

Eugeniusz KORBIEL

PROBLEMY I REALIZACJA INTERFEJSU IEC
W UKŁADACH MIKROPROCESOROWYCH

Streszczenie. W artykule przedstawia się problemy współpracy urządzeń zbudowanych na bazie mikroprocesora, w systemie zbierania danych opartego na standardowym systemie interfejsu IEC BUS.

Omówione są programowe i hardware'owe korzyści płynące z tej współpracy oraz podany jest schemat układu interfejsu dla szybkości transmisji ≤ 10 kbajt/s, oparty na przedstawionym programie kontroli przerw interfejsu.

1. WPROWADZENIE

Na przestrzeni ostatnich kilkunastu lat we wszelkich zastosowaniach elektroniki dał się głęboko odczuć wpływ rozwoju technologicznego związanego z układami scalonymi małej i średniej skali integracji. Związane było to jednak głównie ze zmniejszeniem poboru mocy i gabarytów urządzeń oraz spadkiem cen, co znakomicie wpłynęło na możliwość praktycznego obcowania z niedostępnymi dawniej urządzeniami.

Zmiany te jednak w dużo mniejszym stopniu dotyczyły podstawowych funkcji i zasad działania tych urządzeń, gdyż zazwyczaj realizowano na innych technologicznie elementach te same koncepcje układowe.

Jakościowym skokiem naprzód jest dopiero zamocowanie urządzeniom możliwości programowej organizacji jego pracy oraz możliwości analizy i opracowywania danych, już na poziomie prostych urządzeń pomiarowych.

Dalszym krokiem byłoby stworzenie możliwości budowy systemu zbierania danych oraz zdalnego programowania pracy tych urządzeń, a więc stworzenie im możliwości porozumiewania się między sobą.

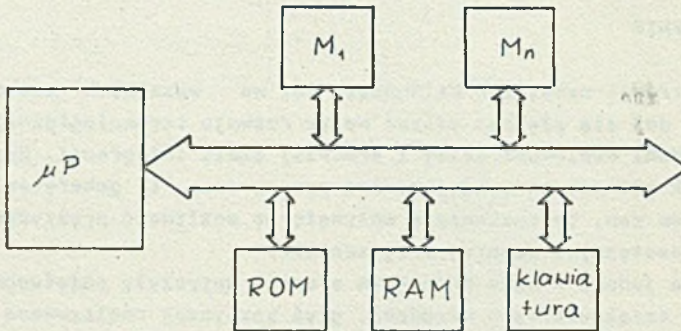
Rozwiązaniem pierwszego problemu stał się mikroprocesor, a drugiego standardowy system interfejsu IEC BUS.

Niniejszy artykuł omawia problemy wzajemnego powiązania mikroprocesorów z IEC BUS i przedstawia koncepcję realizacji standardowych funkcji interfejsu poprzez program kontroli przerw interfejsu.

2. MIKROPROCESORY W URZĄDZENIACH POMIAROWYCH

Zastosowanie mikroprocesora w przyrządach pomiarowych umożliwia m.in.:

- wspomaganie metody pomiarowej: możliwość uśredniania wyników pomiarów przez mikroprocesor pozwala na osiągnięcie takich parametrów, jak rozdzielczość 1ps w przypadku częstotściomierza HP 5345 czy 1 μ V, 1 μ A, 1 μ Ω dla multimetru DATRON 1071, a wprowadzenie programu autokalibracji pozwoliło na usunięcie błędów systematycznych pomiaru,
- sterowanie z klawiatury alfanumerycznej zamiast z pokręteł,
- automatyczne skalowanie przyrządu,
- konwersja danych lub parametrów,
- autotestowanie,
- naturalność współpracy z maszyną cyfrową ze względu na cyfrowy charakter przetwarzania danych w przyrządzie.



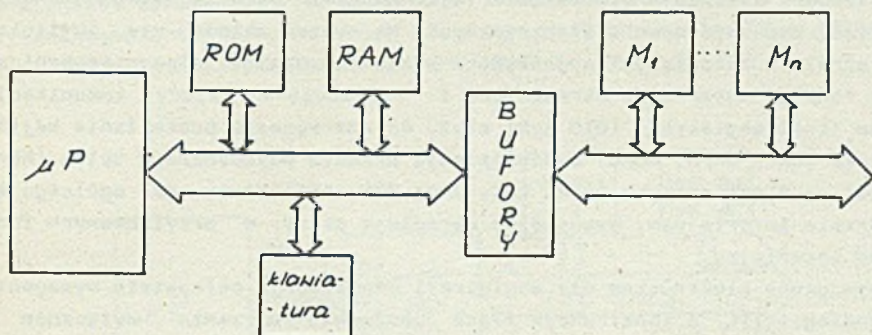
Rys. 1. Elementarna konfiguracja układu przyrządu z mikroprocesorem M_1 - moduł przyrządu (karta, wkładka)

Najprostszą konfigurację przyrządu z mikroprocesorem przedstawia rys. 1 [6].

W tej konfiguracji moduły M_1 przyrządu podłączone są jak urządzenia zewnętrzne do maszyny cyfrowej i komunikacja pomiędzy modułami odbywa się za pośrednictwem mikroprocesora. Układ ten ma jednak tę istotną wadę, że w przypadku konieczności szybkiej transmisji międzymodułowych, typowe mikroprocesory wykonane w technologii MOS mogą okazać się zbyt wolne, aby móc je obciążać. Można to obejść poprzez wprowadzenie osobnych magistral międzymodułowych, co pociąga jednak za sobą komplikacje w postaci dodatkowych łącz i układów pośredniczących.

Tych wad nie posiada konfiguracja z rys. 2.

Transmisje pomiędzy modułami odbywają się na magistrali przyrządu odseparowanej buforami od wewnętrznej magistrali mikroprocesora. Zapewnia to możliwość współbieżnej i niezależnej pracy mikroprocesora i modułów,



Rys. 2. Konfiguracja przyrządu z rozdzielonymi magistralami

jakkolwiek w dalszym ciągu mikroprocesor pełni rolę dyrygenta. Nie można przy tym wykluczyć, że poszczególne moduły mogą być wyposażone we własne szybkie (bipolarne) mikroprocesory, realizujące np. w oscyloskopie dyskretną transformatę Fouriera sygnału wejściowego [6].

Analogiczne problemy występują na następnym hierarchicznie wyższym poziomie komunikacji pomiędzy urządzeniami, w systemie zbierania danych.

Zapewnienie komunikacji pomiędzy n zazwyczaj różnymi urządzeniami, w skrajnym przypadku prowadzi do konieczności zbudowania $n(n-1)/2$ układów pośredniczących. Pojęcie o tej sytuacji daje wyobrażenie sobie konferencji, na której każdy uczestnik mówi innym językiem, a każdej parze rozmówców trzeba zapewnić tłumacza. Narzucającym się rozwiązaniem tej sytuacji jest przyjęcie jednego wspólnego języka konferencyjnego. Innymi słowy konieczna jest w systemie standaryzacja zasad komunikacji pomiędzy urządzeniami.

Wprowadzenie jednolitego protokołu komunikacyjnego zredukowałoby po stronie hardware'owej liczbę koniecznych układów pośredniczących (tłumaczy) do n , przy czym każdy układ mógłby być zaprojektowany bez konieczności znajomości całego systemu, a po stronie software'owej podobnie ujednoliciłoby programy we/wy.

Najpoważniejszą próbą realizacji takiego standardu jest niewątpliwie tzw. standard IEC BUS, powstały w oparciu o opracowania firmy Hewlett-Packard i zwany także HPiB, IEEE 488, PLUS BUS, GPIB, ANSI Std MC 1.1.

3. IEC BUS

Projekt tego standardu został przyjęty równocześnie po obu stronach Atlantyku w 1975 r. Datujący się od tamtej pory dynamiczny rozwój zastosowań podkreślił jego zalety, choć również obnażył braki, co zmusiło jego projektantów do dalszej pracy, której rezultaty oczekiwane są już w najbliższych miesiącach [2].

Standard definiuje system interfejsu poprzez podanie wymagań funkcjonalnych, mechanicznych i elektrycznych. Na system składa się 16-liniowa magistrala z określonymi sygnałami i poziomami napięć, złącza mechaniczne oraz funkcje interfejsu określające i realizujące zasady komunikacji. Ośmiem linii magistrali (DIO 1-8) służy do szeregowego przesyłania bajtów, 3 linie (DAV, NRFD, NDAC) synchronizują przesył pojedynczego bajtu (handshake), a 5 pozostałych (ATN, EOI, SRQ, REN, IFC) służy do ogólnego zarządzania interfejsem, wymuszając określone stany w przyłączonych funkcjach interfejsu.

Wymagania elektryczne dla magistrali odpowiadają całkowicie wymaganiom technologii TTL, a konstrukcja złączy umożliwia tworzenie wyłącznie za pomocą kabli dowolnej konfiguracji magistrali, której jedynymi ograniczeniami są łączna długość kabli (20 m) oraz ilość urządzeń (15).

Podstawowe charakterystyki operacyjne systemu:

max szybkość transmisji - 0,5 Mbajt/s na dł. 20 m
 - 1 Mbajt/s na dł. 10 m

słowo - bajt (8 bitów)

przerwania - szybka identyfikacja 8 urządzeń

- wolna identyfikacja 14 urządzeń

rodzaj transmisji - asynchroniczna, trójprzewodowa, umożliwia transmisję do wielu odbiorców.

Całkowita niezależność systemu od funkcji urządzeń pozwala poprzez odpowiednie funkcje interfejsu na współpracę z dowolnym urządzeniem. Żle się jednak stało, że obecna wersja systemu dopuszcza dowolne kody sterujące funkcjami urządzeń. W tym wypadku troska o uniwersalność systemu doprowadziła do sytuacji, w której łatwo o niekompatybilność pochodzących z różnych firm urządzeń. Uniemożliwia to także ujednoczenie programów we/wy, utrudniając znacznie pracę programisty [11].

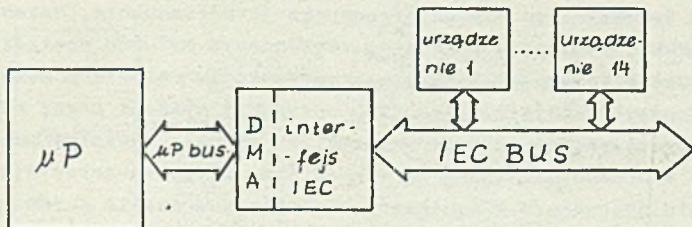
Szczegółowy opis standardu IEC BUS można znaleźć [1, 3, 4, 5, 12, 17]. Dla dalszych rozważań autor zakłada dobrą znajomość tego standardu przez Czytelnika.

4. MARIAŻ MIKROPROCESORA Z IEC BUS

Porównanie charakterystyk operacyjnych IEC BUS z charakterystykami typowych mikroprocesorów wskazuje na naturalny związek pomiędzy nimi.

Długość słowa magistrali IEC jest typową długością dla większości mikroprocesorów a wielokrotnością lub podzielnikiem pozostałych, tak więc przetwarzanie słowa magistrali nie wprowadza żadnych dodatkowych trudności programowych czy hardware'owych. Również ograniczenia na długość i ilość urządzeń nie stanowią problemu w mikrosystemach, gdyż rzadko obsługują one więcej niż kilka urządzeń zewnętrznych. Rzadko również, którekolwiek z urządzeń w mikrosystemie zdolne jest transmitować dane z prędko-

cię kilkaset tysięcy bajtów na sekundę, która narzucałaby ostre wymagania projektantowi interfejsu. Wiążący się z tym problem kanału DMA w mikrosystemie znajduje niezwykle naturalne rozwiązanie (rys. 3) [13].



Rys. 3. Kanał DMA w przypadku wykorzystania IEC BUS jako magistrali we/wy mikroprocesora

Jednolity protokół komunikacyjny umożliwia mikrokomputerowi transmisję DMA ze wszystkimi urządzeniami przyłączonymi do magistrali IEC poprzez tylko jeden układ interfejsu. Z kolei problem różnej szybkości urządzeń, istotny w przypadku jednoczesnej transmisji do wielu urządzeń, może być łatwo rozwiązany poprzez proste buforowanie danych.

Dodatkową korzyścią ofiarowywaną przez IEC BUS mikrosystemom jest możliwość niezależnych transmisji pomiędzy urządzeniami zewnętrznymi, nieangażujących jednostki centralnej. Należy zauważyć, że każde z urządzeń zewnętrznych może zawierać własny mikroprocesor, stanowiąc w ten sposób mikrosystem sam w sobie. Tak więc te rozważania mogą być ważne dla każdego z urządzeń przyłączonych do magistrali.

5. ZAGADNIENIE PROJEKTOWANIA INTERFEJSU POMIĘDZY MAGISTRALĄ IEC A MIKRO-PROCESOREM

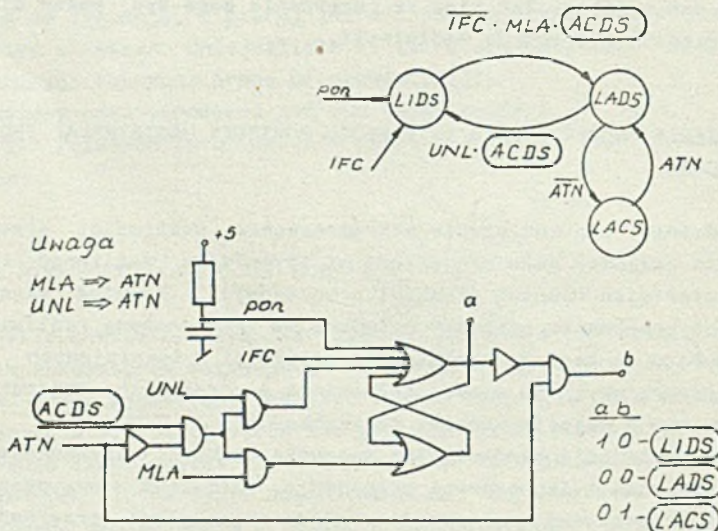
Wprowadzenie, poprzez użycie mikroprocesora, możliwości przetwarzania na poziomie urządzeń dało projektantowi interfejsu możliwość realizacji funkcji interfejsu również środkami programowymi. Powstał więc problem optymalnego kompromisu pomiędzy hardware'ową i programową realizacją. Możliwości wyboru są bardzo szerokie, od adaptacji istniejących rozwiązań czysto hardware'owych do niemal całkowicie programowych. Konieczność kompromisu wynika z relacji kosztów do szybkości.

Droższa realizacja hardware'owa zapewnia szybkość transmisji ograniczoną praktycznie tylko czasami propagacji, natomiast w przypadku tańszej realizacji programowej konieczność programowej obsługi przerw związanych z transmisją, a następnie operacje nad bitami statusowymi wymagają wykonania kilkudziesięciu instrukcji. Oznacza to stratę czasu rzędu setek

mikrosekund, ograniczając w ten sposób szybkość transmisji w najlepszym razie do kilku kilobajtów na sekundę. Trzeba jednak zauważyć, że realizacja czysto hardware'owa czy programowa jest zresztą niemożliwa, gdyż w pierwszym przypadku, ze względu na sterującą rolę mikroprocesora w urządzeniu, będzie on odgrywał rolę łącznika pomiędzy funkcjami interfejsu a funkcjami urządzeń, co wymaga stosownego oprogramowania. Natomiast w drugim przypadku, pomijając trywialne zagadnienie buforów pomiędzy magistralą IEC a wewnętrzną magistralą mikroprocesora, niektóre wymagania standardu IEC przekraczają możliwości programowej obsługi przez mikroprocesory. Dotyczy to czasów reakcji na niektóre sygnały ogólnego zarządzania magistralą, a przede wszystkim na sygnał ATN, kiedy to interfejs musi przygotować się do odbioru komunikatów kontrolera w czasie ≤ 200 ns. Tak więc nawet dla silnie programowej realizacji interfejsu występuje układ szybkiej logiki, umożliwiający pracę interfejsu zgodnie z wymaganiami standardu.

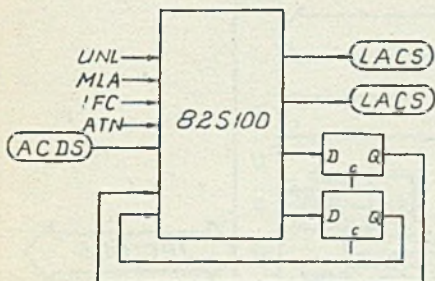
W obu przykładach bufory magistrali oraz układy we/wy mikroprocesora mogą być te same. W przypadku buforów magistrali mogą być to układy MC 3440/3441/3443/3448 [10], natomiast układy we/wy związane są z typem mikroprocesora, np. dla Intela 8080 jest to układ 8212 lub 8225, a dla Motorola M 6800 układ MC 6820 [9].

Ściśle hardware'owy interfejs można zrealizować na kilka sposobów, zależnie od dostępnych elementów. Najbardziej pracochłonna jest realizacja z ogólnie dostępnych elementów SSI i MSI. Przykład realizacji funkcji odbiorcy (L) na tych elementach przedstawiony jest na rys. 4.



Rys. 4. Realizacja funkcji odbiorcy (L) w wersji podstawowej (basic listener) na elementach małej skali integracji (oznaczenia wg [1]). Dla sygnałów \overline{pon} , IFC , ATN obowiązuje logika negatywna. Sygnały MLA i UNL zachodzą tylko w koniunkcji z ATN

Innym sposobem realizacji funkcji interfejsu jest wykorzystanie elementów FPLA (np. Signetica 82S100) z dołączonymi przerzutnikami typu D, jako elementami pamięciowymi [7]. Funkcję odbiorcy zrealizowaną na tych elementach przedstawia rys. 5.



Rys. 5. Realizacja funkcji (L) w wersji podstawowej na elemencie FPLA Signetica 82S100

Zważywszy, że na jednym elemencie 82S100 można zrealizować więcej niż jedną funkcję interfejsu, a także układ dekodera rozkazów i adresu, ten sposób realizacji daje istotny zysk na pracochłonności, zajmowanym obszarze i mocy pobieranej.

Logicznym uwieńczeniem realizacji hardware'owej jest oczywiście pojedynczy układ LSI, zapewniający dopasowanie dowolnego urządzenia do magistrali IEC poprzez realizację funkcji interfejsu na jednym "chipie".

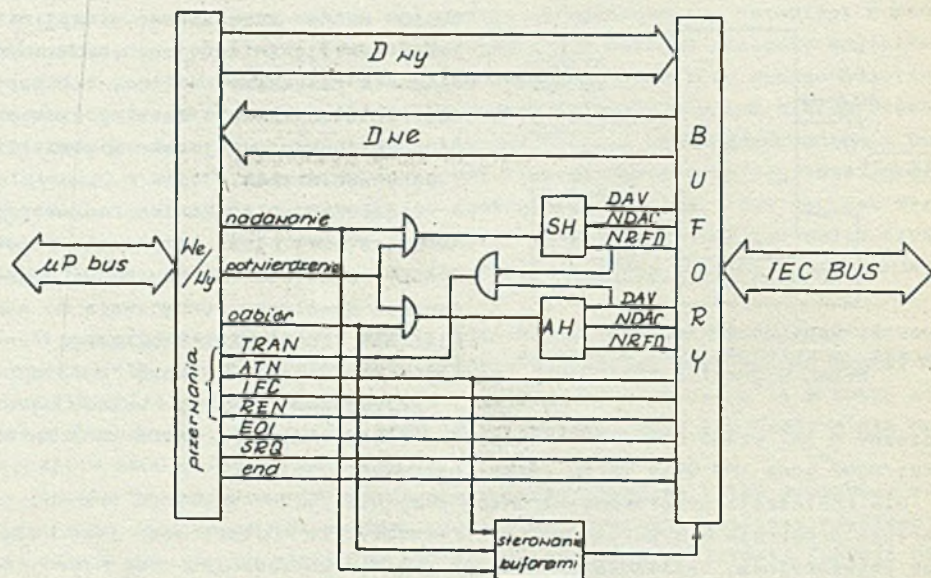
Przykładem takiego układu jest opisany w [8] układ HEF 4738 Philipsa, a także układy MC 68488 [18] Intel 8291/8292 oraz TMS 9914 firmy Texas Instrument [15].

Dla realizacji programowo-hardware'owej nie tylko szybkość transmisji określa proporcje w wyborze środków. Zauważmy, że mikroprocesor jest układem sekwencyjnym, realizującym kolejno pojedyncze zadania, tak więc zaangażowany w zarządzanie interfejsem, nie może w tym czasie kontrolować pracy urządzenia i na odwrót. W takim przypadku przerwanie przychodzące z interfejsu może spowodować np. stratę aktualnie realizowanego pomiaru lub przerwanie operacji sterującej, na co w wielu przypadkach nie można sobie pozwolić. Wspomniany wyżej układ szybkiej logiki wysterowuje bufory magistrali do odbioru komunikatów interfejsu w wymaganym czasie ≤ 200 ns bez interwencji mikroprocesora, co pozwala na zakończenie nieprzerwalnych procedur. Jest to bardzo istotne ze względu na tryb kontroli szeregowej, kiedy kolejne zmiany stanu sygnału ATN mogłyby dezorganizować pracę urządzeń nie żądających obsługi. Idąc dalej można, jeśli zaszczyłyby potrzeba, wprowadzić dodatkowe układy zdolne nadać status również bez zaangażowania mikroprocesora [14].

Istotnym problemem jest także realizacja funkcji inicjatora [SH] i akceptora współpracy [AH], gdyż uczestniczą one w każdej pojedynczej transmisji.

Realizacja hardware'owa tych funkcji, wymagająca zaledwie kilku bramek, w połączeniu z układem szybkiej logiki zapewniłaby już możliwość pracy z szybkością rzędu 10 kbajtów/s, nawet dla całkowicie programowej realizacji pozostałych funkcji. Ponieważ większymi wymaganiami na szybkość

charakteryzują się praktycznie tylko szybkie pamięci masowe, proponowane realizacja ma charakter dość uniwersalny, biorąc pod uwagę koszt realizacji i możliwości aplikacyjne. Rys. 6 przedstawia tę realizację interfejsu na poziomie schematu blokowego.

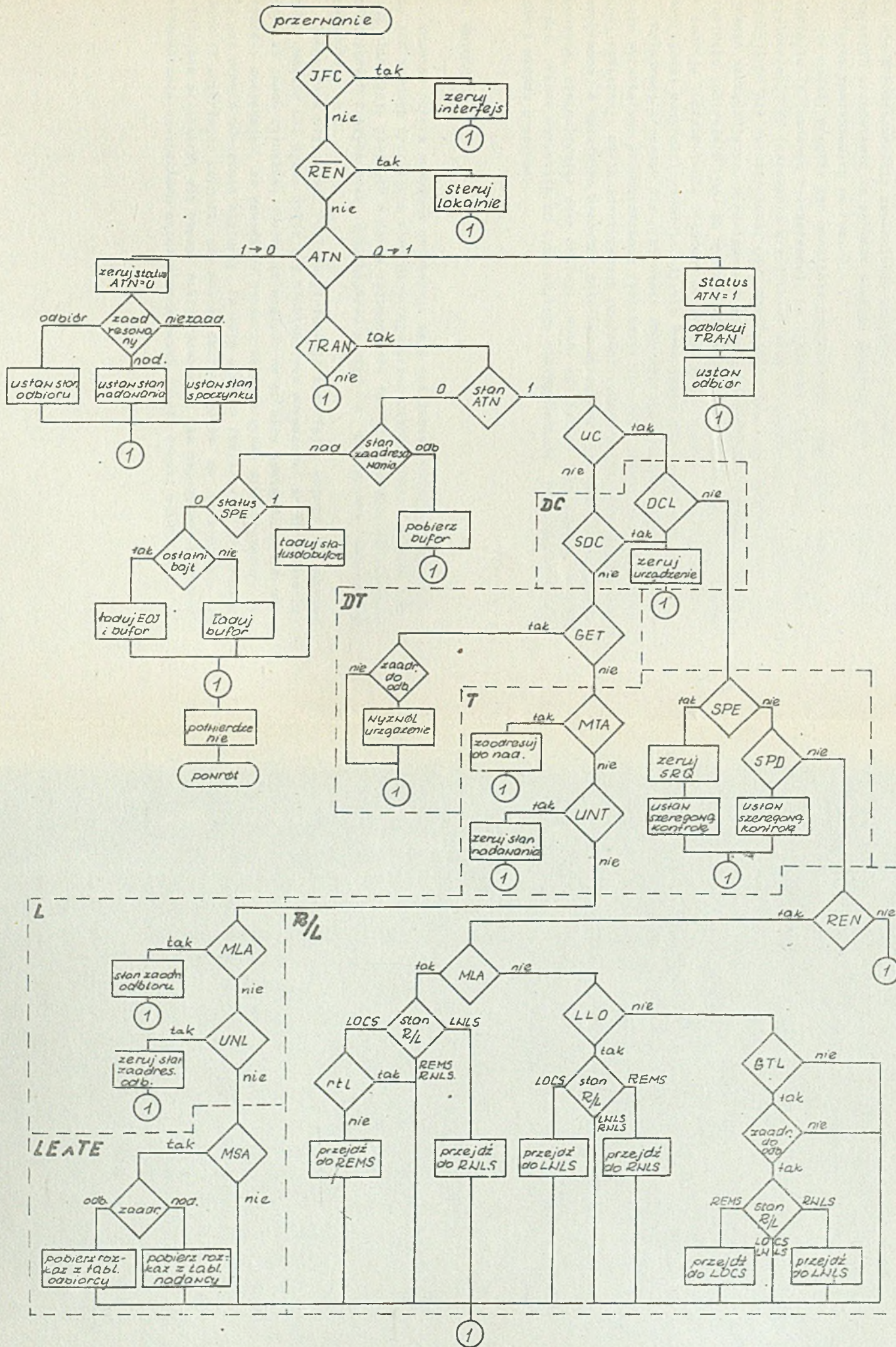


Rys. 6. Projekt uniwersalnego interfejsu dla średnich szybkości transmisji (część hardware'owa)

Rozważmy na tym przykładzie problemy programowe.

Realizuje on wszystkie funkcje interfejsu, z wyjątkiem funkcji kontroler, jako nietypowej dla większości urządzeń i funkcji kontroli równoległej (PP), jako zbyt cennej ze względu na średnią szybkość. Przerwania w tych układach będą związane z sygnałami ogólnego zarządzania magistralą, z wyjątkiem sygnału SRQ (żądanie obsługi), który jest przerwaniem dla funkcji kontroler oraz EOI (identyfikacja) ze względu na brak kontroli równoległej. Pozostałe przerwania związane są z sygnałami współpracy (handshake) określającymi stan transmisji bajtu. Konkretnie w przypadku odbierania komunikatów z magistrali będzie to sygnał DAV (dane ważne) żądający przyjęcia danych, a dla nadawanie komunikatów sygnał NRFD oznaczający gotowość przyjęcia kolejnego komunikatu przez magistralę.

Spośród tych sygnałów najwyższy priorytet będą miały sygnały IFC (zewnętrzny interfejs) oraz REN (koniec zdalnego sterowania), o ile oczywiście będą w danym przypadku zastosowane. Wynika to z nieprzekraczalnego czasu ich obsługi $\leq 100 \mu s$, podczas gdy układ szybkiej logiki pozwala obsłużyć przerwania od sygnału ATN bez specjalnych ograniczeń czasowych.



Rys. 7. Schemat blokowy programu kontroli przerwania interfejsu (oznaczenia poza TRAN wg [1])

Ostatni poziom przerwń tworzą sygnały transmisji (TRAN) DAV i NRFD, co wynika z faktu, że są one odblokowywane (lub sprawdzane) dopiero po ustaleniu się sygnału ATN. Należy przy tym zauważyć, że przerwania od sygnału ATN związane są z każdą zmianą jego stanu (0 → 1 i 1 → 0), co wiąże się z koniecznością zapewnienia odmiennych warunków pracy interfejsu. Wyraźnie przedstawia to schemat blokowy programu kontroli przerwń z interfejsu zamieszczony na rys. 7.

Jak widać, program ten umożliwia jednocześnie realizację funkcji interfejsu TE (nadawca rozszerzony), LE (słuchacz rozszerzony), R/L (zdalny-lokalny), DC (zerowanie urządzenia) oraz DT (wyzwalanie urządzenia). Funkcji SR nie zamieszczono, gdyż realizowana jest ona w nadrzędnym programie zarządzającym, w następstwie sytuacji wymagającej zawiadomienia aktualnego kontrolera, np. po wystąpieniu błędu, gotowości itp. Jej związek z funkcją T przedstawia fragment programu, gdzie rozkaz SPE (zezwolenie szeregowej kontroli) powoduje zdjęcie komunikatu SRQ z magistrali.

Kolejność procedur dla przerwania TRAN została wybrana przez autora w celu wyrazistego przedstawienia realizacji funkcji interfejsu. W konkretnej realizacji wiele identycznych procedur będzie zastąpione jednym podprogramem, a kolejność procedur może być określona np. częstością występowania, minimalizując czas obsługi. Wybór optymalnej realizacji zależy przede wszystkim od typu użytego mikroprocesora, jego listy rozkazów i układu przerwń.

6. WNIOSKI

Zdaniem autora powyższe propozycje dają możliwość realizacji interfejsu IEC każdego urządzenia kontrolowanego przez mikroprocesor, poprzez zwykłą eliminację zbędnych w danym zastosowaniu funkcji. Dotyczy to urządzeń pracujących z szybkością ≤ 10 kbajt/s, a więc z pamięciami kasetowymi włącznie.

Jakkolwiek przyszłość realizacji funkcji interfejsu należy do omawianych układów LSI przez kilka lat problematyka poruszona przez autora będzie na pewno aktualna, choćby ze względu na problem dostępności tych elementów, pamiętając, że wchodzące dopiero na nasz rynek mikroprocesory mają już prawie 10-letnią historię. Ponadto problem realizacji programowej funkcji interfejsu powróci, gdy wzorem makrosystemów do zarządzania interfejsem przeznaczy się osobny mikroprocesor, o ile oczywiście będzie to ekonomicznie uzasadnione aktualnymi w danym czasie cenami elementów.

LITERATURA

- [1] IEEE Standard 488-1975 - Digital interface for programmable instrumentation - The IEEE, Inc. N.Y. 1975.
- [2] D.C. Loughry: The state of the art for IEEE 488 - direction and trends - Electronics Test, p. 8, June 1979.
- [3] J. Klaus: Wie funktioniert der IEC Bus - Elektronik, heft 4, pp. 72-78, heft 5, pp. 73-78, 1975.
- [4] D.W. Ricci, G.E. Nelson: Standard instrument interface simplifies system design - Electronics pp. 95-106, Nov. 14, 1974.
- [5] Donald C. Loughry, M.S. Allen: IEEE Standard and Microprocessor Synergizm - Proceedings of the IEEE, pp. 162-171, vol. 66, no 7, Feb. 1978.
- [6] W.C. Randle, N. Kerth: Microprocessors in instrumentation-Proceedings of the IEEE, vol. 66, pp. 172-181, no. 2, Feb. 1978.
- [7] Vlado Lipovac: Design an IEEE - 488 bus into an FPLA - Electronic Design 24, pp. 104-111, Nov. 22, 1977.
- [8] J. Pieper, R.J. Grossi: LSI streamlines instrument interface with standard IEEE - 488 bus - Electronics, pp. 145-150, April 26, 1979.
- [9] K. Badźmiroweki, J. Pieńkos, W. Piestrzyński: Cyfrowe układy MOS-LSI. WKiŁ, Warszawa 1979.
- [10] Digital interface bus standard - Motorola Semiconductor Products Inc., Switzerland. 1977.
- [11] A. Santoni: What's wrong with 488 - Electronic Design 24, pp. 48-51, Nov. 22, 1977.
- [12] F. Wagner: Projektowanie urządzeń cyfrowych. WNT, Warszawa 1978.
- [13] B.E. Forbes - IEEE-488: A proposed microcomputer I/O bus standard - Computer Design, Nov. 1978, pp. 170-174.
- [14] Yiu K. Wo: A versatile microcomputer based controller with an instrumentation bus interface for data acquisition in automated experimentation - IEEE Trans. on Industrial Electronics and Control Instrumentation, August 1978, pp. 221-225.
- [15] Monolithic interface bus adapter fulfills IEEE 488 requirements - Computer Design, Sept. 1979, p. 182.
- [16] T. Jones, T. Peter: Challenges in microprocessor system design - Computer Design, Nov. 1976, pp. 109-118.
- [17] D.C. Loughry: Interface standard 488 in action: concepts and capabilities - ELECTRO 76 Professional Program.
- [18] J. Farrel: Mating micros to the IEEE-488 bus doesn't take many connections - Electronic Design, vol. 26, no. 24, 22 Nov. 1978, pp. 152-157.

Recenzent

Doc. dr hab. inż. Ferdynand Wagner

Wpłynęło do Redakcji 10.11.1979 r.

ПРОБЛЕМЫ И РЕАЛИЗАЦИЯ ИНТЕАФЕЙСА IEC В МИКРОПРОЦЕССОРНЫХ СХЕМАХ

Р е з ю м е

В статье представлены проблемы совместной работы устройств, построенных на базе микропроцессора, в системе сбора данных, которая основана на стандартной системе IEC BUS. Обсуждены выгоды, вытекающие из совместной работы средств программирования и аппаратной части, а также представлена схема системы интерфейса для скорости передачи 10 байтов, исходя из представленной программы контроля протвинаний интерфейса.

THE MICROPROCESSORS AND IEC BUS - PROBLEMS AND REALIZATION

S u m m a r y

This paper presents the microprocessor and IEC BUS alliance problems and describes the interface functions realisation concept in the interrupt interface handler form.