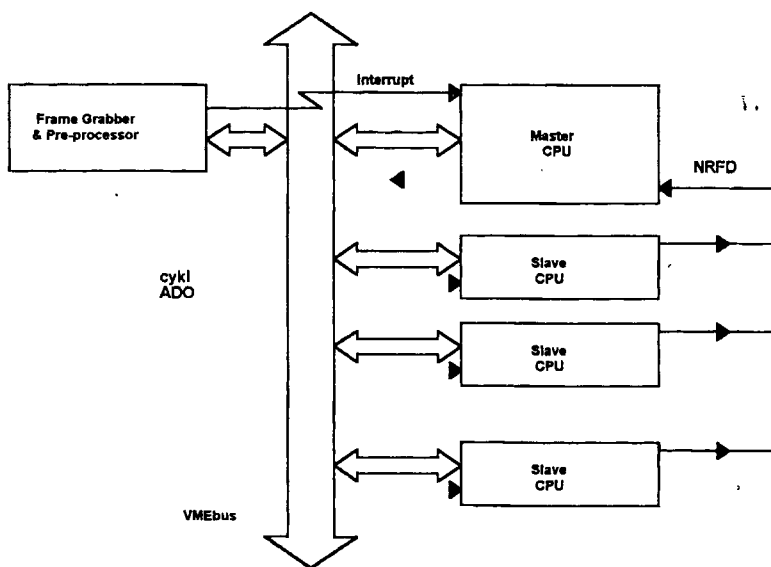
- moduł sterujący RTB wyposażony w możliwości realizacji cykli D32 i A32 (z zaimplementowanym Master RTB),
- moduł będący źródłem przesyłanych danych wyposażony w możliwości realizacji transferów blokowych D32, A32 zgodnie z VMEbus rev.D (bez żadnych sprzętowych zmian w tym module),
- modułów równocześnie odbierających dane wyposażony w możliwości realizacji cykli D32 i A32 (z zaimplementowanym RTB).

Przesłanie danych w trybie RTB składa się z dwóch faz: pierwszej - adresacji i inicjacji, oraz drugiej - transferu podobnie jak w przypadku transferu blokowego.

Podczas fazy pierwszej moduł Master RTB po ustaleniu źródła danych do transferu w trybie RTB przeprowadza adresację modułów mających odbierać dane. W tym celu przeprowadza cykl typu ADO (address only) zgodny z VME rev.D Wysterowuje linie /AS=0 i /IACK=1 i umieszcza na liniach modyfikatorów adresu AM0-5 kod transferu RTB. (Kod ten nie został ustalony ze względu na możliwość wprowadzenia przez VITA nowego standardu - w rozwiązaniach praktycznych planuje się umieszczanie przełączników ustalających ten kod *). Na liniach A2 - A22 RTB Master podaje dla jakich modułów będą przesyłane dane w trybie RTB. Każdy z odbiorców ma przyporządkowaną jedną z linii adresowych o numerze odpowiadającym zajmowanemu slotowi. Oczywiście każdy z tych modułów może ten numer zmieniać w sposób sprzętowy (przełącznik na płycie) lub programowy (rejestr wewnętrzny). Pojawienie się logicznej jedynki na odpowiedniej linii adresowej świadczy o wybraniu danego modułu do transferu w trybie RTB. Stan linii adresowych A23-28 w tej fazie nie ma znaczenia. Stan linii A29 i A30 zależy od szerokości danych zapewnianych przez źródło danych do transferu w trybie RTB. ("00" - transfer o szerokości 1 bajtu, "01" - 2 bajtów, "11" - 4 bajtów). Stan linii A31 odzwierciedla status RTB. Gdy jest to "1" to następny transfer blokowy będzie transferem RTB. Gdy jest to "0" to jest to koniec transferów RTB. W ten sposób przełączany są w modułach odbierających dane w trybie RTB mechanizmy potwierdzenia transferu /DTACK i /NRFD - będzie to omówione dalej. Należy zaznaczyć, że zgodnie z VMEbus rev.D cykl ADO nie zakłada żadnej odpowiedzi od modułów adresowanych w ten sposób. Master przeprowadzający cykl ADO nie śledzi stanu linii /DTACK ani /BERR. Jeżeli jakiś moduł nie jest do transmisji RTB gotowy to nie ma on w tej chwili szansy na zgłoszenie tej niegotowości. Ta niedogodność ma uzasadnienie w chęci jak najmniejszego ingerowania w standard VME. Oczywiście sytuację niegotowości do transferu rozwiąże faza druga RTB, gdzie moduł niegotowy wysteruje linię /BERR i /RETRY podobnie jak przy zatwierdzonym standardzie transferu blokowego zgodnym z rev.D. Moduł wybrany do odbioru danych w trybie RTB sam ustala sobie adres początkowy w lokalnej pamięci i kierunek składowania danych (w stronę rosnących lub malejących adresów)..

Po zakończeniu fazy pierwszej następuje druga w której Master RTB przeprowadza standardowy cykl transferu blokowego. Jednak aby zapewnić możliwość jednoczesnego odbioru danych przez kilka modułów wprowadzono mechanizm linii /NRFD (Not Ready For Data). Wykorzystano do tego celu linię przeznaczoną dla użytkownika z dolnego złącza

magistrali. (Z powodów już wspomnianych wyżej zostawiono użytkownikowi wybór między lokacjami :A17,A25,C17,C25 -wybór za pomocą przełączników umieszczonych na płycie.) Master RTB po zaadresowaniu modułu będącego źródłem danych do RTB i odpowiednim wysterowaniu linii /DS0, /DS1 i /LWORD śledzi stan linii /DTACK i /NRFD. Linia /DTACK steruje modułem źródłowym, linią /NRFD wszystkie moduły odbierające dane. Obie linie są w standardzie OPEN COLLECTOR. Niski stan linii /DTACK świadczy o tym, że na magistrali danych znalazły się dane z modułu źródłowego. Każdy z odbiorców po stwierdzeniu tego faktu wczytuje dane do swoich wewnętrznych buforów i zaprzestaje sterować linią /NRFD (normalnie stan "0" tzn. wysterowanie). Moduł Master RTB po stwierdzeniu stanu nieaktywnego na tej linii (świadczącego o tym, że każdy z odbiorców przeczytał dane) zwalnia wysterowane linie strobów danych DS1 lub/i /DS0 i ewentualnie /LWORD kończąc cykl przesłania danych. Następne cykle odbywają się według powyższych reguł. Należy przypomnieć, że zgodnie z VMEbus rev.D transfer blokowy nie może przekraczać ramek adresowych xxxxxx00 hex, co ogranicza jednorazowe przesłanie w tym trybie do 256 bajtów. Jednak dzięki takiemu ograniczeniu nie dochodzi do "zmonopolizowania" systemu wieloprocessorowego przez jeden z modułów. Bowiem zakończenie cyklu (tj. zwolnienie linii /AS) umożliwia arbitraż w wieloprocessorowych systemach opartych o VME. Inicjalizacja transferu następowałaby poprzez zgłoszenie gotowości danych do transmisji ze źródła przez uaktywnienie przerwania. Reakcja modułu RTB Master rozpoczynałaby transfer w sposób wyżej opisany.



Rys. 4. Przesyłanie danych w trybie "Message Broadcast".

Do dnia napisania niniejszego sprawozdania nie został ustalony żaden standard przesyłania danych w trybie blokowym do kilku odbiorców jednocześnie po magistrali VME. Podczas ostatniego spotkania VITA, które odbyło się w dniach 10-14 stycznia 1994 w Salt Lake City USA, kilku producentów zgłosiło po raz kolejny swoje propozycje rozwiązania tego problemu. Były to propozycje:

- Mercury Computer Systems - RACEway Interlink,
[Barry Isenstein at Mercury Computer Systems (barry@mc.com)]
- Sky Computers - SKYChannel,
- INMOS - IEEE P1355,
[Colin Whitby-Stevens at INMOS (colinws@isnet.inmos.co.uk)]
- National Semiconductor's - CSPI QuickRing technology.
[Sean Long at National Semiconductor (tselc@tevm2.nsc.com)]

Jak dotychczas VITA nie zaakceptowała żadnej z tych propozycji jako swojego standardu. (Szczegółowe informacje na temat proponowanych standardów nie są w tej chwili dostępne wprost - podano e-mail do reprezentatów każdego z producentów).

5. Podsumowanie

Projekt przewiduje opracowanie oraz realizację sprzętową systemu czasu rzeczywistego do zastosowań w komputerowej analizie obrazów. Istota koncepcji polega na wykorzystaniu procesorów specjalizowanych do potokowej obróbki wstępnej danych obrazowych i specjalnie zaprojektowanej architektury wieloprocessorowej w standardzie VME, optymalizowanej ze względu na wymagania techniki przetwarzania, analizy i rozpoznawania danych obrazowych. Ponieważ głównym celem budowy systemu jest dostosowanie jego możliwości do śledzenia i rozpoznawania obiektów ruchomych, przewiduje się możliwość rekonfiguracji systemu w celu dopasowania do rodzaju śledzonego obiektu i charakteru jego ruchu. System będzie analizował dwojakiego rodzaju ruchome obrazy: nieruchome kadry wewnątrz których zachodzi ruch pewnych obiektów (np. zastosowania w robotyce przemysłowej i wizyjnej analizie ruchu drogowego) oraz obrazy pochodzące z ruchomej kamery (np. zamontowanej na ramieniu robota lub na ruchomym pojeździe). W tym drugim przypadku wymagania odnośnie czasowych parametrów systemu będą musiały sprostać wymaganiom wynikającym z realizacji algorytmów typu "optical flow".

W projekcie zostaną zastosowane najnowsze technologicznie komponenty: procesory specjalizowane firm HARRIS i INMOS, pamięci wieloportowe firm MICRON, CYPRESS i TOSHIBA, mikroprocesory 32-bitowe i procesory sygnałowe firmy MOTOROLA, profesjonalne kontrolery magistrali VME firmy NEW-BRIDGE, układy programowalne FPGA i EPLD firmy XILINX.

Obszarem zastosowań proponowanych w projekcie architektur systemu wieloprocessorowego do obróbki i analizy obrazów w czasie rzeczywistym jest wysoko zaawansowana robotyka oraz układy śledzenia i naprowadzania obiektów ruchomych (ruch drogowy, wojskowość).

Literatura

- [1] Alexandridis N.: Design of Microprocessor-Based Systems. Prentice Hall - Englewood Cliffs, New Jersey 1993.
- [2] Decegama A.L.: The Technology of Parallel Processing - Parallel Processing Architectures and VLSI Hardware. Prentice Hall International Inc. - New Jersey 1989.
- [3] Dewilde P.M., Deprettere E.F.A., Nouta R. (1985): Parallel and Pipelined Implementation of Signal Processing Algorithms. Prentice-Hall.
- [4] Dibble P.C.: OS-9 INSIGHTS an advanced programmers guide. Microware 1992.
- [5] Duff M.J.B.: Parallel Processors for Digital Image Processing. Advances in Digital Image Processing. Plenum Press 1979.
- [6] Haralick R.M., Shapiro L.G.: Computer and Robot Vision, Addison-Wesley Publishing - California 1992.
- [7] Hwang K., Briggs F.: Computer Architecture and Parallel Processing. Mc Graw-Hill 1984.
- [8] Iwata Y., Kawamata M., Higuchi T.: Design of fine grain VLSI array processor for real-time 2-D digital filtering. IEEE International Symposium - Chicago 3-6.5.1993.
- [9] Lo K.C. (1993): Parallel random sampling with multiprocessor system. IEEE International Symposium - Chicago 3-6.5.1993.
- [10] Mikrut Z., Tadeusiewicz R.: Automatyczna wizualna identyfikacja obiektów na zrobotyzowanych stanowiskach pracy. Elektrotechnika. Tom 9. Zeszyt 3-4. 1990.
- [11] Pachowicz P., Tadeusiewicz R.: Porównanie systemu CESARO z wybranymi systemami analizy i rozpoznawania obrazów. Elektrotechnika, tom 1, nr 2. 1982.
- [12] Peterson W.D.: The VMEbus Handbook. A Vita Publications 1990.
- [13] Ravn O.: Robot Vision. TEMPUS Summer School - Kraków 1992.
- [14] Tadeusiewicz R.: Systemy wizyjne robotów przemysłowych. WNT Warszawa 1992.
- [15] Tadeusiewicz R., Flasiński M.: Rozpoznawanie obrazów. PWN Warszawa 1991.
- [16] VITA: VMEbus Revision D. 1990.