



ÚŘAD PRO VYNÁLEZY
A OBJEVY

POPIS VYNÁLEZU K AUTORSKÉMU OSVĚDČENÍ

209751

(11)

(B1)

(51) Int. Cl.³
G 06 F 9/22

/22/ Přihlášeno 18 04 80
/21/ /PV 2738-80/

(40) Zveřejněno 31 03 81

(45) Vydáno 15 02 83

(75)

Autor vynálezu

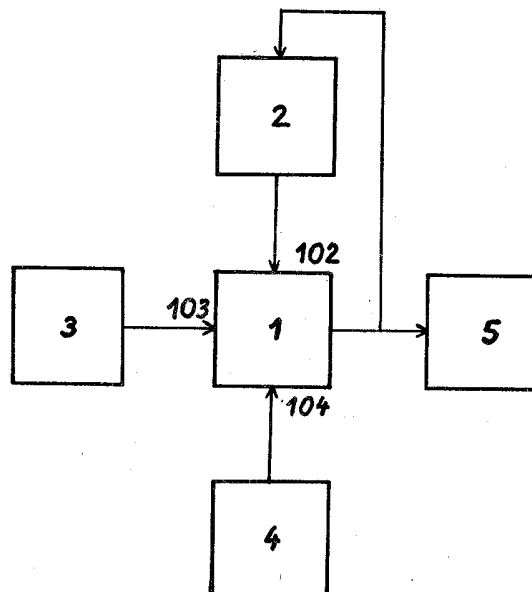
JELÍNEK OLDŘICH ing. a JANDA PETR ing., PRAHA

(54) Obvod pro rychlý rozsok do větví mikroprogramu podle délky a polohy operandu

Obvod pro rychlý rozsok do větví mikroprogramu podle délky a polohy operandu, určený pro mikroprogramem řízené procesory. Účinek je zvýšení rychlosti styku operačního procesoru počítače s pamětí počítače.

Vynález umožňuje v jediné mikroinstrukci modifikovat adresu další mikroinstrukce délkou operandu, pro který má dojít ke styku, a sudostí nebo lichostí adresy začátku toho operandu. Tím dojde k rozsokmu mikroprogramu do větví, které obsahují posloupnost povelů ke styku, vytvořenou přesně pro danou délku a polohu operandu. Zrychlení styku spočívá v tom, že uvedený rozsok se uskuteční najednou, během jediné mikroinstrukce.

Obor použití je výpočetní technika, mikroprogramem řízené operační procesory. Je vhodné jen pro jednodušší procesory, které neumožňují více paralelních činností v jediné mikroinstrukci.



Vynález se týká obvodu, který v návaznosti na vhodný mikroprogram umožní rychlý rozskok do větvi mikroprogramu, podle délky operandu a sudé nebo liché adresy jeho začátku.

Většina počítačů vykonává mimo jiné jednodušší operace též skupinu operací s proměnlivou, případně navzájem různou délkou operandů, které jsou čteny z operační paměti, v operační jednotce zpracovány a výsledek je zapisován zpět do operační paměti. Čtení operandů a zápis výsledku proběhne na obecný počet styků s pamětí, závislý na konstrukci počítače a délce operandů. Řídící mikroprogram musí obsahovat cykly pro styk s pamětí, ve kterých se vede evidence o dosud nepřečtené, respektive dosud nezapsané části operandu a cykl se opakuje až do vyčerpání délky operandu. Před zahájením takového cyklu bývá obvykle nutné ještě zjišťovat polohu začátku operandu vzhledem k útvaru, který je možno z paměti najednou přečíst, respektive do paměti zapsat. U většiny konstrukcí je rozhodující sudost nebo lichost adresy začátku operandu. Obvykle se čte, respektive zapisuje po dvojslabikách, tj. 16 bitů, ale adresa musí být sudá. Je-li adresa lichá, je nutno nejprve číst, respektive zapsat jednu slabiku, tj. 8 bitů, a dále pak pokračovat ve styku po dvojslabikách. Rovněž konec styku s pamětí může být dvojslabičný, končí-li operand na liché adrese, nebo jednoslabičný, končí-li operand na sudé adrese. Styk pro jednu slabiku trvá stejnou dobu jako styk pro dvojslabiku. Styk po slabikách je tedy z hlediska rychlosti nevýhodný. Cyklus pro styk s pamětí vyžaduje přípravu a obsahuje vedlejší činnost, která zpomaluje styk s pamětí. Časová ztráta je zvláště výrazná, jsou-li operandy krátké.

Z toho důvodu se jeví výhodné, styk s pamětí pro krátké operandy řešit rozskokem do větvi, obsahujících jeden nebo několik po sobě následujících příkazů pro styk s pamětí, které přesně vyhovují délce operandu a sudosti, respektive lichosti adresy jeho začátku. Při takovém řešení styku s pamětí je jedinou přípravnou činností volba vhodné větve mikroprogramu. Tuto volbu lze provést postupným testováním délky operandu a testováním adresy začátku operandu na sudost. Takové rozhodování je pomalé a znehodnocuje jinak výhodné řešení styku s pamětí.

Dosavadní obvod pro provádění volby větve mikroprogramu sestává z testovacích obvodů spojených s obvody adresní aritmetiky řídící paměti, které spolu zajistí konečné dosažení adresy začátku vhodné větve mikroprogramu, ovšem stane se tak postupnými změnami adresy na více kroků.

Popsaný nedostatek, to je postupnou a tedy pomalou volbu větve mikroprogramu, odstraňuje obvod pro rychlý rozskok do větvi mikroprogramu podle délky a polohy operandu, podle vynálezu, jehož podstatou je, že aritmetická jednotka je vybavena vstupem pro součet adresy mikroinstrukce s kódem délky operandu, ke kterému je připojen výstup registru kódu délky operandu, dále je aritmetická jednotka vybavena vstupem pro modifikaci adresy mikroinstrukce sudosti nebo lichosti adresy operandu, ke kterému je připojen výstup registru adresy operandu a dále je aritmetická jednotka vybavena dalším vstupem, připojeným na registr adresy mikroinstrukce a výstup aritmetické jednotky je propojen na vstup řídící paměti a na vstup registru mikroinstrukce.

Shora uvedeným propojením částí obvodu pro rychlý rozskok se docílí ovlivnění tvorby adresy mikroinstrukce obsahem registru kódu délky operandu a obsahem vybrané části registru adresy operandu a tím se umožní volba větve mikroprogramu podle délky a polohy operandu v rámci jediné mikroinstrukce. Tato volba je podstatně rychlejší proti postupné volbě, vnučené dosavadním obvodem.

Na připojeném výkrese je schematicky znázorněn obvod podle vynálezu.

Obvod dle vynálezu sestává z aritmetické jednotky 1, z registru 2 adresy mikroinstrukce, z registru 3 kódu délky operandu, z registru 4 adresy operandu a z řídící paměti 5. Na vstup 102 aritmetické jednotky 1 je připojen výstup registru 2 adresy mikroinstrukce, na vstup 103 aritmetické jednotky 1 je připojen výstup registru 3 kódu délky operandu a na vstup 104

aritmetické jednotky 1 je připojen výstup registru 4 adresy operandu. Výstup aritmetické jednotky 1 je připojen na vstup registru 2 adresy mikroinstrukce a na vstup řídicí paměti 5.

Aritmetická jednotka 1, za předpokladu, že současnou mikroinstrukcí je právě rychlý rozsok, vytvoří adresu příští mikroinstrukce sečtením čtyř následujících binárních čísel: jedničky, generované uvnitř aritmetické jednotky 1 adresy současné mikroinstrukce, uložené v registru 2 adresy mikroinstrukce, kódu délky operandu, uloženým v registru 3 kódu délky operandu, proměnné, charakterizující sudost nebo lichost adresy začátku operandu, přičemž tato proměnná je vytvořena uvnitř aritmetické jednotky 1 na základě adresy operandu, uložené v registru 4 adresy operandu.

Takto vytvořená adresa nové mikroinstrukce je jednak uložena do registru 2 adresy mikroinstrukce, jednak je použita v řídicí paměti 5 pro výběr nové mikroinstrukce. Aritmetická jednotka 1 vytváří adresu příští mikroinstrukce shora uvedeným sečtením čtyř čísel jen v případě, že současná mikroinstrukce je rychlý rozsok. Jestliže současná mikroinstrukce je řadová, vytvoří aritmetická jednotka 1 adresu nové mikroinstrukce pouze přičtením vnitřně generované jedničky k adrese uložené v registru 2 adresy mikroinstrukce, což je běžná funkce. Při některých jiných mikroinstrukcích se tvoří adresa nové mikroinstrukce zcela jinak, což není v obrázku vyznačeno. Při vykonávání mikroinstrukce různé od rychlého rozsoku, se uplatní jen známé části obvodu.

V dalším textu je zařazena tabulka TAB. 2, znázorňující příklad mikroprogramu, v němž je využit rychlý rozsok, umožněný obvodem dle vynálezu. Pro tento příklad je zvoleno omezení délky operandu na 4 slabiky. Delší operand by byl čten pomocí cyklu, což není v tabulce TAB. 2 znázorněno. Obvod dle vynálezu připočte při rychlém rozsoku k adrese mikroinstrukce, uložené v registru 2 adresy mikroinstrukce binární přírůstek, který je definován takto:

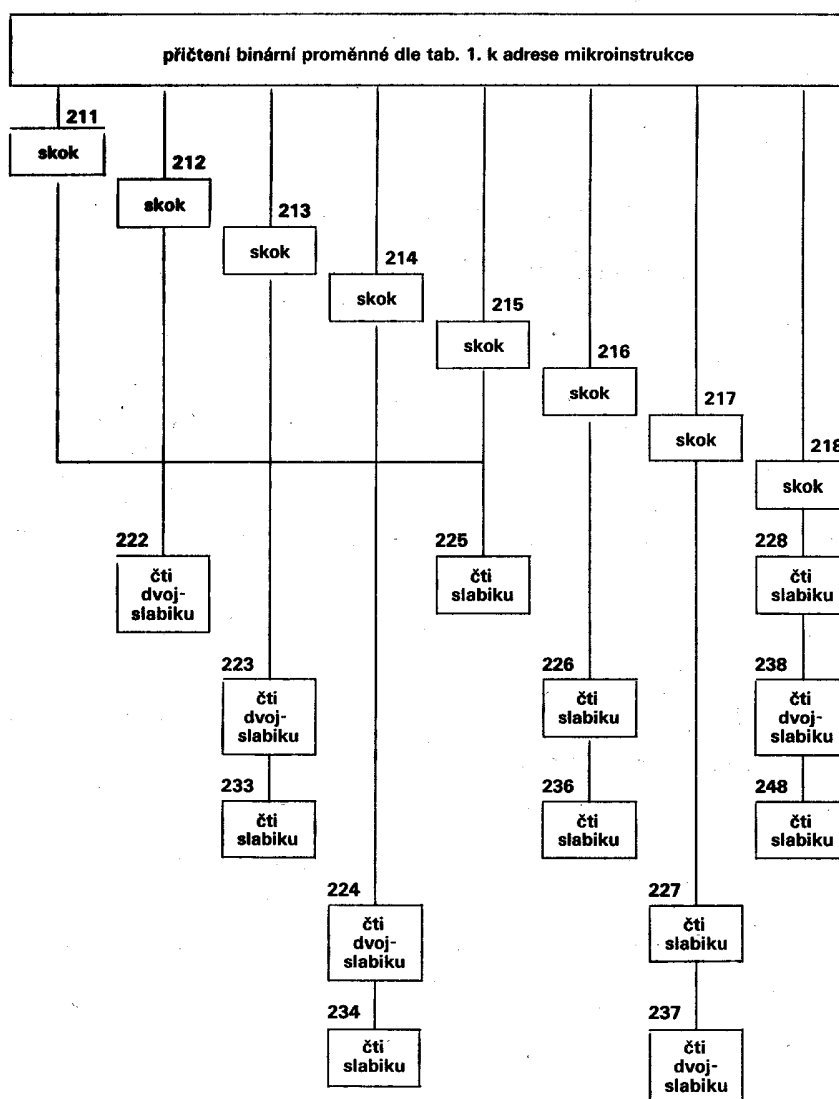
pro sudou adresu začátku operandu ...1 + kód délky + 0000

pro lichou adresu začátku operandu ..1 + kód délky + 0100

Přehled binárních přírůstků adresy mikroinstrukce pro všechny situace uvedeného příkladu podává tabulka TAB. 1.

adresa začátku operandu	délka operandu (slabika)	kód délky	binární přírůstek adresy	dekadický přírůstek adresy
sudá	1	00	0001	1
sudá	2	01	0010	2
sudá	3	10	0011	3
sudá	4	11	0100	4
lichá	1	00	0101	5
lichá	2	01	0110	6
lichá	3	10	0111	7
lichá	4	11	1000	8

TAB. 1



TAB. 2.

Přírůstek adresy v dekadickém tvaru je uveden jen pro lepší srozumitelnost. Obvod dle vynálezu pracuje binárně. Jedinou mikroinstrukcí 201 se uvede v činnost obvod dle obrázku, který přičte k adrese současné mikroinstrukce 201 přírůstek dle TAB. 1. Za mikroinstrukcí 201 následuje v mikroprogramu těsně za sebou osm skoků 211 až 218. Posloupnout adres mikroinstrukcí 201, 211, 212, 213, 214, 215, 216, 217 a 218 je graficky zdůrazněna vždy nižším umístěním v TAB. 2. Začátky posloupností mikroinstrukcí za skoky mohou ležet kdekoli v řídicí paměti a výška jejich umístění v TAB. 2 nic neznamená. Mikroinstrukcí 201 se provede volba jednoho z osmi skoků 211 až 218 a tím i volba vhodné větve mikroprogramu. Například je-li délka operandu 4 slabiky a jeho začátek na sudé adrese, připočte obvod dle obrázku uvedený v činnosti mikroinstrukcí 201 k adrese této mikroinstrukce přírůstek 4, takže se provede skok 214 a po něm mikroinstrukce 224 a 234. Operand je přečten nadvakrát po dvojslabikách. Jiný příklad. Délka operandu je 4 slabiky, operand začíná na liché adrese. Obvod dle vynálezu, uvedený v činnosti mikroinstrukcí 201, připočte k adrese této mikroinstrukce přírůstek 8, takže se provede skok 218 a po něm mikroinstrukce 228, 238 a 248. Operand je přečten natřikrát.

Obvod na provádění rychlého rozskoku do větví mikroprogramu podle délky a polohy operandu je určen pro menší počítače, jejichž technické vybavení běžně neumožňuje provádět paralelně více činností v jediné mikroinstrukci.

PŘEDMĚT VYNÁLEZU

Obvod pro rychlý rozskok do větví mikroprogramu podle délky a polohy operandu, vyznačený tím, že aritmetická jednotka (1) je vybavena vstupem (103) pro součet adresy mikroinstrukce s kódem délky operandu, ke kterému je připojen výstup registru (3) kódu délky operandu, dále je aritmetická jednotka (1) vybavena vstupem (104) pro modifikaci adresy mikroinstrukce sudostí nebo lichostí adresy operandu, ke kterému je připojen výstup registru (4) adresy operandu a dále je aritmetická jednotka (1) vybavena dalším vstupem (102), připojeným na registr (2) adresy mikroinstrukce a výstup aritmetické jednotky (1) je propojen na vstup řídicí paměti (5) a na vstup registru (2) adresy mikroinstrukce.

1 list výkresů

