

mgr inż. MIROSLAW SŁODCZYK
dr inż. ANDRZEJ SYRYCZYŃSKI
Przemysłowy Instytut
Automatyki i Pomiarów MERA-PIAP
W a r s z a w a

WIELOZADANIOWY SYSTEM OPERACYJNY CZASU RZECZYWISTEGO PI 80 DLA URZĄDZEŃ MIKROPROCESOROWYCH INTEL DIGIT PI Część I. Koordynator

W artykule omówiono koordynator wielozadaniowego systemu operacyjnego czasu rzeczywistego PI 80 dla urządzeń mikroprocesorowych INTEL DIGIT PI, przeznaczonych do pomiarów i sterowania. Przedstawiono zastosowania, zasady działania, strukturę i opis funkcjonalny.

1. Wstęp

System operacyjny wielozadaniowy wykorzystuje się tam, gdzie procesor wykonuje szereg zadań, niezależnie inicjowanych w funkcji czasu, lub zdarzeń zewnętrznych. System wielozadaniowy pozwala w sposób istotny uprościć proces przygotowania i uruchamiania oprogramowania użytkowego przez podzielenie oprogramowania użytkowego na szereg zadań stosunkowo prostych, o małym stopniu wewnętrznej i zewnętrznej uciążliwości. Funkcje uruchamiania zadań, ich korelowania czasowego, wzajemnych powiązań pomiędzy zadaniami i powiązań w stosunku do czasu oraz zdarzeń zewnętrznych wykonuje koordynator systemowy w sposób standardowy, nie wymagający pełnego opisu w programie użytkowym, poza wypełnieniem lub modyfikowaniem odpowiednich tabel.

Jedynym ograniczeniem stosowania systemu operacyjnego wielozadaniowego są wymagania czasu obsługi. Koordynator systemu zajmuje określoną część czasu procesora i dlatego istnieje granica szybkości obsługi zadań. W skrajnym przypadku, gdy szybkość następowania zdarzeń osiągnie wartość graniczną, system przestaje nadążać. Przy oprogramowaniu użytkowym nie stosującym systemu można uzyskać większą szybkość obsługi, ale za cenę znacznie większej trudności opracowania programu użytkowego.

W obecne opracowanie, opracowane dla mikroprocesorów INTEL DIGIT PI, które mogą być stosowane w niewielkich zestawach automatyki i pomiarów -- koncentratorach i sterownikach -- mikroprocesorowych, wykorzystuje się jako jednostkę centralną 8 bitowy mikroprocesor typu 8080A, produkowany w krajach RWPG i przewidziany do produkcji krajowej.

Stosunkowo mała moc obliczeniowa mikroprocesora 8080 i jego niski koszt w warunkach zagranicznych mogą sugerować stosowanie oddzielnego mikroprocesora do każdego odrębnego zadania w przypadkach bardziej złożonej aplikacji. Każdy mikroprocesor mógłby wykonywać wówczas jedno, stosunkowo proste, zadanie i nie byłoby potrzeby stosowania systemu wielozadaniowego.

Jednak, gdy szereg mikroprocesorów dzieli między sobą zadania obsługi obiektu, muszą istnieć powiązania pomiędzy ich programami, zapewniające współbieżność działania poszczególnych procesorów wzajemnie oraz w stosunku do zdarzeń zewnętrznych i biegu czasu. Dlatego, zarówno ze względu na uproszczenie oprogramowania użytkowego, jak i na konieczność powiązań poszczególnych mikroprocesorów ze sobą, systemy wielozadaniowe, w odpowiednio uproszczonej postaci, są niezbędne dla uniwersalnych zestawów urządzeń mikroprocesorowych automatyki i pomiarów.

2. Koordynator systemu operacyjnego PI 80

Zadaniem koordynatora wielozadaniowego systemu operacyjnego czasu rzeczywistego jest zarządzanie programami — zadaniami, uwarunkowane czasem rzeczywistym oraz innymi zdarzeniami (przerwaniami) w sterowanym procesie.

Założenia systemu stanowią wyważony kompromis pomiędzy jego prostotą (warunkującą niewielki zajmowany obszar pamięci i małą stratę czasu na działanie samego systemu), a wartością użytkową, związaną z maksymalnym stopniem przejęcia przez system różnorodnych uzależnień międzyzadaniowych i powiązań pomiędzy zdarzeniami zewnętrznymi, a zadaniami. Dlatego też stawia się większe wymagania wobec programów użytkowych niż w przypadku bardziej skomplikowanych programów zarządzających. Przyjęto, że koordynator może obsługiwać do 16 zadań.

Zarządzanie zadaniami odbywa się w opisywanym koordynatorze według regulaminu priorytetów tuz względnych lub priorytetów względnych. Regulamin priorytetów bezwzględnych polega na tym, że w chwili pojawienia się zgłoszenia o priorytecie wyższym niż priorytet programu aktualnie wykonywanego następuje jego przerwanie, zawieszenia i przekazanie sterowania programowi o wyższym priorytecie. Regulamin priorytetów względnych polega na tym, że zadanie rozpoczęte wykonuje się do końca, a zgłaszające się w tym czasie inne zadania zostają zapamiętane i muszą odczekać. Wyboru odpowiedniego regulaminu priorytetów dokonuje użytkownik, pisząc programy obsługi przerwań. Jeżeli program bezpośredniej obsługi przerwania kończy się skokiem do koordynatora systemu, to zrealizowany będzie priorytet bezwzględny. Natomiast, jeżeli po wykonaniu bezpośredniej obsługi przerwania ustawiony zostanie powrót (return), to system przywróci wykonywanie dotychczas wykonywanego, przerwanego zadania i zostanie zrealizowana zasada priorytetu względnego.

3. Zadania i stany zadań

Wszystkie standardowe funkcje wymagające obsługi sygnałów i urządzeń, w tym peryferii, wejść, wyjść, przerwań, powierza się zadaniom, stanowiącym część oprogramowań użytkowych. Zadania są zatem wyodrębnionymi częściami oprogramowania użytkowego. System dopuszcza do 16 zadań, o numeracji 15...0 w kolejności priorytetu. Zadania są sterowane przez koordynator systemu, w funkcji zdarzeń zewnętrznych i czasu.

Każde zadanie może znajdować się w jednym z ośmiu stanów. Stan zadania opisuje jego identyfikator Z, który może przyjmować wartości 0...7. Stany zadań są omówione poniżej, a przejścia pomiędzy nimi przedstawia schemat blokowy programu (rys. 1).

Z = 0 — zadanie niezgłoszone

- a) w ogóle nie ma zadania (nie istnieje),
- b) zadanie nie zostało zgłoszone do wykonania,
- c) zadanie zostało całkowicie ukończone i ustawiło się swoim zakończeniem w stan 0.

Z = 1 — zadanie zgłoszone do wykonania.

Zdarzenie (zegar, bezpośrednia obsługa przerwania, inne zadanie) zażądało wykonania zadania, które zostanie uruchomione przez koordynator, i wówczas stan zadania przybierze wartość 2.

Z = 2 — zadanie wykonywane.

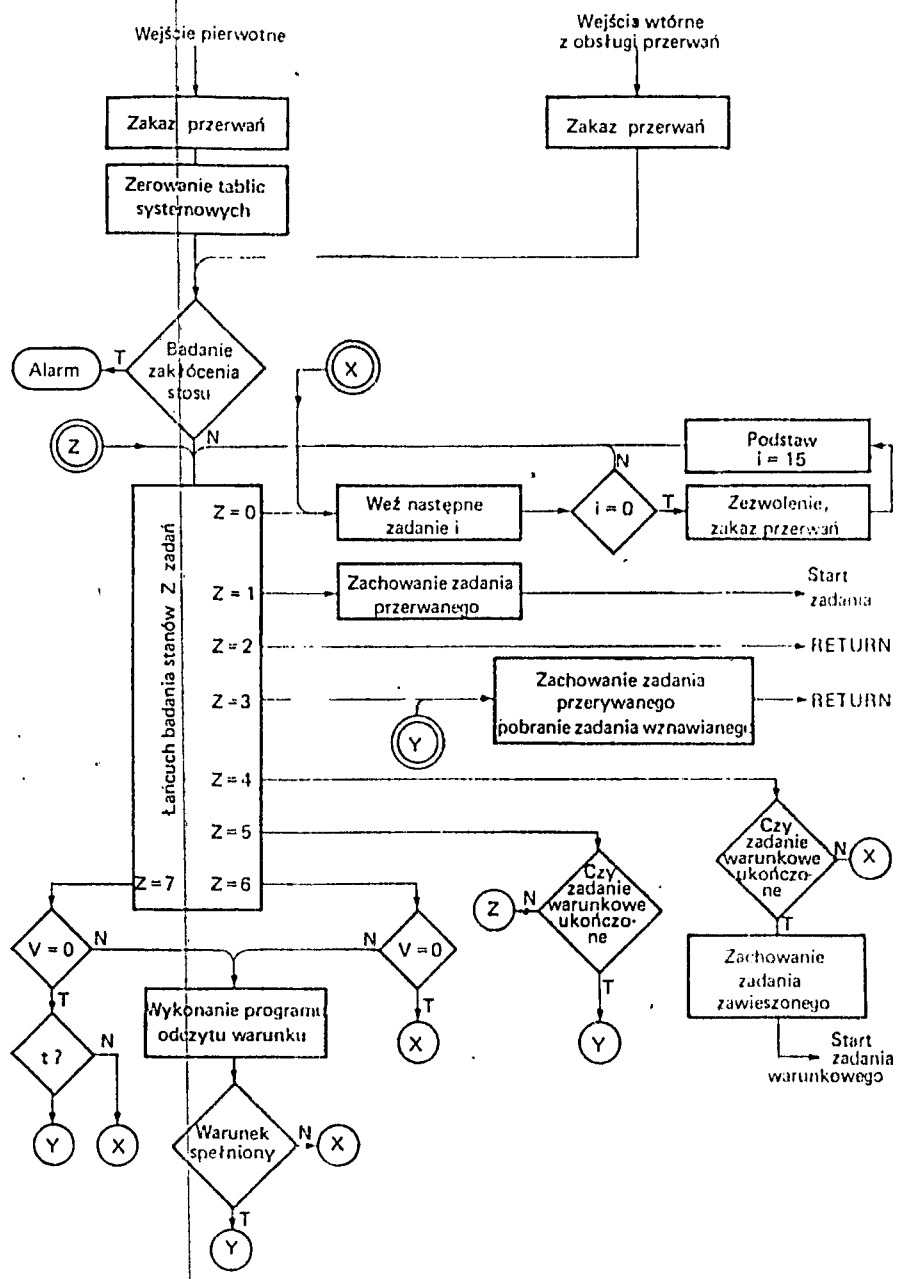
Zadanie przed niniejszym wejściem do koordynatora było wykonywane i nie jest jeszcze zakończone.

Z = 3 — zadanie przerwane.

Zadanie zostało uprzednio rozpoczęte, ale nie jest jeszcze zakończone. Zadanie zostało przerwane przez koordynator w celu wykonania zadania o wyższym priorytecie.

Z = 4 — zadanie zgłoszone do zawieszenia.

Rozpoczęte zadanie zgłasza do koordynatora za pomocą stanu 4 potrzebę wykonania dodatkowego zadania.



Rys. 1. Schemat blokowy programu koordynatora systemu operacyjnego czasu rzeczywistego, X – badanie następnego zadania; Y – wznowienie zadania przerwanego; Z – badanie stanów zadań.

d) cykliczne uruchamianie zadań okresowych kontrolowane obsługą przerw zegarowych
W punktach *a*, *b* i *c* zadanie lub obsługa zdarzenia musi ustawić w tabeli *W* numer zadania warunkowego *a*, programu odczytu warunków *b* lub numer licznika programowego *c* oraz dodatkowo dla punktu *c* wpisać do licznika ilość jednostek czasu, deklarując w ten sposób czas oczekiwania

5. Tablice systemowe

Stan realizacji zadań jest zapamiętany w tablicach systemowych. Tablice systemowe dla 16 zadań zawierają:

- stany zadań *Z*, 16 słów,
- warunki *W*, 16 słów,
- rejstry zadań przerwanych, 10 x 16 słów,
- liczniki zegarowe, 16 słów.

Tablice liczące łącznie 208 słów 8-bitowych są umieszczone w pamięci wewnętrznej pakietu mikroprocesora.

Dalszym elementem o charakterze tablicy stałej są listy adresów startów zadań i startów programów odczytujących warunki. Listy te stanowią część oprogramowania użytkowego i mogą być umieszczone w dowolnym obszarze pamięci. Ponadto program obsługi przerwania zegarowego korzysta z tablicy okresów zadań okresowych, zajmującej 16 słów.

Łącznie, wszystkie powyżej wymienione tablice zajmują 288 komórek pamięci. Program koordynatora natomiast zajmuje 377 komórek pamięci stałej.

W celu ustawienia zadań użytkownik powinien ustawić adresy startów zadań i programów odczytu warunków. Numer zadania, a więc miejsce adresu startu ustala priorytet zadania, przy czym zadaniu numer 15 ma najwyższy priorytet. Pozostałe czynności związane z ustawieniem stanów zadań spoczywają na programach obsługi zdarzeń (przerw) w sterowanym procesie; na programach użytkowych (zadaniach) oraz na samym koordynatorze.

6. Zakończenie

Opracowany system operacyjny cechuje się zastosowaniem jednakowego mechanizmu dla realizacji warunków programowych, warunków zewnętrznych i warunków czasowych, co zapewnia jego dużą wartość użytkową, a zarazem stosunkowo małą objętość wymaganej pamięci. Ważnym systemu jest możliwość pracy zarówno z regulaminem priorytetów względnych, jak i priorytetów bezwzględnych. Koordynator systemu nie nawiązuje do żadnej konkretnej realizacji sprzętowej układu mikroprocesorowego i niezależnie od głównego przeznaczenia, dla zestawów mikroprocesorowych INTEL DIGIT PI, może być stosowany dla dowolnych innych urządzeń mikroprocesorowych, zbudowanych w oparciu o mikroprocesor typu 8080.

W następnych częściach publikacji zostaną przedstawione dalsze części systemu operacyjnego czasu rzeczywistego PI 80, to jest monitor przerw, programy startowe, standardowe programy obsługi pakietów sprzężenia z obiektem, a także ogólna budowa zadań w oprogramowaniu użytkowym. Opisany koordynator systemu operacyjnego PI 80 został poddany badaniom dynamicznym, uzyskano wartości czasów obsługi dla każdego typu przejścia. Uzyskane wyniki świadczą o stosunkowo niskich czasach obsługi, umożliwiających stosowanie systemu dla bardzo wielu aplikacji. Opracowana metoda badania właściwości dynamicznych wielozadaniowych systemów operacyjnych i wyniki badań systemu PI 80 zostaną opublikowane w najbliższym czasie.