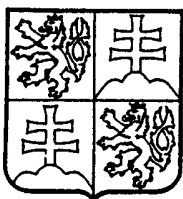


ČESKÁ A SLOVENSKÁ
FEDERATIVNÍ
REPUBLIKA
(19)



FEDERÁLNÍ ÚŘAD
PRO VYNÁLEZY

ZVEŘEJNĚNÁ PŘIHLÁŠKA VYNÁLEZU

(12)

(21) 00933-91.A

(13) A3

(22) 04.04.91

(32) 10.05.90

(31) 90/522291

(33) US

(40) 13.05.92

5(51) G 06 F 15/00,
15/16,
15/20,
15/76

(71) International Business Machines Corporation, Armonk, New York, US

(72) Blaner Bartholomew, Newark Valley, New York, US
Vassiliadis Stamatias, Vestal, New York, US

(54) Zapojení pro paralelní zpracovávání dvou nebo více instrukcí paralelně v číslicovém počítači

(57)

Zapojení obsahuje první paměťový obvod (10), opatřený univerzálním vstupem (10a), slučovací obvod (11) instrukcí opatřený univerzálním vstupem (11a) připojeným k univerzálnímu výstupu (10a) prvního paměťového obvodu (10) a dále opatřený druhým univerzálním vstupem (11b), druhý paměťový obvod (12) opatřený univerzálním vstupem (12a) připojeným ke druhému univerzálnímu výstupu (11b) slučovacího obvodu (11) instrukcí a dále opatřený druhým univerzálním vstupem (12b), opatřený univerzálním výstupem (16) opatřený univerzálním vstupem (16a) připojeným k druhému univerzálnímu výstupu (12b) druhého paměťového obvodu (12) a dále opatřený druhým univerzálním vstupem (16b). Dále obsahuje nejméně dvě funkční jednotky (13, 14, 15) pro zpracovávání instrukcí, opatřené univerzálním vstupem (13a, 14a, 15a), připojeným ke druhému univerzálnímu výstupu (16b) vyvolávacího a vydávacího obvodu (16).

Zapojení pro paralelní zpracovávání dvou nebo více instrukcí paralelně v číslicovém počítači

Oblast techniky

Vynález se týká číslicových počítačů a číslicových datových procesorů a obzvláště číslicových počítačů a datových procesorů schopných zpracovávat paralelně dvě nebo více instrukcí.

Stav techniky

Výkon tradičních počítačů, které vykonávají instrukce po jedné postupným způsobem se v minulosti výrazně zlepšil vzhledem ke zlepšením v obvodové technice. Takové počítače pro vykonávání instrukcí po jedné jsou někdy označovány jako skalární počítače nebo procesory. Jelikož se vývoj obvodové techniky dostal na své hranice, ukázalo se jako nutné, aby návrháři počítačů vyšetřovali jiné prostředky pro získání výrazných zlepšení výkonu.

Nedávno byly navrženy takzvané superskalární počítače, které usilují o zvýšení výkonu vykonáváním více než jedné instrukce současně z jediného proudu instrukcí. Takové navrhované skalární stroje typicky rozhodují v době provádění instrukcí, jestliže může být daný počet instrukcí vykonáván paralelně. Takové rozhodnutí je založeno na operačních kódech instrukcí a na datových závislostech, které mohou existovat mezi přilehlými instrukcemi. Operační kódy určují hardwarové součástky, které každá z instrukcí bude užívat, a všeobecně není možné, aby dvě nebo více instrukcí používaly stejnou hardwarovou součástku ve stejné době, nebo aby vykonávaly instrukci, která závisí na výsledcích předchozí instrukce. Toto je známo pod označením datová závislost. Tyto hardwarové a datové závislosti brání vykonávání některých kombinací instrukcí paralelně. V tomto případě jsou dotřené instrukce místo toho vykonávány samotné neparalelním způsobem. To samozřejmě snižuje výkon superskalárního stroje.

Navržené superskalární počítače přinášejí určitá zlepšení ve výkonu, ale mají také nevýhody, které by bylo žádoucí minimalizovat. Jednak dochází k tomu, že rozhodování v době vykonávání instrukcí o tom, které instrukce mohou být vykonávány paralelně, vyžaduje sice malé, ale znatelné množství času, které nemůže být pohotově maskováno jeho překrytím jinými normálními strojovými operacemi. Tato nevýhoda se stává výraznější s tím, jak roste složitost architektury instrukčního souboru. Jinou nevýhodou je to, že rozhodování musí být opakováno opět pokaždé, kdy se mají tytéž instrukce vykonávat po druhé a po více krátě.

Charakteristika vynálezu

Jako je popisováno v související patentové přihlášce USA č. 08/519 384, jeden z atributů počítače typu SCISM/ Scalable Compound Instruction Set/ je: Neprováděj rozhodování o paralelním vykonávání instrukcí v době vykonávání. Udělej to v dřívějším bodě při celkovém procesu zpracovávání instrukcí. Například, udělej to před vyrovnávací pamětí instrukcí v těch strojích, které mají vyrovnávací paměti instrukcí nebo zásobníky instrukcí. V jiném příkladě, udělej to před rychlou vyrovnávací pamětí instrukcí v těch strojích, kde jsou instrukce vedeny jednotkou s rychlou vyrovnávací pamětí.

Jiným atributem stroje SCISM je zaznamenávat výsledky rozhodování o paralelním vykonávání instrukcí tak, že takové výsledky jsou k dispozici v případě, kdy jsou stejné instrukce používány po druhé a v dalším případě.

Podle vynálezu se zaznamenávání učinění rozhodnutí o paralelním vykonávání instrukcí dosahuje generováním příznaků, které jsou připojovány nebo jsou vkládány do jednotlivých instrukcí v instrukčním toku. Tyto příznaky říkají, zda instrukce mohou být vykonávány paralelně nebo zda potřebují být vykonávány jednotlivě. Tento proces označování příznakem se zde někdy pojmenovává slučování. Slouží skutečně

ke kombinování dvou nebo více individuálních instrukcí do jediné sloučené instrukce pro účely paralelního vykonávání.

V obzvláště výhodném provedení vynálezu je počítač opatřen mechanismem s rychlou vyrovnávací pamětí pro dočasné ukládání strojových instrukcí na jejich cestě od paměťové úrovně počítače o vyšší úrovni do jednotek počítače pro vykonávání instrukcí, a slučování nebo označování instrukcí příznakem se vykonává mezilehle mezi paměťovou jednotkou vyšší úrovně a ukládacím mechanismem s rychlou vyrovnávací pamětí, takže v mechanismu s rychlou vyrovnávací pamětí jsou uloženy jak instrukce, tak i slučovací příznaky. Jak je známo, použití dobře navržené rychlé vyrovnávací paměti, navržené jako samotné, tak i v návaznosti na počítač, slouží pro zdokonalení celkové výkonnosti počítače. Ukládání slučovacích příznaků do rychlé vyrovnávací paměti umožňuje přitom dále, aby příznaky mohly být používány stále znovu, pokud dotyčné instrukce zůstávají v mechanismu s vyrovnávací pamětí. Jak je známo, instrukce často zůstávají v rychlé vyrovnávací paměti dosti dlouho pro to, aby byly používány více než jednou.

Popis obrázků na výkresech

Vynález je blíže vysvětlen v následujícím popise na příkladech provedení s odvoláním na připojené výkresy, ve kterých znázorňuje obr. 1 blokové schema příkladného provedení části číslicového počítačového systému, konstruovaného podle vynálezu, obr. 2 schema úseku instrukčního toku, v němž jsou slučovací příznaky nebo příznaková pole sdruženy s instrukcemi, obr. 3 větší podrobnost vnitřní konstrukce reprezentativního provedení slučovací jednotky instrukcí, která může být použita v počítačovém systému z obr. 1, obr. 4 blokové schema větší podrobnosti reprezentativní vnitřní konstrukce analyzních jednotek slučování z obr. 3, obr. 5 příklad soustavy logických obvodů, která může být použita pro implementaci částí analyzátoru slučování a generátoru příznaků z obr. 3, které vytvářejí slučovací příznaky pro první

tři instrukce v instrukčním toku, obr. 6 tabulku použitou pro vysvětlení příkladu z obr. 5, obr. 7 blokové schema příkladu reprezentativního provedení části číslicového počítačového systému, použité pro vysvětlení, jak mohou být slučované instrukce zpracovávány paralelně funkčními jednotkami na zpracovávání instrukcí, obr. 8 schema obzvláštního sledu instrukcí, který může být zpracováván počítačovým systémem z obr. 7, a obr. 9 tabulku použitou pro vysvětlení zpracování instrukčního sledu z obr. 8 počítačovým systémem z obr. 7.

Provedení vynálezu

Na obr. 1 je znázorněno reprezentativní provedení části číslicového počítačového systému nebo číslicového zpracovávacího systému konstruovaného podle vynálezu. Tento počítačový systém je schopný zpracovávání dvou nebo více instrukcí paralelně. Zahrnuje první paměťový mechanismus pro ukládání instrukcí a dat, která se mají zpracovávat. Tento paměťový mechanismus je provedený jako vysokoúrovňová paměť, je na obr. 1 vyznačen jako první paměťový obvod 10. Tento první paměťový obvod 10 je velkokapacitní paměťový mechanismus s nižší rychlostí a může být tvořen velkokapacitní systémovou paměťovou jednotkou nebo dolní částí univerzálního hierarchického paměťového systému apod.

Počítačový systém z obr. 1 také zahrnuje slučovací mechanismus instrukcí pro přijímání instrukcí z prvního paměťového obvodu 10 přes univerzální výstup 10a a pro sdružování příznakových polí s těmito instrukcemi, přičemž tato příznaková pole udávají, které z těchto instrukcí mohou být zpracovávány paralelně s jinými. Tento slučovací mechanismus instrukcí je reprezentován slučovacím obvodem 11 instrukcí s univerzálním vstupem 11a. Tento slučovací obvod 11 instrukcí analyzuje přicházející instrukce pro určování, které mohou být zpracovávány paralelně. Dále vytváří slučovací obvod 11

instrukcí pro tyto analyzované instrukce příznakovou informací nebo příznaková pole, která udávají, které instrukce mohou být zpracovávány paralelně s druhými a které nesmí být zpracovávány paralelně s jinými.

System z obr. 1 dále zahrnuje druhý paměťový obvod 12 spojený s univerzálním výstupem 11b slučovacího obvodu 11 instrukcí pro přijímání a ukládání analyzovaných instrukcí a jejich přidružených příznakových polí přes univerzální vstup 12a. Tento druhý paměťový obvod 12 je reprezentován rychlou vyrovnávací pamětí sloučených instrukcí. Rychlá vyrovnávací paměť je paměťový mechanismus s menší kapacitou a větší rychlostí typu obvykle používaného pro zlepšování výkonu počítačového systému z hlediska rychlosti snižováním častosti přístupu na první paměťový obvod 10 s nižší rychlostí.

Na obr. 1 systém dále zahrnuje řadu funkčních jednotek pro zpracovávání instrukcí, které pracují vzájemně paralelně. Tyto funkční jednotky pro zpracovávání instrukcí jsou reprezentovány funkčními jednotkami 13, 14, 15 atd. Tyto funkční jednotky 13 - 15, opatřená každým univerzálním vstupem 13a, 14a 15a pracují paralelně souběžně spolu a každá sama je schopná zpracovávat jeden nebo více typů instrukcí strojového kódu. Příklady funkčních jednotek, které mohou být použity, jsou: univerzální aritmetická a logická jednotka, aritmeticko-logická jednotka typu pro generování adres, aritmeticko-logická jednotka pro překonávání datové závislosti, zpracovávací jednotka instrukcí větvení, jednotka pro posouvání dat, zpracovávací jednotka s pohyblivou řádovou čárkou atd. Daný počítačový systém může obsahovat dva nebo více těchto typů funkčních jednotek. Například může daný počítačový systém obsahovat dvě nebo více univerzálních aritmeticko-logických jednotek. Žádný daný počítačový systém také nemusí obsahovat každou nebo všechny z těchto funkčních jednotek různého typu. Obzvláštní uspořádání a konfigurace funkčních jednotek bude záviset na povaze obzvláštního počítačového systému, jaký je uvažován.

Počítačový systém z obr. 1 také zahrnuje mechanismus pro vyvolávání a vydávání, připojený k rychlé vyrovnávací paměti druhého paměťového obvodu 12, pro poskytování přilehlých instrukcí, které jsou v ní uloženy, a pro jejich poskytování různým funkčním jednotkám 13 - 15 pro zpracovávání instrukcí, když příznaková pole instrukcí indikují, že mohou být zpracovávány paralelně. Tento mechanismus je znázorněn jako vyvolávací a vydávací obvod 16 s univerzálním vstupem 16a a univerzálním výstupem 16b. Vyvolávací a vydávací jednotka 16 vyvolává přes univerzální vstup 16a instrukce z rychlé vyrovnávací paměti druhého paměťového obvodu 12, vyšetřuje jejich příznaková pole a pole operačních kódů a na základě takových vyšetření vysílá přes univerzální výstup 16b instrukce do odpovídající z funkčních jednotek 13 - 15.

Jestliže je požadovaná instrukce přítomna v rychlé vyrovnávací paměti slučovaných instrukcí, tj. ve druhém paměťovém obvodu 12, je odpovídající adresa vyslána do rychlé vyrovnávací paměti pro to, aby z ní byla vyvolána požadovaná instrukce. Toto se někdy označuje jako "cache hit" (úspěšné vyvolání). Není-li požadovaná instrukce v rychlé vyrovnávací paměti, tj. druhém paměťovém obvodu 12, potom musí být vyvolána z paměti vyšší úrovně, tj. prvního paměťového obvodu 10, a zavedena do rychlé vyrovnávací paměti. To se někdy označuje jako "cache miss" (neúspěšné vyvolání z rychlé vyrovnávací paměti). Když dojde k neúspěšnému vyvolání, je adresa požadované instrukce vyslána do prvního paměťového obvodu 10 vyšší úrovně. Na základě tohoto první paměťový obvod 10 začne vyjímání nebo čtení řady instrukcí, která obsahuje požadovanou instrukci. Tyto instrukce jsou přenášeny na univerzální vstup 11a slučovacího obvodu 11 instrukcí, který analyzuje tyto vstupující instrukce a generuje odpovídající příznaková pole pro každou instrukci. Instrukce opatřené příznaky jsou po té vedeny do rychlé vyrovnávací paměti slučovaných instrukcí, tj. druhého paměťového obvodu 12 a jsou zde uloženy pro následující pou-

žití, v případě potřeby, funkčními jednotkami 13, 14 a 15.

Analýza instrukcí vykonávaná ve slučovacím obvodu 11 instrukcí vyžaduje určité relativně malé množství času. Analýza slučování instrukcí se však provádí pouze v případě výskytu neúspěšného vyvolávání a je tak relativně málo častá.

Obr. 2 ukazuje část proudu sloučených instrukcí opatřených příznaky, jak se mohou objevovat na výstupu slučovacího obvodu 11 instrukcí z obr. 1. Jak je patrné, každá instrukce ("INSTR") má příznakové pole (TAG), připojené slučovacím obvodem 11 instrukcí. Instrukce s příznaky, jako jsou instrukce znázorněné na obr. 2, jsou ukládány do rychlé vyrovnávací paměti sloučených instrukcí, tj. druhého paměťového obvodu 12. Podle potřeby jsou instrukce s příznaky v rychlé vyrovnávací paměti vyvolávány vyvolávacím a vydávacím obvodem 16 instrukcí. Když jsou instrukce s příznaky přijímány vyvolávacím a vydávacím obvodem 16, jejich příznaková pole jsou vyšetřována pro určení, jestli mohou být zpracovávány paralelně a jejich pole operačních kódů jsou vyšetřována, která z funkčních jednotek, které jsou k dispozici, se nejlépe hodí pro jejich zpracovávání. Jestliže příznaková pole indikují, že dvě nebo více instrukcí jsou vhodné pro zpracovávání paralelně, jsou vysílány do odpovídající z funkčních jednotek v souladu s kódováním jejich polí operačních kódů. Takové instrukce jsou pak zpracovávány souběžně jedna s druhou jejich odpovídajícími funkčními jednotkami.

Zjistí-li se instrukce, která není vhodná pro paralelní zpracovávání, vyšle se potom do odpovídající funkční jednotky, jak je určeno jejím operačním kódem a potom se zpracovává samotná zvolenou funkční jednotkou.

V nejdokonalejším případě, kde je vždy zpracováváno paralelně více instrukcí, by rychlost vykonávání instrukce počítačového systému byla N krát větší, než v případě, kdy se instrukce vykonávají vždy po jedné, přičemž N je počet instrukcí ve skupinách, které se zpracovávají paralelně.

Obr. 3 ukazuje podrobněji vnitřní konstrukci reprezentativního provedení slučovacího obvodu instrukcí v souladu s vynálezem. Tento slučovací obvod instrukcí, označený v podrobnostech jako slučovací jednotka 20, je vhodný pro použití jako slučovací obvod 11 instrukcí z obr. 1. Slučovací jednotka 20 instrukcí z obr. 3 je určena pro případ, kdy se současně zpracovává maximální množství dvou instrukcí paralelně. V tomto případě se použije jednobitový příznak. Hodnota příznakového bitu "jedna" znamená, že instrukce je "první" instrukce. Hodnota příznakového bitu "nula" znamená, že instrukce je "druhá" instrukce a může být vykonávána paralelně s předcházející první instrukcí. Instrukce mající hodnotu příznakového bitu jedna může být vykonávána buď sama nebo současně a paralelně s příští instrukcí, v závislosti na hodnotě příznaku pro takovou příští instrukci.

Každé párování instrukcí, mající hodnotu příznakového bitu jedna, s následující instrukcí, mající hodnotu příznakového bitu nula, tvoří sloučenou instrukci pro účely paralelního vykonávání, tj. že instrukce v takové dvojici mohou být vykonávány vzájemně paralelně. Když mají příznakové bity pro dvě po sobě následující instrukce každá hodnotu jedna, vykonává se první z těchto instrukcí sama neparalelním způsobem. V nejhorším možném případě by všechny instrukce ve sledu měly příznakovou bitovou hodnotu jedna. V tomto nejhorším případě by všechny instrukce byly vykonávány současně a neparalelním způsobem.

Slučovací jednotka 20 instrukcí z obr. 3 obsahuje vstupní registr 21 plurálních instrukcí pro přijímání více po sobě následujících instrukcí z paměťové jednotky vyšší úrovně, tj. prvního paměťového obvodu 10. Vstupní registr 21 má nejméně tři paměťové pole INST0, INST1, INST2 a je opatřen univerzálním vstupem 21a připojeným k univerzálnímu výstupu 10a prvního paměťového obvodu 10. Slučovací jednotka 20 instrukcí také obsahuje skupinu analyzních mechanismů, založených na pravidle. Každý takový analyzní mechanismus instruk-

cí analyzuje rozdílnou dvojici instrukcí ležících vedle sebe ve vstupním registru 21 instrukcí a vytváří signál slučovatelnosti, který indikuje, zda tyto dvě instrukce v této dvojici mohou být zpracovávány paralelně či nikoliv. Na obr. 3 je znázorněno několik analyzních obvodů 22 - 25 slučovatelnosti. Každý z těchto analyzních obvodů 22 - 25 slučovatelnosti obsahuje dva právě zmíněné analyzní mechanismy instrukce. Každý z těchto analyzních obvodů 22 - 25 tak vytváří dva ze signálů slučovatelnosti. Například první analyzní obvod 22 slučovatelnosti vytváří na prvním výstupu M01 první signál slučovatelnosti, který indikuje, zda Instrukce 0 a 1 mohou být zpracovávány paralelně. Analyzní obvod 22 slučování také vytváří na druhém výstupu M12 druhý signál slučovatelnosti, který indikuje, zda mohou být Instrukce 1 a 2 zpracovávány paralelně.

Podobným způsobem vytváří druhý analyzní obvod 23 slučování první signál slučovatelnosti na prvním výstupu M23, který indikuje, zda Instrukce 2 a 3 mohou být zpracovávány paralelně, a druhý signál slučovatelnosti na druhém výstupu M34, který indikuje, zda Instrukce 3 a 4 mohou být zpracovávány paralelně. Třetí analyzní obvod 24 slučování vytváří na prvním výstupu M45 první signál slučovatelnosti, který udává, zda Instrukce 4 a 5 mohou být zpracovávány paralelně a na druhém výstupu M56 druhý signál slučovatelnosti, který udává, zda Instrukce 5 a 6 mohou být zpracovávány paralelně. Čtvrtý analyzní obvod 25 slučovatelnosti vytváří na prvním výstupu M67 první signál slučovatelnosti, který udává, zda Instrukce 6 a 7 mohou být zpracovávány paralelně, a druhý signál slučovatelnosti na druhém výstupu M78, který udává, zda Instrukce 7 a 8 mohou být zpracovávány paralelně.

Slučovací jednotka 20 instrukcí dále obsahuje obvod 26 generování příznaků, citlivý na signály slučovatelnosti, objevující se na výstupech analyzních obvodů 22 - 25 pro generování individuálních příznakových polí pro různé instrukce ve vstupním registru 21 instrukcí. Obvod 26 pro ge-

nerování příznaků má nejméně dva vstupy připojené ke dvěma výstupům MO1, M12 analyzního obvodu 22 slučovatelnosti. Dále má výstupy TAGO, TAG1, TAG2, TAG3, TAG4, TAG5, TAG6, TAG7 a TAG8 pro generování příznaků připojené k odpovídajícím vstupům výstupního registru 27. Výstupní registr 27 má nejméně tři paměťová pole, v daném případě paměťová pole TO, INSTO, T1, INST1, T2, INST2, ... T8, INST8.

Individuální příznaková pole pro různé instrukce ve vstupním registru instrukcí jsou přiváděna do výstupního registru 27, jako jsou instrukce samotné, získávané ze vstupního registru 21 instrukcí. Tímto způsobem je ve výstupním registru 27 slučovací jednotky paměťové pole TO příznaku pro Instrukci 0, paměťové pole T1 příznaku pro Instrukci 1, atd.

V daném provedení je každé paměťové pole TO, T1, T2 příznaku atd. tvořeno jediným binárním bitem. Hodnota příznakového bitu "jedna" indikuje, že bezprostředně následující instrukce, ke které je připojen, je "první instrukce". Hodnota příznakového bitu "nula" indikuje, že bezprostředně následující instrukce je "druhá instrukce". Instrukce mající hodnotu příznakového bitu jedna, následovaná instrukcí, mající hodnotu příznakového bitu nula, indikuje, že tyto dvě instrukce mohou být vykonávány paralelně jedna vůči druhé. Instrukce opatřené příznaky ve výstupním registru 27 slučovaných instrukcí jsou vedeny na vstup rychlé vyrovnávací paměti slučovaných instrukcí, tj. druhého paměťového obvodu slučovaných instrukcí z obr. 12 a jsou ukládány do druhého paměťového obvodu 12.

Je třeba poznamenat, že množství registrového hardwaru znázorněného na obr. 3 může být redukováno ukládáním slučovaných instrukcí přímo do rychlé vyrovnávací paměti slučovaných instrukcí.

Na obr. 4 je znázorněna do větších podrobností vnitřní konstrukce použitá pro analyzní obvod 22 slučovatelnosti instrukcí z obr. 3. Druhé analyzní obvody 22 - 25 slučovatelnosti mají podobnou konstrukci. Jak ukazuje obr. 4, analyz-

ní obvod 22 slučovatelnosti zjišťovací obvod 30 slučovatelnosti instrukcí pro vyšetřování operačního kódu Instrukce 0 a operačního kódu Instrukce 1 a pro určování, zda tyto dva operační kódy jsou kompatibilní pro účely paralelního vykonávání. Zjišťovací obvod 30 je konstruován v souladu s předem určenými pravidly pro volbu toho, které páry operačních kódů jsou kompatibilní pro vykonávání paralelně. Konkrétněji obsahuje zjišťovací obvod 30 logické obvody pro implementování pravidel, která definují, jaké typy instrukcí jsou kompatibilní pro paralelní vykonávání v určité hardwarové konfiguraci, použité pro uvažovaný počítačový systém. Jestliže operační kódy pro Instrukci 0 a Instrukci 1 jsou kompatibilní, potom zjišťovací obvod 30 vytváří na svém výstupu 30a výstupní signál úrovně binární jedničky. Jestliže nejsou kompatibilní, zjišťovací obvod 30 vytváří binární nulovou hodnotu na svém výstupu 30a. Jak je patrné z obr. 4, má zjišťovací obvod 30 dva vstupy OP0 a OP1, které jsou připojeny k univerzálnímu výstupu 21b vstupního registru 21 z obr. 1.

Analýzní obvod 22 slučovatelnosti signálů dále obsahuje zjišťovací obvod 31 slučitelnosti druhé instrukce pro vyšetřování operačních kódů Instrukcí 1 a 2 a určování, zda jsou kompatibilní pro paralelní vykonávání. Zjišťovací obvod 31 je konstruován stejným způsobem jako zjišťovací obvod 30 pro volbu, které dvojice operačních kódů jsou kompatibilní pro paralelní vykonávání pro případ Instrukcí 1 a 2. Zjišťovací obvod 31 tak zahrnuje soustavu logických obvodů pro implementování pravidel, která definují, které typy instrukcí jsou kompatibilní pro paralelní vykonávání, přičemž tato pravidla jsou stejná, jaká jsou použita ve zjišťovacím obvodu 30. Jestliže jsou operační kódy pro Instrukce 1 a 2 kompatibilní, potom zjišťovací obvod 31 vytváří na svém výstupu 31a výstupní signál na binární úrovni jedna. Jinak vytváří výstupní signál na binární úrovni nula. Obdobně jako u zjišťovacího obvodu 30 jsou i zde vstupy OP1, OP2, připojené ke vstupnímu registru 21.

Analyzní obvod 22 slučovatelnosti signálů dále obsahuje zjišťovací obvod 31 slučitelnosti druhé instrukce pro vyšetřování operačních kódů Instrukcí 1 a 2 a určování, zda jsou kompatibilní pro paralelní vykonávání. Zjišťovací obvod 31 je konstruován stejným způsobem jako zjišťovací obvod 30 pro volbu, které dvojice operačních kódů jsou kompatibilní pro paralelní vykonávání pro případ Instrukcí 1 a 2. Zjišťovací obvod 31 tak zahrnuje soustavu logických obvodů pro implementování pravidel, která definují, které typy instrukcí jsou kompatibilní pro paralelní vykonávání, přičemž tato pravidla jsou stejná, jaká jsou použita ve zjišťovacím obvodu 30. Jestliže jsou operační kódy pro Instrukce 1 a 2 kompatibilní, potom zjišťovací obvod 31 vytváří na svém výstupu 31a výstupní signál na binární úrovni jedna. Jinak vytváří výstupní signál na binární úrovni nula. Obdobně jako u zjišťovacího obvodu 30 jsou i zde vstupy OP1, OP2, připojené ke vstupnímu registru 21.

Analyzní obvod 22 slučovatelnosti dále obsahuje první zjišťovací obvod 32 závislosti registrů pro zjišťování konfliktů v použití univerzálních registrů označených poli R1 a R2 Instrukcí 0 a 1. Tyto univerzální registry budou podrobněji rozebírány níže. Mimo jiné může být zjišťovací obvod 32 závislosti konstruován pro zjišťování výskytu podmínky datové závislosti, při níž druhá instrukce (Instrukce 1) potřebuje užívat výsledků získaných vykonáváním předcházející instrukce (Instrukce 0). V tomto případě může být buď druhá instrukce vykonávána hardwarem pro překonávání závislosti, takže je vykonávána paralelně s první instrukcí, anebo musí vykonávání druhé instrukce vyčkat na dokončení vykonávání předchozí instrukce, a tedy nemůže být provedeno paralelně s předcházející instrukcí. (Je třeba poznamenat, že technika pro překonávání některých datových závislostí bude rozebrána níže.) Neexistují-li závislosti registrů, které brání vykonávání Instrukcí

0 a 1 paralelně, potom se vyšle na výstup 32b zjišťovací jednotky 32 binární hodnota jedna. Je-li závislost, potom se vyšle na výstup 32b binární hodnota 0. Vstupy 32a zjišťovací jednotky 32 jsou připojeny k univerzálnímu výstupu vstupního registru 21.

Analyzní obvod 22 slučovatelnosti dále zahrnuje druhý zjišťovací obvod 33 závislosti registrů pro zjišťování konfliktů v použití univerzálních registrů označených poli R1 a R2 Instrukcí 1 a 2. Tento zjišťovací obvod 33 je stejné konstrukce, jako výše popsany zjišťovací obvod 32, a vytváří výstupní signál binární úrovně jedna na výstupu 33b, neexistují-li závislosti registrů nebo závislosti registrů mohou být pokryty hardwarem pro překonávání závislosti registrů, anebo v jiném případě je na výstupu 33b binární úroveň nuly. Druhý zjišťovací obvod 33 má opět vstupy 33a spojené se vstupním registrem 21.

Výstupní vedení od zjišťovacího obvodu 30 slučovatelnosti instrukcí a zjišťovacího obvodu 32 závislosti registrů jsou připojena ke dvěma vstupům 34a, 34b prvního součinného obvodu 34. Výstupní vedení prvního součinného obvodu 34 má binární hodnotu jedna, jestliže jsou oba uvažované operační kódy kompatibilní a jestliže nejsou závislosti registrů. Tato binární jedničková hodnota na výstupním vedení prvního součinného obvodu 34 indikuje, že obě uvažované instrukce jsou slučovatelné, tj. jsou vykonávatelné paralelně. Jestliže naproti tomu má výstupní vedení prvního součinného obvodu 34 binární hodnotu nula, potom obě instrukce nejsou slučovatelné. Je tak na výstupní vedení prvního součinného obvodu 34 vytvářen na výstupu MO1 první signál slučovatelnosti, který indikuje, zda Instrukce 0 a 1 mohou být zpracovávány paralelně. Tento signál z výstupu MO1 je veden do generátoru 26 příznaků.

Výstupní vedení druhého zjišťovacího obvodu 31 slučovatelnosti a druhého zjišťovacího obvodu 33 závislosti jsou připojeny ke dvěma vstupům 35a, 35b druhého součinného ob-

vodu 35. Součinný obvod 35 vytváří na výstupu M12 druhý signál kompatibility, který má binární hodnotu jedna, jestliže oba uvažované operační kódy (operační kódy pro Instrukce 1 a 2) jsou kompatibilní a jestliže neexistují závislosti registrů pro Instrukce 1 a 2 nebo existují závislosti registrů, které mohou být překonány pomocí hardwaru pro překonávání závislosti. Jinak má druhý součinný obvod 35 výstupní vedení binární hodnotu nula. Výstupní vedení od druhého součinného obvodu 35 vede do druhého vstupu obvodu 26 generování příznaků.

Ostatní analyzní obvody 23 až 25 slučovatelnosti, znázorněné na obr. 3, mají stejnou vnitřní konstrukci, jaká je znázorněna na obr. 4 pro první analyzní obvod 22 slučovatelnosti.

Na obr. 5 je znázorněn příklad soustavy logických obvodů, které je možno použít pro implementaci analyzního obvodu 22 slučovatelnosti a části obvodu 26 generování příznaků, který je použit pro generování prvních tří příznaků, a to příznaku TAG 0, TAG 1 a TAG 2. V příkladě z obr. 5 se předpokládá, že jsou dvě kategorie instrukcí, které jsou označeny jako kategorie A a kategorie B. Pravidla pro slučování těchto kategorií instrukcí se předpokládají následující:

- (1) A se může vždy slučovat s A
- (2) A se nikdy nemůže slučovat s B
- (3) B se nikdy nemůže slučovat s B
- (4) B se může vždy slučovat s A
- (5) Pravidlo (4) má přednost před pravidlem (1).

Je třeba si povšimnout, že tato pravidla jsou citlivá na pořadí, v němž se instrukce vyskytují.

Dále se předpokládá, že tato pravidla jsou takové, že když jsou dodržována, nebudou problémy se závislostmi registrů, protože pravidla implicitně indikují, že v případě, kdy se objevuje nějaké vzájemné blokování, jsou takto blo-

kované instrukce vždy proveditelné hardwarem pro překonávání datové závislosti. Jinými slovy se pro příklad na obr. 5 předpokládá, že nejsou zapotřebí zjišťovací obvody 32 a 33 z obr. 4. V takovém případě také nejsou zapotřebí součinné obvody 34 a 35 a výstup zjišťovacího obvodu 30 se stává signálem na výstupu MO1 a výstup zjišťovacího obvodu 31 se stává signálem na výstupu M12.

Pro tyto předpoklady ukazuje obr. 5 vnitřní soustavu logických obvodů, které mohou být použity pro zjišťovací obvod 30 slučovatelnosti instrukcí a pro zjišťovací obvod 31 slučovatelnosti instrukcí z obr. 4. Jak ukazuje obr. 5, obsahuje zjišťovací obvod 30 slučovatelnosti instrukcí dekodéry 40 a 41, součinné obvody 42 a 43 se dvěma vstupy 42a 42b, 43a, 43b a první součtový obvod 44. Zjišťovací obvod 31 slučovatelnosti druhé instrukce obsahuje dekodéry 41 a 45, součinné obvody 46 a 47 a druhý součtový obvod 48. Střední (druhý) dekodér 41 je sdílen oběma zjišťovacími obvody 30 a 31. Ve schématu na obr. 5 jsou tedy zapojeny první dekodér 40 se vstupem OPO a výstupy 40a a 40b, druhý dekodér 41 se vstupem OP1 a výstupy 41a a 41b a třetí dekodér 45 se vstupem OP2 a výstupy 45a a 45b.

První zjišťovací obvod 30 vyšetřuje operační kódy na vstupech OPO a OP1 Instrukcí 0 a 1 pro určování jejich kompatibility pro účely paralelního vykonávání. To se děje v souladu s pravidly (1) - (4) uvedených výše. První dekodér 40 sleduje operační kód první instrukce a jestliže jde o opatření kód kategorie A, je výstup 40a prvního dekodéru 40 nastaven na úroveň jedna. Je-li na vstupu OPO operační kód kategorie B, potom je výstup 40b prvního dekodéru 40 nastaven na úroveň jedna. Nepatří-li kód na vstupu OPO ani do kategorie A ani do kategorie B, potom jsou oba výstupy 40a 40b prvního dekodéru 40 na binární úrovni nula. Druhý dekodér 41 s výstupy 41a, 41b provádí podobný druh dekódování pro druhý operační kód na vstupu OP1.

První součinnový obvod 42 implementuje výše uvedené pravidlo (1). Je-li operační kód kategorie A a na vstupu OP1 je také operační kód kategorie A, potom první součinnový obvod 42 vytváří výstupní signál na výstupu 42c úrovně jedna. Jinak je na výstupu 42c prvního součinnového obvodu 42 binární úroveň nula. Druhý součinnový obvod 43 implementuje výše uvedené pravidlo (4). Je-li první operační kód operační kód kategorie B a druhý operační kód je operační kód kategorie A, potom druhý součinnový obvod 43 vytvoří na výstupu 43c výstupní úroveň jedna. Jinak vytváří výstup úroveň nula. Jestliže buď první součinnový obvod 42 nebo druhý součinnový obvod 43 vytváří výstup úroveň jedna, uvádí to výstup prvního součtového obvodu 44 na úroveň jedna a signál slučovatelnosti na výstupu MO1 má v tomto případě hodnotu jedna. Tato hodnota jedna indikuje, že první a druhé instrukce (Instrukce 0 a 1) jsou kompatibilní pro účely paralelního vykonávání.

Je-li dekodéry 40 a 41 zjištěna jakékoli jiná kombinace operačních kódů, potom výstupy 42c, 43c součinnových obvodů 42 a 43 zůstávají na nulové úrovni a signál slučovatelnosti na výstupu MO1 má hodnotu nula indikující neslučitelnost. Objevení se kombinací označených pravidly (2) a (3) tak neuspokojuje součinnové obvody 42 a 43 a signál na výstupu MO1 zůstává na nulové úrovni. V případě výskytu dalších kategorií operačních kódů, jiných než jsou kategorie A a B, jejich výskyty v toku instrukcí neaktivují výstupy dekodérů 40 a 41. Bude to tak mít podobně za následek, že hodnota signálu slučovatelnosti na výstupu MO1 bude nula.

Zjišťovací obvod 31 slučovatelnosti druhé instrukce vykonává podobný typ analýzy operačního kódu pro druhou a třetí instrukci (Instrukce 1 a 2). Je-li operační kód na vstupu OP2 je operační kód kategorie A, potom vzhledem k pravidlu (1) třetí součinnový obvod 46 se vstupy 46a, 46b vytváří výstupu 46c výstupní signál úrovně jedna a druhý signál slu-

čovatelnosti na výstupu M12 je uveden na binární úroveň jedna indikující slučovatelnost. Jestliže naproti tomu na vstupu OP1 je operační kód kategorie B a na vstupu OP2 je operační kód kategorie A, potom v důsledku pravidla (4) je čtvrtý součinnový obvod 47 se vstupy 47a, 47b aktivován pro vytváření binární úrovně jedna pro druhý signál slučovatelnosti na výstupu M12. Pro jiné kombinace operativních kódů než kombinace stanovené v pravidlech (1) a (4) má signál na výstupu M12 hodnotu nula. Jak ukazuje obr. 5, má čtvrtý součinnový obvod 47 výstup 47c, připojený ke vstupu 48b druhého součtového obvodu 48, zatímco jeho druhý vstup 48a je spojen s výstupem 46c třetího součinnového obvodu 46.

Signály slučovatelnosti na výstupech MO1 a M12 jsou dodávány do obvodu 26 generování příznaků. Obr. 5 ukazuje soustavu logických obvodů, která může být použita v obvodu 26 generování příznaků pro reagování na signály slučovatelnosti na výstupech MO1 a M12 pro vytváření požadovaných hodnot příznakových bitů pro příznaky TAG 0, 1 a 2. Tabulka na obr. 6 ukazuje logiku, která je implementována obvodem 26 generování příznaků pro příznaky TAG 0, 1 a 2. Hodnota příznakového bitu jedna udává, že přidružená instrukce je "první instrukce" pro účely paralelního provádění. Hodnota příznakového bitu nula indikuje, že přidružení instrukce je "druhá" instrukce pro účely paralelního vykonávání. Jediné páry instrukcí, které jsou slučovány a vykonávány paralelně jsou ty, pro které první instrukce ve dvojici má hodnotu příznakového bitu jedna a druhá instrukce ve dvojici má hodnotu příznakového bitu 0. Jakákoli instrukce mající hodnotu příznakového bitu jedna, po které následuje jiná instrukce mající hodnotu příznakového bitu jedna, je vykonávána samostatně jednotlivým způsobem a nikoliv paralelně s následující instrukcí.

Pro případ z první řádky na obr. 6 mají všechny tři příznakové bity hodnotu jedna. To znamená, že každá z instrukcí 0 a 1 bude vykonána jednotlivým, neparalelním způsobem.

Pokud jde o druhou řádku na obr. 6, instrukce 0 a 1 budou vykonány paralelně, protože příznak 0 má požadovanou hodnotu jedna a příznak 1 má požadovanou hodnotu 0. Pokud jde o třetí řádku na obr. 6, instrukce 0 bude vykonávána jednotlivým způsobem, zatímco instrukce 1 a 2 budou vykonávány vzájemně spolu paralelně. Pokud jde o čtvrtou řádku, instrukce 0 a 1 budou vykonávány vzájemně spolu rovnoběžně.

Pro případy, kde má příznak 2 binární hodnotu jedna, je status jemu přidružené instrukce 2 závislý na binární hodnotě pro příznak 3. Jestliže má příznak 3 binární hodnotu 0, potom instrukce 2 a 3 mohou být vykonávány paralelně. Jestliže naproti tomu má příznak 3 binární hodnotu jedna, potom instrukce 2 bude vykonávána jednotlivým neparalelním způsobem. Je třeba poznamenat, že logika implementovaná pro obvod 26 generování příznaků neimplementuje výskyt dvou po sobě následujících příznakových bitů majících binární hodnoty nula.

Vyšetřování obr. 6 ukazuje logiku, jakou je třeba implementovat částí obvodu 26 generování příznaků, znázorněnou na obr. 5. Jak ukazuje obr. 6, bude mít příznak 0 vždy binární hodnotu jedna. Toto je dosahováno tím, že se přivádí konstantní binární hodnota jedna na výstupní vedení 50 obvodu generování příznaků, které tvoří výstupní vedení příznaku 0 (TAG 0). Prohlídka obr. 6 dále ukazuje, že bitová hodnota pro příznak 1 je vždy opačná bitová hodnotě signálu slučovatelnosti na výstupu M01. Tento výsledek je dosahován připojením výstupního vedení 51 pro příznak 1 k výstupu prvního invertoru 52, jehož vstup je připojen k vedení signálu na výstupu M01.

Binární úroveň na výstupním vedení 53 příznaku 2 je určována součtovým obvodem 54 a druhým invertorem 55. Jeden vstup součtového obvodu 54 je připojen k výstupu M01. Jestliže signál na výstupu M01 má hodnotu jedna, potom příznak 2 má hodnotu jedna. To se týká hodnot příznaku 2 na druhém a čtvrtém řádku obr. 6. Druhý vstup součtového obvodu 54 je

připojen přes druhý invertor 55 k výstupu M12. Jestliže signál na výstupu M12 má binární hodnotu nula, tato hodnota je invertována druhým invertorem 55 na přivádění hodnoty binární jedničky na druhý vstup součtového obvodu 54. To působí, že výstupní vedení 53 příznaku 2 má binární hodnotu jedna. To se týká hodnoty příznaku 2 pro řádek jedna na obr. 6. Je třeba si povšimnout toho, že pro případ řádky 3 musí mít příznak hodnotu nula. K tomu dojde proto, že v tomto případě bude mít signál M01 hodnotu nula a signál M12 bude mít hodnotu jedna, která je invertována druhým invertorem 55 pro získání nuly na druhém vstupu součtového obvodu 54.

Implicitní v logice z obr. 6 je prioritní pravidlo pro případ řádku čtyři, kde každý ze signálů na výstupech M01 a M12 mají binární hodnotu jedna. Případ tohoto řádku čtyři může být získán sledem instrukčních kategorií BAA. To může být implementováno sledem příznaků 101, jak je uvedeno v obr. 6, nebo alternativně sledem příznaků 110. V daném provedení se postupuje podle pravidla 5 a je zvolen sled 101 znázorněný na obr. 6. Jinými slovy je párování BA dávana přednost před párováním AA.

Kombinace 1,1 pro signál na výstupu M01 a M12 může být také dosažena sledem operačních kódů AAA. V tomto případě se opět zvolí sled 101 příznaků z obr. 6. To je lepší, protože zajišťuje hodnotu jedna pro příznak 2 a tedy patencielně umožňuje slučovat Instrukci 2 a Instrukci 3, jestliže Instrukce 2 je kompatibilní s Instrukcí 3.

Na obr. 7 je znázorněn detailní příklad, jak může být konstruován počítačový systém pro používání slučovacích příznaků podle vynálezu pro zajištění paralelního zpracovávání počítačových instrukcí ve strojovém kódu. Slučovací jednotka 20 instrukcí, použitá na obr. 7, se předpokládá být typu popsaného na obr. 3 a jako taková přidává ke každé instrukci jednobitové příznakové pole. Tato příznaková pole jsou používána pro identifikování, které dvojice instrukcí mohou

být zpracovávány paralelně. Tyto instrukce opatřené příznaky je přiváděny a ukládány do druhého paměťového obvodu 12 s rychlým přístupem. Vyvolávací a vydávací řídicí jednotka 60 vyvolává instrukce opatřené příznaky z rychlé vyrovnávací paměti druhého paměťového období 12 podle potřeby a uspořádává je pro jejich zpracovávání odpovídajícími ze skupiny funkčních jednotek 61, 62, 63 a 64 pro zpracovávání instrukcí. Vyvolávací a vydávací jednotka 60 vyšetřuje příznaková pole a pole operačních kódů vyvolávaných instrukcí. Jestliže příznaková pole indikují, že dvě po sobě následující instrukce mohou být zpracovávány paralelně, potom vyvolávací a vydávací jednotka 60 je přiděluje do odpovídajících z funkčních jednotek 61 až 64, jak je určováno jejich operačními kódy a jsou zpracovávány paralelně zvolenými funkčními jednotkami. Jestliže příznaková pole indikují, že se má obzvláštní instrukce zpracovávat jednotlivým, neparalelním způsobem, potom vyvolávací a vydávací jednotka řídicí 60 ji přiděluje k určité funkční jednotce, jak je určováno jejím operačním kódem a je zpracovávána nebo vykonávána samostatně.

Jednotky 60 až 64 představují konkrétnější provedení spodní části schématu znázorněného na obr. 1. Jednotka 60 odpovídá vyvolávacímu a vydávacímu obvodu 16 a funkční jednotky 61, 62, 63, 64 jsou konkrétními příklady provedení jednotlivých funkčních jednotek 13, 14, 15 z obr. 1.

První funkční jednotka je zpracovávací jednotka 61 instrukcí větvení pro zpracovávání instrukcí typu větvení. Druhá funkční jednotka je třívstupová aritmetická a logická jednotka 62 pro generování adres, která je používána pro výpočet adres ukládání pro instrukce, které přenášejí operandy do paměti nebo k paměti. Třetí funkční jednotka je univerzální aritmeticko-logická jednotka 63, která je použita pro vykonávání operací matematického a logického typu. Čtvrtá funkční jednotka v daném příkladě je aritmeticko-logická jednotka 64 pro překonávání datové závislosti (collapsing data dependency) typu popsaného ve výše uvedeném související-

cím americkém patentovém spise č. _____ (patentová přihláška 07/504 910 - IBM Docket EN9-90-014). Tato aritmeticko-logická jednotka 64 pro překonávání závislosti je třívstupová aritmeticko-logická jednotka schopná vykonávat dvě aritmeticko-logické operace v jediném strojním cyklu.

Provedení počítačového systému z obr. 7 také zahrnuje soubor univerzálních registrů 65 pro použití při vykonávání určitých instrukcí ve strojové úrovni. V typickém případě jsou tyto univerzální registry 65 použity pro dočasné ukládání datových operandů a adresových operandů nebo jsou používány jako čítače nebo pro jiné účely zpracovávání dat. V typickém počítačovém systému je použito šestnáct takových univerzálních registrů. V daném provedení se předpokládá, že univerzální registry 65 jsou typu s více branami, přičemž je možno přistupovat současně na dva nebo více registrů.

Počítačový systém z obr. 7 dále obsahuje vysokorychlostní datový mechanismus s vyrovnávací pamětí 66 s rychlým přístupem pro ukládání datových operandů, získávaných z prvního paměťového obvodu 10 vyšší úrovně. Data ve vyrovnávací paměti 66 s rychlým přístupem mohou být také přenášena zpět do paměťové jednotky paměťového obvodu 10 vyšší úrovně. Datová vyrovnávací paměť 66 s rychlým přístupem může být známého typu a její operace vzhledem k paměti paměťového obvodu 10 vyšší úrovně může být vedena známým způsobem.

Obr. 8 ukazuje příklad instrukčního sledu, slučovaného nebo opatřeného příznaky, který může být zpracováván počítačovým systémem z obr. 7. Příklad z obr. 8 je složen z následujících instrukcí v následujícím sledu: zaveď, přičti, porovnej, proveď podmíněné větvení a ulož. Tyto instrukce jsou označeny jako instrukce I1 až I5 v uvedeném pořadí. Příznakové bity pro tyto odpovídající instrukce jsou 1, 1, 0, 1 a 0. Vzhledem k organizaci stroje znázorněného na obr. 7, se zpracovává instrukce "zaveď" jednotlivým způsobem samostatně. Instrukce "přičti" a "porovnej" jsou považovány jako sloučená instrukce a jsou prováděny vzájemně spolu paralelně. In-

strukce "proved" podmíněné větvení" a "ulož" jsou také považovány za sloučenou instrukci a jsou také zpracovávány vzájemně spolu paralelně.

Tabulka z obr. 9 poskytuje delší informaci o každé z těchto instrukcí z obr. 8. Sloupec R/M na obr. 9 udává obsah prvního pole v každé instrukci, který se v typickém případě používá pro identifikování obzvláštního z univerzálních registrů 65, který obsahuje první operand. Výjimka v tomto případě je instrukce "proved podmíněné větvení", kde pole R/M obsahuje masku podmínkového kódu. Sloupec R/X na obr. 9 udává obsah druhého pole každé instrukce, přičemž toto pole je v typickém případě použito pro označování druhého z univerzálních registrů 65. Takový registr může obsahovat druhý operand nebo může obsahovat hodnotu adresového indexu (X). Sloupec B na obr. 9 udává obsah třetího možného pole v každé instrukci, které může identifikovat obzvláštní registr z univerzálních registrů 65, který obsahuje základní hodnotu adresy. Nula ve sloupci B udává nepřítomnost pole B nebo nepřítomnost odpovídající adresové složky v poli B. Podle D na str. 9 udává obsah dalšího pole v každé instrukci, které, pokud je použito pro účely generování adres, obsahuje hodnotu posunu adresy. Nula ve sloupci D může také udávat nepřítomnost odpovídajícího pole v obzvláštní uvažované instrukci nebo alternativně nulovou hodnotu posunu adresy.

Uvažujeme-li nyní zpracovávání instrukce "zaveď" z obr. 8, určuje vyvolávací a vydávací řídicí jednotka 60 z příznakových bitů pro tuto instrukci "zaveď" a následující instrukci "přičti", že instrukci "zaveď" je třeba zpracovávat jednotlivým způsobem samotnou. Činnost, která má být vykonána touto instrukcí "zaveď", je vyvolat operand z paměti, v tomto případě datové rychlé vyrovnávací paměti 66 a uložit takový operand do univerzálního registru R2. Paměťová adresa, z níž má být tento operand vyvolán, je určena sečtením hodnoty indexu v registru X, základní hodnoty v registru

B a hodnoty posunu v D. Vyvolávací a vydávací řídicí jednotka 60 přiděluje tuto operaci generování adresy aritmeticko-logické jednotce 62 pro generování adres. V tomto případě aritmeticko-logická jednotka 62 sečte hodnotu indexu adresy v registru X (nulovou hodnotu v daném příkladě), základní adresovou hodnotu obsaženou v univerzálním registru R7 a hodnotu posunu adresy (nulová hodnota v daném příkladě), obsažené v samotné instrukci. Výsledná vypočítaná paměťová adresa, která se objevuje na výstupu aritmeticko-logické jednotky 62 je vedena na adresový vstup datové rychlé vyrovnávací paměti 66 pro přístup k požadovanému operandu. Operand, k němuž je zajišťován přístup, je zaváděn do univerzálního registru R2 v souboru registrů 65.

Pokud jde o zpracovávání instrukcí "přečti a porovnej", jsou tyto instrukce vyvolávány vyvolávací a vydávací řídicí jednotkou 60. Vyvolávací a vydávací řídicí jednotka 60 vyšetřuje slučovací příznaky pro tyto dvě instrukce a zjišťuje, že mohou být vykonávány paralelně. Jak je patrné z obr. 9, instrukce "porovnej" má zjevnou datovou závislost na instrukci "přečti", protože instrukce "přečti" musí být hotova dříve, než může být R3 porovnáváno. Tato závislost však může být zpracována aritmeticko-logickou jednotkou 64 pro překonávání datové závislosti (collapsing data dependency). V důsledku toho mohou být tyto obě instrukce zpracovávány paralelně v uspořádání z obr. 7. Konkrétně přiděluje řídicí vyvolávací a vydávací jednotka 60 zpracování instrukce "přečti" do aritmeticko-logické jednotky 63 a přiděluje zpracování instrukce "porovnej" do aritmeticko-logické jednotky 64 pro překonávání závislosti. Aritmeticko-logická jednotka 63 přičítá obsah univerzálního registru R2 k obsahu univerzálního registru R3 a ukládá výsledek sčítání zpět do univerzálního registru. Současně aritmeticko-logická jednotka 64 pro překonávání závislosti vykonává následující matematickou operaci:

$$R3 + R2 - R4$$

Podmínkový kód pro výsledek této operace je vysílán do registru podmínkového kódu umístěného ve zpracovávací jednotce 61 instrukcí větvení. Datová závislost je překonávána, protože aritmeticko-logická jednotka 64 totiž skutečně vypočítává součet $R3 + R2$ a potom srovnává tento součet s $R4$ pro určení podmínkového kódu. Tímto způsobem aritmeticko-logická jednotka 64 nemusí čekat na výsledky z aritmeticko-logické jednotky 63, která vykonává instrukci sčítání. V tomto konkrétním případě numerické výsledky vypočítané aritmeticko-logickou jednotkou 64 a objevující se na výstupu aritmeticko-logické jednotky 64 nejsou dodávány zpět do univerzálního registru 65. V tomto případě aritmeticko-logická jednotka 64 pouze nastavuje podmínkový kód.

Pokud jde o zpracovávání instrukce provedení podmíněného větvení a ukládání na obr. 8, jsou tyto instrukce vyvolávány z rychlé vyrovnávací paměti 12 sloučených instrukcí řídicí vyvolávací a vydávací řídicí jednotkou 60. Vyvolávací a vydávací řídicí jednotka 60 určuje z příznakových bitů pro tyto instrukce, že nemusí být zpracovávány navzájem spolu paralelně. Dále z operačních kódů obou instrukcí, určuje že instrukce podmíněného větvení by měla být zpracovávána zpracovávací jednotkou 61 instrukcí větvení a instrukce ukládání by měla být zpracovávána aritmeticko-logickou jednotkou 62 pro generování adres. V souladu s tímto určením pole M masky a pole D posunu instrukce podmíněného větvení jsou vedeny do jednotky větvení 61. Podobně se hodnota indexu adresy v registru X a základní hodnota adresy v registru B pro tuto instrukci podmíněného větvení získají z univerzálních registrů 65 a jsou dodávány do jednotky 61 větvení. V daném příkladě je hodnota registru X nula a základní hodnota se získá z univerzálního registru R7. Hodnota D posunu má hexadecimální hodnotu dvacet, zatímco pole M masky má hodnotu polohy masky osm.

Jednotka 61 větvení začíná počítat potenciální adresu větvení ($0 + R7 + 20$) a současně srovnává podmínkový kód

získaný z předchozí instrukce "srovnej" s maskou M podmínkového kódu. Je-li hodnota podmínkového kódu stejná, jako je hodnota kódu masky, je potřebná podmínka větvení splněna a adresa větvení vypočítaná jednotkou 61 větvení je na to uložena do počítadla instrukcí v řídicí jednotce 60. Toto počítadlo instrukcí řídí vyvolávání instrukcí z rychlé vyrovnávací paměti 12 slučovaných instrukcí. Jestliže naproti tomu podmínka není splněna, tj. podmínkový kód nastavený předchozí instrukcí nemá hodnotu osm, potom nedochází k větvení a žádná adresa větvení není dodávána do počítače instrukcí v řídicí jednotce 60.

Současně s tím, kdy je jednotka 61 větvení zaměstnána prováděním svých zpracovávacích činností pro instrukci větvení, aritmeticko-logická jednotka 62 pro generování adres je zaměstnána výpočtem adresy $(0 + R7 + 0)$ pro instrukci "ulož". Adresa vypočítaná aritmeticko-logickou jednotkou 62 je dodávána do datové rychlé vyrovnávací paměti 66. Jestliže není provedeno žádné větvení jednotkou 61 větvení, potom instrukce "ulož" provádí uložení operandu v univerzálním registru R3 do datové rychlé vyrovnávací paměti 66 na adrese vypočítané aritmeticko-logickou jednotkou 62. Jestli naproti tomu je podmínka větvení splněna a k větvení dochází, potom obsah univerzálního registru R3 není uložen do datové rychlé vyrovnávací paměti 66.

Výše uvedený sled instrukcí z obr. 8 je určen pouze jako příklad. Počítačový systém podle obr. 7 je rovněž schopný zpracovávat různé a jakékoli jiné sledy instrukcí. Příklad z obr. 8 však jasně ukazuje užitečnost příznaků slučovaných instrukcí při určování, které instrukce mohou být zpracovány paralelně navzájem splu.

I když to, co bylo popsáno, je považováno za výhodná provedení vynálezu, bude zřejmé odborníkům v oboru, že v rámci vynálezu je možné provést různé změny a obměny, aniž by se opustila jeho myšlenka. Vynález proto všechny takové změny a modifikace kryje, které jsou v jeho duchu a spadájí do jeho myšlenky.

Příl.	PROVÁDĚNÍ A OBJEVY	ÚŘAD PRO VYŘÍZENÍ PATENTŮ	12. XII 91	058241	PV 933-91A

1. Zapojení pro paralelní zpracování dvou nebo více instrukcí v číslicovém počítači vyznačené tím, že obsahuje první paměťový obvod (10) opatřený univerzálním výstupem (10a) slučovací obvod instrukcí (11) opatřený univerzálním vstupem (11a) připojeným k univerzálnímu výstupu (10a) prvního paměťového obvodu (10) a dále opatřený druhým univerzálním výstupem (11b), druhý paměťový obvod (12) opatřený univerzálním vstupem (12a) připojeným ke druhému univerzálnímu výstupu (11b) slučovacího obvodu (11) instrukcí a dále opatřený druhým univerzálním výstupem (12b), vyvolávací a vydávací obvod (16) opatřený univerzálním vstupem (16a) připojeným k univerzálnímu výstupu (12b) druhého paměťového obvodu (12) a univerzálním výstupem (16b), a dále dvě nebo více funkčních jednotek (13, 14, 15) pro zpracovávání instrukcí, opatřené každá univerzálním vstupem (13a, 14a, 15a) připojeným ke druhému univerzálnímu výstupu (16b) vyvolávacího a vydávacího obvodu (16).

2. Zapojení podle nároku 1 vyznačené tím, že slučovací obvod (11) instrukcí obsahuje vstupní registr (21) instrukcí s nejméně třemi paměťovými poli (INST0, INST1, INST2), opatřený univerzálním vstupem (21a) připojeným k univerzálnímu výstupu (10a) prvního paměťového obvodu (10), a dále opatřený nejméně jedním delším výstupem (21b), dále nejméně jeden analyzní obvod (22) slučovatelnosti opatřený vstupem (22a) připojeným k výstupu (21b) vstupního registru (21) instrukcí a dále opatřený nejméně dvěma výstupy (M01, M12), dále obvod (26) generování příznaků, mající nejméně dva vstupy připojené ke dvěma výstupům (M01, M12) analyzního obvodu (22) slučovatelnosti, a dále nejméně třemi výstupy (TAG0, TAG1, TAG2), a dále výstupní registr (27) instrukcí s nejméně třemi paměťovými poli (T0, INST0, T1, INST1; T2, INST2), opatřený nejméně třemi vstupy připojenými ke třem výstupům (TAG0, TAG1, TAG2) obvodu (26) generování příznaků a dále opatřený univerzálním výstupem (27a) připojeným k univerzálnímu vstu-

pu (12a) druhého paměťového obvodu (12).

3. Zapojení podle nároku 2 vyznačené tím, že analyzní obvod (22) slučování obsahuje první zjišťovací obvod (30) slučitelnosti instrukcí, mající dva vstupy (OPC, OP1) připojené k univerzálnímu výstupu (21b) vstupního registru (21) instrukcí a dále mající výstup (30a), dále druhý zjišťovací obvod (31) slučitelnosti instrukcí, opatřený dvěma vstupy (OP1, OP2) připojenými k univerzálnímu výstupu (21b) vstupního registru (21) instrukcí, a výstupem (31a), dále první zjišťovací obvod (32) závislosti registrů a druhý zjišťovací obvod (33) závislosti registrů, které jsou opatřeny každý odpovídajícími vstupy (32a, 33a) připojenými k univerzálnímu výstupu (21b) vstupního registru (21) instrukcí, a dále jsou opatřeny odpovídajícím výstupem (32b, 33b), a dále první součinný obvod (34) se dvěma vstupy (34a, 34b) připojenými k výstupům (30a, 32b) prvního zjišťovacího obvodu (30) slučitelnosti instrukcí a prvního zjišťovacího obvodu (32) závislosti registrů a dále s výstupem připojeným ke vstupu (obvodu (26) generování příznaků, a dále druhý součinný obvod (35) mající dva vstupy (35a, 35b) připojené k výstupům (31a, 33b) druhého zjišťovacího obvodu (31) slučitelnosti a druhého zjišťovacího obvodu (33) závislosti registrů a opatřený dále výstupem připojeným ke vstupu obvodu (26) generování příznaků.

4. Zapojení podle nároku 3 vyznačené tím, že první a druhý zjišťovací obvod (30, 31) slučitelnosti instrukcí společně obsahují první dekodér (40), opatřený vstupem (OPC) připojeným k univerzálnímu výstupu (21b) vstupního registru (21) instrukcí, a dále opatřený dvěma výstupy (40a, 40b), dále druhý dekodér (41) opatřený vstupem (OP1) připojeným k univerzálnímu výstupu (21b) vstupního registru (21) instrukcí a opatřený dále dvěma vstupy (41a, 41b), dále třetí dekodér (45) opatřený vstupem (OP2) připojeným k univerzálnímu výstupu (21b) vstupního registru instrukcí (21) a dále opatřený dvěma výstupy (45a, 45b), dále první součinný obvod (42) obvod se dvěma vstupy (42a, 42b) připojenými k výstupu (40a)

prvního dekodéru (40) a výstupu (41a) druhého dekodéru (41) a dále s výstupem (42c), dále druhý součinnový obvod (43) se dvěma vstupy (43a, 43b) připojenými k výstupu (40b) prvního dekodéru (40) a výstupu (41a) druhého dekodéru (41) a dále s výstupem (43c), dále třetí součinnový obvod (46) se dvěma vstupy (46a, 46b) připojenými k výstupu (41a) druhého dekodéru (41) a k výstupu (45a) třetího dekodéru (45) a dále s výstupem (46c), dále čtvrtý součinnový obvod (47) se dvěma vstupy (47a, 47b) připojenými k výstupu (41b) druhého dekodéru (41) a k výstupu (45a) třetího dekodéru (45) a dále s výstupem (47c), dále první součtový obvod (44), mající dva vstupy (44a, 44b) připojené k výstupům (42c, 43c) prvního a druhého součinnového obvodu (42, 43) a mající dále výstup připojený ke vstupu (M01) obvodu (26) generování příznaků a dále obsahuje druhý součtový obvod (48) mající dva vstupy (48a, 48b) spojené s výstupy (46c, 47c) třetího a čtvrtého součinnového obvodu (46, 47) a dále opatřený výstupem připojeným ke vstupu (M12) obvodu (26) generování příznaků.

5. Zapojení podle kteréhokoli z nároků 2 až 4 vyznačené tím, že obvod (16) generování příznaků obsahuje první invertor (52) opatřený vstupem spojeným s výstupem (M01) analyzního obvodu slučování (22) a výstupem spojeným se vstupem (TAG1) výstupního registru (27) instrukcí, dále obsahuje druhý invertor (55) opatřený vstupem spojeným s výstupem (M12) s výstupem analyzního obvodu (22) slučování a dále opatřený výstupem (55a), a dále obsahuje součtový obvod (54) opatřený dvěma vstupy spojenými s výstupem (M01) analyzního obvodu (22) slučovatelnosti a s výstupem (55a) druhého invertoru (55) a dále opatřený výstupem spojeným se vstupem (TAG2) výstupního registru (27) instrukcí.

6. Zapojení podle kteréhokoli z nároků 1 až 5 vyznačené tím, že funkční jednotky (13, 14, 15) ~~na~~ obsahují zpracovávající jednotku (61) instrukcí větvení, aritmetickou a logickou jednotku (62) pro generování adres, univerzální aritmetickou a logickou jednotku (63) a aritmetickou a logickou jednotku (64) pro překonávání datové závislosti.

8. Zapojení podle kteréhokoli z nároků 1 až 7 vyznačené tím, že druhý paměťový obvod (12) sestává z malckapacitní vysokorychlostní paměti.

JUDr. ~~Petr Pleskavsky~~

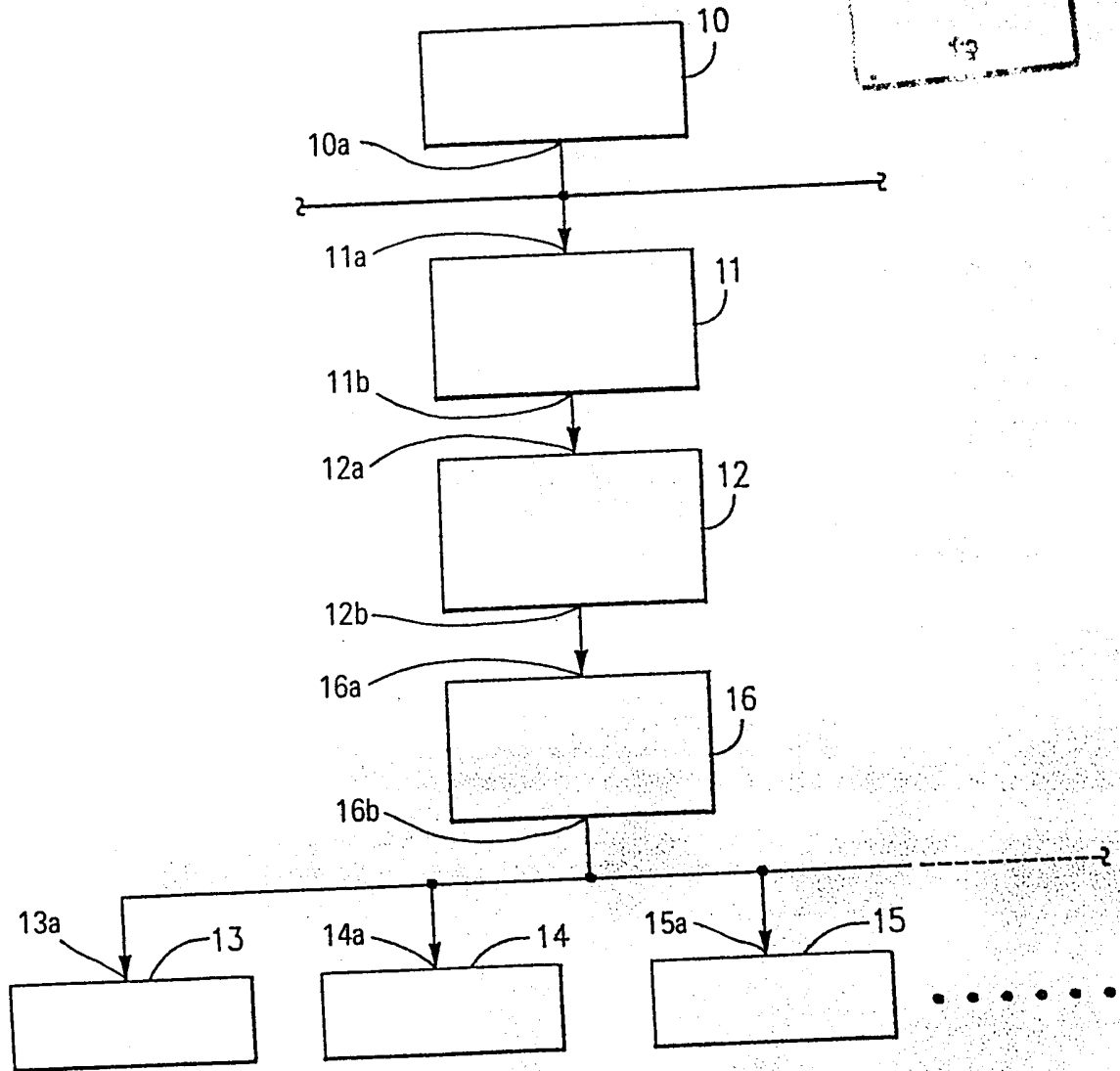
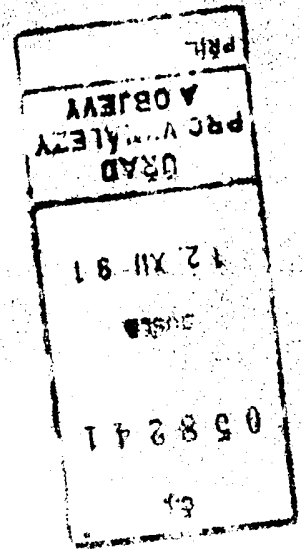


FIG.1

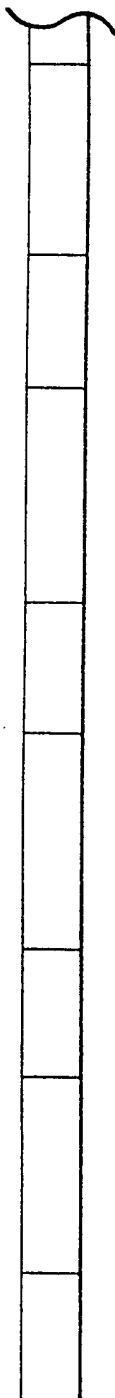
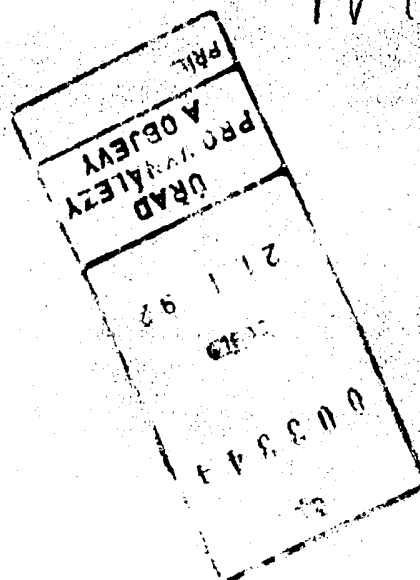


FIG.2

M01	M12	0	1	2
0	0	1	1	1
1	0	1	0	1
0	1	1	1	0
1	1	1	0	1

FIG.6

2/7



PV 933-91A

JUDr. Petr HALENSKY
advokát

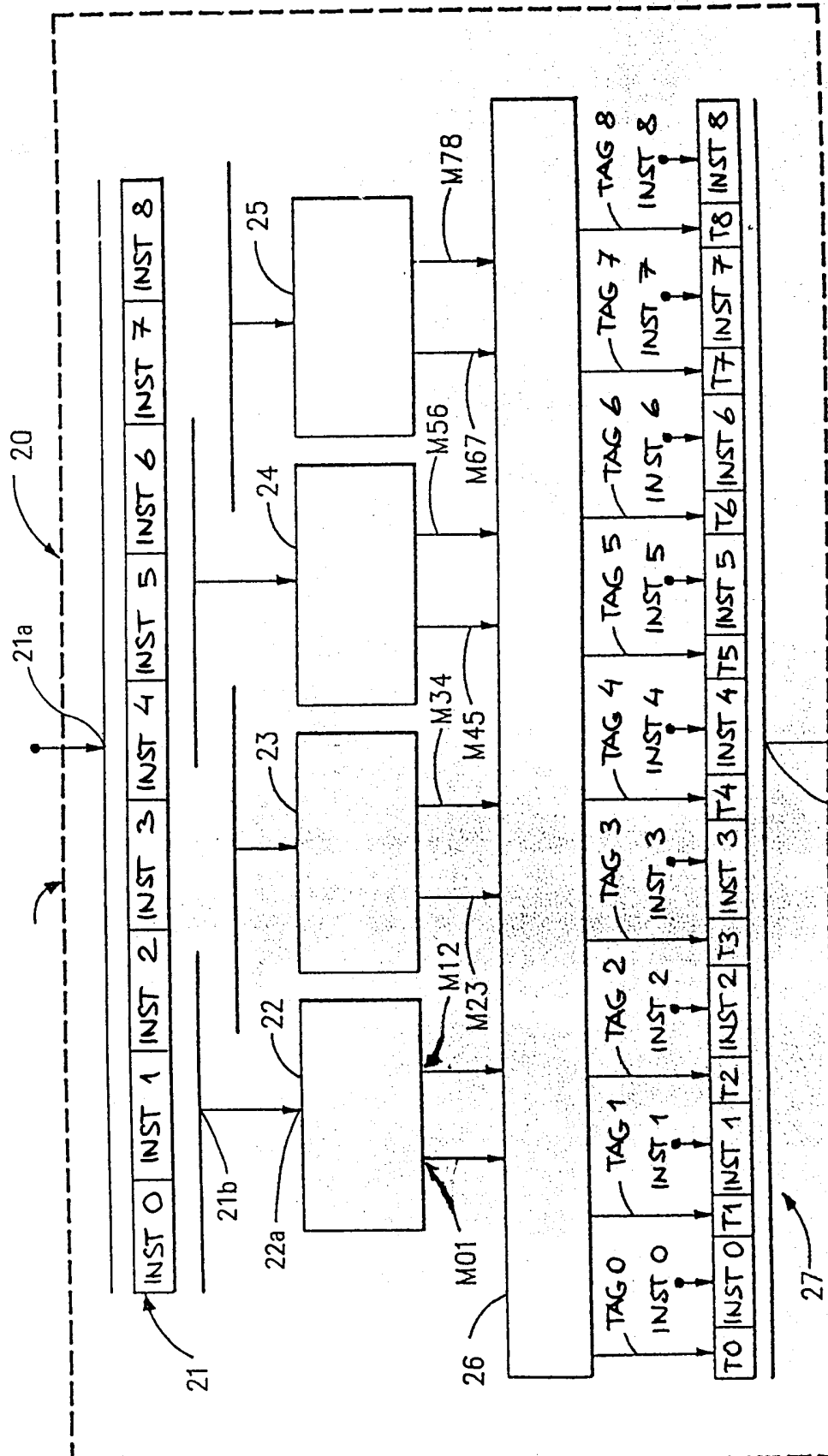
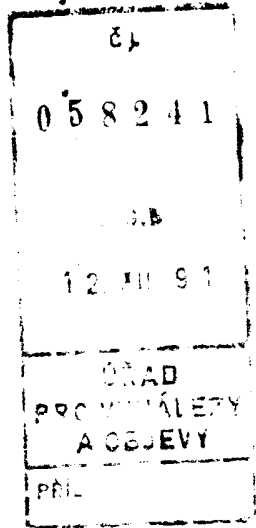


FIG. 3

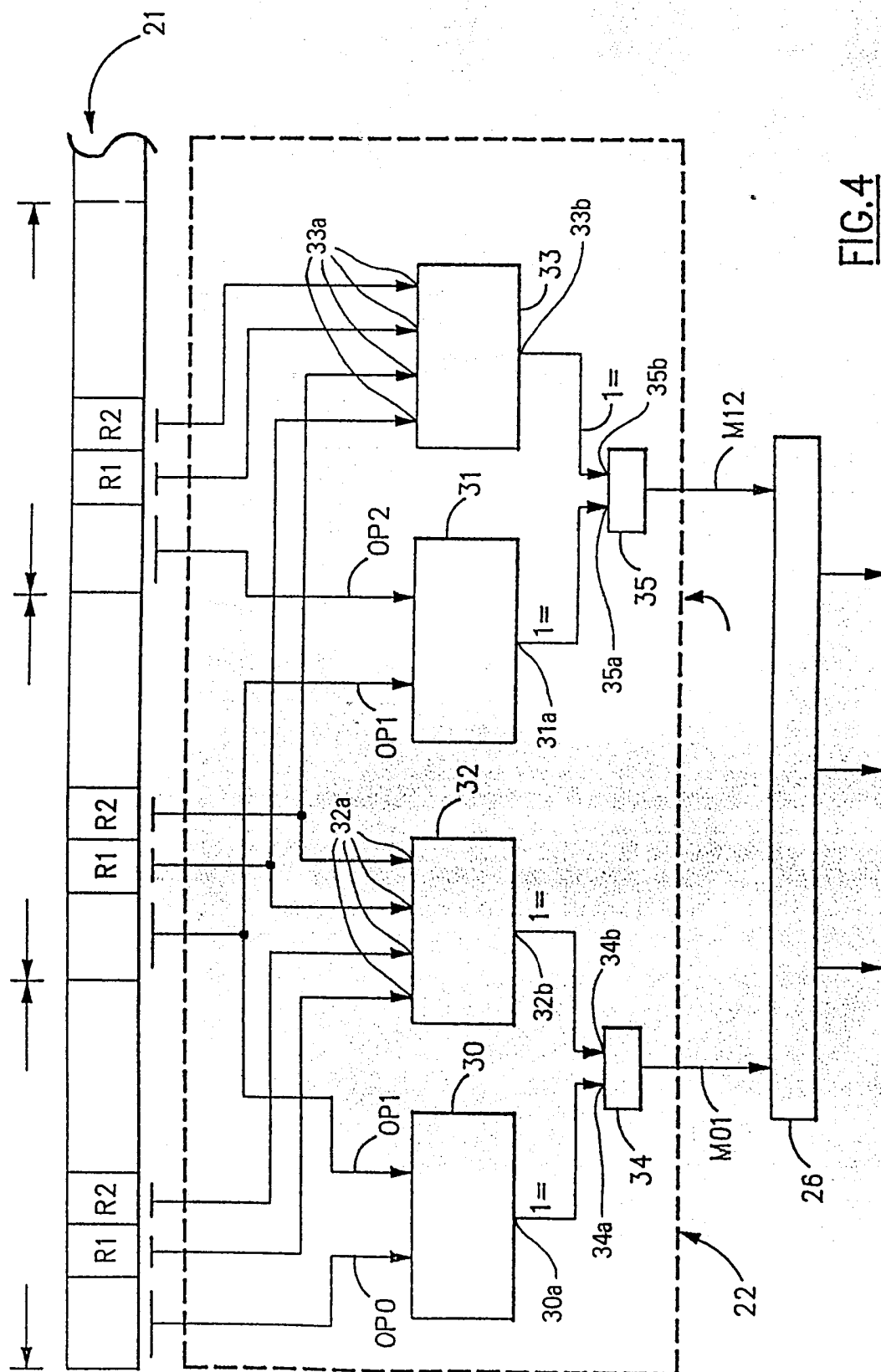


FIG. 4

U 5 8 2 4 1
12. XII 91
ÚRAD
PRO VÝSLEDKY
A OBJEVY
PRIL

058241

12. XII 91

ÚŘAD
PRO VYHLEDÁVÁNÍ
A OBJEVY

PŘÍL.

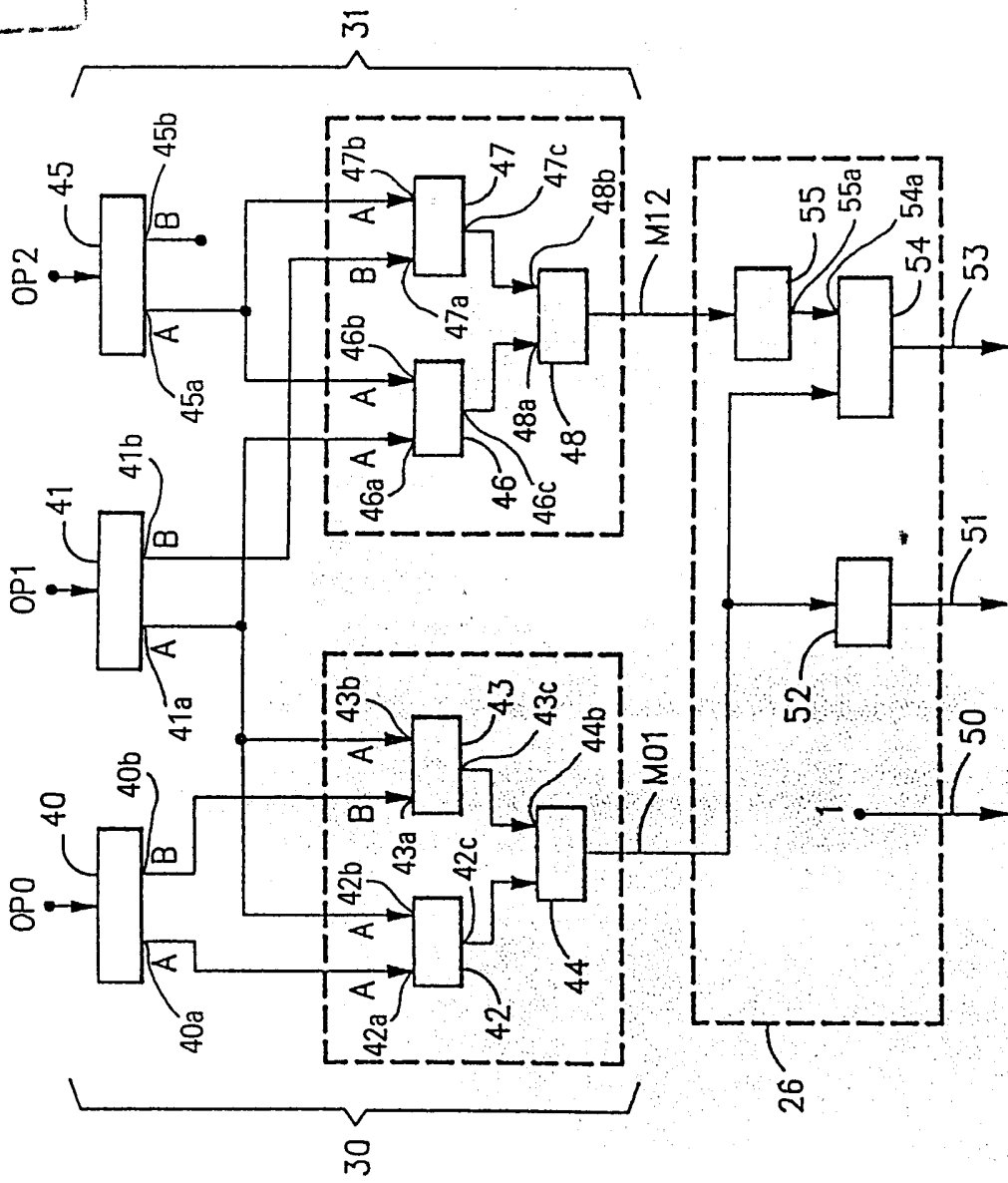
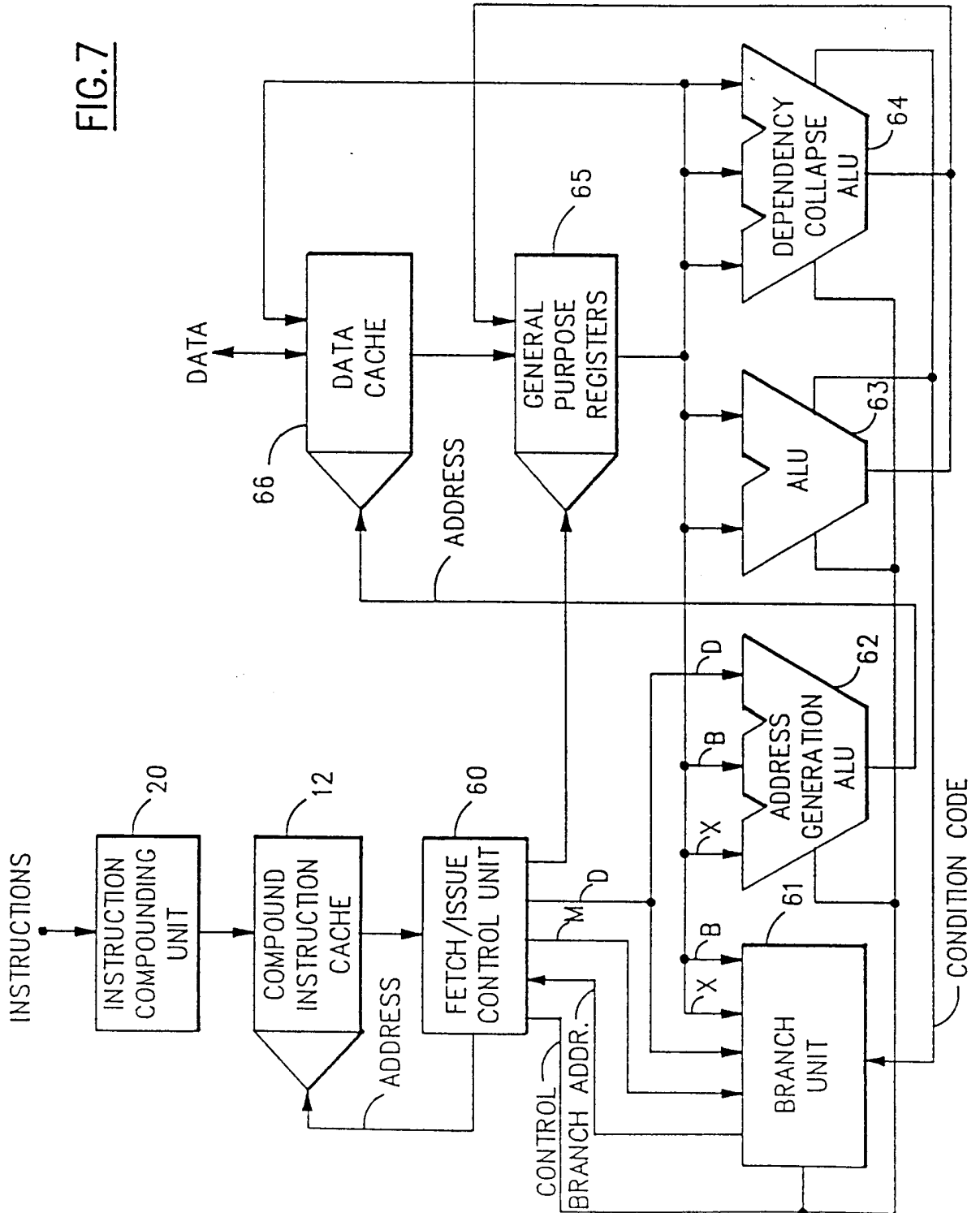


FIG.5

JUDr. Petr KAHENSKÝ
advokát

6 / 7

00000000
101192
URAD PRO JUD. A OBJEVY
Přil.



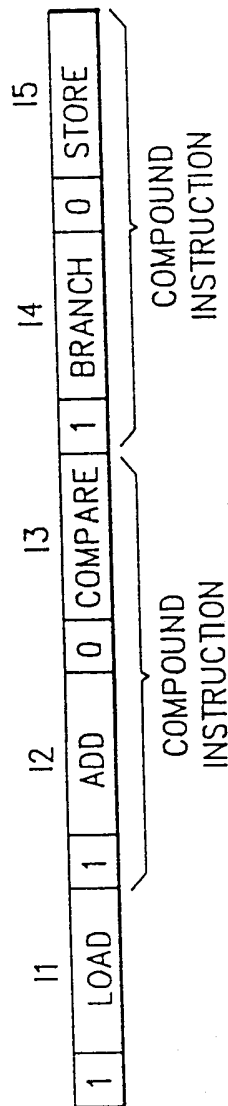


FIG.8

NO.	OP CODE	R/M	R/X	B	D	ACTION
I1	LOAD	R2	0	R7	0	STORAGE $\rightarrow$ R2 (STORAGE ADDRESS = $0 + R7 + 0$)
I2	ADD	R3	R2	0	0	$R3 + R2 \rightarrow R3$
I3	COMPARE	R3	R4	0	0	R3 (c) R4 $\rightarrow$ CONDITION CODE
I4	BRANCH ON CONDITION	8	0	R7	20	ADDRESS ($0 + R7 + 20$) TO INSTR. COUNTER IF CONDITION MET (CC=8)
I5	STORE	R3	0	R7	0	R3 $\rightarrow$ STORAGE (ADDR. = $0 + R7 + 0$) (IF CONDITION NOT MET)

FIG.9

