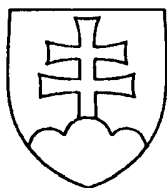


SLOVENSKÁ REPUBLIKA

(19)

SK



ÚRAD
PRIEMYSELNÉHO
VLASTNÍCTVA
SLOVENSKEJ REPUBLIKY

**ZVEREJNENÁ PRIHLÁŠKA
VYNÁLEZU**

- (22) Dátum podania: 13.11.91
(31) Číslo prioritnej prihlášky: 611 594
(32) Dátum priority: 13.11.90
(33) Krajina priority: US
(43) Dátum zverejnenia: 12.01.95
(86) Číslo PCT:

(21) Číslo dokumentu:

3440-91

(13) Druh dokumentu: **A3**

(51) Int. Cl. 6:

G 06 F 17/00
G 06 F 19/00

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

(72) Pôvodca vynálezu: Dieffenderfer James Warren, Owego, NY, US;
Kogge Peter Michael, Endicott, NY, US;
Wilkinson Paul Amba, Appalachin, NY, US;
Schoonover Nicholas Jerome, Tioga Center, NY, US;

(54) Názov prihlášky vynálezu: **Paralelný procesorový systém**

(57) Anotácia:

Obsahuje množinu procesorových piketových jednotiek (100), pričom každá procesorová piketová jednotka (100) obsahuje miestnu pamäť (102), ktorej výstup je pripojený ku vstupu B-registra (104), ktorého výstup je pripojený ku vstupu (B) bitového paralelného procesorového prvku (101). Jeho výstup je pripojený ku vstupu A-registra (105), ktorého výstup je pripojený ku vstupu Q-registra (106), ktorého výstup je pripojený ku vstupu (A) bitového paralelného procesorového prvku (101), ku vstupu B-registra (104) a ku vstupu miestnej pamäti (102). Každá procesorová piketová jednotka (100) môže ďalej obsahovať zbernicu (103) vysielaných dát/adries, pripojenú k A-registru (105) a k miestnej pamäti (102), ďalej vľavo a vpravo prenášajúcu zbernicu pripojenú k B-registru (104), a ďalej riadiaci/stavový register (107) pripojený k procesorovému prvku (101).

PV 3440-91

PRO VYNÁLEZY A OBJEVY	URAD PATENTU ČESKOSLOVENSKA	13. XI. 91	052608
--------------------------	-----------------------------------	------------	--------

PARALELNÍ ASSOCIATIVNÍ PROCESOROVÝ SYSTÉM

Oblast techniky

Vynález se týká paralelního procesního systému obsahujícího množství piketových jednotek, kde každá piketová jednotka má bitový paralelní procesní prvek kombinovaný s místní pamětí připojenou ke procesnímu prvku pro paralelní zpracování informace ve všech piketových jednotkách asociativním způsobem, kde každá piketová jednotka je uzpůsobena ke provádění jednoho prvku asociativního procesu.

Přehled souvisejících přihlášek vynálezu

Související přihlášky vynálezu podané přihlašovatelem předkládané přihlášky vynálezu jsou tyto:

Přihláška vynálezu Spojených států amerických 07/250,595 podaná 27.zářím 1988, nyní opoštěna ve prospěch její pokračovací přihlášky seriového čísla 07/519,332 podané 4.května 1990, přihlašovatel James L. Taylor, název "SIMD Array Processor" (tato přihláška vynálezu byla původně zveřejněna jako přihláška EPO seriového čísla 86307855/88-A 3.května 1989, přihláška vynálezu Spojených států amerických 07/193,990 podaná 13.května 1988, přihlašovatel H. Li, název "Methods and Circuit for Implementing an Arbitrary Graph on a Polymorphic Mesh", přihláška vynálezu Spojených států amerických 07/426,140 podaná 24.října 1989, přihlašovatelé R. Jaffe a spol., název "Two-dimensional Input/Output Scheme for Massively Parallel SIMD Computers" a přihláška vynálezu Spojených států amerických 07/439,758 podaná 21.listopadu 1989, přihlašovatelé W.C. Dietrich, Jr. a spol, název "Method and Apparatus for Performing Memory Protection Operations in a Parallel Processor System".

Dosavadní stav techniky

Patentový spis Spojených států amerických vydaný 27.října 1970, autoři R.A. Stokes a spol., popisuje řadový počítač s paralelními procesory a jednou programovatelnou řídicí jednotkou, množstvím registrů pro uložení doplňkových vektorů, maskovacích registrů a prostředků uzpůsobených pro odezvu na sled jedné nebo několika instrukcí řídicí jednotky pro odpovídající operace s daty v registrech vektorů. Během roku 1970 se paralelní procesory popsané Stokesem staly známými jako

JIMD (Jedna Instrukce Množství Dat) stroje. Takovéto stroje mohou být také popsány tak, že sestávají z programovatelné řídicí jednotky ovládající řadu n paralelních procesorů, kde každý procesor má paměť, aritmetickou jednotku, dekodér programu a vstupní/výstupní jednotku. Tyto systémy představovaly velké skříně, které byly někdy sdruženy s vnějším velínem. Významný rozdíl mezi JIMD a obecnějšími procesory byl ten, že v takových systémech mohou všechny JIMD procesory mít v přidruženém procesoru rozdílnou množinu dat, avšak všechny procesory jsou ovládány společným řadičem. JIMD počítače jsou také odlišné od obecnějších procesorů Von Neumanna v tom, že každá instrukce zpracovává spíše vektor dat než jediný operand.

Nejobecnější typy mnohprocesorových systémů jsou systémy Mnoho Instrukcí Mnoho Dat, kde každý procesor může provádět zvláštní program pracující se zvláštní množinou dat. Procesory v MIMD systému mohou provádět oddělené úkoly nebo mohou provádět rozličné dílčí úkoly společného hlavního úkolu.

S pokrokem přemýšlení o JIMD paralelních procesorech se uvažovalo, jak je ukázáno v patentovém spise Spojených států amerických číslo 4,435,758, autoři Lorie a spol., datum vydání 6. března 1984, název "Method for Conditional Branch Execution in SIMD Vector Processors", že může být výhodné, když úkoly, které má systém provádět, jsou vysoce nezávislé a prosté sporů, avšak když úkoly mají spor pro zdroje, potom může být označena síť synchronních procesorů pracujících v JIMD. Patentový spis Spojených států amerických číslo 4,435,758 popsal tento problém a také popsal zlepšení, které provedl ve srovnání s obsahem patentového spisu Spojených států amerických číslo 4,101,960 ze dne 18. července 1978, autoři R.A. Stokes a spol., s ohledem na provádění podmíněných větví.

Stalo se normou popisovat nejpokrokovější JIMD stroje jako seriové procesory se synchronními bity uspořádané v matici $N \times N$, jak je ukázáno příkladem mnohprocesorového systému popsaným v pojednání autorů H. Li a M. Maresca o názvu "Polymorphic-Torus Network", Proceedings of the International Conference on Parallel Processing, str. 411-414. Metody násobení maticových vektorů pro takovou masivně paralelní architekturu popsané od H. Li a spol. pro $N \times N$ procesory propojené fyzicky smyčkovou topologií a se smyčkovou sítí překrytou jinou spínací sítí pro účely zpětné konfigurace byly podrobně popsány v časopise IBM Technical

Disclosure Bulletin, sv.32, číslo 3A srpen 1989 pro zlepšení rychlosti provádění výpočtu násobení řádkové matice s vektorem.

Existuje publikace 1990, která ukazuje, že bylo také věnováno úsilí vytvořit JIMD a JIMD/MIMD stroje, které pracují s mnoha řadami procesorů, kde všechny procesory dané řady provádějí přesně stejnou instrukci, avšak každá řada je naprogramována odlišně. Viz příklad popsany od H. Li, P. Don a R. Jaffe v IBM Technical Disclosure Bulletin, sv.32, 8B, leden 1990, kde je popsána architektura s touto konfigurací, která byla nazvána Parallel Local Operator Engine (FLOE) pro zpracování specifických opakovacích inspekčních funkcí paměti.

Tato přihláška plánuje vývoj vzduchem chlazeného systému, který může být vyráběn stávající technologií, která může vytvořit na čipu množinu s nově uspořádanou architekturou úkoly, které byly uvažovány pro JIMD stroje, jakož i úkoly, které byly uvažovány pro stroje podle Von Neumanna.

Na základě přehledu dosavadního stavu techniky se zjišťuje, že jiní popsali použití procesoru s pamětí ROM pro statické instrukce a registry pro ukládání a přístup k datům na jediném křemíkovém čipu, viz například patentový spis Spojených států amerických číslo 4,942,516 zveřejněný 17.července 1990, původce G. Hyatt, název "Single Chip Integrated Circuit Computer Architecture", tento druh práce však neměl vztah ke komplexním aplikacím JIMD typu.

Jiní autoři popsali rozličné prostředky pro různé úkoly, na které má být náš systém také zaměřen. Je známo, že násobení matic může být prováděno paralelně. Jsou známy publikace, které se týkají aplikací ke kterým může náš systém být použit, které souvisí s umělou inteligencí. Obsahově adresovatelné (viz obr.3) nebo asociativní paměti byly adresovány na univerzální úrovni s rozličnými procesními čipy. Bylo zde zjištěno, že pro AI aplikace je v některých případech je vhodné založit volbu řádku na výsledcích předešlých vyhledávacích operací, se řádkovou logikou. Viz VLSI for Artificial Intelligence, Jose G. Delgado-Frias a Will R. Moore, vydavatelství Kluwer Academic Publishers, 1989, str.95-108, "VLSI and Rule-Based Systems", Peter Kogge, Mark Brule a Charles Stormon, pro diskuzi aplikací. Nicméně, elegantní návrhy jiných autorů zdánlivě směřují obor jiným směrem, než činí původci vynálezu. Jeden způsob spočívá

na oxfordských čípech "Intelligent Memory" pro masivní paralelismus, které byly popsány v publikaci VLSI Systems Design, December 1988, str.18-28 v pojednání, jehož autorem je Senior Editor Bob Cushman, a které má název "Matrix Crunching with Massive Parallelism. Jiní autoři se snažili o paralelní provádění procesů na asociativní paměti VLSI za účelem popsání čipu asociativní paměti VLSI vhodného pro implementaci přestavitelného, úplně paralelního asociativního schematu provádění procesů. Tento přístup spočíval na úvaze, že použití klasické architektury asociativní paměti by vyžadovalo nepřiměřený počet výstupů pro zavedené přenosy dat. Viz například pojednání "Parallel Processing on VLSI Associative Memory", autorů S. Hengen a I. Scherson, podpořené NSF cenou ECS-h404627 a přednesené autory na Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106.

Problém, se kterým jsme se setkali, je potřeba vytvořit kompaktní procesory pro komplexní aplikace a při bližším zkoumání tohoto problému jsme zjistili, že dřívější návrhy, které byly omezeny na bitové seriové implementace s nejvýše několika málo tisíci bitů paměti na jeden procesní prvek a s malým počtem procesních prvků na jeden elektronický čip byly nedostačující. Jako výsledek jsme dospěli k potřebě vymezit architekturu, která by mohla být vyráběna s dramaticky vyššími hustotami a přesto byla vhodná pro prostředí chlazené vzduchem a mohla být balena v lehké a kompaktní jednotce.

Jako výsledek jsme přišli na řešení, které představuje N rozměrné pole procesních jednotek bytové šíře, z nichž každá je spojena s odpovídajícím segmentem paměti bytové šíře a se řídicí logikou. Rozdělitelná sekce pole obsahující několik procesních jednotek je obsažena na křemíkovém čipu. To bude podrobně popsáno dále. Naše vlastní řešení způsobilo přídavné problémy k řešení za účelem zlepšení chování systému a my jsme vynalezli řešení za účelem dalšího vyvíjení navržené architektury. Některá z těchto řešení umožňují nezávislé chování procesních prvků. V této souvislosti při zpětném pohledu jsme zjistili, že dosažený stav techniky popsany v evropské přihlášce vynálezu EP-A-208,457 popisoval řadu procesorů, ve které každý procesní prvek v řadě měl možnost zvolit prvek, ze kterého dostává svůj vstupní signál.

Závislá přihláška vynálezu 07/519,332 podaná 4.května 1990, přihlašovatel James L. Taylor, pokračující k číslu 07/250,595

podanému 27.září 1988 s názvu "SIMD Array Processor" popisuje mnohem větší pole procesních prvků, které má zvýšený stupeň flexibility k umožnění lepšího využití potenciálu pro paralelní provádění procesů bez úsazení nákladů a složitosti SIMD procesoru. Tato aplikace byla původně poprvé zveřejněna 3. května 1989 jako evropská přihláška vynálezu EPO 88/307885/88-1.

Zde popisovaný systém vysílá globální instrukci pro místní bitové seriové provádění podél sběrnice spojující řídící logiku v rozličných paralelních procesních prvcích a programově přeměňuje zvolené bity globální instrukce pro použití v místních bitových linkách, kde se přeměněné bity dekódují.

Zpětný pohled také identifikoval jednu diskuzi o SIMD procesorových prvcích bytové šíře o názvu "Design of SIMD Microprocessor Array", jejíž autoři jsou C.R. Jesshope, R. O'Gorman a spol., a která byla zveřejněna v IEEE Proceedings sv.136, květen 1989. Toto pojednání popisuje školní studii SIMD architektury autory, kteří jsou z Departement of Electronics and Computer Science na The University, Southampton, UK. Mnohé výtvoary jsou podobné těm, které diskutujeme, a popisují procesor, který má SIMD architekturu bytové šíře. Pojednání navrhuje procesní prvky s prvky a architekturou osmibitového procesního akumulátoru, omezenou RAM paměť na kartě (1kbyt) a množství PE na čipu (4) jakož i určitý významný stupeň místní autonomie. Nicméně, při použití takové navrhované struktury není zajištěno asociativní provádění procesů. Bude ukázáno, že navrhovaná struktura nezajišťuje sousední komunikaci bytové šíře, jako tomu je v případě předloženého vynálezu. Jiné výtvoary, které jsme vyvinuli, také nejsou navrženy. Tudíž zde existuje potřeba našeho piketového systému, který je proveditelný na čipu integrovaného obvodu za účelem pokroku ve stavu techniky paralelního provádění procesů. Viz pojednání "Summary of Parallel Computer Architectures" jehož autorem je R. Duncan a které je v časopise IEEE Computer v únoru 1990.

Definice

Ze účelem snadného postupu při popisu předloženého vynálezu zde následuje obecný, neomezující popis nové terminologie oboru. Piket - prvek procesního pole sestávající přednostně z kombinovaného procesního prvku s místní pamětí pro bitové provádění procesů s paralelními byty informace v hodinovém cyklu. Přednostní

provedení sestávající z procesoru bytové říše pro proud dat, 32 kbyťů nebo více paměti, základních řízení, má vazbu pro komunikace s jinými pikety.

Piketový čip - obsahuje množství piketů na jediném křemíkovém čipu.

Piketový procesorový systém (nebo podsystém) - celkový systém sestávající z pole piketů, komunikační sítě, I/O systému a JIMD kontroléru obsahujícího mikroprocesor, procesoru uložené rutiny a mikrokontroléru který spouští pole.

Piketová architektura - přednostní provedení JIMD architektury s výtvary, které usnadňují některé různé druhy problémů, jako:

- asociativní provádění procesů s množinami
- paralelní intenzivní provádění numerických procesů
- provádění procesů s fyzickými poli podobnými obrazům

Pole piketů - soubor piketů uspořádaných v geometrickém pořádku.

Výraz "piket" pochází od Toma Sawyera a jeho bílé ohrady ačkoliv bude také funkčně pochopeno, že analogie s vojenskou hlídkovou linií je výstižná.

Podstata vynálezu

Architektura, kterou vypracujeme, může být označena jako "piketová" architektura. Zahrnuje jisté struktury procesorových prvků podobných v určitém ohledu dřívějším JIMD procesním prvkům (PE), které spolu působí, když je žádáno, podobně jako hlídková linie, které však byly zlepšeny a jsou schopny nezávislé Von Neumannovy operace. Tato architektura byla vyvinuta pro použití N-rozměrného pole procesních jednotek uzpůsobených pro paralelní provádění procesů a majících řídicí logiku v každé procesní jednotce, kterou označujeme jako piket.

Přednostní systém je schopen počítat podobně jako JIMD proudový stroj, avšak chování bylo zlepšeno implementací více paralelních procesních prvků. Problémy vznikající závislostí dat byly vyloučeny. V operacích strojů JIMD nemůže mít žádný procesor nebo funkce v sobě žádnou datovou závislost, která by způsobila, že jeden procesní prvek by vyřadil různý počet cyklů.

Vyvíjení architektury, které jsme provedli, měly za výsledek systém s množstvím piketových jednotek, obecně piketů, majících bitový paralelní procesní prvek s místní pamětí svázanou s procesním prvkem pro paralelní zpracování informace asociativním způsobem, kde každý piket je uzpůsoben provádět jeden prvek

asociačního procesu. Vytvořili jsme cestu pro vodorovnou asociaci s každým piketem. Paměť piketových jednotek je uspořádána v poli. Pole piketů tudíž takto uspořádané obsahuje množinovou asociativní paměť.

Vynález množinového asociativního paralelního procesního systému na jediném čipu umožňuje vyjmout menší množinu "dat" z větší množiny z paměti, kde na ní může být provedena asociativní operace. Tato asociativní operace, kterou je typicky nějaké exaktní srovnání, se provádí na celé množině dat paralelně použitím piketové paměti a prováděcí jednotky.

V piketovém poli má každý piket část dat z větší množiny. Přídavně každý piket zvolí jeden kus dat z jejích částí. Jeden kus dat v každém z množiny piketů takto obsahuje množinu dat, na které se provádí všemi pikety paralelně asociativní operace.

Návrh, který může být v současnosti implementován až do 50 kgate proudu dat a řídicí logiky a s množstvím megabitů paměti DRAM na jediném čipu. Každý z těchto čipů má konfiguraci takovou, že obsahuje množství piketů nebo procesních jednotek. Ve přednostním provedení pro zpracování textu, které je schopné užívat grafiky, je 16 piketů se 32 kbyty paměti DRAM pro každý piket na jediném piketovém čipu a systém obsahuje řadu 64 takových piketových čipů, poskytující řadu 1024 procesních prvků. Tato piketová architektura může být vyráběna technologií CMOS, která umožňuje uložit 4 miliony bitů DRAM v opakovaných vzorech na jediném čipu a zbývající povrch čipu je vyplněn logickými standartními buňkami, které budou používat až 50 k logických prvků a které mohou vytvářet proud dat a řídicí logiku pro vytváření piketů na čipu. Uspořádali jsme systém tak, že pikety mohou zpracovávat data s místní autonomií a vytvořili jsme "skluz" mezi pikety.

Technologie piketů je schopna expanze a se 128 kbyty DRAM v každém jednotlivém piketu (čip se 16 Mbyty paměti DRAM) architektura piketu obsáhne úplnou 24-bitovou barevnou grafiku stejným způsobem jako text a 8-bitové barevné nebo šedá stupnicové grafika jsou prováděny s naším obvyklým přednostním provedením. Pokusná výrobní technologie ukazuje, že tuto hustotu je možno pro blízkou budoucnost předvídat jako konsistentní výrobek schopný pracovat v prostředí chlazeném vzduchem. Pro barevnou grafiku by naše přednostní architektura piketů měla zvýšit množství DRAM

na čipu na 128 kbytu pro jeden piket při zachování 16 piketů na jednom čipu. Alternativně by pro procesory plně barevné grafiky mohlo být použito 24 piketových jednotek na jednom piketovém čipu s 96 kbytu paměti.

Přehled obrázků na výkresech

Přednostní provedení předloženého vynálezu je znázorněno na výkresech, kde obr.1 je schematické znázornění dosavadního JIMD procesoru, který je popsán v závislé přihlášce vynálezu, která může být považována za representativní objekt dosavadního stavu techniky, obr.2 znázorňuje pár základních piketových jednotek, které jsou uspořádány na křemíkové bazi s procesorem, pamětí, řídicí logikou a asociativní pamětí s bytovou komunikací s jinými pikety v poli, obr.3 znázorňuje asociativní provádění procesu v paměti, obr.4 znázorňuje základní 16 (n) uspořádání piketů pro podsystém JIMD, který používá mikroprocesorový kontrolér, pevně zadrátovaný sekvenční kontrolér pro uložené rutiny a pole piketů a které tvoří základní paralelní piketový procesorový systém, který může být také oddělenou jednotkou, obr.5 znázorňuje mnohonásobný piketový procesorový systém, který zahrnuje množství piketových procesorů z obr.4, obr.6 je funkční blokový diagram podsystému, obr.7 znázorňuje uspořádání kontroléru podsystému s kartami podle obr.5.

Příklady provedení vynálezu

Obr.1 znázorňuje typický JIMD systém podle dosavadního stavu techniky typu popsaného obecně v evropské přihlášce vynálezu Taylora a IBM číslo 88307855/88-A a v britské přihlášce vynálezu UK-A-1,445,714. V takových zařízeních podle dosavadního stavu techniky je JIMD počítač počítač s jednou instrukcí a množstvím dat mající paralelní řadový procesor obsahující množství paralelně spojených bitových seriových procesorů, z nichž každý je sdružen s jednou z množství pamětí JIMD. Vstupní/výstupní systém působí jako etapový systém pro jednotku JIMD a obsahuje dočasnou ukládací oblast pro dvousměrný dvourozměrný přenos dat mezi vnějším počítačem (což může být hlavní panel nebo mikroprocesor) a JIMD počítačem. Vstupní/výstupní systém obsahuje vstupní a výstupní procesní prostředky pro řízení proudu dat mezi vnějším počítačem a dočasnou ukládací oblastí a množstvím pamětí JIMD, což obvykle jsou organizované pufrové

sekcce nebo části větší paměti. Vstupní operace systému vstup/výstup tudíž způsobí přenos dat z paměti vnějšího počítače do dočasné ukládací oblasti a ve druhém kroku do JIMD paměti a pro výstup je zde také proces o dvou krocích, který přenáší přes dvourozměrnou sběrnici data mezi vnějším počítačem a JIMD počítačem. Vstupní/výstupní systém pro vstup/výstup dat může být oddělená jednotka, podjednotka vnějšího počítače, nebo často je to jednotka uvnitř JIMD počítače, kde JIMD kontrolér působí jako řídicí orgán pro dočasné ukládání v pufru při vstupu/výstupu dat.

Samotný JIMD počítač obsahuje řadu procesorů mající množství procesních prvků a sít, která propojuje jednotlivé procesní prvky a množství obvyklých oddělených JIMD pamětí. JIMD počítač je paralelní řadový procesor mající velký počet jednotlivých procesních prvků spojených a pracujících paralelně. JIMD počítač zahrnuje řídicí jednotku, které vyvíjí proud instrukcí pro procesní prvky a také vytváří potřebné časovací signály pro počítač. Sít, která propojuje rozličné procesní prvky obsahuje určitou formu propojovacího schéma pro jednotlivé procesní prvky a toto propojení může být provedeno rozličnými topologiemi, jako je smyčková, polymorfni torus nebo hyperkubická. Množství pamětí je pro bezprostřední ukládání bitových dat pro jednotlivé procesní prvky a je zde korespondence jedna-na-jednu mezi počtem procesních prvků a počtem pamětí, které mohou být výše uvedenou pufrovou částí větší paměti.

Například, jak je znázorněno v obr.1, je zde vnější procesor 28. Tento vnější procesor 28 je použit pro ukládání programů v mikrokódu do řádkového kontroléru 14 (který obsahuje pufr pro dočasné uložení) pro výměnu dat s ním a pro záznam jeho stavu přes datovou sběrnici 30 vnějšího kontroléru adresovou a řídicí sběrnici 31. Vnější procesor 28 v tomto příkladu by mohl být jakýkoli vhodný universální počítač, jako vnější panel nebo osobní počítač. V tomto příkladu podle dosavadního stavu techniky je řada procesorů řady znázorněna jako na 2-D bazi, avšak řada může být organizována i odlišně, například na 3-D nebo 4-D skupinovém uspořádání. JIMD řadový procesor obsahuje pole 12 procesních prvků $P(i, j)$ a řádkový kontrolér 14 pro vydávání proudu globálních instrukcí procesním prvkům $P(i, j)$. Ačkoliv to v obr.1 není znázorněno, příklad podle dosavadního stavu techniky má procesní prvky, které pracují na jednom bitu v jednom čase a má sdružený blok ukládání, který je částí paměti sdružené

s procesním prvkem. Procesní prvky jsou propojeny tak zvanou SVZJ (Sever, Východ, Západ, Jih) sítí s jejich příslušnými sousedy dvousměrnými bitovými linkami. Procesní prvek $P(i, j)$ je takto spojen s procesními prvky $P(i-1, j)$, $P(i, j+1)$, $P(i, j-1)$ a $P(i+1, j)$ ve směrech Sever, Východ, Západ a Jih. V tomto typickém příkladu je síť SVZJ u svých okrajů propojena toroidálně, takže severní a jižní okraje jsou propojeny dvousměrně a západní a východní okraje jsou propojeny podobně. Aby data mohla vstupovat do a vystupovat z pole procesorů je k síti SVZJ připojena datová sběrnice 26 řádkového kontroleru. Jak je znázorněno, je připojena k východo-západnímu okraji pole. Místo toho nebo přidavně může být připojena k severo-jížnímu okraji dvousměrnými třístavovými ovladači, které jsou připojeny k toroidálnímu východo-západnímu SVZJ spojení. Jako v přednostním provedení, které bude popsáno, 1024 procesních prvků by mohlo být docíleno dosavadním stavem techniky kdyby počet procesních prvků v tomto příkladu byl 32×32 místo znázorněných 16×16 . Ve znázornění jedna linka označuje jednobitovou linku, zatímco dvojitá linka spojující funkční prvky je užita pro znázornění množství spojovacích linek nebo sběrnic.

V tomto příkladu podle dosavadního stavu techniky řádkový kontroler vysílá instrukce paralelně k procesním prvkům přes instrukční sběrnici 18 a vysílá volbu řádku a volbu sloupce linkami 20 volby řádku popřípadě linkami 22 volby sloupce. Tyto instrukce způsobí, že do procesních prvků jsou zavedena data ze zásoby a jsou zpracována a potom uložena ještě jednou v zásobě. Pro tento účel má každý procesní prvek přístup k bitové části (sekce nebo pufr) hlavní paměti. Logicky je proto hlavní paměť řádkového procesoru rozdělena do 1024 částí pro řadu 1024 procesních prvků. To znamená, že až třicet dvě 32-bitová slova mohou být přenášena do nebo ze zásoby v jednom čase v přenosovém kroku. K provedení operace čtení nebo psaní je paměť adresována ve výrazech indexové adresy, která se přivádí k adresovým linkám paměti přes adresovou sběrnici 24 a instrukce čtení nebo psaní se přivádí ke každému procesnímu prvků paralelně. Během operace čtení signály volby řádku a sloupce na linkách volby řádku a sloupce identifikují procesní prvek, který má provést operaci. V popisovaném příkladu je tudíž možno číst jedno 32-bitové slovo z paměti do třiceti dvou procesních prvků ve vybraném řádku když pole je 32×32 . Procesní prvek je sdružen s částí

nebo blokem paměti (i, j) široké jeden bit. Zatímco část nebo blok paměti je logicky sdružen na bázi jedna-na-jednu se sdruženým jednotlivým procesním prvkem, může být a typicky je fyzicky oddělen na jiném čipu. Nevíme jak s touto dřívější architekturou by popisovaný řadový procesor mohl být vyroben, jako může být náš piket vyroben se řadou procesorů a odpovídající pamětí na jediném čipu druhu dále popsaného.

Procesní prvky $P(i, j)$ příkladu z dosavadního stavu techniky mají být chápány tak, že obsahují ALU, se vstupními a výstupními registry, které zahrnují přenos, a každý z nich je schopen uchovat jeden bit informace. Je zde multiplexer, který je připojen k ALU vstupu, výstupu a také ke dvousměrnému datovému portu části paměti (i, j) sdružené s jednotlivým procesním prvkem $P(i, j)$.

Jsou zde zvláštní instrukční a datové sběrnice a řadový kontroler má zásobník mikrokódu, do kterého se uloží mikrokód definující proces, který má být proveden řadou procesorů, vnějším procesorem 28 při použití datové sběrnice 30 a adresové a řídicí sběrnice 31. Když byla operace řadového kontroleru začata vnějším procesorem 28, je sekvence mikrokódu řízena řídicí jednotkou mikrokódu, která je připojena k zásobníku mikrokódu uvnitř řadového kontroleru 14. ALU a sada registrů řadového kontroleru jsou použity při vyvíjení adres řadové paměti, čítání smyček, výpočtu skoků adres a registrových operací pro obecné účely, které jsou vysílány na adresovou sběrnici řadového kontroleru. Řadový kontroler také má maskovací registry pro dekódování maskovacích kódů řádku a sloupce a specifické operační kódy přicházejí ke procesním prvkům přes instrukční sběrnici. V tomto příkladu by řadový kontroler mohl mít uvnitř datový pufr, avšak funkčně mezi datovou sběrnici vnějšního kontroleru a datovou sběrnici řadového kontroleru. Z tohoto pufru jsou data zaváděna pod kontrolou mikrokódu v kontrolním zásobníku do řady procesorů a naopak. Pro tento účel je pufr uspořádán jako dvousměrný FIFO pufr pod kontrolou kontroly mikrokódu v řadovém kontroleru. Více podrobností takových systému podle dosavadního stavu techniky je možno nalézt ve výše citovaných příkladech, zejména v přihlášce vynálezu Spojených států amerických U.S.S.N. 07/519,332, přihlašovatel Taylor, název "SIMD Array Processor".

Přehlídká předchozích dřívějších pokusů může být srovnána s přednostními provedeními předloženého vynálezu. Obr.2 znázorňuje základní piketovou jednotku 100 obsahující kombinovaný

procesní prvek ALU 101 s místní pamětí 102 k němu připojenou pro zpracování jednoho bytu informace v hodinovém cyklu. Jak je znázorněno, piketová jednotka je vytvořena na čipu křemíkové baze nebo piketovém čipu s lineárním polem piketů s postranními sousedy (vlevo a vpravo v obrázku), takže na čipu křemíkové baze je vytvořeno piketové procesní pole s množstvím místních pamětí, jednou pro každý proud procesních dat bytové šíře, uspořádaných v logickém řádku nebo v lineární řadě se sousedními komunikačními sběrnicemi pro průchod dat dvousměrně vpravo a vlevo. Soubor piketů v piketovém čipu je uspořádán v geometrickém seřazení, přednostně vodorovně na čipu. Obr.2 znázorňuje typickou implementaci dvou piketů piketové řady na piketovém čipu s množstvím pamětí a proudem dat zahrnujícím komunikační cesty mezi každým procesním prvkem piketu a pamětí. V našem přednostním provedení komunikační cesty dat mezi pamětí jedna-na-jednu s procesními prvky řady má šíři bytu a napříč, vlevo nebo vpravo se sousedem nebo se "skluzem" pro komunikaci se vzdálenějšími piketovými procesory.

"Skluz" může být definován jako prostředek pro přenos informace v jednom cyklu do nesousední polohy umístěním piketové adresy, které by normálně byla schopna přijmout informaci kdyby nebyla průhledná pro zprávu vyslanou dokud nepříjde a není přijata u nejbližšího aktivního souseda, který ji přijme. Skluz tedy působí tak, že vysílá informaci do nesousední polohy přes "vyřazené" pikety. Nechť piket "A" má vyslat informaci do vzdáleného piketu "G". Před tímto cyklem jsou mezilehlé pikety učiněny průhlednými vyřazením piketů "B" až "F". Potom v následujícím jediném cyklu "A" vyšle svou zprávu vpravo a tato projde pikety "B" až "F", které jsou průhledné, protože jsou vyřazeny a "G" přijme zprávu, protože je stále v činnosti. Při normálním použití informace "skluz" je zpráva přenášena lineárně napříč struktury, skluz může však také pracovat s dvourozměrnou smyčkou nebo s mnohazměrnou řadou.

Přístup procesních prvků v našem přednostním provedení není bitově seriový v operaci, nýbrž spíše bytově seriový. Každý procesor má přístup k jeho vlastní připojené paměti spíše než aby měl přístup ke bloku místní paměti a některé přidružené její části nebo stránce. Sběrnice šíře znaku nebo násobku znaku je použita místo šíře jednoho bitu. Místo jednoho bitu se zpracovává byt (nebo v budoucích systémech s uvažovaným

dvojnásobným výkonem znakového bytu několik bytů) informace v jednom hodinovém cyklu. Mezi každým procesním prvkem piketu může takto protékat 8, 16 nebo 32 bitů k vyrovnání šíře přidružené paměti. V našem přednostním provedení má každý piketový čip 8 (9) bitů šíře paměti ze 32 Kbytů a přednostně 16 piketů s těmito 32 Kbyty zásoby na piketový uzel lineární řady. V našem přednostním provedení je každá přidružená paměť vytvořena v CMOS jako DRAM a znakový byte je 9 bitů (které fungují jako osmibitový znak s vlastní zkouškou).

Paralelní cesta bytové šíře proudu dat sběrnice mezi pikety a mezi procesním prvkem a jeho pamětí je podstatné zlepšení oproti seriové bitové struktuře systémů dosavadního stavu techniky avšak je třeba uvážit, že zvýšený paralelismus má za následek přidavné problémy, které vyžadují řešení když se porozumí implikacím nově dosažené architektury. Významná řešení jsou zde popsána.

Vlastnost, která bude hodnocena, je, že přidavně ke přenosům vlevo a vpravo na sousední místo a mechanismu skluzu, který byl popsán s přihlédnutím k výkresům, jsme vytvořili také vysílací sběrnici šíře dvou byte, takže všechny pikety mohou vidět totáž data ve stejném čase. Řízení piketu a propagace adresy se také přenáší na tuto vysílací sběrnici. Tato sběrnice dodává srovnávací data když se provádějí asociační operace s množinami a jiné srovnávací nebo matematické operace.

Úlohy které mají vysoce paralelní datové struktury, které jim dávají možnost zpracování v piketových procesních prvcích dat při řízení proudu jedné instrukce nacházejí použití ve skládání vzorů umělé inteligence, spojování čidla a stopy v optimálním určování s více čidly, zkoumání kontextu a zpracování obrazu. Nicméně, mnohé z těchto použití nyní možných nebylo užito v JIMD procesech s ohledem na seriové bitové provádění procesů pod jedním časem hodin. Tak například tradiční seriový procesní prvek JIMD stroje provádí jeden bit operace ADD pro každý cyklus procesoru, zatímco 32-bitový paralelní stroj může provést 32 bitů operace ADD v jednom cyklu.

Uspořádání 32 KB na procesní prvek přivede mnohem více logicky dosažitelné paměti ke každému procesnímu prvku než tradiční JIMD stroj.

Počet kolíků na čipu byl udržen nízký protože data, které jdou do a ze čipu byla udržena minimální. Paměť DRAM je obvyklá

řadová paměť CMOS a jedna která má přístup "řádek-sloupec" potlačením sloupcové demultiplexace na zadní straně paměťové řady a vytvořením adresy řádku, která čte řádek paměťové řady ke proudění dat paralelně.

Přídavně k datům paměť obsahuje "tri-bits" nebo "trit", takže zde jsou tři stavy rozeznávané logikou místo tradičního binárního digit, tedy buď logická 1, logická 0 nebo neznám. Tento neznám v překrytém poli kryje buď logickou 1 nebo logickou 0. Trit je obsažen v postupných rístech zásoby v zásobní řadě. Masky jsou jiná forma dat uložených v paměti, které jsou směřována na maskový registr piketového procesního prvku.

Jelikož řada zásoby může obsahovat příkazy, umožňuje to, že jeden piket může provádět jinou operaci než jiný piket. Čipová kontrola jednotlivých piketů během operací zahrnujících mnoho piketů, ne však nutně všechny, umožňuje implementace, které jsou možné pouze pro JMD operace. Jedna vytvořená jednoduchá kontrolní funkce spočívá v tom, že jsou závislé operace v kterémkoli piketu, jehož stavový výstup splňuje nějakou specifickou podmínku. tak nenulová podmínka může znamenat klid. Klid je podmínka, která zastavuje operace a uvádí piket do neaktivního, avšak vědomého stavu. Jiný vytvořený příkaz je zamezení nebo dovolení zápisu do paměti spočívající na podmínkách v piketu nebo na příkazu daného sběrnici před operací skluzu.

Použitím piketového čipu se 16 výkonnými pikety, z nichž každý má 32 KB paměti, pouze 64 čipy vytvoří 1024 procesorů a paměť 32768 KB. Řada piketů obsahuje množinovou asociativní paměť. Vynález je také užitečný pro numerické intenzivní zpracování analyzy obrazu jakož i provádění procesů s vektory. Tato výkonná piketová procesní řada může být dnes balena pouze na dvou malých kartách! Buďte ukázáno, že tisíce piketů mohou být výhodně zabaleno v obalu s nízkým výkonem a lépe přenosném, což umožňuje provádění aplikací zpracování obrazu s minimálním zpožděním nebo v rozsahu času jednoho videosnímku, například během letu dopravního letadla bez významných platebních zatížení.

Výkonnost piketů vede na možnost použití velkých přidružených paměťových systémů zabaleno do smětaných pouzder a umožňuje použití procesní výkonnosti v řadě aplikací když projektant systému si navykne používat nový systém.

Obr.3 znázorňuje strukturu, které může být označena jako úplně asociativní paměť tehdy, když se asociace řádá, srovnávané

hodnota je předána do všech míst paměti a všechna místa paměti současně odpovídají se svými stykovými linkami. Asociativní paměť je v oboru o sobě známá. V systému zde popisovaném je při použití paralelních piketů paměti a procesních prvků, které mají bytové přenosy za účelem provádění průzkumu, vstup dat a maska pro průzkum za účelem zjištění místa slova K mezi N slovy v paměti. Všechny stýkající se pikety zdvihnou stavovou linku a potom oddělená operace čte nebo vybere první styk K. Tato operace, obecně zvaná asociativní množina, může být opakována nahoru přes piketovou paměť. Podobně je dosaženo zápisu přes vysílací operaci, ve které zdvižená vybraná linka označuje participaci a vysílaná data jsou kopírována do všech vybraných piketů.

Jiné provedení, ačkoliv není jediné přednostní, snižuje velikost paměti DRAM dosažitelné pro každý piket pro umožnění inkluze některé sekce plně asociativní paměti typu znázorněného v obr.3. Kdyby bylo zahrnuto 512 bytů úplně asociativní paměti, každý piket by obsahoval množinu průzkumných indexů a v jedné operaci 512 krát 1024 piketů dává 512k srovnání na operaci nebo 512 Gig-srovnání za sekundu při jedné mikrosekundě pro operaci. S roztažitelností může být koncept posunut do rozsahu mnoha Tera-srovnání. Toto provedení zesiluje asociativní úlohy, které zavádějí extensivní průzkum informace s možnostmi vysoce přesahujícími současné výpočetní možnosti.

Když tato asociativní operace užívající paměť a v šíři bytu spojené procesní prvky, jak je znázorněno v obr.2, přidavně k použitím odlišných algoritmů nebo operací, umělé inteligenci a paralelnímu programování byla zkoušena v JIMD situacích, jsou zde mnohé přidavné aplikace nyní použitelné pro stroj s uspořádáním čipu, které popisujeme, v prostředí JIMD, které může obsahovat: Jednoduché paralelizovatelné aritmetické úlohy včetně násobení matic a jiné úlohy, které mohou být prováděny ve strojích se specializovanou pamětí;

Úlohy krytí obrazů a zpracování obrazů, které mohou být prováděny ve Von Neumannových strojích, avšak které mohou být značně zrychleny s aplikacemi přizpůsobitelnými krajnímu paralelismu, například krytí vzorů trojrozměrného obrazu;

Dotazové funkce podložené daty;

Krytí vzorů v oblasti umělé inteligence;

Kontrola sítí v mřístcích pro rychlou identifikaci zpráv, které jdou k uživateli na druhé straně mřístku sítě;

Simulace úrovně hradel;

a kontroly porušení základních pravidel VLSI.

Procesní úlohy, které využívají výhody banky pamětí a přidružených procesních prvků se objeví aplikačním programátorům ať najdou jak získat výhodu výkonnosti nové systémové architektury.

Proces udržení popisu digitálního systému může být usnadněn použitím řady k jednomu hradlu nebo logickému prvku na jeden piket 100. V takovém systému by proces mohl začít označením každého popisu hradla jako výčtu signálů, které hradlo přijímá jako vstupy a pojmenováním signálu, který vyvíjí. Požaduje se změna signálu v každém čase, jeho jméno se vysílá na sběrnici 103 ke všem piketům a je srovnáváno paralelně se jmény očekávaných vstupních signálů. Když se nalezne krytí, zaznamená se v piketu nová hodnota signálu v bitu registru proudu dat. Když byly zaznamenány všechny změny signálu, způsobí se ve všech piketech paralelní čtení kontrolního slova, které sdělí jejich proudu dat, jak se má využít běžná množina vstupů pro výpočet výstupu. Způsobí se, aby tyto výpočty byly provedeny paralelně, s výsledky srovnány se starými hodnotami z místního hradla. Zaznamenají se ve stevovém bitu proudu dat všechny hradla piketů, jejichž výstupy se mění. Způsobí se, aby vnější kontroler přezkoušel všechny pikety a vyžádá se následující hradlo, které se změnilo. Potom se vyšle odpovídající jméno signálu a hodnota z piketu do všech ostatních piketů, jak byly původně určeny, a cyklus se opakuje tak dlouho, až nenastávají žádné změny signálu nebo se proces zastaví.

Jiný proces je vyhledání jména slovníku. Jména jsou uschována v piketové paměti 102 tak, že první písmeno všech jmen může být srovnáno s prvním písmenem žádaného vysílaného jména na adresové sběrnici 103 vysílaných dat. Všechny pikety prosté krytí jsou vyřazeny kontrolní vlastností, které použijeme. potom se srovná druhé písmeno a srovnávací a vyřazovací procedura se opakuje pro po sobě následující písmena (znaky) tak dlouho, až nezůstává žádné piketové jednotka aktivní nebo byl dosažen konec slova. V tomto okamžiku se všechny zbývající piketové jednotky podrobí dotazu a index žádaných dat se přečte sekvenčním obvodem.

Obr.4 znázorňuje základní uspořádání piketů množství paralelních procesorů a pamětí, piketových jednotek uspořádaných v řadě na jediném křemíkovém čipu jako část paralelní řady,

které může být uspořádáno jako JMD podsystém, znázorňující řídicí seskupení takového systému. Zde je také znázorněn řídicí procesor a mikroprocesor supervisoru. V obr.4 je znázorněna paměť a logika paralelního procesního prvku na stejném čipu, které jsou v obr.4 znázorněny v sekci označené Array of Pickets. Každá paměť je v šíři n bitů, přednostně v šíři znaku, 8 (9) bitů, jak bylo uvedeno, avšak koncepčně také může mít šíři slova několika byte. Paměťová část paralelního procesního prvku piketu bude mít přednostně šíři 8 (9) bitů nebo alternativně 16 nebo 32 bitů. Při běžné CMOS technologii dáváme přednost použití asociativní paměti šíře 8 bitů nebo šíře znaku (šíře 9 bitů byte s vlastní kontrolou) pro každý piketový procesní prvek. Paměti jsou přímo sdruženy jedna-na-jednu se spojeným procesním prvkem, který obsahuje ALU, maskovací registry (A & Q užité pro maskování), a zámek 104 (SP v obr.4) jakož i stavové registry 107 a registry A 105 a Q 106 proudu dat (DF v obr.4), které jsou podrobněji znázorněny v diagramu piketu v obr.2. DRAM a logika každého piketového procesoru nemají žádnou zátěž nějaké propojovací sítě, nýbrž je zde přímé jedna-na-jednu přidružení mezi pamětí DRAM mnohabitové šíře a jejím procesním prvkem na samotném čipu. K obr.4 je třeba uvést, že zámek (SR) 104 registru skluzu B je umístěn logicky mezi pamětí a přidruženou logikou ALU procesního prvku a zámek je v podstatě vazební port pro každý procesní prvek podél piketové řady. Každý piketový čip obsahuje množství paralelních piketových procesních prvků uspořádaných v lince (která je znázorněna jako přímá sběrnice) pro komunikaci s piketovým řízením. Sběrnice adres vektorů je společná k paměti a adresový registr vektoru dat kontroluje jaká data procházejí přes každou paměť.

Obr.4 také znázorňuje propojení mezi hlavní nebo mikroprocesorovou kartou MP, která v našem přednostním provedení je mikroprocesor 386 uspořádaný jako PS/2 systém s kontrolerem podsystému, kterým procházejí globální instrukce do procesoru uložených rutin CRP, který jsme opatřili, který opatřuje instrukce pro sekvenční obvod 402 instrukcí a řízení 403 provádění, které provede specifický mikrokód vyvolaný sekvenčním obvodem 402 instrukcí. Tento sekvenční obvod 402 instrukcí by mohl být ve funkci přirovnán ke kontroleru. Nicméně také v procesoru CRP uložených rutin jsme zavedli místní registry 405, které spolu s (neznázorněnými) lokálními registry ALU zajišťují bazi pro

všechno adresování, které se vysílá do všech piketů v řadě 406 piketů. Takto se výpočty adres provádějí pro všechny pikety v jednom ALU bez použití piketových zdrojů neby myslitelně bez použití piketových prováděcích cyklů. Tento významný přídavek přidává piketové řadě pružnost řízení tím, že umožňuje klid, potlačení a jiné řídicí funkce pro specializované úkoly, které mají být prováděny, a umožňuje oddělení piketů od jakékoli vysílané instrukce nebo datové funkce.

Sekvenční obvod 402 se zavedeným mikrokódem 407 vysílá do řady piketů pro provádění pod JIMD sled instrukcí určený hlavním programovým mikroprocesorem MP a uloženými rutinami procesoru CRP uložených rutin knihovnu 408 doby běhu k umožnění provádění procesů JIMD s daty obsaženými v řadě piketů.

Instrukce dodané mikroprocesoru MP přes rozhraní podsystému jsou vytvořeny jako procesní příkazy vysoké úrovně, které mohou obsahovat Start Procesu, Zápis Pozor. a Čti Výsledek, které se zavedou do mikroprocesoru kontrolerem podsystému mikroprocesoru MP. Mikroprocesor může být uvažován jako hlavní systém nebo řídicí procesor v podsystému v uspořádání znázorněném v obr.4, 5, 6 a 7. Je pochopitelné, že tato jednotka by mohla být oddělenou jednotkou s přidáním (neznázorněného) periferního vstupního zařízení jako klávesnice a displeje. V tomto odděleném uspořádání by systém MP mohl být uvažován jako komerční PS/2, ke kterému se karty obsahující kartu se sekvenčním obvodem (které obsahuje procesor uložených rutin) a karty řady procesorů jsou vloženy podél linek znázorněných v obr.7. Knihovna 411 rutin může obsahovat sledy rutin pro celkové řízení procesu, jako CALL (,); Kalman, Convolve, a Nav. Update. Výběr těchto rutin se děje přes uživatelský program a tedy celkové provádění procesů může být řízeno vnějším počítačem nebo uživatelským programem 412 uloženým v MP. V paměti MP je uložen datový pufr 413 pro přenos dat do a z paralelního systému piketových procesorů. Sekvenční obvod 402 instrukcí je uspořádán, aby prováděl řídicí proud z MP jakož i uložené rutiny, které jsou v paměti 408 knihovny časů běhu uložených rutin. Některé z těchto rutin zahrnují CALL(,), Load Block, Sin, Cos, Find, Min, Range Comp. a Matrix Multiply z uložených rutin knihovny 408 časů běhu uložených rutin.

Uvnitř CRP je také mikrokód 407 pro provádění řízení funkcí nižší úrovně jako Load, Read, Add, Multiply, a funkce krytí.

Dáváme přednost a zajišťujeme vnější řízení FOR/UNT pro každou procesní jednotku. Zajišťujeme také normalizovanou implementaci bytu deterministické pohyblivé řádové čárky.

Použití deterministického přístupu k vývoji makro pro systém, který jsme zajistili, dovoluje uspořádat pikety do skupin a skupinovou kontrolu. Místní funkce klidu je zajištěna pro umožnění změny provádění procesů jednotlivých piketů.

Kdyby uživatelský program vyžadoval provádění řadou procesorů jsou pro řadu piketových procesorů zajištěny primitivní příkazy, adresy a vysílané data.

Zvláštní funkce, kterou využívá každá část systému je určena úlohou, která má být prováděna, a označena během překladu uživatelského programu.

Fružnost podsystemu může být osvětlena spíše obecným problémem. Vezmeme například problém násobení matic... $[x] \times [y] = [z]$.

Problém může být zapsán takto:

$$\begin{array}{ccccccc}
 & x_1 & x_{R+1} & \dots & & y_1 & y_{M+1} \\
 & \vdots & \vdots & & & \vdots & \vdots \\
 R & \cdot & \cdot & & X & M & \cdot & \cdot \\
 & \vdots & \vdots & & & \vdots & \vdots \\
 & x_R & x_{2R} & \dots & x_{RM} & & y_M & y_{M+1} \dots y_{MC} \\
 & & & & & & C & \\
 & & & & & & z_1 & z_{R+1} \dots \\
 & & & & & & \vdots & \vdots \\
 & & & & & & \vdots & \vdots \\
 & & & & & & z_R & z_{2R} \dots z_{R+C}
 \end{array}$$

Který by byl řešen následujícím údajem vedle kterého je osvětlen počet průchodů a počet hodinových cyklů na jeden průchod je podán prostřednictvím příkladu.

		průchodů	cyklů/ průchod
01	Call Matrix Mult Fx (R,M,C,Xaddr, Yaddr, Zaddr)	1	c
02	xSUB = ySUB = zSUB = 1	1	3
03	DO I = 1 to C	1	3
04	DO J = 1 to R	C	3
05	z = 0	CxR	5/6x
06	DO K = 1 to M	CxR	3
07	***Převeď na asociativní paralelní procesor***		
08	Zz = Xx x Yy + Zz	CxRxM	204/345x
09	***Vrať výsledek***		
10	xSUB = xSUB + R	CxRxM	2
11	ySUB = ySUB + 1	CxRxM	2
12	NEXT K	CxRxM	3

		průchod	cyklů/ průchod
13	$xSUB = xSUB - xR + 1$	CxR	2
14	$ySUB = ySUB - M$	CxR	2
15	$zSUB = zSUB - 1$	CxR	2
16	NEXT J	CxR	3
17	$xSUB = 1$	C	2
18	NEXT z	C	3
19	END Cell	1	1

Poznámka * Fevná čárka (4 byte)/ Pohyblivá čárka (1 + 4 Byte)

Z prohlídky výše uvedeného příkladu je zřejmé, že úloha identifikovaná výše uvedeným příkazem 08 spotřebuje asi 98 % cyklického času. Je tedy převedena na JIMD organizaci paralelního piketového procesoru. Ostatní procesy potřebují pouze 2 % cyklického času a jsou udrženy v architektuře uvnitř mikroprocesoru.

Také prohlídka příkladu násobení matic svědčí pro provádění (a každý údaj po překladu by způsobil provedení ve specifickém lokačním systému) buď na MP, CRP, LR nebo řadě piketů.

Ve výše uvedeném příkladu násobení matic příkaz 01 by měl být přiřazen hlavnímu procesoru MP, zatímco příkazy 02, 05, 10, 11, 13, 14, 15 a 17 by měly být přiřazeny místnímu registru LR, a příkazy 03, 04, 06, 12, 16, 18 a 19 by měly být přiřazeny pro provádění procesorem CRP uložených rutin, s jinek čas spotřebujícím prováděním procesů s maticemi, tudíž přiřazeny pro provádění pod jednou instrukcí řadě piketů s přiřazením příkazu 08 řadě piketů.

Obr.5 znázorňuje mnohonásobný paralelní systém piketových procesorů 510, který zahrnuje množství paralelních piketových procesorů. Pro aplikace jako stopování mnohonásobných cílů, spojování čísel a dat, zpracování signálů, umělé inteligence, zpracování obrazů satelitů, sledování vzorů/úkolů, operace kódování a dekódování metodou Reed-Solomon byl námi vytvořen systém, který může být uspořádán v přednostním provedení jako JIMD systém s 1024 paralelními procesory se dvěma ke 4 SLM E kartami 511 (zde reprezentovány jako 4 karty pro jeden systém) pro každých 1024 procesorů. Jednotlivé karty 512 jsou zasouvateľné do varového montážního oddílu 513 s kluznými lištami 514 a karty jsou opatřeny zasouvacími/vysouvacími pákami 516, takže když se uzavře víko 517, uzavře účinně systém v oddílu se zásobou 32 nebo 64 Mbyte s prováděcí schopností asi 2 bilionů operací za sekundu. Systém je kompaktní a řada množství piketů je zasazena do karty 518 zadního panelu, která má v sobě logiku, která umožňuje propojení množství karet.

Procesor se 32 K byte záschy je vytvořen na čtyřech kartách SLM E a systém má hmotnost pouze asi 10 kg. Napájení je provedeno zdrojem 519, jak je znázorněno. Příkon vzduchem chlazeného procesoru takové výkonnosti je podle odhadu pouze asi 280 W. Každý JIMD systém má dva I/O porty 520 pro komunikaci kanálovým adaptérem s přidruženými hlavními panely nebo jinak se zbytkem světa. Se znázorněným vícenásobným paralelním piketovým procesorem, z nichž každý sestává ze 4 logických stránek a s použitím standartního letectvého balení a sběrnicové struktury pro připojení k nějaké vnější paměti (například sběrnice PI, TM a IEEE 488) může být procesor připojen přes I/O port k paměťové sběrnici nějakého misijního procesoru a může být uvažován jako prodloužení paměťového procesoru misijního procesoru.

Se znázorněným mnohonásobným paralelním piketovým procesorem obsahujícím 1024 paralelních procesních prvků má každý procesor 32 kbyte místní paměti a přidružená cesta k piketovému paralelnímu procesoru je paralelní šíře 8 bitů nebo znaku (9 bitů).

Procesory uvnitř každého piketu si vyměňují data s jinými sousedními procesory a mezi stránkami přes zadní propojovací síť, přednostně křížový přepínač, alternativně skluzový křížový přepínač, přihrádkovou síť, bazi 3 N-krychle nebo bazi 8 N-krychle.

Jednotlivé piketové procesory systému jsou obsaženy uvnitř svazku dvou karet ze čtyř karet, a mikroprocesor PS/2 na jedné kartě, zatímco sekvenční obvod procesoru uložených rutin je obsažen na jiné z těch čtyř karet, které obsahují systém schematicky znázorněný na obr. 5, 6 a 7. Jednotlivé pikety 100 karet 512 piketů mohou být uspořádány ve společném útvaru s procesorem CRP uložených rutin pro vstup do nebo výstup z operací založených na podmínkách dat a řízených architekturou zámku 104 a místními registry 405, které jsou připojeny k řízení provádění procesů karty CRP 703 sekvenčního obvodu. Takto mohou piketové procesory provádět nezávisle srovnávací a normalizační operace, které jsou přidruženy k operacím v pohyblivé řádové čárce.

Procesory jsou řízeny paralelně společným sekvenčním obvodem, jak je zde popsáno. Karta 703 sekvenčního obvodu obsahuje kontroler CRP piketového procesoru a může způsobit, že piketový procesor provede jedinou posloupnost operací kódovanou pro provedení na řadě JIMD procesorů v bytové sekvenci podobné klasickému bitovému seriovému provádění procesů. Kontroler má tři vrstvy. Mikro-kontrola pro pikety je v mikroódu podobně jako to je

u moderních procesorů, a je přenesen paralelně do všech piketů. Mikro-kontrola a pikety jsou synchronizovány na stejný hodinový systém CLK, takže funkce řízené sekvenčním obvodem mohou být prováděny ve stejném hodinovém čase. Zaváděcí příkazy do sekvenčního obvodu mikro-kontroly jsou funkcí procesoru uložených rutin. Karta 703 sekvenčního obvodu je pevně zadrátovaný kontroler, který při většině funkcí provádí smyčkové řídicí příkazy a rekurzivně startuje nové sekvence mikro-kontroly. Tento kontroler se svou knihovnou 408 uložených rutin a se svou smyčkovou funkcí udržuje pikety dobře zásobovány a nesvázány příkazy. Procesor kontroleru CRP uložených rutin obsahuje velkou sbírku makro, která jsou vyvolávána hlavním systémem, který působí v podsystemu jako primární supervizorový kontroler piketů. To je vrcholový kontrolní systém řady piketů. Je to 386 mikroprocesorů, které ovládají působnost řady piketů. V daném okamžiku mohou všechny pikety řady provádět tutéž instrukci, ačkoliv podmnožiny procesorů mohou reagovat individuálně na kontrolní proud.

Je zde více odchylek individuálních reakcí, takže místní autonomie působením bytové řídicí funkce pro každý piket (klid, potlačení atd.) je zde místní autonomie, jejíž výhodu může přivodit programování a které může být provedena pod kontrolou systému během překladu programu.

Navíc, jak bylo popsáno, je zde autonomie adresování místní paměti. JIMD kontroler sekvenčního obvodu dodává společnou adresu pro všechny užívané pikety. Každý piket může tuto adresu místně zvýšit ke zvýšení její schopnosti přístupu k datové závislé paměti.

Navíc, některý piket se může nebo nemusí podílet na působení řady v závislosti na místních podmínkách.

S těmito vlastnostmi je nyní možné zavést koncepci skupin do JIMD provádění procesů opatřením prostředků pro každý piket pro jeho přiřazení k jedné nebo více z určitých skupin a provádění procesů může probíhat v závislosti na těchto skupinách, kde změna uspořádání se může odehrávat v podstatě letmo. V jednom provedení může být pouze jedna skupina nebo kombinace skupin aktivní v jednom čase a každá provede tentýž proud JIMD instrukcí. Některé operace vyžadují práci pouze s podmnožinou nebo skupinou piketů. Programování může získat výhodu z této schopnosti. Místní participeční autonomie je určena k této práci.

Je zřejmé, že čím více piketů počítá, tím lépe.

Jedna cesta ke zvýšení počtu piketů, které se podílejí na provádění procesů, spočívá v tom, že se umožní, aby každý piket prováděl jeho vlastní proud instrukcí. To je v podstatě MIMD uvnitř JIMD. Nyní je možné v podstatě uspořádat tentýž JIMD stroj jako MIMD systém nebo jako stroj úplně jiného uspořádání. To je možné proto, že je možné programovat piket, aby pracoval se svým vlastním sledem instrukcí.

Protože každý piket může být vytvořen tak, aby měl svůj vlastní sled, dekodování a je možná velmi jednoduchá množina instrukcí na úrovni piketu a to umožňuje mnohem extenzivnější místní provádění procesů. Oblasti, ve kterých se nejsnadněji najde počáteční aplikace jsou v komplexním rozhodování, nicméně jednoduché provádění procesů v pevné řádové čáře představuje další zájmovou oblast programátorů.

Takový jednoduchý program by zavedl bloky piketového programu, například ne větší než 2K, do piketové paměti 102 a tyto mohou být prováděny když JIMD karta 703 začne místní provádění řízením provádění začínajícím u některé specifikované adresy xyz. To by pokračovalo když kontroler buď napočítá příslušný počet hodinových cyklů nebo provede test pro úplný signál úlohy ovládním stavového kanálu (SF) registry znázorněnými v obr.4.

Stavový kanál (SF obr.4) používá zámek 104 pro každý piket. Každý piket má zámek 104, který může být naložen, aby udával stavovou podmínku piketu. JIMD kontroler může testovat kolektivní hodnotu v těchto zámcích (jeden na jeden piket) ovládním linky stavu řady. Tato linka stavu řady je logická kombinace hodnot z každého ze zámků stavu piketů.

V následujícím příkladu předpokládáme, že chceme nastavit hodnotu vyšší než 250, která je v rozsahu 500×250 . Následující rutina by použila stavový kanál pro zjištění, že úloha byla splněna.

```
If VALUE 500 then TURN YOUR PICKET OFF
STAT - PICKET OFF CONDITION
IF STAT FUNNEL = OFF then finished - - - -
VALUE - VALUE - 250
Repeat
```

Tak může být uspořádání mnohonásobných paralelních piketových procesorů uspořádáno rozličnými způsoby, a jako JIMD procesor.

Takový JI'D stroj v přednostním provedení je programován, aby prováděl jediný sled instrukcí klasickým způsobem a je kódován, aby prováděl operace na řadě JI'D procesorů sekvenčním postupem podobným klasickým procesorům, při obecném řízení JI'D kontroleru nebo sekvenčního obvodu. Na aplikační úrovni je to splněno vektorem a vektorem podobnými instrukcemi a vektory mohou být zpracovány procesory a přes procesor. Vektorové instrukce mohou být sloučeny s mikroinstrukcemi, typicky se 6 až 10 takovými instrukcemi.

S takovým přednostním provedením systém se jeví schematicky jak je znázorněn ve funkčním blokovém diagramu podsystemu paralelních procesorů znázorněném v obr.6. Přes I/O porty systému řízené řízením vnějšího rozhraní 413 funguje sekvenční obvod podsystemu podobně jako JI'D program s vysoce funkčními makro bloky řídicími funkce procesních prvků. Adresování paměti dovoluje proud dat 8 bitů, šíře byte a je užita aritmetická logika modulo 8 pro funkce (logické, add, multiply a divide). Je účinně opatření pro formát pohyblivé řádové čárky a pro autonomní operaci piketů s individuálním spánkovým a klidovým modelem a odděleným adresováním.

Uspořádání kontroleru podsystemu je znázorněno v obr.7. Každá z karet 512 řady procesorů (znázorněno jako 4 v tomto znázornění podsystemu, avšak je možné provést redukci na dvě SEM E karty) je připojena k sekvenčnímu obvodu CRP 703, který je připojen ke kontroleru 702 podsystemu, který je spojen buď se systémem hlavní paměti nebo s jiným podsystemem v uspořádání přes rozhraní čipu 705 ke přidružené sběrnici 706 mikrokanálu. V přednostním provedení je kontroler podsystemu obecně účelová mikroprocesorová jednotka PS/2 společnosti IBM (ochranná známka IBM) a používá procesní čip Intel 386 a paměť 4 Mbyte. Mikroprocesor MP 702 osobního počítače je připojen ke kartě sekvenčního obvodu přes sběrnici 705 706 mikrokanálového typu uvnitř podsystemu.

Je zřejmé, že jsou možné mnohé modifikace a variace předloženého vynálezu ve světle těchto výkladů a je tedy třeba uvážít, že připojené patentové nároky umožňují, že vynález může být vytvořen v praxi i jinak než jak byl popsán.

Speciálně následující výtvořky mohou být alternativně kombinovány nebo spojeny s opatřeními definovanými v nárocích:

- 1) Skluzový prostředek umožňuje přenos informace v jediném cyklu do sousední polohy piketové jednotky umístěním adresy piketu, který by normálně byl schopen přijmout informaci, kdyby nebyl průhledný pro zprávu vyslanou dokud nepřijde a je přijata nejbližším aktivním sousedem, který ji přijme;
- 2) Skluzový prostředek umožňuje skluzovou funkci vysláním informace do sousední polohy přes "vyřazené" pikety umožňuje první piketové jednotce přenést informaci do vzdáleného piketu když před přenosovým cyklem jsou bránící pikety učiněny průhlednými jejich vyřazením a potom v přenosovém cyklu řídicí prostředek způsobí, že první piketová jednotka vyšle informaci do místa určení v odlehlé piketové jednotce;
- 3) jsou opatřeny prostředky pro přenos zpráv lineárně přes strukturu, přes dvourozměrnou síť, nebo ve trojrozměrném směru v řadě;
- 4) množství piketových jednotek má místní autonomii a je možné je uspořádat jako svázanou jednotku do řady zmíněných piketových jednotek;
- 5) systém je možno uspořádat jako JIMD a MIMD systém a skupiny množství procesních jednotek jsou přiřazeny ke programovatelnému uspořádání, ve kterém individuální jednotky systému mají určitý stupeň místní autonomie;
- 6) Je zahrnut hlavní procesorový systém, který komunikuje s vnějším kontrolním sekvenčním obvodem přes sběrnici a hlavní systém má prostředek pro způsobení provádění globálních instrukcí paralelním procesním systémem;
- 7) Vnější řídicí sekvenční obvod je připojen k mikrokódové paměti, která je programovatelná uloženými rutinami pro funkce;
- 8) Sekvenční obvod je opatřen vysokofunkčními makro bloky pro řízení funkcí procesních jednotek, které jsou k sekvenčnímu obvodu připojeny sběrnici, a kde adresování paměti místních pamětí systému umožňuje proud dat šíře byte a aritmetickou logiku modulo 8 užít pro funkce logické, add, multiply a divide, a kde je učiněno opatření pro operace v pohyblivé řádové čáře se zmíněnými paralelními jednotkami a pro individuální módy spánku a klidu s odděleným adresováním individuálních piketových procesních jednotek;
- 9) provádění operací s funkcemi je programovatelně umístitelné k provádění uvnitř systému hlavního procesoru, s vnějším řídicím sekvenčním obvodem s uloženými rutinami, s prostředky místního registru, nebo uvnitř množství piketových procesních jednotek,

- a kde jedny instrukce vyžadující rozsáhlé provádění procesů na množství dat jsou přiřazeny ke zmíněnému množství procesních jednotek, které je uspořádáno pro JIND provádění procesů;
- 10) procesní jednotky systému jsou spojeny do řady procesních jednotek s procesními jednotkami řady majícími programovatelnou místní autonomii, a kde piketové procesní jednotka může pracovat se svým vlastním sledem instrukcí a může vstoupit do nebo vystoupit z operací spojených s jinými procesními jednotkami na základě podmínek dat, a kde procesní jednotky systému mohou nezávisle provádět zařazení a normalizaci v operacích, které jsou přidruženy k operacím v pohyblivé řádové čárce;
- 11) zmíněný vnější řídicí sekvenční obvod je uspořádán pro provádění operací jako CALL (,), Load, Elock, Sin, Cos, Find, Min, Range, a Matrix Multiply uloženými rutinami zajištěnými knihovnou času běhu uložených rutin;
- 12) uložené rutiny jsou zajištěny pro provádění kontrolních, Load, Read, Add, Multiply a Match funkcí;
- 13) nějaký procesní prvek vymění data v řadě sousedních procesních jednotek řady a mezi stránkami ve zmíněném systému přes propojovací síť;
- 14) je zde opatřen vnější řídicí procesor pro řadu piketů, a kde mikro-kontrolní kód je přenášen paralelně ke všem piketům skupiny piketových procesorů v řadě, a kde řídicí procesor a piketové procesní jednotky jsou synchronizovány na tentýž hodinový systém, takže funkce řízené vnějším řídicím procesorem mohou být prováděny ve stejném čase hodin;
- 15) nad vnějším řídicím procesorem pro řadu piketů je vrcholový řídicí systém jako mikroprocesor svázaný s vnějším řídicím procesorem přes sběrnici mikrokanálu, kterýžto vrcholový mikroprocesor ovládá aktivitu řady piketů, a kde je systém spojen tak, že v daném okamžiku všechny piketové procesní jednotky řady mohou provádět tutéž instrukci, ačkoliv podmnožiny procesorů mohou reagovat individuálně na řídicí proud;
- 16) množství piketových paralelních procesních jednotek je uspořádáno podél adresové sběrnice pro komunikaci s vnějším piketovým kontrolerem, a kde je vektorová adresa společná lokální paměti piketového systému a kde je registr vektorové adresy dat, aby kontroloval, jaká data jdou do každé místní paměti piketového systému. Komunikace se řízením piketů. Sběrnice vektorové adresy je společná paměť a registr vektorové adresy dat

- zjišťuje, jaká data jdou přes každou asociativní paměť množiny;
- 17) je zde opatřeno množství piketových jednotek uspořádaných v řadě a cesty pro proud dat mezi piketovými jednotkami majícími komunikační cesty vatvářející způsoby a prostředky pro bitovou paralelní komunikaci se všemi ostatními pikety v řadě při opatření vazební cesty pro přenos zpráv z jednoho piketu do některého jiného piketu;
- 18) zprávy mohou být přenášeny z jednoho piketu do některého jiného piketu přes N-rozměrnou řadu implementovanou užitím drátu-nebo nebo bodově spojené pikety;
- 19) je implementována malá sekce místní paměti takže každé místo v této sekci současně bude provádět srovnání krytí proti danému vzoru.

P A T E N T O V É N Á R O K Y

1. Paralelní procesní systém obsahující množství piketových jednotek, kde každá piketová jednotka má bitový paralelní procesní prvek kombinovaný s místní pamětí připojenou ke procesnímu prvku pro paralelní zpracování informace ve všech piketových jednotkách asociativním způsobem, kde každá piketová jednotka je uzpůsobena ke provádění jednoho prvku asociativního procesu.
2. Paralelní procesní systém podle bodu 1, vyznačující se tím, že paměti množství piketových jednotek jsou uspořádány v poli a pole piketů s jejich pamětí tvoří sadu asociativní paměti.
3. Paralelní procesní systém podle bodu 1 nebo 2, vyznačující se tím, že dále obsahuje množství piketových jednotek vytvořených na čipu s křemíkovou bází a uspořádaných v poli se sousedy pro piketové jednotky ke straně přilehlých piketových jednotek tak, že na čipu baze je vytvořeno piketové procesní pole s množstvím pamětí, jednou pro každý z procesních prvků schopných působení jako bytové adresovatelná paměť velkého obsahu sloupců uspořádaná v logické řadě nebo poli se sousedními komunikačními cestami pro přenos dat dvousměrně mezi piketovými jednotkami a z vnějšku do nich.
4. Paralelní procesní systém podle kteréhokoli z bodů 1 až 3, vyznačující se tím, že je opatřen pamětovým čipem pole piketového procesoru s množstvím piketových jednotek a pamětí a majícím cesty přenosu dat mezi každým piketovým procesním prvkem a pamětí a mezi piketovými jednotkami.
5. Paralelní procesní systém podle kteréhokoli z bodů 1 až 4, vyznačující se tím, že je opatřen množstvím piketových jednotek uspořádaných v poli a cestami pro přenos dat mezi piketovými jednotkami majícími paměť jedna-na-jednu s každým procesním prvkem pole křížem, vlevo nebo vpravo do nebo z přilehlého souseda piketové jednotky a se skluzovým prostředkem pro zajištění skluzové komunikace se vzdálenými piketovými jednotkami.
6. Paralelní procesní systém podle kteréhokoli z bodů 1 až 5, vyznačující se tím, že každá piketová jednotka má procesor, který má přístup ke vlastní připojené místní paměti, a kde rozšířený znak nebo data násobného rozšíření znaku a instrukce jsou přeneseny mezi piketovými jednotkami v jednom hodinovém cyklu systému.
7. Paralelní procesní systém podle kteréhokoli z bodů 1 až 6, vyznačující se tím, že každý piketový čip má vlastní místní paměť pro zajištění asociativního uložení rozšířené sady znaku v poli

pro zajištění asociativního uložení rozšířené sady znaku v poli piketových jednotek s alespoň 32 Kbyty zásoby pro každou místní paměť a kde je šestnáct piketových jednotek vytvořeno jako uzly lineárního pod-pole.

8. Paralelní procesní systém podle kteréhokoli z bodů 1 až 7, vyznačující se tím, že je opatřen vysílací sběrnici pro komunikaci mezi piketovými jednotkami, která je mnohobytové šířky, takže všechny pikety mohou přijímat tatáž data ve stejném čase, a je opatřen prostředky pro převedení řízení piketů a přenosů propagace adres na tuto vysílací sběrnici.

9. Paralelní procesní systém podle kteréhokoli z bodů 1 až 8, vyznačující se tím, že místní paměť piketové jednotky je DRAM CMOS paměť v paměťovém poli, a která umožňuje přístup řádek-sloupec potlačením demultiplexace sloupce na zadní straně paměťového pole, a která vytváří řádkovou adresu, která čte ze řádku paměťového pole pro způsobení paralelního přenosu dat.

10. Paralelní procesní systém podle kteréhokoli z bodů 1 až 9, vyznačující se tím, že paměť přídatně k datům obsahuje "tri-bits" nebo "trit", takže zde jsou tři stavy rozlišované logikou, znamenající buď logickou 1 nebo logickou 0 nebo nerozumím a trit je uložen v postupných zásobních místech v zásobním poli asociativní paměti sady.

11. Paralelní procesní systém podle kteréhokoli z bodů 1 až 10, vyznačující se tím, že je opatřen řídicím prostředkem piketové jednotky pro zajištění řídicí funkce pro jednotlivou operaci piketovou jednotkou.

12. Paralelní procesní systém podle kteréhokoli z bodů 1 až 11, vyznačující se tím, že místní paměť má vícebitovou binární referenční ukládací adresu.

13. Paralelní procesní systém podle kteréhokoli z bodů 1 až 12, vyznačující se tím, že je opatřen vnějším řídicím a ukládacím prostředkem pro řízení funkcí, který v jednotlivé piketové jednotce nahrazuje operace v piketové jednotce, která má výstupní status, který splňuje zvláštní podmínku, kteréžto řídicí funkce zajišťují čekací funkci a potlačovací funkci a dovolený zápis do paměti založený na podmínkách v piketové jednotce a řídicí funkce, které jsou přiděleny piketové jednotce po odpojení od vnějšího řídicího a ukládacího prostředku.

14. Paralelní procesní systém podle kteréhokoli z bodů 1 až 13, vyznačující se tím, že každá místní paměť a procesní prvek piketové jednotky jsou opatřeny bytovým přenosovým prostředkem a prostředkem pro vstup dat a maskou pro umístění informace v paměti, a dále je opatřen prostředkem pro adresování paměti za účelem provedení vyhledání a prostředkem pro vstup dat a maskou pro vyhledání za účelem umístění slova mezi N slovy v paměti, a kde styčná místa zvyšují styčnou linku a zvláštní operace čte nebo volí první styk, a kde je vysílací prostředek pro vysílací operaci, ve které zvýšená zvolená linie označuje participaci a vysílaná data jsou kopírována na všechna zvolená místa slov.

15. Paralelní procesní systém podle kteréhokoli z bodů 1 až 14, vyznačující se tím, že procesní jednotka obsahuje ALU, maskové registry a zámek, jskož i stavové registry (SR) a registry přenosu dat (DF), které jsou připojeny k paměti jedna-na jednu pro každou procesní jednotku.

16. Paralelní procesní systém podle kteréhokoli z bodů 1 až 15, vyznačující se tím, že místní paměť je mnohabitová DRAM a logika pro každou piketovou procesní jednotku je vytvořena na téže křemíkovém čipu jako místní paměť DRAM, a kde je přímé spojení jedna-na-jednu mezi místní pamětí a jejím procesním prvkem, přičemž zmíněná místní paměť má buňky, které mají mnohabitovou adresu.

17. Paralelní procesní systém podle kteréhokoli z bodů 1 až 16, vyznačující se tím, že každá procesní jednotka je opatřena maskovými registry a zámkem, který je umístěn tak, aby působil jako spojovací port pro každou procesní jednotku podél komunikační linky, které je společná zmíněnému množství procesních jednotek.

18. Paralelní procesní systém podle kteréhokoli z bodů 1 až 17, vyznačující se tím, že je opatřen vnějším řídicím sekvenčním obvodem a místním řídicím registrem pro řízení stavu jednotlivých piketových procesních jednotek systému.

19. V paralelním procesním systému majícím vnější řídicí obvod a množství procesních jednotek s logickými hradly a registry a v nich obsaženými pamětmi, proces pro udržování popisu digitálního systému zahrnující kroky:

záznam každého popisu hradla digitálního systému jako výčet signálů, které popsané hradlo přijme jako vstupy a pojmenování signálu, který vyvíjí,

ořadevek, aby v každém čase se signál změnil, jeho jméno se

vyšle do všech procesních jednotek a je srovnáno paralelně se jménem očekávaného vstupního signálu,

pokračování pro určení, zda byl nalezen styk a provedení záznamu v procesní jednotce o nové hodnotě signálu v registru přenosu dat, pokračování, až do zaznamenání všech změn signálu a potom způsobení, aby všechny procesní jednotky přečetly paralelně řídicí slovo, které sdělí jejich přenosu dat jak se použije průběžné sada vstupů pro výpočet výstupu,

způsobení paralelního provedení těchto výpočtů s výsledky srovnávanými se starou hodnotou z místního hradla a

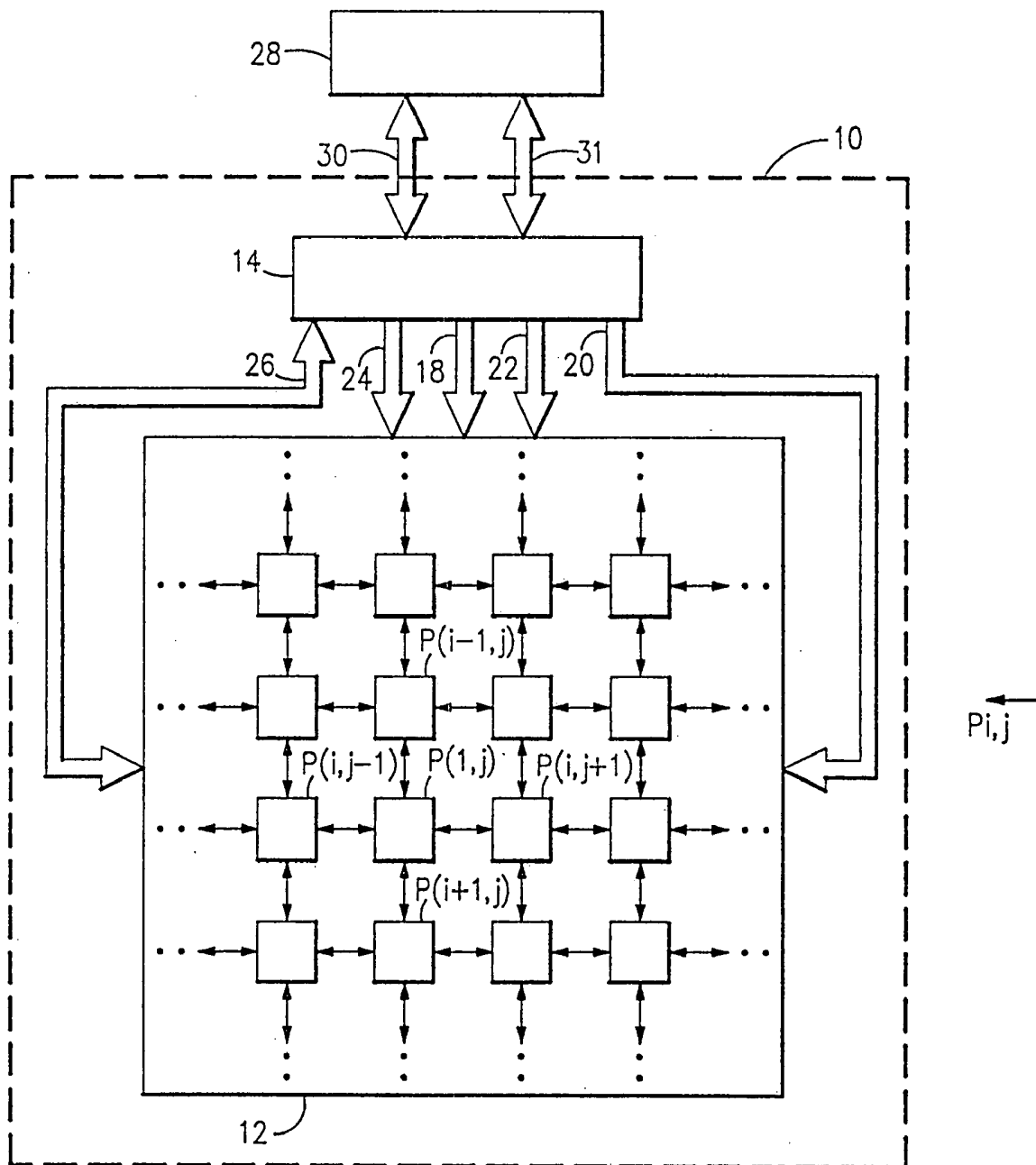
zaznamenání stavu registru přenosu dat všech z těch hradel procesních jednotek, jejichž výstupy se mění, a

způsobení, aby vnější řídicí prostředek vyšetřil všechny procesní jednotky a vyžádal změnu následujícího hradla a

vyslání odpovídajícího jména signálu a hodnoty z procesní jednotky do všech ostatních procesních jednotek a opakování cyklu až již nenastává žádná změna signálu nebo proces je zastaven.

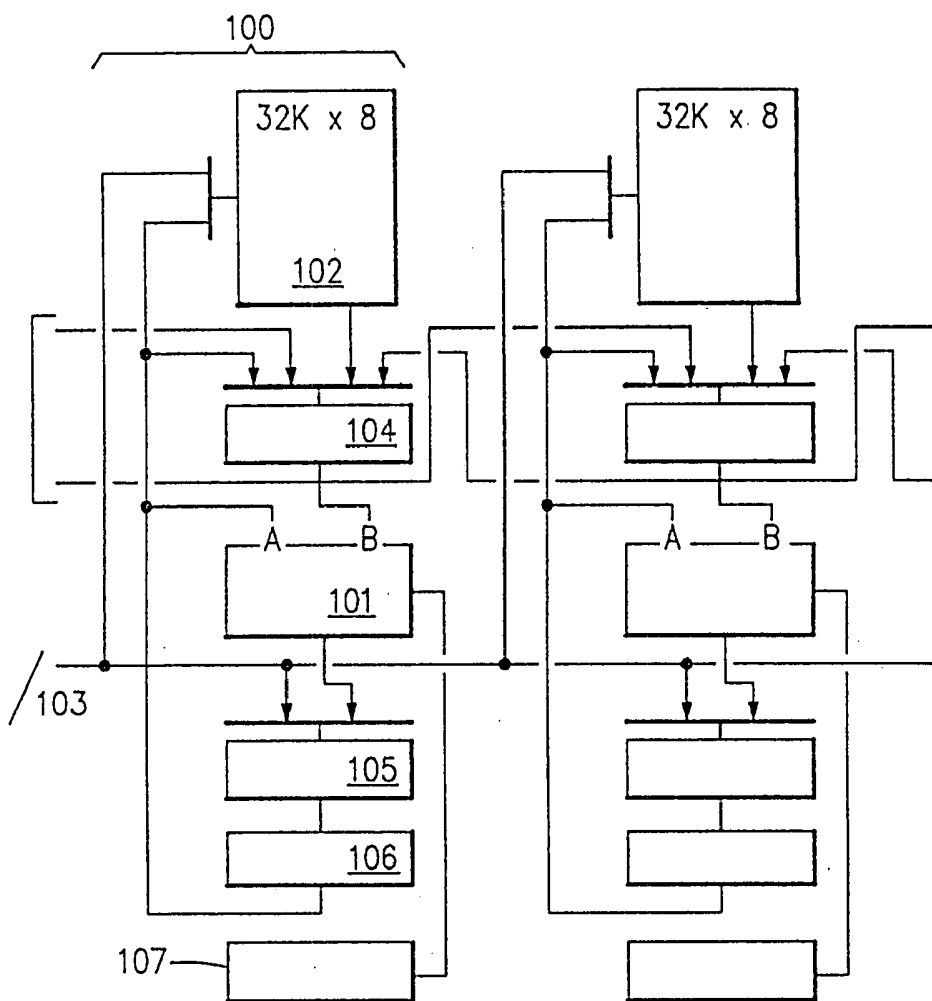
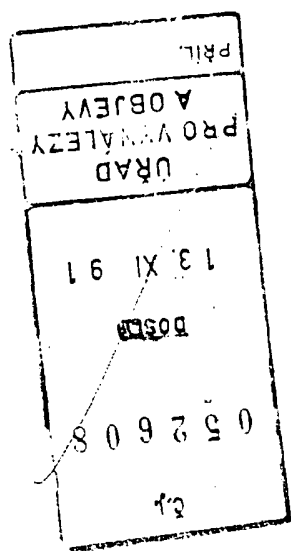
20. Způsob použití paralelního procesního systému majícího množství procesních jednotek a místních pamětí spojených pro přenos dat a řízení mezi nimi, vyznačující se tím, že se uloží slovníková jména v paměti procesní jednotky, takže první písmeno všech jmen může být porovnáno s oním žádaného vyslaného jména vyslaného do množství procesních jednotek paralelního procesního systému a kde všechny procesní jednotky bez styku jsou vyřazeny se zajištěnou řídicí charakteristikou, načež se porovná druhé písmeno jména a porovnávací a vylučovací procedura se opakuje pro po sobě následující písmena (znaky) dokud nezbývá žádná aktivní procesní jednotka nebo až byl dosažen konec slova, a potom se provede dotaz na zmíněné procesní jednotky a potom se způsobí čtení indexu žádaných dat z procesních jednotek.

1/5

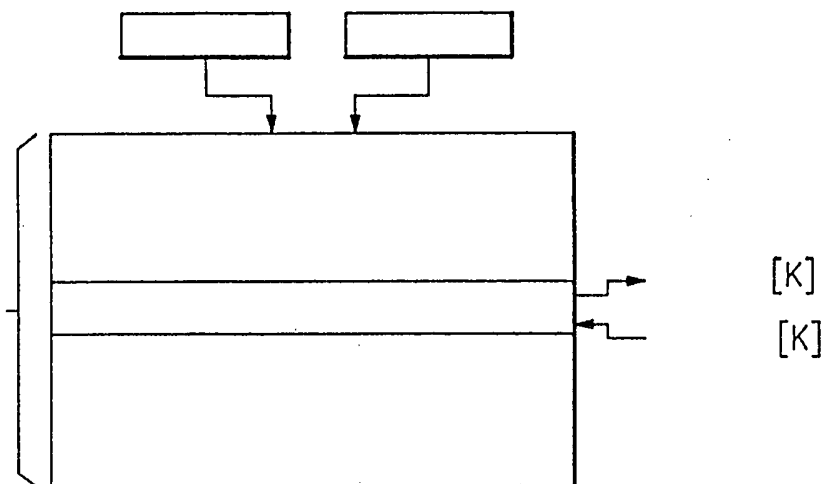


Obr. 1

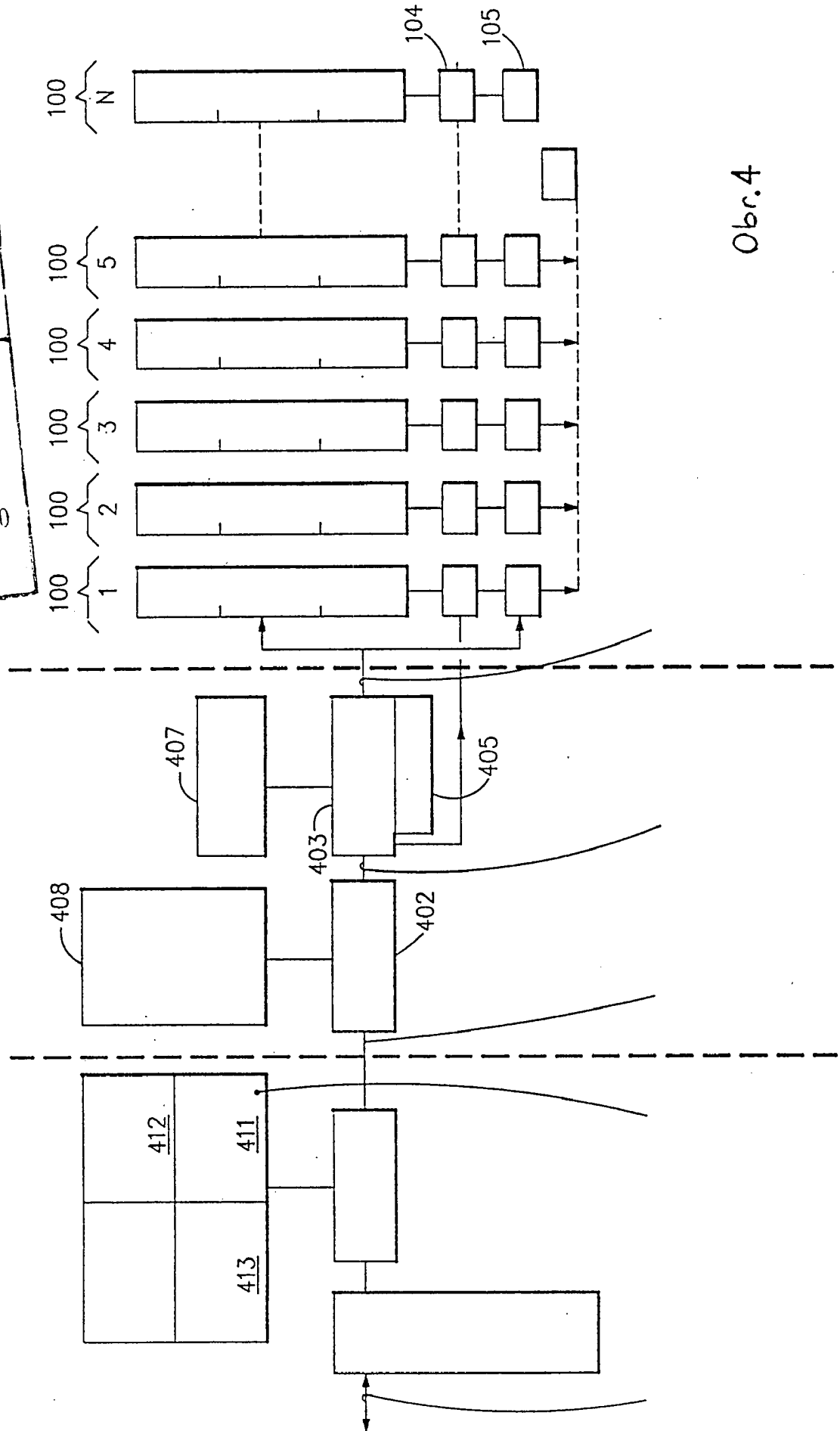
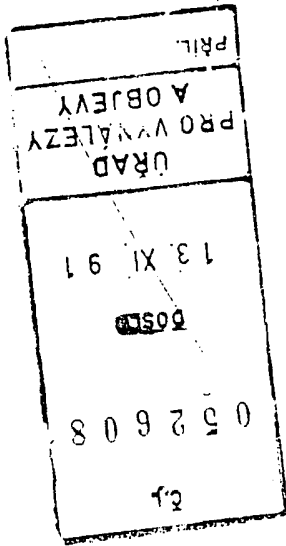
052608	13 XI 91	URAD	PROV. VY. ALEZY	AOBJEVY	PRIL.
--------	----------	------	-----------------	---------	-------



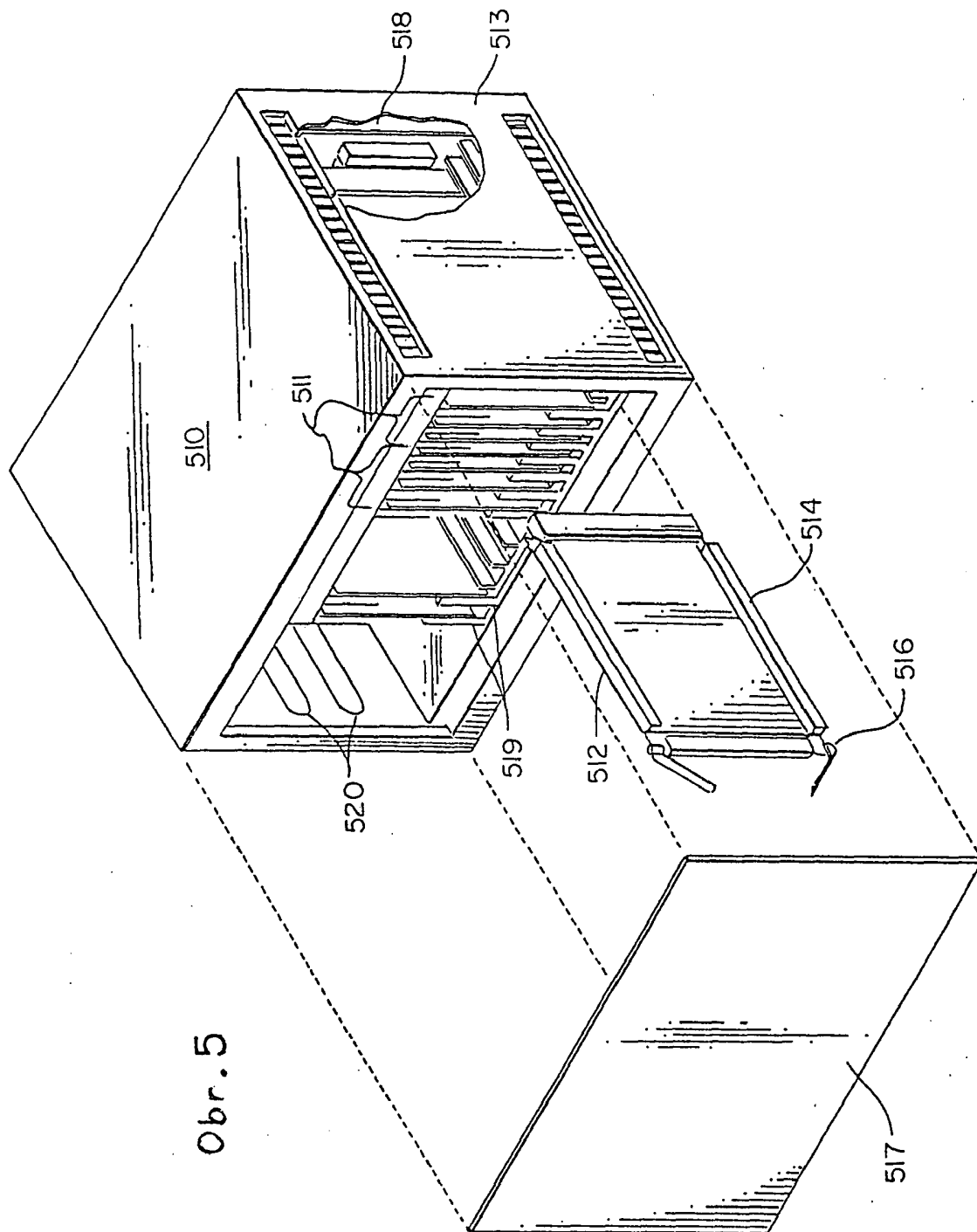
Obr.2



Obr.3



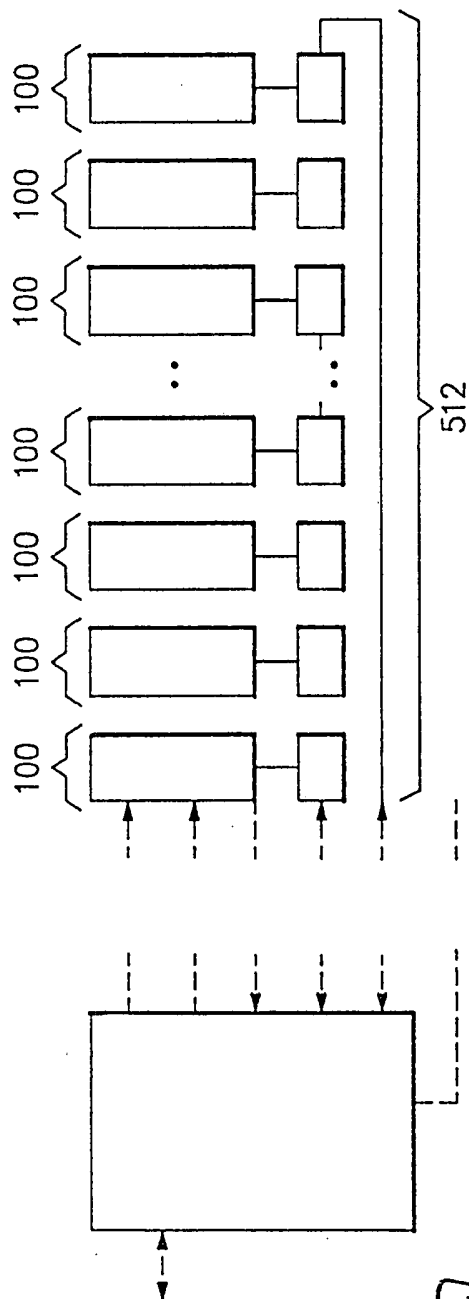
Obr. 4



Obr. 5

PRIL.	URAD PRO VYVÁLEZY A OBJEVY	13. XI. 91	809290	13
-------	----------------------------------	------------	--------	----

Obr. 6



Obr. 7

