ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1980
Seria: AUTOMATYKA z. 50 Nr kol. 626

Bożena NOWAK-SZYMURA, Leon IBROM

PROGRAMOWA REALIZACJA DZIAŁAŃ ARYTMETYCZNYCH PRZEZ MINIKOMPUTER MERA 302 - ANALIZA MOŻLIWOŚCI, OSZACOWANIA, WNIOSKI

Streszczenie. W artykule przedstawiono analizę możliwości programowej realizacji działań arytmetycznych na słowach wielokrotnej precyzji. Procedury opracowano i wykonano dla minikomputera MERA 302. Dla każdej procedury podano minimalne i maksymalne czasy realizacji oraz zajętość pamięci.

Wstęp

Prezentowany artykuł stanowi omówienie jednego z etapów pracy naukowo-badawczej dotyczącej wykorzystania minikomputera do sterowania procesem termicznego cięcia blach. Podstawowym działaniem w tym procesie jest prowadzenie palnika po określonym konturze - ręczne lub automatyczne. Automatyzacja polega na wykorzystaniu układu śledzącego wytrasowany kontur lub na zaprogramowaniu ruchu palnika. W tym przypadku zadanie wycięcia zadanego konturu sformułowane jest w postaci programu. Stosowane języki kodowania konturów są na ogół bardzo proste, programy w tych językach mają najczęściej sztywną strukturę. Np instrukcje w języku ESSI [1] zawierają współrzędne punktów charakterystycznych dla konturu i linii łączących te punkty. Punktami charakterystycznymi są przede wszystkim punkty, które stanowią krańce łuku lub odcinka stanowiącego element konturu oraz inne punkty umożliwiające zdefiniowanie parametrów łuku (np. dla okręgu - współrzędne jego środka).

Program cięcia przetwarzany jest przez minikomputer. Podstawowym działaniem realizowanym w trakcie przetwarzania jest interpolacja linii tworzących kontur linią łamaną oraz przesyłanie sygnałów sterujących do elementów wykonawczych, którymi są silniki krokowe.

Dla tak postawionego zadania przeprowadzono analizę możliwości wykorzystania minikomputera MERA 302 do sterowania. Minikomputer MERA 302 jest małą jednostką o prostym procesorze, krótkim słowie maszynowym i pamięci operacyjnej o małej pojemności.

Oprogramowanie dostarczane przez producenta pozwala wykorzystać go w charakterze minikomputera biurowego ewentualnie do zbierania i lokalnego przetwarzania danych. Wykorzystanie minikomputera do sterowania wymaga odpowiedniej adaptacji środków urządzeniowych oraz gruntownej przebudowy oprogramowania.

Dla dokładniejszego zbadania celowości budowy specjalistycznego oprogramowania przeprowadzono próbę zaprogramowania procedur realizujących działania arytmetyczne. Pakiet tych procedur umożliwiłby wykonanie zasadniczego zadania stawianego przed minikomputerem - interpolacji łuków krzywych. Artykuł stanowi relacje z działań prowadzonych w tym kierunku.

2. Charakterystyka procesora Momik 8b

Procesor minikomputera MERA 302 - Momik 8b [2] operuje na słowie maszynowym 8-bitowym, posiada tylko jeden rejestr uniwersalny - akumulator 8-bitowy. Lista rozkazów zawiera 34 rozkazy, które są jedno - lub bezadresowe. Podstawową jednostką informacji jest bajt o długości 8 bitów. Można w nim przechować jeden znak alfanumeryczny, dwie cyfry dziesiętne lub liczbę całkowitą z zakresu [-128,127]. Liczby ujemne przedstawione są w kodzie uzupełnieniowym do 2. Pamięć operacyjna o pojemności 8k bajtów podzielona jest na strony po 32 bajty na jednej stronie. Rozkazy maszynowe zapisywane są w zależności od typu w 1 lub 2 bajtach i zawierają (w przypadku rozkazów argumentowych) adres argumentu lub argument bezpośredni.

Z działań arytmetycznych urządzeniowo realizowane są tylko rozkazy dodawania dwóch bajtów (z sygnalizacją przepełnienia) i przesunięć zawartości bajtu.

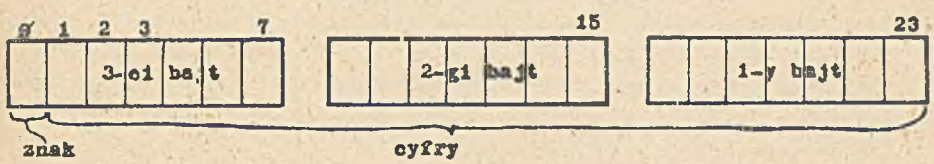
Powyższa charakterystyka pozwala stwierdzić, że dla realizacji obliczeń numerycznych należy przede wszystkim zapewnić możliwość wykonywania podstawowych działań arytmetycznych na liczbach przedstawianych z żądaną dokładnością i w odpowiednim zakresie.

Momik 8b wyposażyć można w dostarczany przez producenta pakiet procedur umożliwiających realizację działań arytmetycznych stałoprzecinkowych na słowach o długości 1,2,3 i 4 bajtów i konwersję liczb z układu dziesiętnego do dwójkowego i na odwrót. Pakiet ten dołączony może być do asemblerów MOTIS, MACROMOTIS, ASSEBLAN.

3. Struktura słowa liczbowego

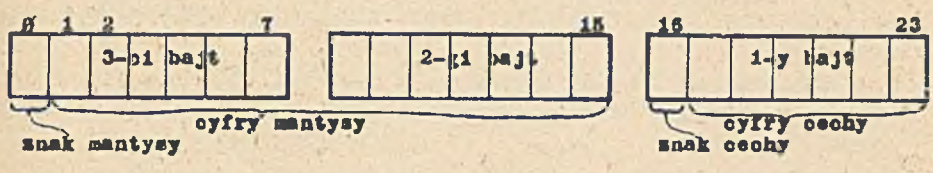
Dla potrzeb systemu sterowania procesem cięcia termicznego blach należy zapewnić możliwość ustawienia palnika w dowolnym punkcie płaszczyzny cięcia, przy czym maksymalny wymiar liniowy sięgać może kilkunastu metrów a dokładność ustawienia powinna wynosić 0,1 mm. Stanowiło to czynnik decydujący o wymaganym zakresie dla przetwarzanych wielkości liczbowych. Ponieważ w czasie realizacji pracy (rok 1975/76) nie dysponowano jeszcze możliwością korzystania z pakietu procedur arytmetycznych operujących w słowach wielokrotnej precyzji dla potrzeb systemu opracowano i wykonano procedury realizujące operacje arytmetyczne dla liczb binarnych w zapisach stało- i zmiennoprzecinkowych o długości 3 bajty. Zapis liczb stałoprze-

cinkowych i zmiennoprzecinkowych w pamięci procesora Momik 8b przedstawiają rys. 1 i 2. Liczby stałoprzecinkowe mają długość 24 bity i są przedsta-



Znaczenie pozycji słowa liczbowego:
bit Ø - pozycja znakowa liczby
bity 1-23 - cyfry liczby

Rys. 1. Zapis liczby stałoprzecinkowej w pamięci procesora Momik 8b



Znaczenie pozycji słowa liczbowego:
bit Ø - pozycja znakowa mantysy
bity 1-15 - cyfry mantysy
bit 16 - pozycja znakowa cechy
bity 17-23 - cyfry cechy

Rys. 2. Zapis liczby zmiennoprzecinkowej w pamięci procesora Momik 8b

wione w kodzie uzupełnieniowym (uzupełnienie do 2), w związku z tym zakres liczb reprezentowanych w formacie stałoprzecinkowym wynosi $[-2^{+24}]; [2^{+23}-1]$. Liczby zmiennoprzecinkowe mają długość 24 bity. Mantysa i cecha przedstawione są w kodzie uzupełnieniowym (uzupełnienie do 2). Mantysa jest 15-bitowym ułamkiem właściwym, a cecha 7-bitowa liczbą z przedziału $[-64;63]$. Zakres liczb zmiennoprzecinkowych przedstawiony w sposób nieznormalizowany wynosi $[-1;1-2^{-15}]. 2^{+63}$.

Realizacja działań na słowach liczbowych o opisanej powyżej strukturze wymaga ich podziału na części o długości równej długości argumentów rozkazów arytmetycznych procesora Momik 8b i wykorzystania tych rozkazów w celu otrzymywania wyników częściowych. Metodę tę zastosowano przy realizacji procedur działań dodawania, odejmowania, mnożenia, dzielenia, negacji i przesuwania liczb stało- i zmiennoprzecinkowych. Poniżej przedstawiono krótki opis zrealizowanych procedur. Celem zilustrowania zastosowanej metody omówiono dokładniej procedurę mnożenia stałoprzecinkowego MNOS. Podane zostały również parametry zrealizowanych procedur, takie jak długość zajmowanego przez procedurę obszaru pamięci operacyjnej i czas realizacji procedur. Podane czasy realizacji procedur zostały obliczone w oparciu o listę czasów wykonywania rozkazów.

4. Opis procedur działań arytmetycznych na liczbach stałoprzecinkowych

a) Procedura dodawania liczb stałoprzecinkowych SUMS. Struktura argumentów i wyniku operacji dodawania przedstawiona jest na rys. 1. Procedura dodawania zaczyna się od dodania dwóch najmniej znaczących bajtów liczb stałoprzecinkowych. W procedurze występuje sygnalizacja błędu nadmiaru. Procedura zajmuje 24 komórki pamięci operacyjnej, a czas realizacji wynosi 77,2 - 84,2 μs.

b) Procedura tworzenia liczb ujemnych w arytmetyce uzupełnieniowej (uzupełnienie do 2) NEG polega na zanegowaniu każdego bajtu liczby stałoprzecinkowej (począwszy od najmniej znaczącego) i dodaniu 1 do najmłodszej pozycji najmłodszego bajtu liczby.

c) Procedura odejmowania liczb stałoprzecinkowych ODES. W procedurze odejmowania liczb stałoprzecinkowych wykorzystano dodawanie liczb stałoprzecinkowych, po uprzednim zastosowaniu procedury NEG przy obliczaniu odjemnika.

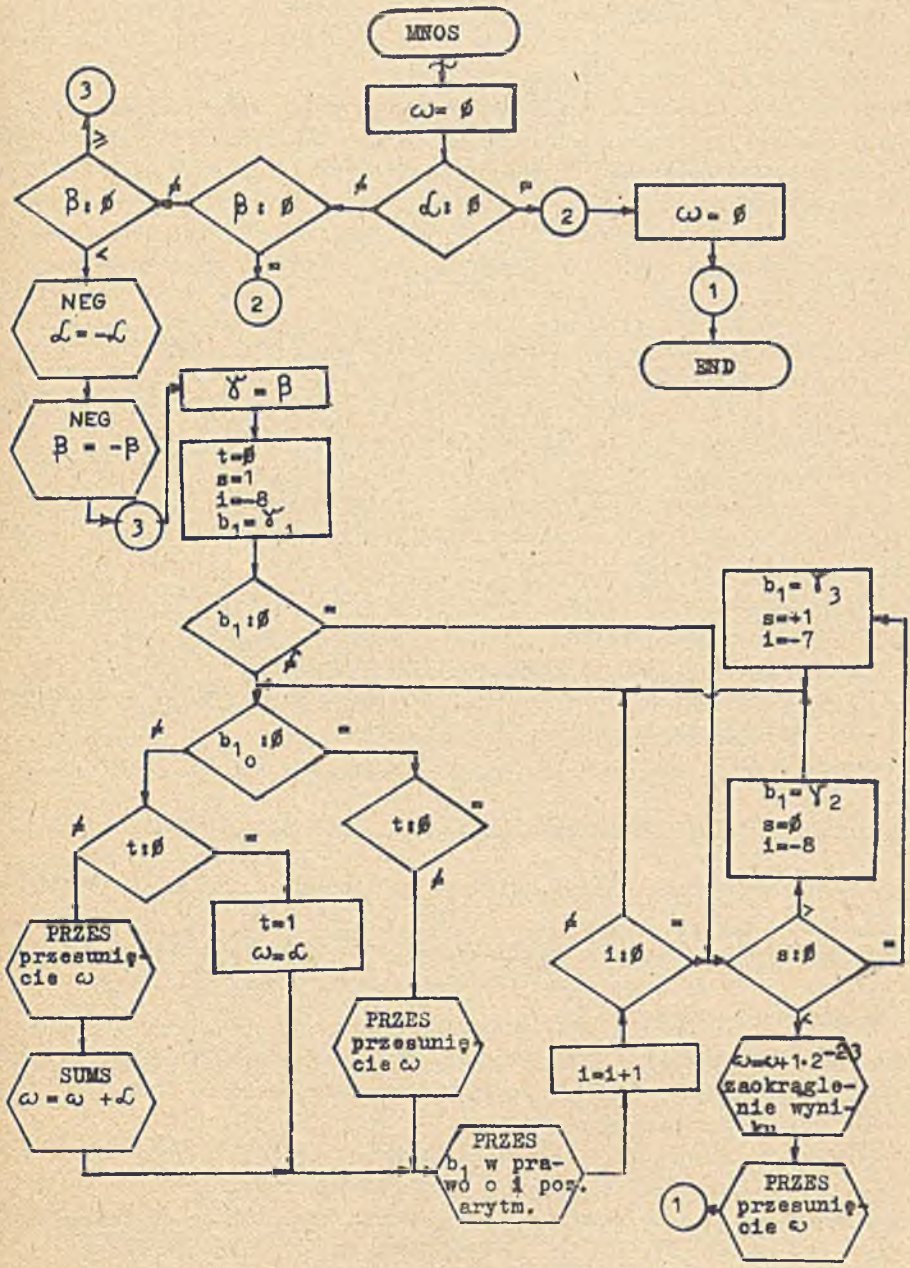
W procedurze występuje sygnalizacja nadmiaru. Procedura zajmuje 47 komórek pamięci operacyjnej, czas realizacji mieści się w granicach 105,0 - 111,8 μs.

d) Procedura mnożenia liczb stałoprzecinkowych MNOS. W procedurze MNOS operacje mnożenia realizuje się jako ciąg dodawania i przesunięć, lub samych przesunięć, przy czym ich liczbę i kolejność określają wartości bitów mnożnika.

W czasie mnożenia mnożna i mnożnik przedstawione są jako argumenty 3-bajtowe, a wynik mnożenia jest również liczbą 3-bajtową.

Zastosowana w procedurze metoda mnożenia dla mnożnika ujemnego polega na wykonaniu mnożenia poprzednio zanegowanej wartości mnożnej i mnożnika [3]. Zatem, główna część algorytmu mnożenia wykonuje obliczenia zawsze dla mnożnika większego od zera. W każdym kroku mnożenia przesuwa się sumę częściową o jedną pozycję w prawo (mnożnik przesuwa się względem mnożnej o jedną pozycję w lewo), przy czym każdorazowo pomija się ostatni bit sumy częściowej. W ten sposób sumowanie wykonuje się na słowie 3-bajtowym, w związku z czym można wykorzystać procedurę SUMS (procedura dodawania słów 3 bajtowych).

Dla liczb zawierających na końcu ciąg zer (w praktyce najczęściej spotykane) zastosowano przyspieszenie algorytmu polegającego na jednorazowym przesunięciu sumy częściowej o liczbę pozycji odpowiadającej liczbie zer. W procedurze zastosowano procedurę NEG, procedurę SUMS i procedurę PRZES (podprogram przesuwania mnożnej o jedną pozycję w lewo). Schemat blokowy algorytmu mnożenia liczb stałoprzecinkowych przedstawia rys. 3.



Rys. 3. Schemat blokowy algorytmu mnożenia liczb stałoprzecinkowych

α - mnożna
β - mnożnik
ω - iloczyn lub suma częściowa
γ - zmienna pomocnicza
i - numer bitu w kolejnym bajcie mnożnika
t - zmienna pomocnicza($t=\phi$ - nie wystąpił bit = 1; $t\neq\phi$ - wystąpił bit=1)
s - zmienna określająca aktualnie rozpatrywany bajt mnożnika (gdy $s>\phi$ - to 1-szy bajt mnożnika, $s=\phi$ - to 2-gi bajt mnożnika $s<\phi$ - to 3-ci bajt mnożnika)
b_1 - wartość aktualnie analizowanego bajtu mnożnika. Procedura mnożenia zajmuje 212 komórek pamięci operacyjnej, a czas jej realizacji wynosi 286,2 - 5633,8 μs.

e) Procedura dzielenia liczb stałoprzecinkowych DZIS. W procedurze zastosowano metodę dzielenia nierestytucyjnego [3]. Dzielenie wykonuje się za pomocą kolejnych odejmowań i przesunięć reszty częściowej. W każdym kroku dzielenia przesuwa się resztę częściową o jedną pozycję w lewo (dzielnik przesuwa się względem dzielnej o jedną pozycję w prawo). W czasie dzielenia dzielna i dzielnik przedstawione są jako argumenty 3-bajtowe, iloraz jest również liczbą 3-bajtową reprezentującą najbardziej znaczące pozycje wyniku.

W procedurze wykorzystano podprogramy ODES, SUMS, NEG oraz PRZES (przesuwanie reszty częściowej i aktualnego bajtu ilorazu w lewo).

Występuje sygnalizacja nadmiaru (wykrycie dzielenia (-1)÷(-1)).

Procedura dzielenia mieści się w 204 komórkach pamięci operacyjnej, a czas jej realizacji wynosi 162,0 - 4804,0 μs.

5. Opis procedur działań arytmetycznych na liczbach zmiennoprzecinkowych

Operacje zmiennoprzecinkowe działają na liczbach przedstawionych na rys. 2.

Przyjęto również "zero znormalizowane", dla którego mantysa równa się 0.0000...0, a cecha przyjmuje najmniejszą z wartości, tj. 1.0000000 (w kodzie uzupełnieniowym do 2) [3].

a) Procedura normalizacji liczb zmiennoprzecinkowych NORM normalizuje mantysę wynikową po wykonaniu operacji stałoprzecinkowych na mantysach liczb zmiennoprzecinkowych.

b) Procedura dodawania liczb zmiennoprzecinkowych SUMZ. Dodawanie liczb zmiennoprzecinkowych polega na dodaniu mantys dwóch liczb o tych samych cechach. W przypadku, gdy cechy dwóch liczb są różne, zwiększa się cechę mniejszej z liczb, aż do zrównania jej z cechą większej liczby. Każde zwiększenie o "1" cechy związane jest z przesunięciem mantysy o jedną pozycję w prawo. Może się zdarzyć, że w następstwie przesuwania mantysy w prawo, wszystkie jej bity znajdą się poza rejestrem (w reje-

strze pozostaną same zera). Za wynik dodawania przyjmuje się wtedy bezpośrednio większą z liczb. W procedurze wykorzystano procedury SUMS, ODES i NORM (wynik dodawania liczb zmiennoprzecinkowych podlega znormalizowaniu).

Procedura zajmuje 211 komórek pamięci operacyjnej, a czas jej realizacji mieści się w granicach 697,0 - 710,8 μs.

c) Procedura odejmowania liczb zmiennoprzecinkowych ODEZ wykorzystuje procedurę dodawania liczb zmiennoprzecinkowych SUMZ po uprzednim zastosowaniu procedury NEG przy obliczaniu odjemnika.
Zajętość pamięci operacyjnej - 226 komórek.
Czas wykonania procedury - 736,6-750,4 μs.

d) Procedura mnożenia liczb zmiennoprzecinkowych MNOZ. Mnożenie dwóch liczb zmiennoprzecinkowych polega na mnożeniu ich mantys i dodaniu cech. W procedurze sygnalizuje się nadmiar, który może wystąpić w czasie dodawania cech. Sygnalizowane jest również wykrycie zerowej mnożnej lub zerowego mnożnika. Procedura wykorzystuje procedury MNOS i NEG. Wynik operacji mnożenia jest znormalizowany. Procedura zajmuje 317 komórek pamięci operacyjnej, a czas jej wykonania mieści się w granicach 666,0 - 6013,6 μs.

e) Procedura dzielenia liczb zmiennoprzecinkowych DZIZ. Dzielenie liczb zmiennoprzecinkowych polega na dzieleniu mantys dwóch liczb i odejmowaniu ich cech. Nadmiary występujące przy zerowym dzielniku oraz przy dzieleniu mantys i odejmowaniu wykładników są sygnalizowane. Występuje również sygnalizacja wykrycia zerowej dzielnej. Wynik dzielenia jest znormalizowany. Procedura wykorzystuje procedurę DZIS i NORM. Zajętość pamięci operacyjnej-313 komórek. Czas realizacji procedury-550,8-5192,8μs

6. Wnioski

1. Dla przedstawionych procedur czasy ich realizacji zawarte są w granicach 77,2 - 6013,6 μs, a zajętość pamięci mieści się w przedziale 24 - 317 komórek.
2. Struktura arytmometru procesora Momik 8b nie jest przystosowana do realizacji złożonych obliczeń numerycznych.

LITERATURA

[1] WOŹNIAK A., PANCEWICZ A.: Sterowanie cyfrowe do cięcia termicznego. Praca niepublikowana (Instytut Spawalnictwa - Gliwice 1974).
[2] Dokumentacja systemu Momik 8b. Warszawa 1972.
[3] IVAN FLORES: Arytmetyka maszyn cyfrowych. WNT, Warszawa 1970.
[4] YAOHAN CHU: Maszyny cyfrowe, zasady projektowania. PWN, Warszawa 1967.

ПРОГРАММНОЕ ОСУЩЕСТВЛЕНИЕ АРИФМЕТИЧЕСКИХ ДЕЙСТВИЙ
В МИНИ-ЭВМ МЭРА-302 - АНАЛИЗ ВОЗМОЖНОСТЕЙ, ОЦЕНКИ, ВЫВОДЫ

Р е з ю м е

В статье представлен анализ возможностей программного осуществления арифметических действий с многократной точностью. Процедуры разработаны для мини-ЭВМ Мэра-302. Для каждой процедуры оценены минимальное и максимальное времена работы и нужный для нее объем памяти.

MINICOMPUTER MERA-302 PROGRAM REALIZATION OF ARITHMETIC OPERATIONS - THE ANALYSIS OF POSSIBILITIES; ESTIMATION; CONCLUSION

S u m m a r y

The paper presents the possibility analysis of program realization of the arithmetic operations on words with multiple precision. The procedures have been carried out for the minicomputer MERA - 302.

For each procedure, the minimum and maximum realization times as well as the memory occupation have been worked out.