



Patent dodatkowy
do patentu _____

Zgłoszono: 02.VI.1964 (P 104 738)

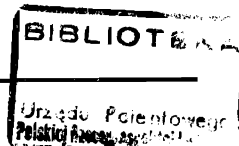
Pierwszeństwo: _____

Opublikowano: 20.I.1967

42 m³ 3/10
Kl. 42 m, 14

MKP G 06 f 3/10

UKD



Współtwórcy wynalazku: mgr inż. Jacek Bańkowski,
mgr inż. Konrad Fiałkowski

Właściciel patentu: Politechnika Warszawska Katedra Budowy Maszyn
Matematycznych, Warszawa (Polska)

**Sposób kodowania rozkazów w elektronicznych maszynach
cyfrowych mikroprogramowanych poziomo, oraz sposób kontroli
operacyjnej poprawności tych rozkazów**

1

Przedmiotem wynalazku jest sposób kodowania rozkazów w elektronicznych maszynach cyfrowych mikroprogramowanych poziomo oraz sposób kontroli operacyjnej poprawności tych rozkazów, który zapewnia formowanie rozkazów operacyjnie poprawnych z możliwością sygnalizacji błędów.

Z organizacji maszyn cyfrowych mikroprogramowanych poziomo wynika, że pewnych mikrooperacji nie można używać w jednym rozkazie. Rozkaz zawierający takie mikrooperacje nie jest poprawny.

Dotychczas stosowane sposoby kodowania rozkazów wymagały od użytkownika maszyny cyfrowej mikroprogramowanej poziomo pamięciowego opanowania dużej ilości reguł tworzenia rozkazów poprawnych. W wypadku pomyłki użytkownika, przy dotychczas stosowanym sposobie kodowania rozkazów, formowany był rozkaz błędny, którego wykrywanie powodowało znaczne trudności przy uruchamianiu programu.

Stosując sposób według wynalazku uzyskuje się sygnalizację błędu w wypadku próby utworzenia niepoprawnego rozkazu. Wszystkie rozkazy zakodowane, przy których sygnalizacja błędu nie wystąpiła, są więc poprawne i uruchomienie programu składającego się jedynie z takich rozkazów wymaga zużycia mniejszej ilości czasu maszyny i użytkownika.

Istotą wynalazku jest tworzenie słowa kon-

2

trolnego w oparciu o umieszczoną na stałe w pamięci maszyny cyfrowej macierz, określaną dalej jako macierz przejść dozwolonych. Macierz ta jest macierzą kwadratową, a wyrazy jej przyjmują wartości 0 lub 1. Wymiar macierzy odpowiada liczbie wszystkich możliwych różnych mikrooperacji. Wiersze macierzy ponumerowane są kolejnymi liczbami, $i = 1, 2, \dots, N$, a kolumny macierzy liczbami $j = 1, 2, \dots, N$. Wiersz o numerze „i” odpowiada mikrooperacji o numerze „i”, i zawiera 1 w tych kolumnach, których numery odpowiadają numerom mikrooperacji tworzących z mikrooperacją o numerze „i” rozkaz niepoprawny.

Macierz zapisywana jest w pamięci wierszami, przy czym każdy wiersz zajmuje jedną komórkę pamięci. Wiersze macierzy wykorzystywane są przy formowaniu słowa kontrolnego, to znaczy słowa, badanie którego określa poprawność formowanego rozkazu.

Słowo kontrolne formowane jest w miarę tworzenia rozkazu, przy czym początkowo, przed rozpoczęciem tworzenia rozkazu wszystkie pozycje słowa kontrolnego mają wartość 0. Dodanie do tworzonego rozkazu mikrooperacji o numerze i powoduje dodanie logiczne i- tego wiersza macierzy do słowa kontrolnego.

Przed dodaniem do tworzonego rozkazu nowej mikrooperacji o numerze i należy sprawdzić, czy pozycja odpowiadająca mikrooperacji i w

poprzednio utworzonym słowie kontrolnym jest równa 0. Jeżeli na pozycji tej jest zapisane 0, nowoutworzony rozkaz jest w dalszym ciągu poprawny, a do słowa kontrolnego można dodać wiersz macierzy odpowiadający i- tej mikrooperacji. W przeciwnym przypadku, gdy odpowiadająca i- tej mikrooperacji pozycja słowa kontrolnego jest równa 1, uformowanie poprawnego rozkazu nie jest możliwe, należy więc zaprzestać dalszego formowania rozkazu i zasygnalizować błąd.

Wynalazek wyjaśniony zostanie bliżej na przykładzie.

W przykładzie przedstawiona jest macierz przejść dla maszyny zawierającej 6 różnych mikrooperacji. Przyjmuje się, że jedna mikrooperacja nie może być użyta w rozkazie wielokrotnie. Wynikiem tego założenia są wartości 1 na przekątnej macierzy.

Dodatkowo przyjęto w przykładzie, że mikrooperacja o numerze 1 nie może występować z mikrooperacjami o numerach 3 i 5, mikrooperacja o numerze 2 może występować z wszystkimi innymi mikrooperacjami, mikrooperacja o numerze 3 nie może występować z mikrooperacjami o numerach 1 i 6, mikrooperacja o numerze 4 nie może występować z mikrooperacjami o numerze 5, mikrooperacja o numerze 5 nie może występować z mikrooperacjami o numerach 1 i 4, mikrooperacja o numerze 6 nie może występować z mikrooperacją o numerze 3.

Macierz przejść dozwolonych dla przykładowej maszyny jest pokazana na fig. 1.

Kolejne postacie słowa kontrolnego powstające w miarę dodawania nowych mikrooperacji pokazane są na fig. 2 oraz fig. 3.

Przed przystąpieniem do formowania rozkazu słowo kontrolne zawiera wyłącznie zera. Próba dołączenia do rozkazu mikrooperacji o numerze „i” (numer podany z lewej strony na fig. 2 i fig. 3) powoduje badanie pozycji słowa kontrolnego o numerze „i”. Pozycja ta na fig. 2 i fig. 3 zaznaczona jest znakiem x. W wypadku, gdy badana pozycja o numerze „i” jest zerem wiersz macierzy przejść o numerze „i” zostaje zsumowany logicznie ze słowem kontrolnym.

W przeciwnym wypadku zamiast kolejnej po-

staci słowa kontrolnego w wierszu zapisane jest słowo: „błąd”.

W przykładzie podanym na fig. 2 możliwe jest utworzenie rozkazu przez dodanie kolejnej mikrooperacji o numerach 1, 2, 4, przy czym rozkaz zawierający te mikrooperacje jest poprawny. W przykładzie podanym na fig. 3 możliwe jest utworzenie rozkazu przez dodanie kolejnej mikrooperacji o numerach 3, 5, natomiast dodanie do tych mikrooperacji mikrooperacji o numerze 1 powoduje powstanie rozkazu operacyjnie niepoprawnego i w kolejnym wierszu zamiast słowa kontrolnego wypisane jest słowo: „błąd”.

Zastrzeżenia patentowe

1. Sposób kodowania rozkazów w elektronicznych maszynach cyfrowych mikroprogramowanych poziomo, **znamienny tym**, że każdej literze kodu nadaje się z góry ustaloną treść operacyjną, którą wykonuje się jako mikrooperacje lub jako zestaw mikrooperacji bez względu na obecność innych liter kodu części operacyjnej rozkazu.
2. Sposób według zastrz. 1, **znamienny tym**, że treść operacyjną każdej litery kodu przechowuje się w pamięci maszyny.
3. Sposób według zastrz. 1 i 2, **znamienny tym**, że w pamięci maszyny oprócz słów określających treść operacyjną wszystkich liter kodu przechowuje się parametry kontrolne wszystkich liter kodu.
4. Sposób kontroli operacyjnej poprawności rozkazów kodowanych według zastrz. 1 — 3, **znamienny tym**, że w czasie formowania części operacyjnej rozkazów z parametrów kontrolnych wszystkich użytych liter kodu tworzy się słowo kontrolne.
5. Sposób według zastrz. 4, **znamienny tym**, że słowo kontrolne jest pozycyjną sumą logiczną parametrów kontrolnych wszystkich użytych liter kodu.
6. Sposób według zastrz. 4 i 5, **znamienny tym**, że badane są określone pozycje lub grupy pozycji słowa kontrolnego i nowa mikrooperacja lub grupa mikrooperacji jest dołączona do tworzonego rozkazu jedynie wskutek pozytywnego wyniku tego badania.

```

1 0 1 0 1 0
0 1 0 0 0 0
1 0 1 0 0 1
0 0 0 1 1 0
1 0 0 1 1 0
0 0 1 0 0 1

```

fig.1.

```

0 0 0 0 0 0
1.  X
    1 0 1 0 1 0
2.  X
    1 1 1 0 1 0
4.  X
    1 1 1 1 1 0

```

fig.2.

```

0 0 0 0 0 0
3.  X
    1 0 1 0 0 1
5.  X
    1 0 1 1 1 1
1.  X

```

btqd

fig.3.