SLOVENSKÁ REPUBLIKA

(19) **SK**

ÚRAD
PRIEMYSELNÉHO
VLASTNÍCTVA
SLOVENSKEJ REPUBLIKY

ZVEREJNENÁ PRIHLÁŠKA VYNÁLEZU

(21) Číslo dokumentu:

934-91

(22) Dátum podania: 04.04.91

(31) Číslo prioritnej prihlášky: 543 464

(32) Dátum priority: 26.06.90

(33) Krajina priority: US

(43) Dátum zverejnenia: ~~09.08.1995~~ 13. sept. 1995

(86) Číslo PCT:

(13) Druh dokumentu: **A3**

(51) Int. Cl.[6] :

G 06 F 13/00

(71) Prihlasovateľ: International Business Machines Corp., Armonk, NY, US;

(72) Pôvodca vynálezu: Blaner Bartholomew , Newark Valley, NY, US;
Vassiliadis Stamatis, Vestal, NY, US;
Eichemeyer Richard James, Endicott, NY, US;

(54) Názov prihlášky vynálezu: **Číslicový počítačový systém na spracovanie najmenej dvoch paralelných inštrukcií**

(57) Anotácia:
Obsahuje hlavnú pamäť (10) pripojenú cez adaptér (8) k mechanizmu skladajúcemu inštrukcie (11). Mechanizmus (11) skladajúci inštrukcie, spojený s adresovou prikazovacou zbernicou (9a) a s textovou zbernicou (9b) je cez rýchlu vyrovnávaciu pamäť (112) pripojený k jednotke (116) vyvolania a vydania, ako i k inštrukčnej zbernici (17), ku ktorej sú pripojené dve alebo viac inštrukčných základných jednotiek (13, 14, 15...).

Číslicový počítačový systém

Oblast techniky

Vynález se týká číslicového počítačového systemu pro zpracování nejméně dvou paralelních instrukcí.

Dosavadní stav techniky

Tradiční počítače,které přijmou sled instrukcí a zpracují jednu instrukci jsou známy.Instrukce provedené těmito počítači pracují s jednoznačnými objekty nazývanými skaláry.

Operační rychlost tradičních skalárních počítačů dosáhla svého vyvrcholení vlivem pokroku v technologii obvodů,v počítačových mechanismech a architektuře počítače. Nicméně,pro každou novou generaci příslušných strojů musí být objeven pro tradiční skalární stroje nový akcelerační mechanismus.

Mechanismus pro zvýšení počítačové rychlosti sekvenčních procesorů byl nedávno nalezen v redukované architektuře instrukčních sad,která využívá omezenou sadu velmi jednoduchých instrukcí. Jiný urychlovací mechanismus spočívá v architektuře komplexní instrukční sady,která je založena na minimálním souboru instrukcí s komplexními vícenásobnými operandami.

Použití každého z uvedených přístupů pro skalární stávající počítač by vyžadovalo základní změnu instrukční sady a architektury stroje. Taková dalekosáhlá transformace je nákladná,způsobuje prostoje a počáteční snížení spolehlivosti a strojové dostupnosti.

Ve snaze použít pro skalární stroje některé z úspěchů dosažených redukcí instrukční sady , byl uskutečněn vývoj tak zvaných superskalárních počítačů. Tyto stroje jsou v podstatě skalárními stroji, jejichž výkonnost je zvýšena jejich přizpůsobením k provedení více než jedné instrukce najednou z instrukčního toku včetně posloupnosti jednotlivých skalárních instrukcí. Tyto stroje typicky rozhodují v době provádění instrukce zda mají být zpracovány paralelně dvě nebo více instrukcí v posloupnosti skalárních instrukcí. Rozhodnutí je založeno na operačních kódech instrukcí a závislostech dat , které mohou mezi instrukcemi existovat. Operační kód označuje technické prostředky požadované pro jednu instrukci. Obvykle není možno současně provádět dvě nebo více instrukcí,které používají stejných technických prostředků nebo stejných operand. Tyto závislosti na technickém vybavení počítače a datech zabraňují paralelnímu zpracování některých instrukčních kombinací. V uvedených případech jsou tyto instrukce zpracovány seriově. Tím je ovšem snížena výkonnost superskalárního stroje.

Superskalární počítače mají četné nevýhody, které je třeba minimalizovat. Konkrétní část doby při zpracování je spotřebována k rozhodnutí, které instrukce lze provést paralelně. Tuto dobu nelze běžně zamaskovat překrytím jinými strojovými operacemi.Uvedený nedostatek se stává ještě výraznější,je-li zvyšována složitost architektury instrukční sady. Rozhodnutí o paralelním zpracování se musí tedy opakovat pokaždé,když jsou prováděny stejné instrukce.

Při prodlužování užitečné doby životnosti stávajících skalárních počítačů je podstatný každý prostředek urychlující zpracování. Nicméně, urychlování pomocí architektury redukované instrukční sady nebo superskalární techniky je potencionálně příliš nákladné či příliš nevýhodné než, aby bylo použito pro stávající skalární počítače. Rychlost zpracování takovým počítačem měla by se raději zvyšovat paralelním nebo souběžným zpracováním ve stávající instrukční sadě aniž by byla požadována změna instrukční sady, změna architektury počítače nebo prodloužení doby potřebné k zpracování instrukce.

Shora uvedené nevýhody jsou odstraněny číslicovým počítačovým systemem podle vynálezu.

Podstata vynálezu

Předmětem vynálezu je číslicový počítačový system pro zpracování nejméně dvou paralelních instrukcí , vyznačující se tím, že výstup-vstup rozhraní je přes adaptér připojen k instrukce skládacímu mechanismu , jehož výstup je spojen s textovou sběrnicí, zatím co jeho vstup-výstup je spojen s adresovou přikazovací sběrnicí, která je připojena jednak na vstup-výstup adaptéru jednak k hlavní paměti, která je vstupem-výstupem spojena s textovou sběrnicí připojenou vstupem-výstupem k rychlé vyrovnávací paměti pro složené instrukce s výstupem připojeným přes jednotku pro vyvolání a vydání instrukce k funkční sběrnici, na kterou je připojen větší počet funkčních jednotek.

Další podstatou je, že vstup-výstup hlavní paměti je spojen s adaptérem přes instrukce skládací mechanismus a dvě paralelní vyrovnávací paměti složených stránek, jejichž společný výstup je připojen k hlavní paměti.

Jinou podstatou je, že instrukce skládací mechanismus obsahuje instrukce skládací jednotku s větším počtem instrukčních registrů, které jsou spojeny s větším počtem skládacích analyzátorů připojených k příznakovému generátoru a příznakovému instrukčnímu registru.

Podstata je také v tom,že první skládací analyzátor obsahuje první instrukční slučitelnou logiku připojenou k prvnímu vstupu prvního součinového obvodu, jehož druhý vstup je připojen k prvnímu registru závislé logiky, zatím co druhá instrukční slučitelná logika a druhý registr závislé logiky jsou připojeny k prvnímu a druhému vstupu druhého součinového obvodu.

Další podstatou je,že první slučitelná logika obsahuje první dekodér, jehož první výstup je spojen s prvním vstupem prvního součinového obvodu,jehož druhý vstup je připojen k druhému dekodéru, zatím co druhý výstup prvního dekodéru je spojen s prvním vstupem druhého součinového obvodu, jehož druhý vstup je připojen k druhému dekodéru, přičemž výstupy obou součinových obvodů jsou připojeny k obvodu logického součtu, jehož výstup je připojen k obvodu invertoru v příznakovém generátoru a současně k prvnímu vstupu prvního obvodu logického součtu, jehož druhý vstup je přes druhý obvod logického součtu připojen k obvodu logického součtu v druhé instrukční slučitelné logice, k

jehož prvnímu vstupu je připojen výstup pátého součinového obvodu, jehož druhý vstup je připojen k šestému součinovému obvodu, přičemž vstupy obou součinových obvodů jsou připojeny k druhému dekodéru a třetímu dekodéru.

Podstatou posléze je, že instrukce skládací jednotka je přes hlavní paměť a rychlou vyrovnávací paměť připojena k řídící jednotce vyvolání-vydání, která je spojena s první funkční jednotkou, druhou funkční jednotkou, třetí funkční jednotkou a čtvrtou funkční jednotkou,které jsou připojeny k univesálnímu registru a rychlé vyrovnávací paměti.

Za účelem lepšího porozumnění číslicovému počítačovému systemu podle vynálezu včetně jeho výhod a charakteristik je v následujícím popise brán zřetel k výkresům jak následuje.

Přehled obrázků na výkresech

Obr.1 charakteristické zapojení části číslicového počítačové-systemu podle vynálezu

Obr.2 2A-2D znázorňují alternativní realizaci pro ukládání složené příznakové informace do hlavní paměti.

Obr.3 podrobnější struktura toku dat mezi vstupním a výstupním adaptérem a hlavní pamětí počítačového systemu.

Obr.4 časový diagram pro převod instrukce do struktury toku dat.

Obr.5a znázorňuje délku instrukčního toku s příznaky složení nebo příznaková pole přidružená k instrukcím;

Obr.5b znázorňuje délku instrukčního toku s instrukčními

Obr.6 podrobné vnitřní charakteristické zapojení části instrukce skládací jednotky,kterou lze použít v číslicovém počítačovém systemu podle vynálezu.

Obr.7 charakteristická vnitřní konstrukce každého ze skládacích analyzátorů.

Obr.8 zapojení logických obvodů,které lze použít pro realizaci skládacího analyzátoru a částí příznakového generátoru pro vytvoření složených příznaků pro první tři instrukce instrukčního toku.

Obr.9 tabulka vysvětlující pracovní postup podle Obr.8

Obr.10 charakteristické zapojení části číslicového počítačového systemu a jeho použití k vysvětlení jak lze složené instrukce paralelně zpracovat vícenásobnými funkčními instrukcemi skládacích jednotek.

Obr.11 příklad specielní posloupnosti instrukcí,které lze zpracovat číslicovým počítačovým systemem podle Obr.10

Obr.12 tabulka pro vysvětlení postupu zpracování instrukční posloupnosti v Obr.11 číslicovým počítačovým systemem podle vynálezu.

Příklad provedení vynálezu

Na obr.1 je charakteristické zapojení části číslicového počítačového systemu podle vynálezu s hierarchicky uspořádanou paměťovou strukturou. Hlavní paměť 10 je připojena k paměťové sběrnici 9 sestávající z adresové přikazovací sběrnice 9a a z textové sběrnice 9b. Adaptér 8 je připojen prvním vstupem-

výstupem k rozhraní 7 , zatím co druhý vstup-výstup je spojen s adresovou přikazovací sběrnicí 9a. Výstup adaptéru 8 je přes instrukce skládací mechanismus 11 spojen s textovou sběrnicí 9b připojenou vstupem-výstupem k rychlé vyrovnávací paměti 112 pro složení instrukce,jejíž výstup je přes jednotku pro vyvolání a vydání instrukce 116 připojen k funkční sběrnici 17. Na funkční sběrnici je připojen větší počet funkčních jednotek 13,14,15... .

Hlavní paměť 10 je vstupem-výstupem spojena s adaptérem 8 přes instrukce skládací mechanismus 11 . Výstup tohoto mechanismu je připojen ke dvěma paralelním vyrovnávacím pamětím 19a , 19b složených stránek, které jsou společným výstupem spojeny s hlavní pamětí 10.

Instrukce skládací mechanismus 11 obsahuje instrukce skládací jednotku 20, jejíž zapojení je v obr. 6. Tato jednotka je opatřena větším počtem instrukčních registrů 21, které jsou spojeny s větším počtem skládacích analyzátorů 22,23,24.... spojených s příznakovým generátorem 26 a příznakovým registrem 27.

Hlavní paměť 10 převádí instrukce a data pomocí paměťové sběrnice 9 na druhý vstup-výstup adaptéru 8,který je spojen svým prvním vstupem-výstupem k rozhraní 7. S tímto rozhraním může být spojena jedna nebo více pomocných pamětí,které nejsou vyobrazeny. Adaptér 8 převádí data přes rozhraní 7 podle ukládací paměťové informace a obdržené paměťové informace z pomocných pamětí a zpět. Adaptér 8 vyměňuje rovněž programovaná data na paměťové sběrnici 9 s hlavní pamětí 10 poskytováním instrukcí a dat a přijímáním instrukcí a dat z hlavní paměti 10

přes paměťovou sběrnici 9. Adaptér 8 ukládá instrukce a data mezi rozhraní 7 a vyrovnávací paměť 9 ,které mohou mít různé rychlosti a formát. Adaptér 8 posléze provádí také testování a zjišťování chyb funkcí.

Hlavní paměť 10 s poměrně velkou kapacitou má paměťový mechanismus se střední rychlostí,který je připojen přes paměťovou sběrnici 9 k nízkokapacitní rychlé vyrovnávací paměti.

Číslicový počítačový systém v obr.1 obsahuje také instrukce skládací mechanismus 11 pro příjem instrukcí z adaptéru 8 a přiřazování těmto instrukcím složenou příznakovou informaci ve tvaru příznakových polí indikujících,která z těchto instrukcí má být paralelně zpracována. Instrukce skládací mechanismus 11 analyzuje přicházející instrukce a určuje,která z nich má být paralelně zpracována. Kromě toho instrukce skládací mechanismus 11 vytváří pro tyto analyzované instrukce složenou příznakovou informaci ve tvaru příznakových polí,která indikují které instrukce lze paralelně zpracovat jednu s druhou a které nelze vzájemně paralelně zpracovat.

V obr.1 jsou instrukce předávány do počítačového systemu z pomocných pamětí přes adaptér 8, instrukce skládací mechanismus 11 a hlavní paměť 10. Hlavní paměť 10 přijímá a ukládá analyzované instrukce a jejich přidružená příznaková pole. Potom postupuje tyto údaje do rychlé vyrovnávací paměti 12 složených instrukcí. Rychlá vyrovnávací paměť 12 má menší kapacitu a vyšší rychlost než hlavní paměť 10 a je to běžně používaný druh

pro zvýšení výkonnosti počítačového systemu snížením četnosti přístupu do hlavní paměti 10 .

Číslicový počítačový system v obr.1 obsahuje rovněž pluralitu funkčních jednotek 13,14,15.... . Tyto jednotky pracují paralelně jedna s druhou souběžným způsobem a každá sama o sobě je způsobilá zpracovat jeden nebo více typů instrukcí strojové úrovně. Příklady funkčních jednotek,které lze použít obsahují univesální aritmetickou a logickou jednotku,typ této jednotky pro sdružená závislá data,základní jednotku pro větvené instrukce,jednotku s posuvem dat,základní jednotku s pohyblivou čárkou atd. V daném případě takový počítačový system může obsahovat dva i více některých typů funkčních jednotek. Specielní konfigurace funkčních jednotek bude záviset zejména na soustavě uvažovaného počítače.

Číslicový počítačový system v obr.1 obsahuje také jednotku 16 pro vyvolání a vydání instrukce připojenou k vyrovnávací rychlé paměti 12 pro přivádění sousedních instrukcí zde uložených k různým dalším funkčním jednotkám 13 - 15, jestliže instrukční příznaková pole indikují,že je lze zpracovat paralelně. Jednotka 16 vyvolá instrukce z rychlé vyrovnávací paměti 12 přezkouší jejich příznaková pole a pole operačních kódů, a na základě těchto zkoušek vyšle instrukce k některým z příslušných funkčních jednotek 13 - 15 . Je-li žádaná instrukce residentní v rychlé vyrovnávací paměti 12 , je do ní vyslána adresa,aby z ní vyvolala uvedenou instrukci. Jestliže žádaná instrukce není residentní v paměti 12 , musí být vyvolána z hlavní paměti 10 a přivedena do rychlé vyrovnávací paměti 12 . Následkem toho

hlavní paměť 10 začne převádět nebo číst z řádky instrukcí, která obsahuje žádanou instrukci, společné s příznakovými poli instrukcí v řádce.

Výpadek vyrovnávací paměti způsobí, že je do hlavní paměti na výpadek poukázáno, aby bylo zjištěno zda je žádaná instrukce v paměti 10 obsažena. Se zřetelem na to jsou instrukce uloženy do hlavní paměti v blocích nazvaných stránky a zařízení ovládající paměť počítačového systemu je schopné rozhodnout podle žádané instrukce zda stránka, na které je obsažena, je v hlavní paměti. Je-li stránka v hlavní paměti, je řádka obsahující instrukci převedena nebo odečtena z hlavní paměti 10 do vyrovnávací paměti 12. Jestliže stránka obsahující žádanou instrukci není v hlavní paměti 10, dojde k výpadku stránky a je třeba chybějící stránku vyvolat z pomocné paměti a přemístit ji do hlavní paměti 10. Když je stránka vyvolána je identifikace chybějící stránky vyslána do adaptéru 8, který ji vyhledá a potom postoupí přes paměťovou sběrnici 9 k uložení do hlavní paměti 10. Podle vynálezu stránky vyvolané k uložení do hlavní paměti 10 přesouvány na vstup instrukce skládací mechanismus 11, který tyto přicházející instrukce postupně analyzuje a pro každou instrukci generuje příslušné příznakové pole. Příznaky a instrukce jsou potom předány do hlavní paměti 10 a v ní uloženy pro případnou potřebu dodatečného přemístění do vyrovnávací paměti 12 pro složené instrukce.

Ačkoliv instrukce skládací mechanismus 11 je zobrazen v Obr.1 jako zapojen mezi adaptér 8 a hlavní paměť 10, je považován za samostatný vývod ze sběrnice 9 nebo připojení

na vstup hlavní paměti 10.

Ukládání složených instrukcí do hlavní paměti 10 může být realizováno větším počtem cest,z nichž některé jsou zobrazeny v Obr.2A - 2D. Příklady v Obr.2A-2D předpokládají 8 bytů širokou textovou sběrnici 9b plus extra řádky pro příznakovou informaci. Všeobecně se předpokládá,že základní paměťový přesun mezi hlavní pamětí 10 a vyrovnávací pamětí 12 složených instrukcí obsahuje 64-bytové řádky vyrovnávací paměti s jedním příznakovým bitem pro každé dva byty instrukčního textu. Jedna řádka vyrovnávací paměti je znázorněna v každém z příkladů na Obr.2A-2D. Obecně počet příznakových bitů je určován maximálním počtem instrukcí ke složení a informací vhodnou pro instrukce skládacího mechanismu 11 .

Nejjednodušší realizace uložení příznaku z hlediska řízení je zobrazena v Obr.2A. Předpokládá se,že složení je omezeno na dvě instrukce,přičemž je požadován nejméně jednobitový příznak pro každé dvě slabiky instrukčního textu. Tak pro řádku uloženou v paměti na Obr.2A je pro každých 64bitů to je každých osm slabik zapotřebí čtyř slabik příznakové informace na složení. Jak zobrazeno na Obr.2A uložení této informace obsahuje zvětšení velikosti slov z 64 na 68 bitů.Další nepovinné příznakové bity by zvětšily rozsah prodloužených slov.

Druhý přístup,mnohem slučitelnější s dosažitelnou paměťovou technologií je zobrazen v Obr. 2B. V Obr.2B je poskytnuta k uložení instrukcí a sdružené příznakové informace ke slo-

žení samostatná textová a příznaková paměť. V Obr.2B příznaková paměť pracuje paralelně s textovou pamětí. Z paměťové struktury obr.2B vyplývá požadavek na zvláštní sadu příznakových řádek tvořících příznakovou sběrnici v paměťové sběrnici 9 pro zajištění paralelního zpracovní textových a příznakových pamětí. Z toho plyne několik výhod proti pojetí prodloužených slov podle Obr.2A. Provozní systém zaprvé používá určité části paměti jen pro datové stránky místo instrukčních stránek ,přičemž pro tyto části není nutně zapotřebí příznaků. Odlišení datových od instrukčních stránek může být rozhodnuto technickým vybavením nebo programovým vybavením a implementovanými příkazy příznakové paměti, které indikují, že určité stránky obsahují pouze data a proto nevyžadují, aby adresy paměťových stránek byly promítnuty do příznakových paměťových adres pro tyto stránky. Druhou výhodou je, že příznakovou paměť lze odstranit ve snaze po snížení nákladů systemu. Tím se rozšíří možný rozsah výkonnosti soustavy počítačů. Jeli třeba více příznaků než jak bylo požadováno pro více než dvoucestné složení je nahrazena příznaková paměť v Obr.2B novou příznakovou pamětí beze změny návrhu hlavní paměti. Každá paměť kromě toho může být opatřena svou vlastní korekcí chyb.

Se zřetelem na Obr.2A-2D je třeba konstatovat, že jednou generované složené příznaky skládací jednotkou provázejí instrukční tok v paměti ať vetkáním do toku, přidáním do jeho sekcí nebo paralelním udržením s ním.

Ostatní přístupy k realizaci ukládání příznaků jsou zobrazeny v Obr.2C a 2D. V Obr. 2C první sekce hlavní

paměti obsahuje příznakové tabulky a druhá ukládání instrukčních textových stránek. V tomto příkladu je požadována podpora pracovního systemu,aby byla zachována příznaková tabulka části paměti a párové paměťové stránky s příznakovými stránkami. V Obr.2D jsou části každé stránky reservovány pro příznaky.Vyžaduje to způsobilost kompilátoru ke konstruování stránek. Například pro řádky vyrovnávací paměti o 64 slabikách by kompilátor použil 60 slabik pro instrukce a 4 slabiky pro příznaky.V Obr.2D jsou příznaky přidány mezi instrukční slabiky do instrukční vyrovnávací paměti,je-li to vyžádáno centrální základní jednotkou.

V implikaci počítačového systemu na obr.1 může instrukce skládací jednotka 11 tvořit část sběrnicového adaptéru 8.V případě,že je převedena jakákoliv stránka jako zůstatek ze vstupu-výstupu systemu, je podrobena skládacímu procesu zahrnutém v jednotce 11 a předána paměťovou sběrnicí 9 do hlavní paměti 10. Odtud je stránková struktura prováděna podle Obr. 2A za předpokladu,že textová sběrnice 9b je 68 bitů široká a hlavní paměť je uspořádána a řízena k ukládání stránek tak jak je znázorněno v Obr.2A. Vyrovnávací paměť 12 pro složení instrukce je samozřejmě uspořádána a řízena tak,aby přijímala řádky obsahující prodloužená slova jak znázorněno v Obr.2A.

V případě chyby na stránce je tato uložena do vyrovnávací paměti stránek v adaptéru 8 a předána k dispozici instrukce skládací jednotce 11 jak dále popsáno. V Obr.3 dvě stránkové paměti 18a a 18b vysílají posloupnosti stránek do instrukce skládací jednotky 11 ,které provede složení přidáním příznakové informace ke složení do stránkových instrukcí.

Stránky zpracované skládací jednotkou 11 jsou podávány do hlavní paměti 10 přes vyrovnávací paměti 19a a 19b složených stránek. Jak znázorněno v Obr.4 skládací jednotka přidá čas k časovému údaji potřebnému k vyvolání časového úseku z pomocné paměti a vsune jej do hlavní paměti 10. Přidaný čas je však relativně malý proti celkovému požadovanému a je asynchronní pro cetrální základní jednotku.

V Obr.4 je každý segment i převeden z pomocného ukládacího zařízení jako je disková jednotka do jedné ze stránkových vyrovnávacích pamětí 18a nebo 18b . Každý z časových segmentů b_i indikuje čas potřebný k převodu textového segmentu i ze stránkové vyrovnávací paměti do hlavní paměti 10. Textový segment i je tak převeden v čase a_i do jedné ze stránkových vyrovnávacích pamětí 18a nebo 18b podle toho,který textový segment i+1 je převeden do ostatních vyrovnávacích pamětí.Bez složení je textový segment i převeden v čase b_i do stránkové vyrovnávací paměti,kterou je běžně uložen do hlavní paměti. Jak ukázáno v Obr.4 je tento čas podstatně kratší než čas požadovaný k vyvolání stránky z jedné z vyrovnávacích pamětí 18a nebo 18b. V praxi podle vynálezu čas potřebný k operaci skládací jednotky 11 ,aby vykonala text segmentu na jedné stránce vyrovnávacích pamětí plus čas spotřebovaný složenou vyrovnávací pamětí 19a nebo 19b je reprezentován složeným časem c_i. V Obr.4 je nyní čas b_i ten,který je potřeba k převodu textového segmentu i ze stránkové vyrovnávací paměti do skládací jednotky 11. Dále je vydán skládací čas c_i ,zatím co textový segment je podroben zpracování skládací jednotkou 11. Jak patrno z Obr.4 je součet času b_i a c_i menší než čas a_a. Připomeňme,že

superskalární počítač musí v době vykonávání instrukce rozhodnout zda instrukce mohou být provedeny paralelně. Toto rozhodnutí je diskretním krokem v provádění instrukce tím, že se v podstatě přidává k prováděcí době v superskalárním počítači. Naproti tomu, jak patrno z Obr.4, složení v počítačovém systemu na Obr.1 neprodlužuje významně dobu žádanou k provedení operací počítače. Instrukce skládací jednotka 11 tak poskytuje větší výkonnost než počítač umístěný v instrukční prováděcí jednotce.

Obr.3 a 4 zobrazují dvě zásadní výhody složení v hlavní paměti. Zaprvé složení může být částí asynchronního procesu výpadku stránky bez prodloužení doby k úplnému dokončení procesu. Zadruhé, složení velkých bloků instrukčního textu, tak jako stránek, poskytuje větší příležitost uvažovat o významu složení, které může ovlivnit mnohem více optimální složení. Následkem toho je, že vstupní paměť instrukce skládající jednotky, tak jak znázorněno v Obr.1, poskytuje provozní výhody, poněvadž centrální základní jednotka bude vždy provádět instrukce, které byly složeny a složení lze lépe optimalizovat než, když by se vykonalo synchronně v menším úseku instrukčního textu.

Operace instrukce skládající jednotky bude nyní vysvětlena z počátku podle Obr.5a. Obr.5a ukazuje část toku složených nebo příznakových instrukcí jak se mají objevit na výstupu instrukce skládací jednotky 11 v Obr.1. Jak patrno každá instrukce Instr. je opatřena k ní přidaným příznakovým polem P pomocí instrukce skládací jednotky 11 . Označené instrukce, jako ty patrné z Obr.5a jsou uloženy do hlavní paměti v

stránkovém bloku na stránku obsahující instrukce. Příznakové instrukce v hlavní paměti 10 v případě potřeby jsou převedeny do vyrovnávací paměti 12 ,když dojde k výpadku . Příznakové instrukce ve vyrovnávací paměti 12 jsou potom vyvolány instrukce vyvolávající a vydávající jednotkou 16. Jakmile jsou označené instrukce přijaty vyvolávací a vydávací jednotkou 16 ,jsou jejich označená pole přezkoušena za účelem rozhodnutí zda mají být zpracovány paralelně a jejich pole operačního kódu jsou přezkoušena s ohledem na určení,které dosažitelné funkční jednotky jsou nejvhodnější k jejich zpracování. Jestliže příznaková pole indikují,že dvě nebo více instrukcí je vhodné paralelně zpracovat,potom jsou vyslány k příslušným funkčním jednotkám v souladu s kódováním jejich operačně kódovaných polí. Takové instrukce jsou pak zpracovány souběžně jedna s druhou jejich odpovídajícími funkčními jednotkami.

Když se vyskytne instrukce,že není vhodná k paralelnímu zpracování, potom je vyslána do příslušné funkční jednotky jak určí její operační kód a na to je zpracována sama o sobě vybranou funkční jednotkou.

V nejdokonalejším případě,kde větší počet instrukcí je vždy paralelně zpracován,je rychlost provedení instrukce počítačového systemu N krát větší než v případě,kde instrukce jsou prováděny jedna po druhé,přičemž N je počet instrukcí ve skupinách,které jsou zpracovávány paralelně.

Tok označených instrukcí v Obr.5a je snazší předběžně zpracovat instrukce skládající jednotkou,existují-li známé referenční body pro indikaci kde instrukce začínají. Takový

refrenční bod poskytne přesné označení kde jsou hranice instrukce. V mnohých počítačových systemech jsou instrukční meze výslovně známy jen pomocí kompilátoru v kompilační době a pouze prostřednictvím centrální základní jednotky, když jsou instrukce vyvolávány. Mezní referenční bod není znám mezi kompilační dobou a vyvoláním instrukce pokud není přijato specielní mezní referenční schema. Takové schema je zobrazeno v Obr.5b s instrukčními mezními bity B . Jak patrno z Obr.5b, mezní bity mohou být umístěny do instrukčního toku kompilátorem v době kompilace k poskytnutí reference zarovnání instrukce právě před jejím složením. Obecně patentové přihlášky s názvem

pojednávají o složení textových toků, ve kterých instrukční meze jsou neurčité. Kde instrukční meze ovšem jsou určitelné z textového toku jakož i kde tok obsahuje pouze instrukce a všechny instrukce jsou téže délky není třeba definice mezí.

Obr.6 znázorňuje mnohem podrobněji vnitřní konstrukci representačního spojení v jeden celek instrukce skládající jednotky podle předloženého vynálezu. Tato instrukce skládací jednotka 20 je vhodná k použití jako instrukce skládací mechanismus 11 v Obr.1. Instrukce skládací jednotka 20 v Obr. 6 je navržena pro případ kde mohou být paralelně zpracovány maximálně dvě instrukce najednou. Tím však není míněno omezení vynálezu jen na složení párů. V příkladu je použito 1-bitové pole příznaku. Hodnota příznakového bitu 1 /jedna/ znamená, že instrukce je první instrukcí. Hodnota příznakového bitu '0

/nula/ znamená,že instrukce je"druhou" instrukcí a může být provedena paralelně s předchozí první instrukcí. Instrukce s hodnotou příznakového bitu 1 může být provedena buď sama nebo v témže čase a paralelně s další instrukcí v závislosti na hodnotě příznakového bitu takové další instrukce.

Každé párování instrukce s hodnotou příznakového bitu jedna s následující instrukcí o hodnotě příznakového bitu nula tvoří složenou instrukci k paralelnímu provedení,to je, že instrukce v takovém páru mohou být jedna s druhou zpracovány paralelně. Když každý z příznakových bitů pro dvě po sobě následující instrukce má hodnotu jedna, je provedena první z těchto instrukcí samostatně, neparalelním způsobem. V nejhorším možném případě všechny instrukce v posloupnosti by mohly mít hodnotu příznakového bitu jedna. V takovém nejhorším případě by všechny instrukce byly provedeny jedna po druhé neparalelním způsobem.

Na vstupu instrukce skládací jednotky 20 přijme instrukce zarovnávající jednotka ze vstupu-výstupu adaptéru instrukční tok,který je třeba složit. Instrukční tok může obsahovat mezní bity B jak znázorněno na Obr.5b. V tomto případě zarovnání instrukce je jednoduchý postup detekce mezních bitů a dekódování instrukčních operačních kódů. Jak známo, v instrukční sadě operační kódy obsahují bity,které udávají délku instrukce ve slabikách nebo půlslovech. Jakmile byla proto instrukce identifikována mezním bitem B, může být další instrukce jednoznačně identifikována výpočtem množství slabik nebo půlslov od mezního bitu. Zarovnání instrukce není znakem tohoto vynálezu,přičemž se rozumí,že instrukční meze jsou iden-

tifikovány jakoukoliv známou metodou včetně použití mezních bitů.

Instrukce skládací jednotka 20 v Obr.6 obsahuje instrukční registr 21 s pluralitou instrukcí pro příjem plurality po sobě následujících instrukcí ze stránkových pamětí 18a a 18b adaptérů. Instrukce skládací jednotka 20 také obsahuje pluralitu mechanismů k analyzování instrukcí podle pravidel. Každý takový instrukce analyzující mechanismus analyzuje různé páry sousedících instrukcí v instrukčním registru 21 a vytváří signál k možnému složení, který indikuje zda mohou být paralelně zpracovány dvě instrukce toho páru nebo nemohou. V Obr.6 je zobrazena pluralita jednotek 22 - 25 analyzátoru složení. Každý z těchto skládacích analyzátorů 22--25 obsahuje dva instrukční analyzační mechanismy, které produkují dva signály složitelnosti. Například první složeninu analyzující jednotka 22 produkuje první signál M01 možného složení, který indukuje zda instrukce 0 a 1 mají být paralelně zpracovány nebo ne. Složeninu analyzující jednotky 22 produkují také druhý signál M12 možného složení, který indikuje zda instrukce 1 a 2 mají být zpracovány paralelně nebo ne.

Podobným způsobem druhá složeninu analyzující jednotka 23 produkuje první signál M23 možného složení, který indikuje zda instrukce 2 a 3 mají být paralelně zpracovány nebo nemají, a druhý signál M34 možného složení, který indikuje zda instrukce 3 a 4 mohou být paralelně zpracovány. Třetí analyzátor 24 složení produkuje první signál M45 složitelnosti, který indikuje zda instrukce 4 a 5 mohou nebo nemohou být paralelně zpra-

covány a druhý signál složitelnosti M56 ,který indikuje zda mohou nebo nemohou být zpracovány paralelně instrukce 5 a 6 . Čtvrtý analyzátor složení 25 produkuje první signál složitelnosti M67,který indikuje zda mohou být nebo nemohou paralelně zpracovány instrukce 6 a 7 a druhý signál složitelnosti M78 indikující zda mohou být paralelně zpracovány instrukce 7 a 8 .

Instrukce skládající jednotka 20 dále obsahuje mechanismus generování příznaků 26 odpovídající signálům složitelnosti,které se objeví na výstupech analyzačních jednotek 22-25 pro generování jednotlivých příznakových polí pro různé instrukce instrukčního registru 21. Tato příznaková pole T0,T1,T2 atd. jsou zasílána do instrukce označujícího registru 27 právě tak jako jsou zasílány vlastní instrukce,které jsou získány ze vstupu instrukčního registru 21 . Tímto způsobem je v jednotce složení výstupního registru 27 k disposici příznakové pole T0 pro instrukci 0,příznakové pole T1 pro instrukci 1 atd.

V předloženém spojení v jeden celek každé příznakové pole T0,T1,T2, atd. je obsaženo v jediném binárním bitu. Hodnota příznakového bitu "jedna" indikuje,že okamžitě následující instrukce,ke které je připojen je "první" instrukcí. Hodnota příznakového bitu "nula" indikuje,že ihned následující instrukce je "druhou" instrukcí. Instrukce s hodnotou příznakového bitu jedna následovaná instrukcí s hodnotou příznakového bitu nula indikuje,že tyto dvě instrukce mohou být provedeny paralelně jedna s druhou. Označené instrukce ve skládací jednotce výstupního registru 27 jsou postoupeny na výstup hlavní paměti 10 v Obr.1 přes jednu nebo ostatní vyrovnávací paměti 19a nebo 19b v Obr.3. Složené instrukce jsou uloženy do hlavní paměti 10 .

V Obr. 7 je zapojení prvního skládacího analyzátoru 22 , ve kterém první slučitelná logika 30 je spojena s prvním vstupem prvního součinového obvodu 34 , jehož druhý vstup je spojen s prvním registrem závislé logiky 32 . Druhá instrukční slučitelná logika 31 a druhý registr závislé logiky 33 jsou připojeny k prvnímu a druhému vstupu součinového obvodu 35 .

První instrukční slučitelná logika 30 v Obr. 8 obsahuje první dekodér 40 ,jehož první výstup A je spojen s prvním vstupem prvního součinového obvodu 42 , jehož druhý vstup je připojen k druhému dekodéru 41 . Druhý výstup prvního dekodéru 40 je spojen s prvním vstupem druhého součinového obvodu 43 , jehož druhý vstup je připojen k druhému dekodéru 41 . Výstupy obou součinových obvodů 42 ,43 jsou připojeny k obvodu logického součtu 44 . Výstup tohoto obvodu je připojen k prvnímu invertoru 52 v příznakovém generátoru 26 a současně k prvnímu vstupu prvního logického součtu 54 . Jeho druhý vstup je připojen přes druhý invertor 55 k obvodu logického součtu 48 v druhé instrukční slučitelné logice 31 . K prvnímu vstupu obvodu 48 logického součtu je připojen výstup pátého součinového obvodu 46 , jehož druhý vstup je připojen k šestému součinovému obvodu 47 . První vstupy obou součinových obvodů 46 , 47 jsou připojeny k druhému dekodéru 41 a třetímu dekodéru 45 .

V Obr.7 je nyní podrobněji znázorněna vnitřní konstrukce prvního skládacího analyzátoru 22 z Obr.6 . Ostatní čtyři skládací analyzátory 23-25 jsou podobné konstrukce. Jak patrno z Obr.7 první skládací analyzátor 22 obsahuje první instrukční slučitelnou logiku 30 pro zkoušení operačního kódu instrukce nula a operačního kódu instrukce 1 a pro určení zda tyto dva kódy jsou slučitelné pro účely paralelního provádění. První instrukční slučitelná logika 30 je navržena podle předem stanovených pravidel k vybírání párů operačních kódů, které jsou slučitelné pro paralelní provádění. První instrukční slučitelná logika 30 zejména obsahuje kromě toho logické obvody pro realizaci pravidel, která definují jaké typy instrukcí jsou slučitelné pro paralelní provádění ve specielním uspořádání technických prostředků počítačového systému. Jsou-li operační kódy pro instrukci 0 a 1 slučitelné, potom první slučitelná logika 30 vytvoří na svém výstupu dvojkový signál úrovně jedna. Nejsou-li slučitelné, první instrukční slučitelná logika 30 vytvoří na svém výstupním řádku dvojkovou hodnotu nula.

První skládací analyzátor 22 složení dále obsahuje druhou instrukčním slučitelnou logiku 31 slučitelnosti instrukcí pro přezkoušení operačních kódů instrukcí 1 a 2 a pro rozhodnutí zda jsou slučitelné pro paralelní provádění. Druhá instrukční slučitelná logika 31 je navržena stejným způsobem jako první instrukční slučitelná logika 30 podle těchže předem stanovených pravidel použitých pro první instrukční slučitelnou logiku 30 k vybírání párů operačních kódů, které jsou slučitelné pro paralelní provádění v případě instrukcí 1 a 2 . Druhá instrukční slučitelná logika 31 tak obsahuje logické obvody k realizaci pravidel , která definují, jaké typy instrukcí jsou slučitelné pro paralelní provádění, přičemž je použito stejných pravidel jako pro první instrukční slučitelnou logiku 30. Jsou-li operační kódy pro instrukce 1 a 2 slučitelné, potom druhá instrukční slučitelná logika 31 vytvoří dvojkový výstup úrovně jedna.

První skládací analyzátor 22 dále obsahuje první registr závislé logiky 32 pro detekci konfliktů při použití univerzálních registrů navržených pro pole R1 a R2 instrukcí 0 a 1. Tyto univerzální registry budou dále probrány mnohem podrobněji. První registr závislé logiky 32 kromě jiného lze navrhnout pro detekci výskytu závislých dat za podmínky, že druhá instrukce -instrukce 1- potřebuje využít výsledků z provozní charakterisitky předchozí instrukce -instrukce 0-. V tom případě může být druhá instrukce buď provedena závisle sdruženými technickými prostředky tak, že je provedena paralelně s první instrukcí, nebo provedení druhé instrukce musí počkat až je dokončeno provedení předchozí instrukce a tedy nemůže být provedeno paralelně s předchozí instrukcí. Je třeba poznamenat, že technika obcházení některých datových závislostí tohoto typu bude probrána až dále. Neexistují-li registrové závislosti, které zabraňují paralelnímu provedení instrukcí 0 a 1, pak výstupní řádek prvního registru logiky 32 udává dvojkovou hodnotu jedna. Jestliže závislost existuje, potom je udána dvojková hodnota nula.

První skládací analyzátor 22 složení dále obsahuje druhý registr závislé logiky 33 pro detekci konfliktů při použití univerzálních registrů navržených pro pole R1 a R2 instrukcí 1 a 2. Tato druhý registr závislé logiky 33 je stejné konstrukce jako před tím popsaná první registr závislé logiky 32 a tvoří výstup dvojkové úrovně jedna, neexistují-li žádné registrové závislosti nebo registrové závislosti lze provést technickými prostředky se závislými sdruženými daty, a jinak výstupem dvojkové úrovně nula.

Výstupní řádky první instrukční slučitelné logiky 30 prvního registru závislé logiky 32 jsou připojeny na vstupy prvního součinového obvodu 34. Výstupní řádek prvního součinového obvodu 34 má dvojkovou

hodnotu jedna, jsou-li dva uvažované operační kódy slučitelné, což znamená, že jsou způsobilé pro paralelní provedení. Má-li na druhé straně výstupní řádek prvního součinového obvodu 34 dvojkovou hodnotu nula, potom tyto dvě instrukce nejsou složitelné. Tím je vytvořen ve výstupním řádku prvního součinového obvodu 34 první signál MO1 složitelnosti, který indikuje zda instrukce 0 a 1 mají nebo nemají být paralelně zpracovány. Tento signál MO1 je dodáván do příznakového generátoru 26.

Výstupní řádky z druhé instrukční slučitelné logiky 31 a druhé instrukční slučitelné logiky 32 jsou připojeny ke dvěma výstupům druhého součinového obvodu 35. Druhý součinový obvod 35 vytvoří ve svém výstupním řádku druhý signál M12 slučitelnosti, který má dvojkovou hodnotu jedna, jestliže uvažované dva operační kódy operační kódy instrukcí 1 a 2 jsou slučitelné a jestliže neexistují žádné registrové závislosti instrukcí 1 a 2 nebo registrové závislosti, které lze provést technickými prostředky se sruženou závislostí dat. Výstupní řádek druhého součinového obvodu 35 má jinak dvojkovou hodnotu nula. Výstupní řádek druhého součinového obvodu 35 je veden ke druhému vstupu příznakového generátoru 26.

Ostatní čtyři skládací analyzátory 23-25 znázorněné v Obr.6 jsou téže vnitřní konstrukce jako v Obr.7 první analyzátor složení.

V Obr.8 je znázorněn nyní příklad logických obvodů, které lze použít k realizaci prvního skládacího analyzátoru 22 a příznakového generátoru 26, kterým jsou generovány první tři označení příznak 0 a příznak 1 a příznak 2. Například v Obr.5 se předpokládá, že existují dvě kategorie instrukcí, které jsou označeny jako kategorie A a kategorie B. Pravidla pro složení těchto kategorií instrukcí jsou předpokládána jak následuje :

/1/ A může se vždy skládat s A

/2/ A nemůže se nikdy složit s B

/3/ B nemůže se nikdy složit s B

/4/ B může se vždy složit s A

/5/ Pravidlo /4/ má přednost před pravidlem /1/ .

Pozornost je třeba věnovat tomu, že tato pravidla jsou velmi citlivá na pořadí výskytu instrukcí.

Dále je předpokládáno, že tato pravidla při bližším pohledu jsou taková, že neexistují žádné problémy s registrovými závislostmi, protože pravidla implicitně naznačují, že v případě jakéhokoliv blokování je toto blokování vždy proveditelné technickými prostředky závislosti dat. V příkladu na Obr.8 se jinými slovy předpokládá, že registrová závislost logiky 32 a 33 v Obr.7 není zapotřebí. V takovém případě první a druhý součinový obvod 34 a 35 nejsou také třeba a na výstupu první instrukční slučitelné logiky 30 je signál M01 a na výstupu druhé instrukční slučitelné logiky 31 je signál M12 .

Za těchto předpokladů jsou v Obr.8 znázorněny vnitřní logické obvody, jež lze použít pro první instrukční slučitelnou a v Obr.7 pro logiku 31 složitelnosti instrukcí. Podle Obr.8 první instrukční slučitelná logika 30 obsahuje první a druhý dekodéry 40 a 41 třetí a čtvrtý součinové obvody 42 a 43 a obvod logického součtu 44 . Druhá instrukční slučitelná logika 31 obsahuje dekodéry 41 a 45, součinové obvody 46 a 47 a obvod logického součtu 48 . První dekodér 41 společně používají obě instrukční slučitelné logiky 30 a 31. První instrukční slučitelná logika 30 zkouší operační kódy 0H0 a 0H1 pro instrukce 0 a 1 za účelem stanovení jejich slučitelnosti pro účely paralelního provedení.

To je prováděno podle zhora stanovených pravidel (1)-(4). První dekodér 40 zjistí operační kod první instrukce a je-li to operační kód kategorie A je nastavena výstupní vedení A dekoderu 40 na úroveň jedna. Je-li operační kód kategorie B potom je nastaveno výstupní vedení B prvního dekoderu 40 na úroveň jedna. Jestliže operační kód nenáleží ani do kategorie A ani do kategorie B, jsou oba výstupy prvního dekodéru 40 na úrovni binární nuly. Druhý dekoder 41 provádí podobný způsob dekodování pro druhý operační kód OP1.

Třetí součinový obvod 42 realisuje zhora uvedené pravidlo 1/. Je-li operační kód OP0 kategorie A a operační kód OP1 je také kategorie A, potom třetí součinový obvod 42 vytvoří výstup jedna. Jinak je výstup třetího součinového obvodu 42 na úrovni binární nuly. Čtvrtý součinový obvod 43 realisuje zhora uvedené pravidlo 4/. Je-li první operační kód kategorie B a druhý operační kód kategorie A , potom čtvrtý součinový obvod 43 vytvoří na výstupu úroveň jedna . Jinak vytvoří na výstupu úroveň nula. Jestliže buď třetí součinový obvod 42 nebo čtvrtý součinový obvod 43 vytvoří na výstupu úroveň jedna, vybudí se na výstupu obvodu OR 44 úroveň jedna, v kterémž to případě složitelnostní signál MO1 má hodnotu jedna. Tato hodnota jedna indikuje, že první a druhá instrukce - 0 a 1 - jsou slučitelné pro paralelní provádění.

Je-li prvním a druhým dekodérem 40 a 41 detekována jakákoliv jiná kombinace kategorií oparačních kódů, potom výstupy třetího a čtvrtého součinových obvodů 42 a 43 zůstávají na úrovni nula a složitelnostní signál MO1 má hodnotu nula indikující nesložitelnost.

Výskyt kombinací uvedených v shora uvedených pravidlech (2) a (3) tedy nesplňují třetí a čtvrtý součinový obvod 42 a 43 a signál M01 zůstává na úrovni nula. Nejsou-li další kategorie operačních kódů, kromě kategorie A a B, jejich výskyt v instrukčním toku neaktivisuje výstupy prvního a třetího dekodéru 40 a 42. Tedy obdobně to má za následek, že hodnota signálu složitelnosti M01 je nula.

Druhá instrukční slučitelná logika 31 provádí podobný typ analysy operačního kódu pro druhou a třetí instrukci - instrukci 1 a 2 -. Je-li druhý operační kód OP1 je-li kategorie A a třetí operační kód OP2 je kategorie A, potom podle pravidla (1) součinový obvod 46 vytváří na výstupu úroveň jedna a druhý složitelnostní signál M12 je vybuzen na úroveň binární jednotky indikující složitelnost. Jestliže na druhé straně OP1 je kategorie B a OP2 je kategorie A, potom podle pravidla (4) šestý součinový obvod 47 je aktivován pro vytvoření úrovně binární jednotky pro druhý složitelnostní signál M12. Pro jakoukoliv jinou kombinaci operačních kódů než je stanoveno v pravidlech (1) a (4) , má signál M12 hodnotu nula.

Složitelnostní signály M01 a M12 jsou vedeny do příznakového generátoru 26. V obr. 8 jsou znázorněny logické obvody, které lze použít v příznakovém generátoru 26 aby reagoval na složitelnostní signály M01 a M12 tak aby byly vytvořeny žádané označené bitové hodnpty pro příznaky 0,1 a 2. Označená bitová hodnota jedna indikuje, že sdružená instrukce je "první" instrukce pro paralelní provádění. Označená bitová hodnota nula indikuje, že sdružená instrukce je "druhá" instrukce pro paralelní provádění. Jediná instrukce v páru má označení

bitové hodnoty nula.Každá instrukce s označením bitové hodnoty jedna, která je následována jinou instrukcí s označením bitové hodnoty jedna je provedena sama o sobě jednotlivě a ne paralelně s následující instrukcí.

V případě prvního řádku na Obr. 9 všechny tři příznakové bity mají hodnotu jedna. To znamená, že každá z instrukcí o a 1 bude provedena jednotlivě ne paralelním způsobem. Pro případ druhého řádku na Obr. 2 instrukce 0 a 1 budou provedeny paralelně poněvadž příznak 0 má požadovanou hodnotu jedna a příznak 1 má požadovanou hodnotu nula. Pro případ třetího řádku na Obr. 9 instrukce 0 bude provedena jednotlivě, zatím co instrukce 1 a 2 budou provedeny jedna s druhou paralelně. Pro čtvrtou řádku budou instrukce 0 a 1 provedeny paralelně jedna s druhou.

V těch případech, kde příznak 2 má binární hodnotu 1 stav jeho sdružené instrukce 2 je závislý na binární hodnotě příznaku 3. Má-li příznak 3 binární hodnotu nula, potom instrukce 2 a 3 mohou být provedeny paralelně.Jestliže na druhé straně příznak 3 má binární hodnotu jedna, potom instrukce 2 bude provedena jednotlivě,neparalením způsobem. Je třeba poznamenat, že logika realisovaná pro příznakový generátor 26 nedovoluje výskyt dvou následných příznakových bitů o binárních hodnotách nula.

Z Obr. 9 vyplývá potřebná logika k realisaci části příznakového generátoru 26 na Obr.8. Jak ukázáno v Obr. 9, příznak 0 bude mít vždy binární hodnotu jedna.To je splněno vytvořením konstantní binární hodnoty jedna na výstupním vedení 50 příznakového generátoru, která obsahuje příznak 0 výstupního vedení. Z obrázku 9 dále vyplývá, že bitová hodnota příznaku 1 je vždy opačná bitová hodnota složitelnostního signálu M01. Tento výsledek je dosažen připojením

prvního prvního
výstupního vedení 51 pro příznak 1 k výstupu invertoru 52,
na jehož vstup je připojeno signální vedení M01,

druhém
Binární úroveň příznaku 2 ve výstupním vedení 53
prvním, logického součtu a druhým, prvního
je určena obvodem 54 a invertorem 55 . Jeden vstup ob-
logického součtu
vodu 54 je připojen k signálnímu vedení M01. Jestliže M01 má hodnotu jedna, potom příznak 2 má hodnotu jedna. To se týká hodnot příznaku 2 ve druhém a čtvrtém řádku na Obr.9.Další vstup
prvního obvodu logického součtu druhý invertor
54 je připojen přes 55 k signálnímu vedení M12. Jestliže M12 má binární hodnotu nula,je tato hodnota invertována druhým
invrtorem 55 na binární hodnotu jedna pro druhý vstup prvního
logického součtu
obvodu 54. Výstupní vedení 53 příznaku 2 následkem toho má
druhé
binární hodnotu jedna. To se týká hodnoty příznaku 2 v řádce jedna na Obr. 9. Je třeba poznamenat, že v případě řádku 3 příznak 2 musí mít hodnotu nula.Je tomu tak proto, že M01 bude mít hodnotu nula a M12 bude mít hodnotu jedna, která je inver-
druhým
tována invertorem 55 , aby vytvořila nulu na druhém vstupu
prvního obvodu 54.

Z logiky na Obr.9 vyplývá pravidlo přednostní priority pro řádek čtyři v případě, že každé vedení M01 a M12 má binární hodnotu jedna. Tento případ čtvrtého řádku může být vytvořen instrukční kategorií posloupností B, A, A. To může být realisováno příznakovou posloupností 1, 0, 1. na Obr. 9, nebo alternativně příznakovou posloupností 1,1,0. V tomto seskupení platí pravidlo 5) a posloupnost 1,0,1 v Obr.9 je vybrána. Jinými slovy párování B,A je dávána přednost před párováním A,A.

1,1
Tvar pro M01 M12 může být také vytvořen sledem operačních kódů A,A,A. V tomto případě sled příznaků 1,0,1

je znovu vybrán podle Obr. 9. To je lepší,protože je k dispozici hodnota jedna pro příznak 2,čímž je potenciálně umožněno složit instrukci 2 s instrukcí 3,je-li instrukce 2 slučitelná s instrukcí 3.

Na Obr.10 je příklad možné konstrukce počítače s použitím příznaku ke složení podle vynálezu,aby bylo docíleno paralelního zpracování instrukcí ve strojovém jazyku počítače . Instrukce skládací jednotka 20 je přes hlavní paměť 10 a rychlou vyrovnávací paměť 12 připojena k řídící jednotce 60 vyvolání-vydání . Tato jednotka 60 je spojena s první funkční jednotkou 61, druhou funkční jednotkou 62 , třetí funkční jednotkou 63 a čtvrtou funkční jednotkou 64. Všechny funkční jednotky 61 až 64 jsou připojeny k universálnímu registru 65 a k rychlé vyrovnávací paměti 66.

Instrukce skládací jednotka 20 je použita za předpokladu,že je typem popsaným v Obr.6 a jako taková připojuje ke každé instrukci pole jednobitového příznaku.Tato příznaková pole jsou použita k identifikaci párů instrukcí, které mají být paralelně zpracovány. Stránky obsahující takto označené instrukce jsou postoupeny a uloženy v hlavní paměti 10 . Je-li takto označených instrukcí třeba,jsou čteny nebo přeneseny do rychlé vyrovnávací paměti 12 .

Řídící jednotka 60 vyvolání-vydání postoupí označené instrukce z rychlé vyrovnávací paměti 12,je-li toho třeba, a uspořádá je pro zpracování jednou nebo více pluralitními funkčními jednotkami 61 až 64. Jednotka 60 zkoumá příznaková pole a pole operačních kódů přivedených instrukcí.Indikují-li příznaková pole,že dvě následné instrukce mají být zpracovány paralelně,jednotka 60 vyvolání-vydání je přiřadí těm funkčním jednotkám 61-64 jak určeno jejich operačními kódy a tyto jednotky je zpracují paralelně. Indikují-li příznaková pole,že specielní instrukci je třeba zpracovat samostatně neparalelním způsobem,potom ji jednotka 60 vyvolání-vydání přiřadí specielní funkční jednotce jak určeno jejím operačním kódem a je zpracována nebo provedena samostatně.

První funkční jednotka 61 je určena k zpracování větvených instrukcí. Druhá funkční jednotka 62 je aritmetickou a logickou jednotkou pro generování adres se třemi vstupy,která je určena k výpočtu paměťových adres instrukcí,které přesouvají operandy do nebo z paměti . Třetí funkční jednotka 63 je universální aritmetická a logická jednotka určená pro matematické a logické operace. Čtvrtá funkční jednotka 64 je jednotkou se sdruženou závislostí dat se třemi vstupy způsobilá k provedení dvou aritmeticko-logických operací v jednoném strojovém cyklu.

Počítačový system v Obr.10 obsahuje také sadu universálních registrů 65 pro zpracování několika instrukcí ve strojovém jazyku.Tyto universální registry jsou použity k typicky dočasnému uložení datových a adresových operand nebo jsou použity jako čítače nebo ke zpracování ostatních dat. Typický system je opatřen šestnácti universálními registry. Předpokládá se, že registry jsou jedním z vícekanálových typů,přičemž lze dosáhnout současně dva nebo více registrů.

Počítačový system v Obr.10 obsahuje dále rychlou vyrovnávací paměť 66 pro ukládání datových operand získaných z hlavní paměti 10 vyšší úrovně . Data lze také přesouvat zpět z rychlé vyrovnávací paměti 66 do hlavní paměti 10 . Rychlá vyrovnávací paměť 66 může být známého typu a její pracovní režim relativně přizpůsoben hlavní paměti 10 řešen známým způsobem.

Na Obr. 11 je znázorněn příklad složené nebo označené instrukční posloupnosti, která může být zpracována počítačovým systémem na Obr. 10. Příklad na Obr. 11 je složen z následujících instrukcí následovně: plnění ,připojení , srovnání , větvení podmíněné a ukládání . Tyto instrukce jsou identifikovány jako instrukce I1-I5. Příznakové bity pro tyto instrukce jsou přiřazeny následovně. 1,1,0,1, a 0. Vzhledem k organisaci stroje znázorněné na Obr, 10 instrukce plnění je spracována jednotlivým způsobem samostatně. Instrukce přípojení a srovnání jsou považovány za složené instrukce a jsou zpracovány paralelně jedna s druhou. Instrukce větvení a ukládání jsou považovány také za složené instrukce a jsou rovněž zpracovány paralelně jedna s druhou.

Tabulka v Obr.12 udává další informace o každé z instrukcí na Obr.11. Sloupec R/M v Obr.12 indikuje obsah prvního pole každé instrukce, která je typicky použita k identifikaci jednoho zvláštního z universálních registrů 65, který obsahuje první operandu. Jako výjimka je případ instrukce podmíněného větvení, v němž pole R/M obsahuje masku podmíněného kódu. Sloupec R/X v Obr.12 indikuje obsah druhého pole v každé instrukci, kteréžto péle je typicky použito pro identifikaci druhého jednoho z universálních registrů 65. Takový registr může obsahovat druhou operandu nebo může obsahovat hodnotu adresního indexu X . Sloupec B v Obr.12 indikuje obsah třetího možného pole v každé instrukci, kteréžto pole může identifikovat jeden zvláštní registr z universálních registrů 65, který obsahuje základní adresovou hodnotu. 0 v sloupci B indikuje nepřítomnost v poli B nebo nepřítomnost

odpovídající adresové složky v poli B. Pole D v Obr.12 indikuje obsah dalšího pole v každé instrukci, která je-li použita ke generování adres obsahuje hodnotu posuvu adresy. Nula ve sloupci D může indikovat také nepřítomnost odpovídajícího pole v e zvláštní instrukci právě uvažované nebo, alternativně, hodnotu 0 pro posuv adresy.

Nyní bude sledováno zpracování instrukcí plnění v Obr.11, řídící jednotkou 60 vyvolání-vydání určené příznakovými bity instrukce plnění a následující instrukce připojení, že instrukci plnění je třeba zpracovat jednotlivě samostatně. K provedení funkce instrukce plnění je třeba vyvolání operandu z paměti, v tomto případě vyrovnávací rychlé paměti 66 a umístnění této operandy do R2 universálního registru. Paměťová adresa, podle které je operand vyvolán je určena přidáním indexové hodnoty v registru X, základní hodnoty v registru B a hodnoty D posuvu. Řídící jednotka 60 vyvolání-vydání přiřadí tuto adresu operaci generování druhé funkční jednotce 62 generování adres. V tomto případě druhá funkční jednotka 62 připojí hodnotu adresového indexu v registru X -hodnota nula v tomto příkladu- a, hodnotu základní adresy obsaženou v universálním registru R7 a hodnotu posuvu adresy -hodnota nula v danném příkladě- obsaženou v samotné instrukci. Výsledná vypočítaná paměťová adresa objevující se na výstupu druhé funkční jednotky 62 je předána na adresový vstup vyrovnávací rychlé paměti 66 pro přístup k žádanému operandu. Dosažený operand je postoupen do universálního registru R2 v sadě registrů 65 .

Sledujeme-li nyní zpracování instrukcí připojení a srovnání jsou tyto instrukce vyvolány řídící jednotkou 60 vyvolání-vydání. Řídící jednotka 60 přezkouší příznaky ke složení těchto instrukcí a zaznamená, že mohou být paralelně provedeny. Jak patrno z Obr.12 instrukce srovnání má zřejmou závislost dat na instrukci připojení, jelikož připojení musí být dokončeno před R3 dříve než se může být provedeno srovnání. Tuto závislost může však zpracovat čtvrtá funkční jednotka 64 se sdruženými závislými daty. Následkem toho tyto dvě instrukce lze zpracovat paralelně podle konfigurace z Obr. 10. Řídící jednotka 60 zejména přiřadí zpracování instrukce třetí funkční jednotce 63 a přiřadí zpracování instrukce srovnáním čtvrté funkční jednotce 64 se sdruženou závislostí.

Třetí funkční jednotka 63 připojí obsahy R2 universálního registru k obsahu R3 universálního registru a umístní výsledek připojení zpět do R3 universálního registru. Současně třetí funkční jednotka 63 se sdruženou závislostí provede následující matematickou operaci: R3 + R2 - R4 Podmiňovací kód pro výsledek této operace je zaslán do registru podmiňovacího kódu umístněného do větvené první funkční jednotky 61 . Datová závislost je združená, protože čtvrtá funkční jednotka 64 vypočítá totiž součet R3 + R2 a ten pak porovná s R4 pro určení podmiňovacího kódu. Následkem toho nemusí čtvrtá funkční jednotka 64 čekat na výsledky z třetí funkční jednotky 63, která provádí instrukci připojení. V tomto zvláštním případě číselné výsledky vypočtené čtvrtou funkční jednotkou 64 se objeví na jejím výstupu a nemusí se předávat zpět do universálních registrů 65 . V tomto případě stanoví čtvrtá funkční jednotka 64 toliko podmiňovací kód.

Sledujeme-li nyní zpracování instrukce větvení a instrukce ukládání znázorněné na Obr.11, jsou tyto instrukce vyvolány z rychlé vyrovnávací paměti 12 pro složení instrukce řídící jednotkou 60 vyvolání-vydání. Řídící jednotka 60 rozhodne podle příznakových bitů těchto instrukcí, že může být jedna s druhou zpracována odděleně. Dále může rozhodnout podle operační kódů těchto dvou instrukcí, že větvená instrukce má být zpracována první funkční jednotkou 61 a ukládaná instrukce má být zpracována druhou funkční jednotkou 62 pro generování adres. V souladu s tímto rozhodnutím je předáno maskovací pole M a posuvné pole D větvené instrukce do větvené první funkční jednotky 61. Podobně hodnota indexu adresy v registru X a hodnota základní adresy v registru B pro větvenou instrukci se získají z universálních registrů 65 a jsou zaslány do větvené první funkční jednotky 61. V daném příkladě hodnota X je nula a základní hodnota je získána z R7 universálního registru. Posuvná hodnota D má hexadecimální hodnotu dvacet, zatím co maskovací pole M má hodnotu polohy masky osm.

Větvená jednotka 61 začne vypočítávat potencialní větvenou adresu (0 + R7 + 20) a současně porovnává podmiňovací kód získaný z předchozí srovnávací instrukce s maskou M podmiňovacího kódu. Je-li hodnota podmiňovacího kódu stejná jako hodnota kódu masky, je dána podmínka potřebného větvení a vypočtená adresa větve první funkční jednotkou 61 větvení je na to zaslána do počítače instrukcí v řídící jednotce 60. Počítač instrukcí zkontroluje vyvolání instrukcí z rychlé vyrovnávací paměti 12 složených instrukcí. Na druhé straně není-li podmínka udána - to je v případě, že podmiňovací kód stanovený předchozí instrukcí nemá hodnotu - ,potom nenastane žádné větvení a ne-

ní žádná větvená adresa zaslána do řídící jednotky 60 počítače instrukcí.

Současně tím, že větvená první funkční jednotka 61 je v činosti provádí se její procesní akce pro instrukci větvení, a druhá funkční jednotka 62 pro generování adresy je v činnosti, přičemž provádí výpočet adresy (0 + R7 + 0) pro instrukci paměti. Adresa vypočtená druhou funkční jednotkou 62 je zaslána do datové rychlé vyrovnávací paměti 66 . Není-li větvenou jednotkou provedeno žádné větvení, potom paměťová instrukce zpracuje operand R3 univesálního registru pro uložená do datové rychlé vyrovnávací paměti 66 na adresu vypočtenou druhou funkční jednotkou 62. Na druhé straně jeli podmínka větvení udána a větvení je provedeno, potom obsahy R3 universálního registru nejsou uloženy do datové rychlé vyrovnávací paměti 66.

V předchozím Obr. 11 posloupnost instrukcí je uváděna jen jako příklad. Celkové zapojení počítačového systému v Obr.10 je rovněž způsobilé k zpracování různých a všelikých jiných instrukčních sledů.Příklad na Obr.11 však jasně ukazuje užitečnost příznaku složených instrukcí při určování, které páry instrukcí mají být paralelně zpracovány s jinými.

Každé párování instrukce mající hodnotu příznakového bitu jedna s následující instrukcí mající hodnotu příznakového bitu nula vytváří složenou instrukci pro účely paralelního zpracování, což znamená, že instrukce mohou být zpracovány jedna s druhou. Kkyž příznakové bity u každé z následujících dvou instrukcí mají hodnotu jedna, první z těchto instrukcí je provedena samostatně neparalelním způsobem.V nejhorším možném případě všechny instrukce v posloupnosti by moh-

ly mít hodnotu příznakového bitu jedna. V nejhorším možném případě všechny instrukce posloupnosti by mohly být zpracovány najednou neparalelním způsobem.

Shora uvedený příklad technických prostředků je uváděn jen pro malou oblast. S ohledem na to je každý pár sousední instrukce analysován, aby se zjistilo zda je možno zpracovat ho paralelně. Paměťové skládání nabízí totiž zkoušení více než dvou instrukcí a vybrání nejlepšího existujícího seskupení.

Technika složení uváděná v příkladech předpokládá, že je známo kde je začátek instrukce. Obecně lze instrukční meze identifikovat kompilátorem jak již zmíněno nebo dekódováním instrukce před zpracováním.

V závěru je třeba se zmínit, že instrukce skládací jednotka byla znázorněna specielně v poloze mezi vstupem-výstupem adaptéru a paměťovou sběrnicí. Tímto příkladem není vyloučeno jakékoliv jiné umístění v paměti , kde instrukce skládací jednotka může pracovat. Může být obsažena ve vstupu-výstupu adaptéru, může pracovat jako samostatná jednotka do paměťové sběrnice 9 nebo může obsahovat jednotku připojenou pouze k hlavní paměti privátním paměťovým kanálem nepřístupným přes paměťovou sběrnici 9 . Číslicový počítačový system podle vynálezu je možno specialisty v oboru přizpůsobit a adaptovat podle potřeby. Proto je požadována ochrana předloženého vynálezu jen v omezení a souladu s patentovými nároky.

PATENTOVÉ NÁROKY

1. Číslicový počítačový system obsahující prostředky k provedení většího počtu instrukcí paralelně a kombinaci vyznačující se tím,že obsahuje vstupní/výstupní propojovací mezičlánek k vytvoření skupiny instrukcí, které mají být zpracovány; mechanismus ke složení instrukcí pro vytvoření příznakové informace ke složení v závislosti na skupině instrukcí,přičemž příznaková informace ke složení indikuje instrukce ze skupiny instrukcí,které mají být provedeny paralelně; a hlevní paměť připojenou na vstupní/ výstupní propojovací mezičlánek a k mechanismu složení instrukcí pro uložení skupiny instrukcí s příznakovou informací ke složení.
2. Kombinace podle nároku 1 vyznačující se tím,že příznaková informace ke složení je obsažena ve větším počtu označených polí,přičemž každé označené pole je přiřazeno příslušné instrukci analyzované mechanismem složení instrukcí.
3. Kombinace podle nároku 1 nebo 2 vyznačující se tím,že počítačový system obsahuje větší počet funkčních instrukčních základních jednotek,které vzájemně spolupracují paralelně,přičemž příznaková informace je použita k vydání dvou nebo více instrukcí pro různé druhy funkčních jednotek.

4. Kombinace podle jednoho z nároků 1 až 3 vyznačující se tím,že mechanismus složení instrukcí obsahuje plurálně-instrukční registr instrukcí pro příjem množství postupných instrukcí; analyzační mechanismus instrukcí založený na pluralitním pravidlu,přičemž každý analyzační mechanismus podle předpisu analyzuje zejména pár sousedících instrukcí v instrukčním registru a vytváří signál ke složení,který indikuje zda dvě instrukce zmíněného páru mají nebo nemají být zpracovány paralelně; a mechanismus generující označení podle signálů ke složení pro vytváření jednotlivých označených polí pro různé instrukce v instrukčním registru.

5. Kombinace podle nároku 4 vyznačující se tím,že počítačový systém je specielní konfigurací k zpracování instrukcí,přičemž každý mechanismus k analyzování instrukce obsahuje logické obvody pro realizování pravidel,která definují, které typy instrukcí jsou způsobilé pro paralelní provedení ve specielní konfiguraci k zpracování instrukcí použité v počítačovém systemu,logických obvodech vytvářejících slučitelný signál pro analyzační mechanismus.

6. Číslicový počítačový system způsobilý k paralelnímu zpracování dvou nebo více instrukcí a kombinace vyznačující se tím,že obsahuje prostředky pro příjem skupiny instrukcí v posloupnosti instrukcí,které mají být zpracovány; mechanismus složení instrukcí připojený k těmto prostředkům,mechanismus složení instrukcí pro příjem skupiny instrukcí a přiřazených polí označených ke složení a ukládání do paměti.

7. Kombinace podle nároku 6 vyznačující se tím,že obsahuje pluralitu funkčních instrukčních základních jednotek, které pracují paralelně jedna s druhou; a mechanismus vydání instrukcí připojený k mechanismu ukládání do paměti pro dodávání v ní uložených instrukcí do různých jiných funkčních instrukčních základních jednotek ,když jejich označená pole indikují,že mají být paralelně zpracovány.

8. Kombinace podle nároku 6 nebo 7 vyznačující se tím,že mechanismus pamatování obsahuje hlavní paměť pro ukládání bloku informace včetně zmíněných instrukcí a mechanismus rychlé vyrovnávací paměti připojené k hlavní paměti a k mechanismu vydání instrukcí.

9. Kombinace podle jednoho z nároků 6 až 8 ,vyznačující se tím,že mechanismus pamatování obsahuje hlavní paměť mající velikost slova dostatečnou k přidání polí označených ke složení instrukcí ze zmíněné skupiny instrukcí.

10. Kombinace podle jednoho z nároků 6 až 9, vyznačující se tí,že mechanismus pamatování obsahuje hlavní paměť pro ukládání skupiny instrukcí a příznakovou paměť pro ukládání složených příznaků.

11. Kombinace podle jednoho z nároků 6 až 10, vyznačující se tím,že mechanismus pamatování obsahuje hlavní paměť, přičemž hlavní paměť obsahuje příznakovou tabulku pro příznaková pole ke složení a samostatnou sekci pro ukládání pluralitních skupin instrukcí.

12. Kombinace podle jednoho z nároků 6 až 11, vyznačující se tím, že mechanismus pamatování obsahuje hlavní paměť přičemž hlavní paměť obsahuje stránkovou sekci mající pr vní sekci pro ukládání skupiny instrukcí a druhou sekci pro ukládání příznakových polí ke složení.

13. Kombinace podle jednoho z nároků 6 až 12, vyznačující se tím, že skupina instrukcí je stránkou instrukcí.

14. Kombinace podle jednoho z nároků 6 až 13, vyznačující se tím, že obsahuje prostředky k provedení instrukcí připojené k paměťovému mechanismu pro vyvolání pluralita instrukcí a v odpovědi na příznaková pole ke složení paralelní provedení pluralitních instrukcí.

15. Kombinace podle nároku 14, vyznačující se tím, že prostředky k provedení instrukcí pracují asynchronně s mechanismem pro složení instrukcí.

16. Číslicový počítačový systém způsobilý k současnému provedení pluralitních instrukcí a kombinace , vyznačující se tím, že obsahuje paměťový mechanismus pro ukládání předem stanovených skupin instrukcí určených k provedení; jednotku pro složení instrukcí připojenou k paměťovému mechanismu pro generování složené příznakové informace, která identifikuje instrukce v předem určené skupině instrukcí, které mají být současně provedeny; prostředky k uložení složené příznakové informace do paměťového mechanismu a připojeného k jednotce složení instrukce pro uložení složené příznakové informace.

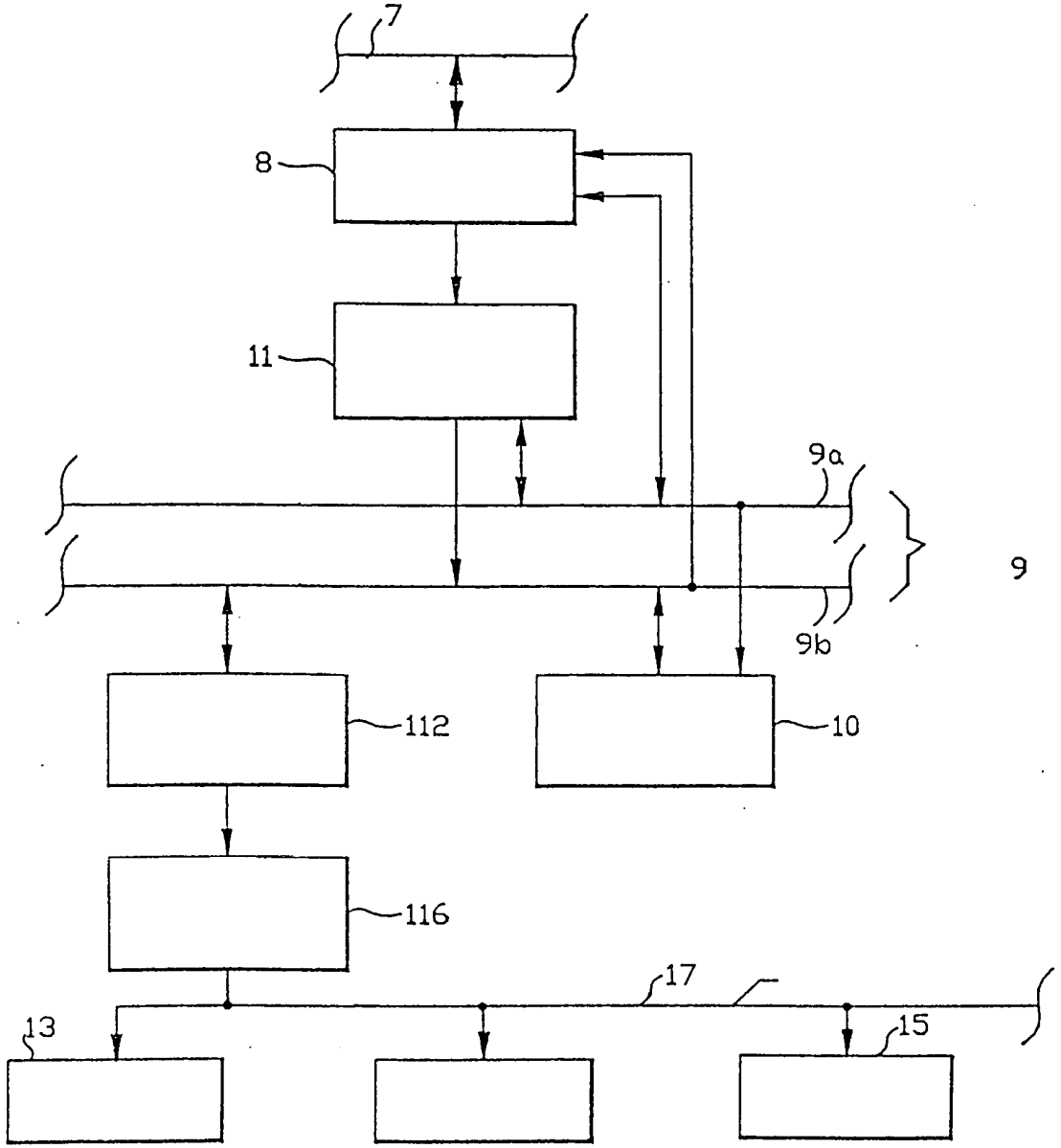
17. Kombinace podle nároku 16, vyznačující se tím, že jednotka ke složení instrukcí je asynchronní s prostředky k provedení instrukce.

18. Kombinace podle nároku 16 nebo 17 , vyznačující se tím, že prostředkem k ukládání složené příznakové informace je příznaková paměť.

19. Kombinace podle jednoho z nároků 16 až 18, vyznačující se tím, že prostředkem k ukládání složené příznakové informace je příznaková tabulka.

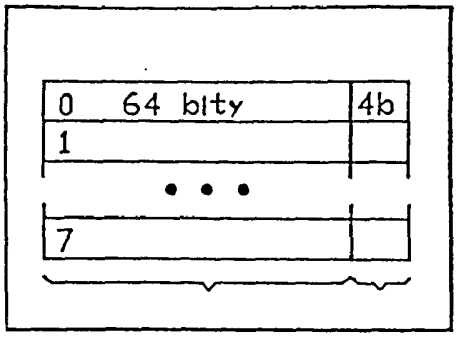
20. Kombinace podle jednoho z nároků 16 až 19, vyznačující se tím, že prostředek k ukládání složené příznakové informace obsahuje znak mezery pamětového mechanismu připojený k předem určené skupině instrukcí.

JUDr. Petr KALENSKÝ
advokát

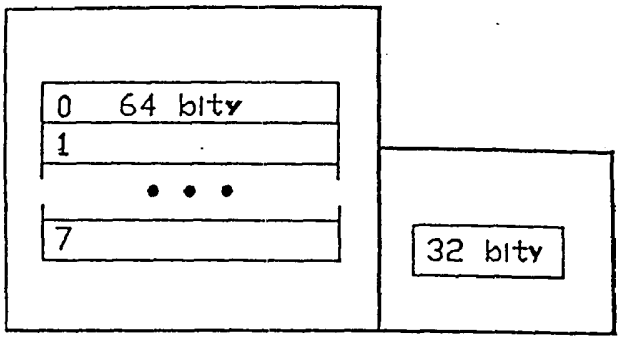


Obr. 1

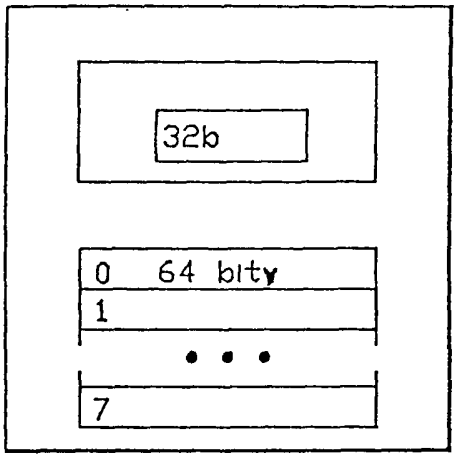
PV 934-91



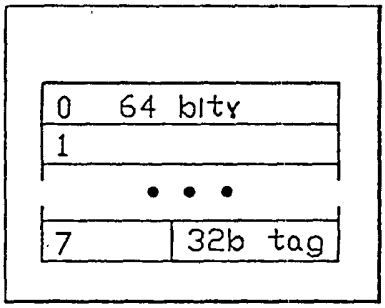
Obr. 2A



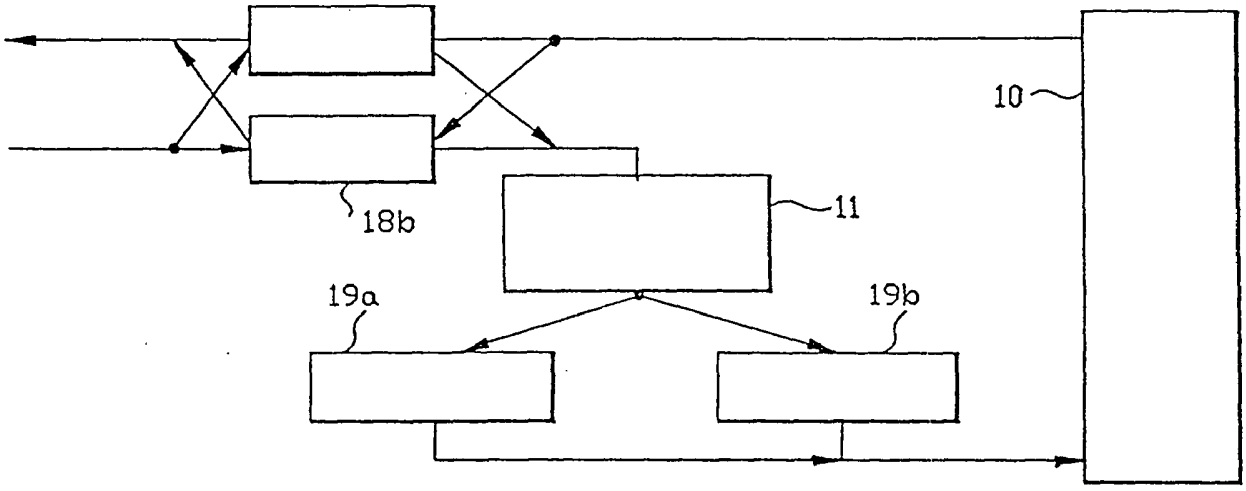
Obr. 2B



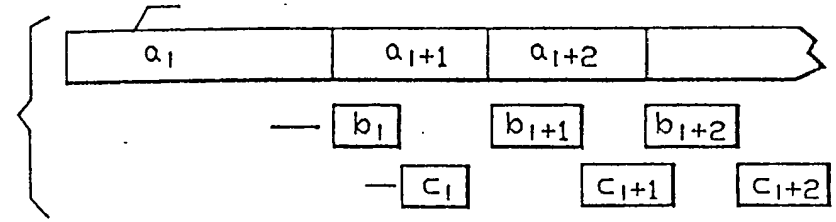
Obr. 2C



Obr. 2D



Obr. 3



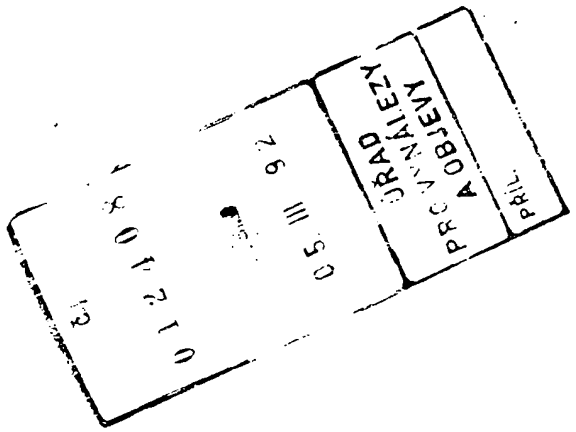
Obr. 4

P 0	Instr. 0	P 1	Instr. 1	P 2	Instr. 2	P 3	Instr. 3

Obr. 5a

B	Instr. 0	B	Instr. 1	B	Instr. 2	B	Instr. 3

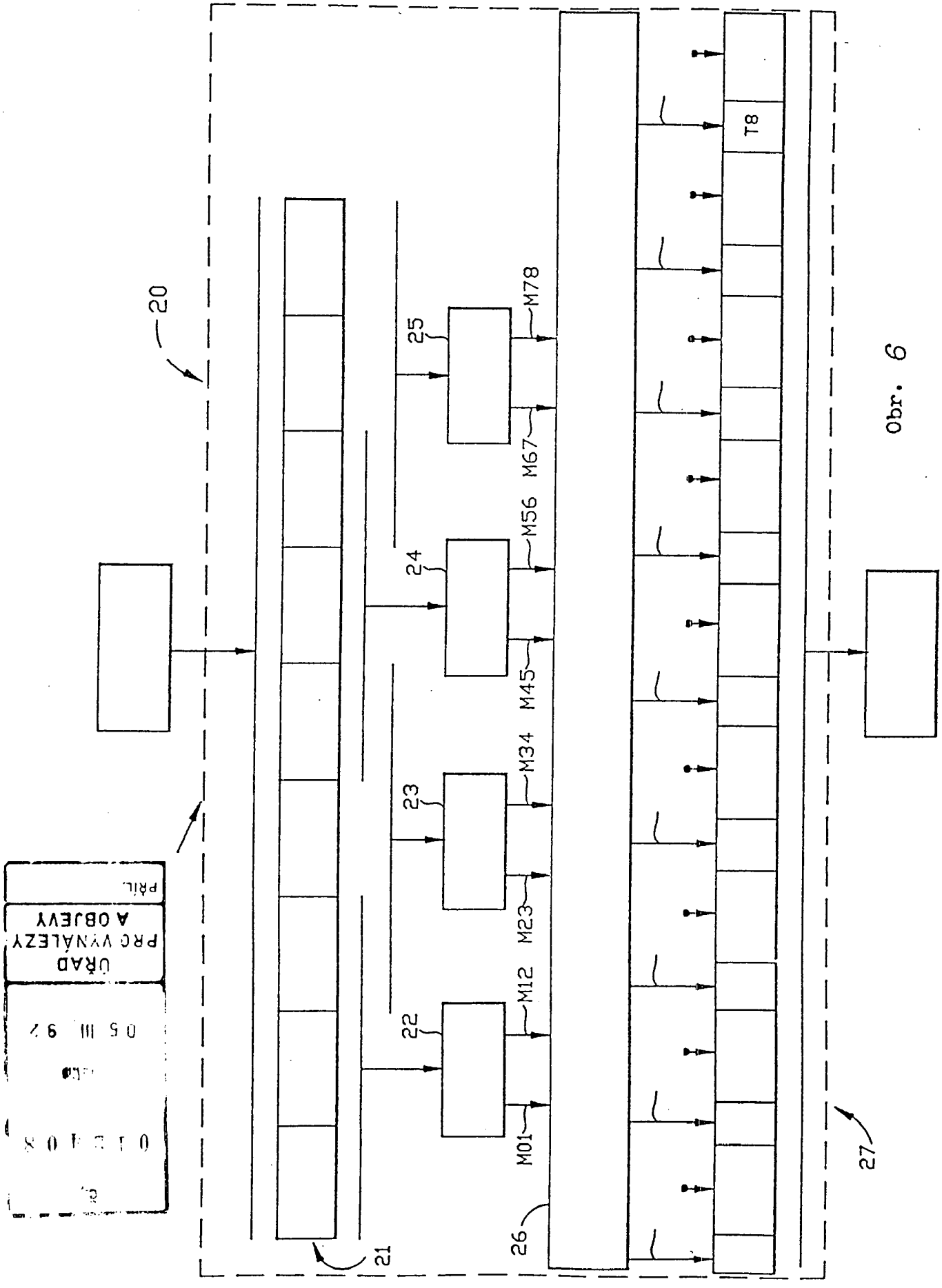
Obr. 5b



Obr. 9

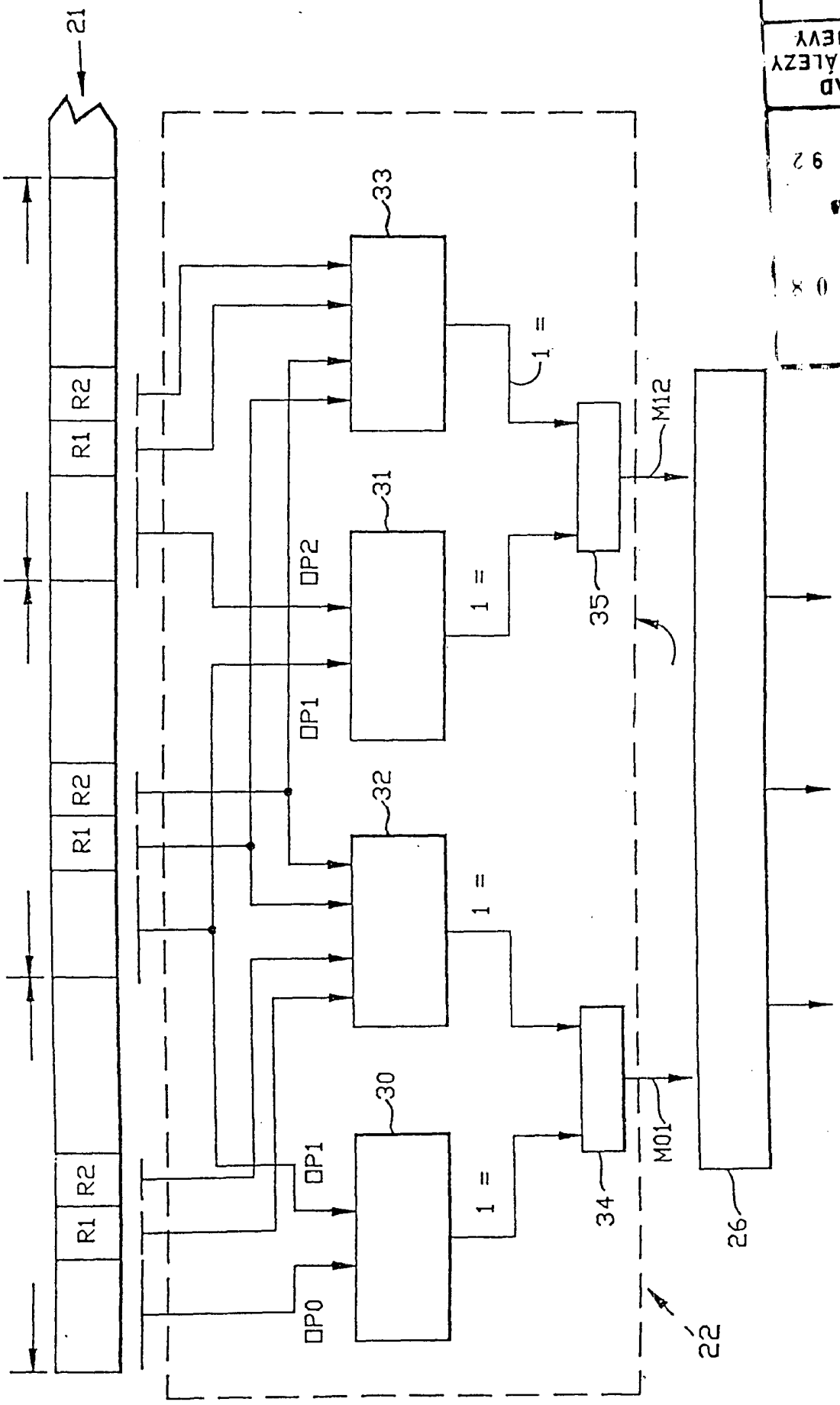
M01	M12	0	1	2
0	0	1	1	1
1	0	1	0	1
0	1	1	1	0
1	1	1	0	1

PV 934-91

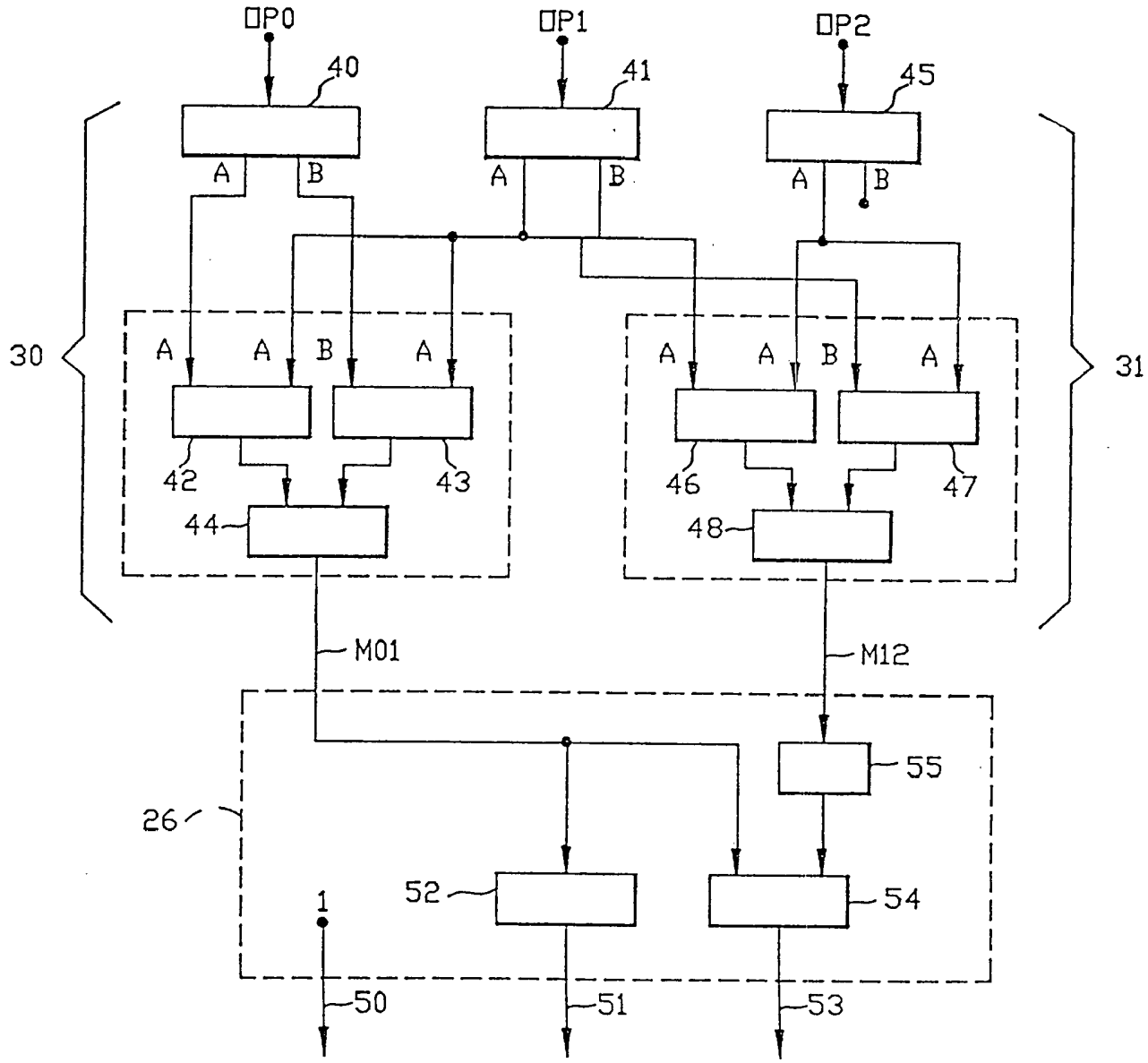


Obr. 6

PV 934-91

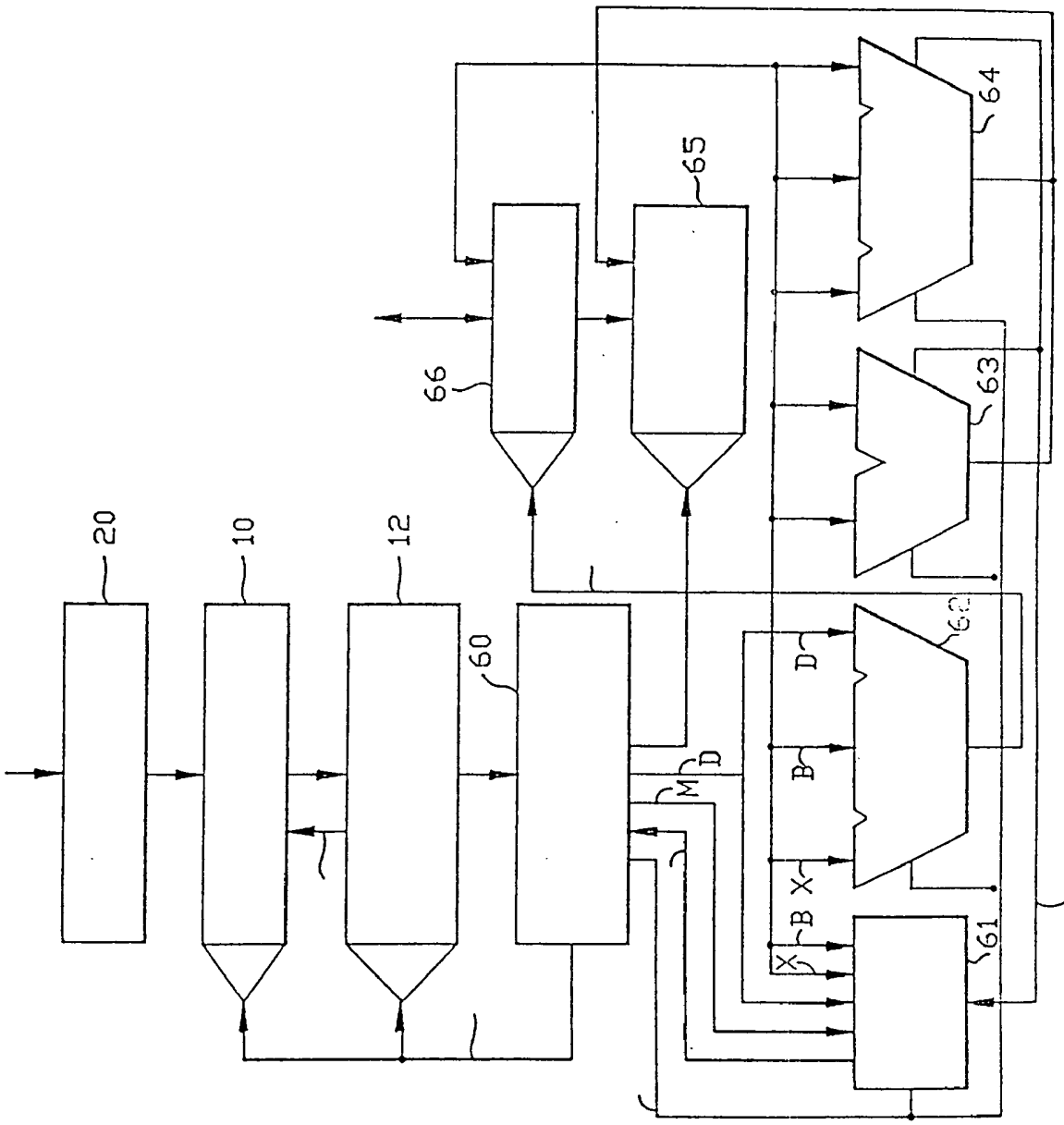


Obr. 7

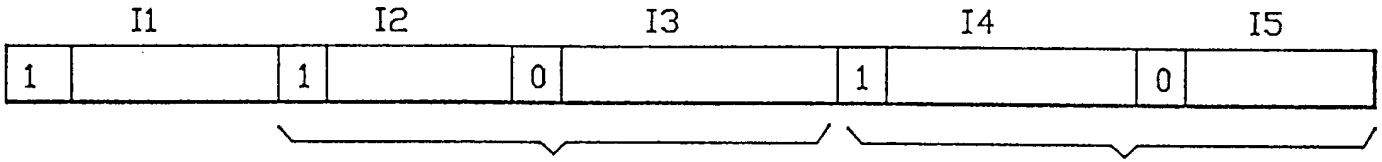


Obr. 8

PV 934 - 91



Obr. 10



Obr. 11

				B	D	
I1		R2	0	R7	0	→
I2		R3	R2	0	0	R3 + R2 → R3
I3		R3	R4	0	0	R3 (C) R4 →
I4		8	0	R7	20	
I5		R3	0	R7	0	R3 →

Obr. 12