

**ARCHITEKTURA PŘERUŠENÍ PRO SYSTÉM ZPRACOVÁNÍ DAT
S NEUNIFORMNÍM PAMĚŤOVÝM PŘÍSTUPEM (NUMA)**

Oblast techniky

Tento vynález se obecně týká zpracování dat a podrobněji zpracování dat v systémech zpracování dat s neuniformním paměťovým přístupem, zkr. NUMA (z angl. Non-uniform Memory Access). Zvláště pak se tento vynález týká architektury přerušení pro systémy zpracování dat NUMA.

Dosavadní stav techniky

V počítačových systémech se přerušení využívá k tomu, aby upozornilo procesor na výskyt události, která vyžaduje speciální ošetření. Přerušení lze například využít k požadování služby od přijímajícího procesoru, hlášení chybového stavu nebo prostě k přenášení informací mezi zařízeními. V jednoprocessorových počítačových systémech je podpora přerušení poměrně jednoduchá, neboť všechna přerušení jsou ošetřována jediným procesorem. V multiprocessorových počítačových systémech však vzniká další úroveň složitosti, protože je zde nutné použít mechanismus pro nasměrování přerušení k ošetření příslušnému procesoru či procesorům.

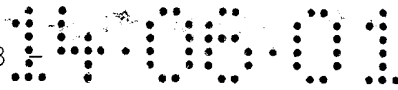
V běžných počítačových systémech se symetrickými multiprocessory (SMP) se přerušení ošetřují různými způsoby, které využívají hardwarové i softwarové mechanismy. Počítačový systém SMP typicky používá globální řadič

přerušeni k výběru procesoru k obsluze přerušeni podle priority přerušeni a priority procesu (pokud nějaká existuje) probíhajícího na každém procesoru. Řadič přerušeni tedy porovná prioritu přerušeni s prioritami procesů probíhajících na procesorech a vybere jako obsluhující procesor takový procesor, který provádí proces, jenž má nižší prioritu než přerušeni. Jelikož jsou procesory v SMP poměrně pevně propojeny, stanovení priorit procesů a přesměrování přerušeni k obsluhujícímu procesoru lze dosáhnout pomocí zařízení, které využívá buďto sdíleného systémového propojení nebo vyhrazených spojů pro přerušeni.

V současné době se objevila topologie multiprocesorového počítačového systému, známá jako neuniformní paměťový přístup (NUMA). Typický počítačový systém NUMA může obsahovat uzlové propojení s vysokou latencí, k němuž je připojeno několik víceprocesorových uzlů, z nichž každý obsahuje lokální systémovou paměť. Protože více procesorů v počítačovém systému NUMA není pevně propojeno, běžné mechanismy obsluhy SMP přerušeni a přenosu nelze v počítačových systémech NUMA aplikovat přímo. Je tedy zřejmé, že v počítačových systémech NUMA je potřebný mechanismus správy přerušeni, který zajišťuje účinné mechanismy pro směrování a komunikaci přerušeni.

Podstata vynálezu

Počítačový systém s neuniformním paměťovým přístupem (NUMA) obsahuje alespoň dva uzly spojené uzlovým propojením, kde alespoň jeden z uzlů obsahuje procesor pro obsluhu přerušeni. Podle tohoto vynálezu, architektura přerušeni počítačového systému NUMA, který obsahuje jak hardwarové,



tak softwarové součásti, rozděluje počítačový systém NUMA na domény vnějšího přerušeni, takže vnější přerušeni je vždy předáváno procesoru v té doméně vnějšího přerušeni, v níž se přerušeni vyskytuje. Přestože každá taková doména vnějšího přerušeni obsahuje typicky pouze jediný uzel, lze implementovat vedení přerušeni (interrupt channelling) nebo slévání přerušeni (interrupt funelling) k nasměrování vnějších přerušeni přes hranice uzlů k předání procesoru.

Po předání procesoru se může spustit k obsluze vnějšího přerušeni na jakémkoli procesoru v systému software správy přerušeni. Architektura přerušeni tohoto vynálezu ve srovnání s dřívějšími technickými metodami výhodně umožňuje softwaru správy přerušeni pohotovou obsluhu vnějších přerušeni snížením velikosti dotazovacího řetězce (stromu) správy přerušeni.

Kromě vnějších přerušeni architektura přerušeni podle tohoto vynálezu podporuje meziprocesorová přerušeni, zkr. IPI (z angl. Inter-Processor Interrupt), kterými může jakýkoli procesor přerušit sám sebe nebo jeden či více procesorů v počítačovém systému NUMA. IPI jsou aktivována zapsáním do paměťově mapovaných registrů v globální systémové paměti, což usnadňuje přenos IPI přes hranice uzlů a dovoluje všesměrovým IPI spustit se prostým vysláním jedné transakce zápisu každému uzlu obsahujícímu procesor, který se má přerušit.

Architektura přerušeni podle tohoto vynálezu se objevuje od malých počítačových systémů NUMA obsahujících několik uzlů po velké systémy obsahující stovky uzlů. Hardware přerušeni v každém uzlu je také rozmístěn za účelem škálovatelnosti, přičemž součásti hardwaru komunikují

transakcemi přerušeni, které jsou přenášeny sdílenými komunikačními cestami (tj. lokální sběrnice a propojení).

Přehled obrázků na výkresech

Vynález bude blíže vysvětlen prostřednictvím konkrétních příkladů provedení znázorněných na výkresech, na kterých představuje

- obr. 1 vzorové provedení počítačového systému NUMA, se kterým může být tento vynález výhodně proveden;
- obr. 2 vzorové provedení mapy fyzické paměti, kterou může používat počítačový systém NUMA znázorněný na obr. 1;
- obr. 3A, 3B vzorová provedení registru zdrojové konfigurace přerušeni a registru nevyřešených přerušeni ve zdrojové jednotce přerušeni, zkr. ISU (z angl. Interrupt Source Unit), podle tohoto vynálezu;
- obr. 4 podrobnější blokové schéma cílové jednotky přerušeni, zkr. IDU (z angl. Interrupt Destination Unit), podle tohoto vynálezu;
- obr. 5 logický vývojový diagram vyšší úrovně pro práci ISU podle tohoto vynálezu;
- obr. 6 logický vývojový diagram vyšší úrovně pro práci IDU podle tohoto vynálezu;
- obr. 7 logický vývojový diagram vyšší úrovně pro

názorné provedení konfigurační rutiny, která konfiguruje zdroje přerušení podle tohoto vynálezu;

obr. 8 logický vývojový diagram vyšší úrovně znázorňující práci softwaru správy přerušení první úrovně, zkr. FLIH (z angl. First Level Interrupt Handler), podle tohoto vynálezu.

Příklady provedení vynálezu

1.0 Přehled počítačového systému NUMA

Pokud jde o obrázky, zejména pak o obr. 1, je zde uvedeno vzorové provedení počítačového systému NUMA podle tohoto vynálezu. Uvedené provedení lze realizovat např. jako pracovní stanici, server nebo samočinný počítač. Podle obrázku počítačový systém 6 NUMA obsahuje několik N ($N \geq 2$) zpracovávajících uzlů 8a-8n, jež jsou propojeny uzlovým propojením 22. Každý zpracovávající uzel 8a-8n obsahuje M ($M \geq 0$) procesorů 10. Pokud se nacházejí uvnitř zpracovávajícího uzlu, jsou procesory 10a-10m přednostně identické a mohou obsahovat procesor z řady procesorů PowerPC od společnosti International Business Machines (IBM), Armonk, New York (PowerPC je ochranná známka společnosti IBM). Kromě registrů, logiky toku instrukcí a prováděcích jednotek využívaných k provádění instrukcí programu, což se souhrnně označuje jako jádro 12 procesoru, obsahuje také každý procesor 10a-10m na čipu hierarchii 14 cache pamětí, která se používá k předčerpávání dat ze systémových pamětí 18 pro příslušné jádro 12 procesoru. Každá hierarchie 14 cache může obsahovat např. mezipaměť

cache první úrovně (L1) a mezipaměť cache druhé úrovně (L2), které mají kapacity 8-32 kilobytů, resp. 1-16 megabytů. Protože data uložená v každé systémové paměti 18 mohou být požadována, čtena a změněna kterýmkoliv procesorem 10 v počítačovém systému 6 NUMA, počítačový systém 6 NUMA implementuje především protokol soudržnosti cache (např. změněná, exkluzivní, sdílená, neplatná, zkr. MESI z angl. Modified, Exclusive, Shared, Invalid, nebo nějaká varianta předchozího) k zajištění soudržnosti jednak mezi cache ve stejném zpracovávajícím uzlu, jednak mezi cache z různých zpracovávajících uzlů.

Podle obrázku zpracovávající uzly 8a-8n dále obsahují příslušný uzlový řadič 20, zapojený mezi lokální propojení 16 a uzlové propojení 22. Každý uzlový řadič 20 slouží jako lokální zprostředkovatel pro vzdálené zpracovávající uzly 8 tím, že plní alespoň dvě funkce. Za prvé, každý uzlový řadič 20 sleduje příslušné lokální propojení 16 a ulehčuje přenos lokálních komunikačních transakcí ke vzdáleným zpracovávajícím uzlům 8. Za druhé, každý uzlový řadič 20 sleduje komunikační transakce na uzlovém propojení 22 a ovládá odpovídající komunikační transakce na příslušném lokálním propojení 16. Komunikace na každém lokálním propojení 16 je řízena přidělovačem 24. Přidělovače 24 regulují přístup k lokálnímu propojení 16 na základě požadování sběrnice, které vysílají procesory 10, a sestavují souvislé odezvy pro sledované komunikační transakce na lokálních propojeních 16.

Přístup do každé systémové paměti 18 v počítačovém systému 6 NUMA je regulován příslušným řadičem 17 paměti, zkr. MC (z angl. Memory Controller). Kromě soustavy obvodů, která přijímá a obsluhuje žádosti o čtení a zápis vysílané

procesory 10a-10m, uzlovým řadičem 20 a jinými zařízeními v daném zpracovávajícím uzlu 8, obsahuje řadič 17 paměti také cílovou jednotku 19 přerušení (IDU), která obsahuje, jak je popsáno níže, několik registrů a příslušnou logiku, které usnadňují směrování a správu přerušení.

Lokální propojení 16 je spojeno prostřednictvím můstku 26 mezilehlé sběrnice s mezilehlou sběrnicí 30, kterou lze například implementovat jako lokální PCI (z angl. Peripheral Component Interconnect, periferní propojení komponent) sběrnicí. Můstek 26 mezilehlé sběrnice zajišťuje jednak cestu s nízkou latencí, kterou mohou procesory 10 přímo přistupovat k zařízením typu V/V zařízení 32 a paměťová zařízení 34 namapovaným v paměti sběrnice nebo ve V/V adresových prostorech, jednak cestu velmi širokého pásma, kterou mohou V/V zařízení 32 a paměťová zařízení 34 přistupovat do systémové paměti 18. V/V zařízení 32 mohou zahrnovat např. zobrazovací zařízení, klávesnici, grafický ukazatel a sériové a paralelní porty pro připojení k externím sítím či připojeným zařízením. Naopak paměťová zařízení 34 mohou zahrnovat optické či magnetické disky, které zajišťují energeticky nezávislý ukládací prostor pro operační systém a aplikační software.

V/V zařízení 32 i paměťová zařízení 34 (stejně jako další mimoprocessorové součásti počítačového systému 6 NUMA) mohou generovat přerušení za několika účely včetně signalizace přijetí nějakého vstupu, hlášení chybového stavu, atd., spojí 35 požadování přerušení. Tato přerušení, která jsou dále zmíněna jako vnější přerušení kvůli zdůraznění, že tato přerušení jsou generována jinou součástí než procesor 10, jsou shromažďována jednou či několika zdrojovými jednotkami 28a, 28b přerušení (ISU). I když jsou

ISU 28a a 28b z důvodu zřetelnosti zobrazeny odděleně, mohou být také začleněny do sady čipů, které tvoří můstek 26 mezilehlé sběrnice. Jak je podrobněji popsáno níže, ISU 28 směřují vnější přerušeni k IDU 19, která poté předává vnější a jiná přerušeni k obslužení lokálním procesorům 10 spojem 36 požadování přerušeni.

Lokální propojení 16 a uzlové propojení 22 lze realizovat pomocí jakékoliv sběrnice přenosové struktury, přepínačové přenosové struktury, přepínačové nepřenosové struktury nebo architektury hybridního propojení, která obsahuje sběrnice i přepínačové složky. Bez ohledu na použitou architekturu propojení, lokální propojení 16 a uzlové propojení 22 podporují především rozdělené transakce, což znamená, že časování adresových a datových částí komunikačních transakcí je nezávislé. Kvůli umožnění identifikace, která data a adresy náleží dané komunikační transakci, se oba adresové a datové pakety, jež společně tvoří transakci, přednostně označují stejnou jmenovkou transakce.

Každý procesor 10 a každé další zařízení připojené na lokální propojení 16, je přednostně jedinečně identifikováno v počítačovém systému 6 NUMA celosystémovým identifikátorem zařízení, který je tvořen složením uzlového identifikátoru toho zpracovávajícího uzlu 8, v němž se zařízení nachází, a lokálního identifikátoru zařízení. Například v provedení, ve kterém jsou maximálně 4 zpracovávající uzly 8 a ke každému lokálnímu propojení 16 lze připojit nejvýše 8 zařízení, lze použít pětibitového identifikátoru zařízení; dva bity vyšších řádů pro uzlový identifikátor a tři bity nižších řádů pro lokální identifikátor zařízení. Každý uzlový identifikátor je přednostně uložen v registru příslušného

uzlového řadiče 20 a lokální identifikátory jsou přednostně uloženy v registrech identifikace zařízení u každého zařízení, které je připojeno k lokálnímu propojení 16. Každého takového celosystémového identifikátoru zařízení lze výhodně použít jako bitovou část vyšších řádů u každé jmenovky transakce, která je generována příslušným zařízením, tak, že je zajištěna jedinečnost jmenovky transakce v počítačovém systému 6 NUMA.

1.1 Mapa fyzické paměti

Na obr. 2 je znázorněna vzorová mapa fyzické paměti, kterou lze využívat v provedení počítačového systému 6 NUMA se čtyřmi zpracovávajícími uzly 8, z nichž každý obsahuje systémovou paměť 18. V provedení znázorněném na obr. 2 sdílí všechna zařízení počítačového systému 6 NUMA společný 16 gigabytový (GB) fyzický adresový prostor 50 zahrnující jednak oblast 52 paměti pro všeobecné účely jednak oblasti 54 systémového řízení a periférií. Každá fyzická adresa v oblasti 52 paměti pro všeobecné účely je vyčleněna pouze pro jedno fyzické místo v jedné ze systémových pamětí 18. Ve znázorněném provedení je oblast 52 paměti pro všeobecné účely rozdělena do 512 MB segmentů, kde je každému ze zpracovávajících uzlů 8 přidělen každý čtvrtý segment. Zpracovávající uzel 8, jenž uchovává ve své systémové paměti 18 určitý údaj, se označuje jako domovský uzel pro tento údaj; naopak ostatní zpracovávající uzly 8a-8n se označují jako vzdálené uzly vzhledem k tomuto určitému údaji.

Oblasti 54 systémového řízení a periférií, stále ještě na obr. 2, které obsahují 2 GB fyzických adres v uvedeném provedení, zahrnují 256 MB oblast 56 systémového řízení, 0,5 GB prostoru 58 pro periferní V/V, 1 GB prostor 60 paměti

periferií a oblast 62 pro zavedení systému, zkr. IPL (z angl. Initial Program Load). Oblast 62 IPL obsahuje adresy vyhrazené k přidělení až 256 MB pro kód IPL (tj. zaváděcí kód), který je typicky uložen v paměti ROM. Kód IPL obsahuje zavaděč operačního systému, jako např. Advanced Interactive Executive (AIX), který je dodáván společností IBM. Podle obrázku je 0,5 GB prostoru 58 pro periferní V/V rozděleno mezi stejně velké segmenty 62, z nichž každý je přidělen příslušnému zpracovávajícímu uzlu 8. Prostor 60 paměti periferií je rozdělen obdobně do stejně velkých 256 MB segmentů 66, z nichž každý je alokován pro příslušný zpracovávající uzel 8.

Stejně jako prostor 58 pro periferní V/V a prostor 60 pro paměť periferií obsahuje i oblast 56 systémového řízení množství segmentů 70, z nichž každý je vyhrazen pro příslušný zpracovávající uzel 8. V uvedeném provedení obsahuje každý segment 70 adresový prostor velikosti 64 MB. Kromě adres určených k uložení řídicích informací o uzlu obsahuje každý segment 70 oblasti systémového řízení také fyzické adresy vyhrazené pro registry přerušeni v IDU 19 a ISU 28 příslušného zpracovávajícího uzlu 8. Jak bude popsáno níže, právě tyto paměťově mapované registry paměti jsou využívány tímto vynálezem k přijímání a směrování vnějších přerušeni, vyvolávání meziprocessorových přerušeni a směrování přerušeni mezi zpracovávajícími uzly 8.

2.0 Přehled architektury přerušeni

Architektura přerušeni podle tohoto vynálezu umožňuje alespoň tři různé třídy přerušeni. Za prvé jsou to vnitřní přerušeni, která jsou spouštěna vnitřní činností procesoru. Vnitřní přerušeni mohou být vyvolána např. programovou

výjimkou nebo přetečením/podtečením registru ve vnitřním procesoru. Za druhé, jak bylo uvedeno výše, vnější přerušeni mohou být generována zařízeními jako např. V/V zařízení a systémové časovače, které leží vně procesorů. Za třetí, tento vynález podporuje také meziprocessorová přerušeni (IPI), která jsou generována jedním procesorem za účelem přerušeni jiného procesoru.

V upřednostňovaném provedení tohoto vynálezu umožňuje počítačový systém 6 NUMA podporu vnějších přerušeni a IPI pomocí architektury přerušeni, která se shoduje s rozšířením standardu OpenPIC (z angl. Open Processor Interrupt Controller). OpenPIC je např. popsán ve studii OpenPIC Programmable Interrupt Controller (PIC) Register Interface Specification Revision 1.2, vydané v říjnu 1995 společnostmi Advanced Micro Devices, Inc. a Cyrix, Inc. a uvedené zde odkazem. I když je upřednostňována slučitelnost s OpenPIC, tento vynález lze použít pro jakýkoliv systém, který má paměťově mapované registry řízení přerušeni, které jsou v celém systému jedinečné.

Architektura přerušeni podle tohoto vynálezu zahrnuje jak hardwarové, tak softwarové součásti, které jsou popsány níže.

2.1 Hardware architektury přerušeni

Navzdory konvenčnímu OpenPIC a jiným implementacím SMP přerušeni, které typicky využívají globálního řadiče přerušeni obsluhujícího jedinou doménu přerušeni, v počítačovém systému 6 NUMA si každý zpracovávající uzel 8 vytvoří vlastní doménu vnějších přerušeni, kde každá doména vnějších přerušeni má vlastní příslušnou IDU 19 a jednu či

více ISU 28, jak je znázorněno na obr. 1. ISU 28 vytvářejí rozhraní systému přerušeni pro zdroje přerušeni a IDU 19 vytváří rozhraní mezi systémem přerušeni a procesory 10. K zajištění účinné správy přerušeni a minimalizace přenosu přerušeni mezi doménami přerušeni jsou vnější přerušeni přijatá ISU 28 přenášena s využitím paketů přerušeni, které jsou rozesílány přes lokální propojení 16 (a mezilehlou sběrnici 30, v závislosti na implementaci) pouze té IDU 19 ležící ve stejné doméně přerušeni (tj. stejném zpracovávajícím uzlu 8), pokud je zpracovávající uzel 8 vybaven procesorem 10, který je nastaven k obsluze přerušeni. Přenos konfiguračních informací, meziprocessorových přerušeni, potvrzení přerušeni, příkazů ukončení přerušeni a jiných informací týkajících se přerušeni mezi doménami přerušeni je však podporován přes paměťově mapované registry v IDU 19, čímž je umožněno celosystémové využití prostředků přerušeni v každém zpracovávajícím uzlu 8.

2.1.1 Složky zdrojové jednotky přerušeni (ISU)

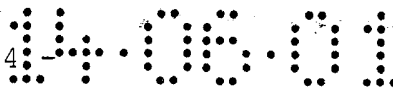
Na obr. 3A a 3B jsou znázorněna vzorová provedení registru zdrojové konfigurace přerušeni, resp. registru nevyřešených přerušeni v každé zdrojové jednotce přerušeni (ISU). Každá ISU 28 přednostně obsahuje alespoň jeden takový registr 72 zdrojové konfigurace přerušeni na každý zdroj přerušeni a jeden registr 82 nevyřešených přerušeni pro všechny zdroje přerušeni podporované danou ISU 28.

Nejprve co se týče obr. 3A, každý registr 72 zdrojové konfigurace obsahuje vektorové pole 73 označující vektor přerušeni pro příslušný zdroj přerušeni, rezervované pole 74 vektoru přerušeni, které může uložit dodatkové bity pro

označení vektoru přerušeni, a pole 75 priorit, které udává prioritu přerušeni generovanou příslušným zdrojem přerušeni. Ve vzorovém provedení jsou priority přerušeni v rozsahu od 0, což je nejnižší priorita, do 15, což je nejvyšší priorita. Možnosti přerušeni jsou přednostně jedinečné v každé doméně přerušeni. Každá doména přerušeni tak má přednostně jenom jedno přerušeni úrovně 1, ale v počítačovém systému 6 NUMA může být až N přerušeni úrovně 1. Samozřejmě lze využít technologických postupů z předchozího stavu techniky k umožnění sdílení přerušeni tak, že několik zdrojů přerušeni v jednom zpracovávajícím uzlu 8 sdílí stejnou úroveň přerušeni.

Registr 72 zdrojové konfigurace přerušeni dále obsahuje dvě rezervovaná pole 76 a 79, senzorový bit 77 pro označení toho, zda byl signál přerušeni spuštěn hranově nebo hladinově, bit 78 polarit pro označení toho, zda je přerušeni aktivní na nižší úrovni (neboli negativní hrany) nebo aktivní na vyšší úrovni (neboli pozitivní hrany), bit 80 aktivity (ACT) označující, zda vektorové pole 73 a pole 75 priority jsou používána a nemohou být změněna, a pole 73 masky (MSK), které umožňuje a znemožňuje jednotce ISU 28 přijetí přerušeni generovaných příslušným zdrojem přerušeni. Jako odezvu na přijetí přerušeni od konkrétního zdroje přerušeni spojem požadování přerušeni tedy ISU může určit odkazem na příslušný registr 72 zdrojové konfigurace oprávnění a prioritu přerušeni pro zdroj přerušeni a stejně tak identifikátor pro vektor přerušeni spojený s daným přerušením.

Jakmile bylo vnější přerušeni přijato a posouzeno jednotkou ISU 28, ISU 28 nastaví bit v registru 82 nevyřešených přerušeni na obr. 3B. Tento bit, který



jedinečně přísluší zdroji přerušeni, značí, že daný zdroj přerušeni má nevyřešené přerušeni. V provedeni na obr. 3B tedy každá ISU 28 může podporovat maximálně 16 zdrojů přerušeni.

2.1.2 Složky cílové jednotky přerušeni (IDU)

Na obr. 4 je znázorněno podrobnější blokové schéma představující IDU 19 v radiči 17 paměti zpracovávajícího uzlu 8. Znázorněné provedeni IDU 19 je slučitelné s OpenPIC a obsahuje tři různé oblasti registrů, globální registry 90, per-processorové registry 92 a registry 133 příkazů meziprocessorových přerušeni (IPI), z nichž každý se nachází v segmentu 70 oblasti systémového řízení zpracovávajícího uzlu na ofsetech (definovaných pomocí OpenPIC) od bazových adres specifikovaných v registru 102 globální konfigurace. Pro účely zjednodušení adresace je ofset mezi bazovou adresou a začátkem segmentu 70 oblasti systémového řízení zpracovávajícího uzlu přednostně stejný pro všechny IDU 19. Např. ve vzorovém provedeni počítačového systému 6 NUMA se čtyřmi zpracovávajícími uzly 8, z nichž každý obsahuje čtyři procesory 10, které společně sdílí 16GB prostor fyzické paměti, mohou být adresové bity 30-63 definovány rozsahem 00000000h-3FFFFFFFh, kde oblast 56 systémového řízení leží na A30..A63 0E000000h-0EFFFFFFh. Pokud číslo uzlu přiřazené zpracovávajícímu uzlu 8 je definováno na A36..A37 s tím, že čísla uzlů jsou v rozmezí b00-b11, pak segment 70 oblasti systémového řízení zpracovávajícího uzlu 8 s číslem uzlu b01 se bude nacházet na A30..A63 0E400000h-0E4FFFFFFh. Ve všech segmentech 70 oblasti systémového řízení se bazová adresa registrů v IDU 19 bude nacházet na běžném přidělovacím ofsetu jako například 000C0000h. Bazovou adresu registrů IDU 19 v rámci čísla uzlu b01 tedy lze získat po

přičtení 0E400000h k 000C0000h, což dává 0E4C0000h. Jednotlivé prostory registrů a registry v IDU 19 v rámci čísla uzlu b01 potom lze adresovat s využitím ofsetů definovaných pomocí OpenPIC následovně:

220000h	=	Pomocí OpenPIC vytvořený ofset bázové adresy specifikované v registru <u>102</u> globální konfigurace na per-processorové registry <u>120</u> procesoru b10 v uzlu b01
+ 0E4C0000h	=	bázová fyzická adresa registrů v IDU <u>19</u> v rámci čísla uzlu b01
<hr/>		
0E4C220000h	=	fyzická adresa per-processorových registrů <u>120</u> procesoru b10 v uzlu b01
0040h	=	Pomocí OpenPIC vytvořený ofset adresy per-processorových registrů <u>120</u> na port 0 příkazů IPI
+ 0E4C220000h	=	fyzická adresa per-processorových registrů <u>120</u> procesoru b10 v uzlu číslo b01
<hr/>		
0E4C220040h	=	fyzická adresa per-processorových registrů <u>120</u> procesoru b10 v uzlu číslo b01

Podle obr. 4, globální registry 90 v každé IDU 19 zahrnují registr 100 hlášení charakteristiky ke čtení i zápisu, registr 102 globální konfigurace ke čtení i zápisu, registr 104 identifikace výrobce pouze ke čtení, jeden registr 106 vektoru meziprocessorového přerušení (IPI) pro

každý port příkazů IPI (popsaný níže) ke čtení i zápisu, registr 108 nevlastního vektoru ke čtení i zápisu a registr 110 inicializace procesoru ke čtení i zápisu. Globální registry 90 jsou definovány pomocí OpenPIC a obsahují následující informace:

Registr 100 hlášení charakteristiky: celkový počet zdrojů přerušení detekovaných kódem IPL ve zpracovávajícím uzlu a celkový počet podporovaných procesorů pro daný zpracovávající uzel.

Registr 102 globální konfigurace: báze adresy oblasti globálních registrů pro zpracovávající uzel.

Registr 104 identifikace výrobce: označuje výrobce čipu integrovaného obvodu, který obsahuje IDU 19, a číslo verze.

Registry 106 vektorů IPI: vektor a informace o prioritě pro každý příslušný registr IPI ve zpracovávajícím uzlu.

Registr 108 nevlastního vektoru: vektor, který je vrácen, pokud je přijato potvrzení přerušení od procesoru a není žádné nevyřešené přerušení pro daný procesor.

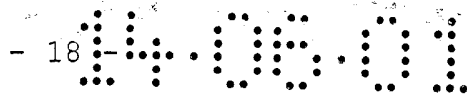
Registr 110 inicializace procesoru: signály nového zavedení softwaru pro každý procesor, který je v daném zpracovávajícím uzlu podporován.

Protože v počítačovém systému 6 NUMA jsou globální registry 90 sdílené všemi procesory 10, využívají se rutiny pro nastavení a správu přerušení softwaru ve vrstvě PAL operačního systému AIX, aby se zachovala soudržnost mezi globálními registry 90 ve všech zpracovávajících uzlech 8a-

8n. Aktualizace registrů, do nichž je povolen zápis, jiných než registr 110 inicializace procesoru je prováděno procesorem, který vyšle na své lokální propojení 16 N oddělených transakcí zápisu. Transakce zápisu mířící do lokální IDU 19 jsou přijaty a obslouženy lokálním řadičem 17 paměti. Zbytek transakcí zápisu je lokálním uzlovým řadičem 20 předán uzlovým řadičům 20 ostatních zpracovávajících uzlů 8, které obratem pošlou transakce zápisu přes lokální propojení 16 své příslušné IDU 19. Přístup do globálních registrů 90 je regulován globálním softwarovým zámekem, aby bylo zaručeno, že v daném čase upravuje globální registry 90 pouze jediný procesor 10. Během úprav globálních registrů 90 jsou všechna přerušení maskována, dokud se neprovedou úpravy v každém zpracovávajícím uzlu 8, aby se předešlo vysílání přerušení se starými nastaveními. Načtení hodnoty z globálních registrů 90 vyžaduje provést čtení lokální kopie globálních registrů 90, protože všechny globální registry 90 jsou synchronizovány.

Per-procesorové registry 92, stále ještě na obr. 4, obsahují M sad 120 registrů, jednu pro každý procesor 10, který může být podporován zpracovávajícím uzlem 8. Per-procesorové registry 92 jsou také definované pomocí OpenPIC a každá sada 120 registrů zahrnuje registr 122 priorit současné úlohy ke čtení i zápisu, registr 124 potvrzení přerušení pouze ke čtení a registr 126 ukončení přerušení pouze ke čtení. Sadu 120 registrů pro konkrétní procesor lze nalézt, jak bylo popsáno výše, s využitím báze adresy obsažené v registru 102 globální konfigurace, identifikátoru procesoru a ofsetu vytvořeného pomocí OpenPIC. Per-procesorové registry 120 mají následující funkce:

Registr 122 priorit současné úlohy: udává relativní úlohovou

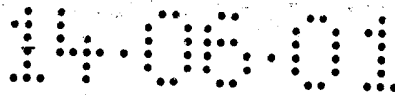


prioritu současné úlohy pokud se neobsluhují žádná přerušení. K vyvolání přerušení procesoru musí být priorita přerušení vyšší než priorita současné úlohy pro daný procesor.

Registr 124 potvrzení přerušení: pokud je čten softwarem pro potvrzení přerušení, hardware dodá vektor přerušení nevyřešeného přerušení pro příslušný procesor; pokud není žádné nevyřešené přerušení, je dodán nevlastní vektor přerušení.

Registr 126 ukončení přerušení, zkr. EOI (z angl. End Of Interrupt): sem software zapisuje příkaz k vydání EOI nejvyššímu obsluhovanému přerušení pro procesor, který vydal příkaz EOI. Zápisem do registru EOI pro vnější přerušení způsobí, že řadič 17 paměti vyvolá na lokálním propojení 16 transakci přerušení EOI.

Třetí oblast registrů v každé IDU 19 je sada registrů 133 příkazů IPI, která zahrnuje jeden registr příkazů IPI pro každou úroveň IPI přerušení, což je 4 v systémech vyhovujících OpenPIC. Každý registr 133 příkazů IPI obsahuje alespoň M bitů, kde pozice každého bitu odpovídá procesorovému identifikátoru jednoho z M lokálních procesorů 10. Zapsání b'1' do konkrétní bitové pozice v registru 133 příkazů IPI způsobí vydání IPI příslušné úrovni stanovenému procesoru 10, jak je popsáno níže. Stav těchto N sad registrů 133 příkazů IPI je hromadně udržován softwarem pro správu přerušení v hlavní sadě registrů příkazů IPI v oblasti paměti pro všeobecné účely. Pokud např. každý ze čtyř zpracovávajících uzlů 8 ve vzorovém počítačovém systému 6 NUMA podporuje maximálně 8 procesorů, pak každá hlavní sada 4 udržovaných registrů příkazů IPI může mít 32 bitů,

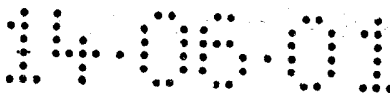


kde bity 0-7 odpovídají procesorům 0-7 zpracovávajícího uzlu 0, bity 8-15 odpovídají procesorům 0-7 zpracovávajícího uzlu 1, atd.

Kromě globálních registrů 90, per-procesorových registrů 92 a registrů 133 příkazů IPI popsanych výše může každá IDU 19 také obsahovat zdroje přerušeni globálního časovače a další registry definované pomocí OpenPIC či jiné registry.

2.1.3 Práce zdrojové jednotky přerušeni (ISU)

Na obr. 5 je znázorněn logický vývojový diagram vyšší úrovně práce jednotky ISU 28 podle tohoto vynálezu. Podle obrázku začíná proces v bloku 140 jako reakce na přijetí vstupu jednotkou ISU 28 a poté pokračuje do bloku 142. Jestliže je vstupem paket přerušeni obdrženy ze sběrnice (tj. z lokálního propojení 16 nebo mezilehlé sběrnice 30), pokračuje proces do bloku 152, který bude popsán níže. Pokud je však vstupem vnější přerušeni (tj. zdrojem přerušeni je vznesen požadavek na spoj požadování přerušeni), pokračuje proces z bloku 142 do bloku 144, který znázorňuje, jak ISU 28 přistupuje do příslušného registru 72 zdrojové konfigurace přerušeni, aby danému přerušeni přiřadila úroveň. ISU 28 pak v bloku 146 určí pohledem do registru 72 zdrojové konfigurace přerušeni, zda jsou přerušeni na úrovni obdrženy vnějšího přerušeni právě maskována nebo ne. Jak je zmíněno výše, v upřednostňovaném provedení tohoto vynálezu je v daném čase v každém zpracovávajícím uzlu 8 aktivní nejvýše jedno přerušeni jakékoliv dané úrovně. Jestliže přerušeni na úrovni přijatého vnějšího přerušeni jsou maskována, ISU 28 neprovede zatím žádnou další akci a zdroj přerušeni musí pokračovat ve vznášení požadavku na



spoj 35 požadování přerušeni nebo vznést požadavek později. Pak se proces vrátí do bloku 142. Pokud se ale v bloku 146 určí, že přerušeni na úrovni přijatého přerušeni nejsou maskována, ISU 28 pošle paket přerušeni lokálním propojením 16 (případně mezilehlou sběrnici 80) k lokální IDU 19 s udáním úrovně přerušeni vektoru přerušeni, jak je znázorněno v bloku 150. Navíc ISU 28 zamaskuje přerušeni na úrovni přijatého přerušeni. Proces se potom vrátí z bloku 150 do bloku 142, který již byl popsán. Pokud tedy není povoleno vedení přerušeni, jak je popsáno níže, jsou všechna vnější přerušeni předávána softwaru hardwarem v tom zpracovávajícím uzlu 8, ve kterém se vnější přerušeni vyskytla.

Co se týče bloku 152, odezvou na přijetí paketu přerušeni na sběrnici ISU 28 určí, zda má nějaké nevyřešené přerušeni na úrovni stanovené v paketu přerušeni. Pokud nemá, je paket přerušeni, který bude zpracován jinou ISU 28, ignorován a proces se vrátí do bloku 142. Pokud se v bloku 152 určí, že ISU 28 má na úrovni stanovené v paketu přerušeni nějaké nevyřešené přerušeni, proces pokračuje do bloku 160. Blok 160 představuje rozhodování, zda transakce přerušeni přijatá jednotkou ISU 28 je či není transakcí EOI nebo zrušení přerušeni. V kladném případě proces postoupí do bloku 162, který znázorňuje, jak ISU 28 smaže masku přerušeni na úrovni stanovené v transakci přerušeni ze sběrnice. Pak se proces vrátí do bloku 142, který byl popsán výše.

Jestliže ISU 28 v bloku 160 naopak určí, že transakce přerušeni přijatá ze sběrnice není transakcí EOI ani zrušení přerušeni, postoupí proces do bloku 170, který představuje rozhodování, zda transakce přerušeni přijatá ze sběrnice je

či není transakcí opakovaného vyslání, která požaduje, aby ISU 28 znovu později vyslala přerušeni na stanovené úrovni. Pokud paket přerušeni přijatý ze sběrnice není transakcí opakovaného vyslání či jinak definovaný paket přerušeni, postoupí proces do bloku 172, který zobrazuje ISU 28, jak provádí funkci správy příslušné chyby. Pokud však transakce přerušeni přijatá ze sběrnice je transakcí opakovaného vyslání, pak proces postoupí do bloku 174. Blok 174 představuje ISU 28, která čeká po dobu časového intervalu v závislosti na implementaci (např. předem stanovený počet hodinových cyklů) než znovu vyšle paket přerušeni k jednotce IDU 19, jak je znázorněno v bloku 150.

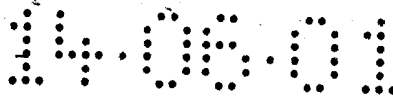
2.1.4 Práce cílové jednotky přerušeni (IDU)

Na obr. 6 je znázorněn logický vývojový diagram vyšší úrovně práce jednotky IDU 19, když zpracovává své vstupy. Jak je vyznačeno, proces začíná v bloku 180 jako reakce na přijetí vstupu jednotkou IDU 19 a poté pokračuje do bloku 182. Blok 182 znázorňuje, jak jednotka IDU 19 určuje, zda je vstupem paket požadování přerušeni vydaný jednotkou ISU 28. Pokud ne, postoupí proces do bloku 200, který je popsán níže. Pokud ale je vstupem paket požadování přerušeni vydaný jednotkou ISU 28, pokračuje proces do bloku 184, který představuje rozhodování, zda úroveň přerušeni stanovená v paketu požadování přerušeni je (1) vyšší než úroveň přerušeni stanovená v registru 122 priorit současné úlohy některého z procesorů 10, který právě neobsahuje přerušeni, v lokálním zpracovávajícím uzlu 8, nebo (2) dostatečně vysoká k získání záznamu ve frontě 130 nevyřešených přerušeni procesoru 10. Pokud není, postoupí proces do bloku 186. Blok 186 znázorňuje jednotku IDU 19, jak vysílá na lokální propojení 16 paket opětovného vyslání přerušeni, který je

přiját a zpracován jednotkou ISU 28, jak je popsáno výše podle obr. 5. Podobný paket opětovného vyslání přerušeni může být poslán, jak je znázorněno v bloku 188, pokud má přerušeni ve frontě 130 nevyřešených přerušeni nižší úroveň než nově přijaté přerušeni a fronta 130 nevyřešených přerušeni je plná, což způsobí, že nevyřešené přerušeni je vytlačeno z fronty 130 nevyřešených přerušeni ve prospěch nového přerušeni.

Po průchodu bloky 184 a 188 pokračuje proces do bloku 190, který znázorňuje IDU 19, jak vznáší požadavek na spoj 36 požadování přerušeni toho procesoru 10, jemuž bylo přerušeni zařazeno do fronty v bloku 184. Navíc, jak znázorňuje blok 192, IDU 19 nastaví v registru 122 priorit současné úlohy příznak „nevyřešeno“ pro úroveň daného přerušeni a nastaví příznak „aktivní“ pro přerušeny procesor. Pak se proces vrátí do bloku 182, který již byl popsán.

Vrátíme-li se do bloku 182, pak pokud vstup přijatý jednotkou IDU 19 není paket požadování přerušeni, IDU 19 v bloku 200 rozhodne, zda přijatá vstupující transakce je transakcí potvrzení přerušeni (ACK) vyslanou na lokální propojení 16 lokálním procesorem 10 k potvrzení přijetí přerušeni. Pokud není, pokračuje proces do bloku 220, který je popsán níže. Pokud však vstup přijatý jednotkou IDU 19 je transakcí potvrzení přerušeni, pak proces pokračuje do bloku 202, který znázorňuje, jak IDU 19 ukončuje svůj požadavek na spoj 36 požadování přerušeni a přesune nevyřešené přerušeni z fronty 130 nevyřešených přerušeni do fronty 132 přerušeni obsluhovaných procesorem tím, že uloží alespoň úroveň přerušeni jako zápis do fronty obsluhovaných přerušeni. Jak je znázorněno v bloku 204, IDU 19 poté vyšle lokálním



propojením 16 transakci přerušeni, která obsahuje úroveň přerušeni a vektor přerušeni, obsluhujícímu procesoru 10. Jestliže je jednotkou IDU 19 z nějakého důvodu přijata transakce přerušeni ACK a neexistuje žádné nevyřešené přerušeni pro daný vysílající procesor 10, je tomuto procesoru 10 dodán nevlastní vektor přerušeni obsažený v registru 108 nevlastního vektoru. Poté se proces vrátí do bloku 182.

Po obsloužení přerušeni vydá obsluhující procesor 10 jednotce IDU 19 transakci zápisu konce přerušeni (EOI), jak je znázorněno na obr. 6, kde proces postoupí z bloku 182 do bloku 200, poté do bloku 220 a poté do bloku 222. Blok 222 zobrazuje, jak IDU 19 zruší příznak „nevyřešeno“ pro úroveň přerušeni obsaženou v transakci zápisu EOI. Jak je uvedeno v bloku 228, IDU 19 také vyšle transakci EOI na lokální propojení 16 ke smazání bitu, který byl danému přerušeni nastaven v registru 82 nevyřešených přerušeni ve zdrojové jednotce ISU 28, jak bylo popsáno výše podle bloků 160 a 162 na obr. 5. Jak je znázorněno v bloku 224, pokud je ve frontě 130 nevyřešených přerušeni přerušeno procesoru 10 přítomno jiné přerušeni, je procesor 10 na přerušeni ve frontě upozorněn a proces postoupí do bloku 190, který již byl popsán. Jinak pokud pro přerušeni procesor 10 nejsou žádná další nevyřešená přerušeni, IDU 19 smaže příznak „aktivní“ v jednotce IDU 19 pro přerušeni procesor 10, jak znázorňuje blok 226. Poté se proces vrátí do bloku 182.

Pokud vstupující transakce přijatá jednotkou IDU 19, stále ještě na obr. 6, není požadavkem přerušeni, transakcí ACK nebo transakcí EOI, IDU 19 v bloku 240 určí, zda vstupující transakce je transakcí zápisu, která směřuje do registru 133 příkazů IPI. Pokud ne, proces postoupí do bloků

260-264, které představují IDU 19, jak provádí další zpracování, jestliže je daný vstup platný, jinak provádí příslušnou činnost zotavení z chyb. Pokud je však vstup transakcí zápisu směřující do registru 133 příkazů IPI, pak ISU 19 rozezná vstup jako spouštěč pro IPI.

Na rozdíl od výše popisovaných vnějších přerušení, IPI může být generováno kterýmkoliv procesorem 10 v počítačovém systému 6 NUMA a může směřovat k sobě samému či k jednomu nebo více dalším procesorům 10 v počítačovém systému 6 NUMA. Takových IPI se typicky využívá k asynchronnímu předávání zpráv mezi procesy běžícími na různých procesorech 10. Aby mohla být přerušení IPI podporována, nastavující software spuštěný při zavedení systému napřed inicializuje úroveň každého ze čtyř podporovaných IPI. Během činnosti počítačového systému 6 NUMA potom zdrojový procesor 10 vybere cílový procesor či procesory 10 jako příjemce zprávy, přičemž v registru 122 priorit současné úlohy tohoto procesoru je udána prahová úroveň IPI každého cílového procesoru 10. Zdrojový procesor 10 pomocí konfiguračních informací a prahové úrovně IPI každého z cílových procesorů určí, které přerušení IPI využije k přerušení zvolených cílových procesorů 10. Zdrojový procesor 10 poté uloží zprávu do oblasti sdílené paměti, k níž lze přistupovat pomocí registru 106 vektoru IPI příslušného vybranému IPI. Nakonec zdrojový procesor 10 vyšle transakci zápisu každému zpracovávajícímu uzlu 8, který obsahuje cílový procesor 8, přičemž každá takováto transakce zápisu směřuje k příslušnému registru 133 příkazů IPI.

Jak bylo pojednáno výše, právě tato transakce zápisu je dekodována jednotkou IDU 19 v bloku 240 na obr. 6. Z bloku 240 postoupí proces do bloku 242, který znázorňuje IDU 19,

jak určuje, jaká priorita (úroveň) přísluší danému cílovému registru 133 příkazů IPI, a jaké procesory 10 přijímají přerušeni této úrovně např. pohledem do registrů 106 vektorů IPI. Jakmile jsou určeny lokální cílové procesory 10, IDU 19 vznesse požadavek na spoj požadování přerušeni cílových procesorů 10, nastaví příznak „nevyřešeno“ pro úroveň přerušeni daného IPI a nastaví příznak „aktivní“ pro cílový procesor 10, jak je uvedeno v blocích 244 a 246. Poté se proces vrací do bloku 182.

2.1.5 Vedení přerušeni (Interrupt Channelling)

Pro některá použití počítačového systému 6 NUMA může být výhodné rozšíření určitých zdrojů jako např. systémové paměti 18, V/V zařízení 32 nebo paměťových zařízení 34, aniž by se rozšířily zpracovávající zdroje počítačového systému 6 NUMA. V takových případech je žádoucí přidat jeden či více uzlů 8 navíc, které neobsahují žádné procesory 10. Avšak vzhledem k výše popsanému rozdělení počítačového systému 6 NUMA na uzlové domény přerušeni je zapotřebí mechanismus ke správě vnějších přerušeni, která jsou generována zdroji přerušeni v uzlech 8 bez procesorů. Podle upřednostňovaného provedení tohoto vynálezu, je dosaženo správy vnějších přerušeni, která jsou generována zdroji přerušeni v uzlech 8 bez procesorů, pomocí vedení přerušeni.

Aby se uskutečnilo vedení přerušeni, lokální IDU 19 (pokud je přítomna) je nečinná a uzlový řadič 20 každého uzlu 8 bez procesorů je nastaven do předávacího módu, ve kterém uzlový řadič 20 uzlu 8 bez procesorů přijímá pakety přerušeni pocházející z lokálních ISU 28 a předává tyto pakety přerušeni určenému „pěstounskému“ uzlu 8, který obsahuje alespoň jeden procesor 10 a jednu IDU 19. Tento

předávací mód může být řízen např. registrem režimu v segmentu 70 oblasti systémového řízení uzlu bez procesoru, který je zapsán konfiguračním softwarem při zavádění systému, přičemž registr režimu obsahuje řídicí bit režimu a identifikátor pěstounského uzlu.

Jako reakci na přijetí transakcí přerušeni předaných přes uzlové propojení 22 spustí uzlový řadič 20 pěstounského uzlu 8 tyto transakce na svém lokálním propojení 16. IDU 19 v pěstounském uzlu 8 poté požaduje pakety přerušeni a předá přerušeni k obslužení lokálním procesorům 10, jak bylo popsáno výše. Jakékoliv pakety přerušeni generované jednotkou IDU 19 v pěstounském uzlu 8 jsou též vysílány zdrojovým jednotkám ISU 28 v uzlu 8 bez procesorů. S využitím vedení přerušeni jsou takto zdroje přerušeni a jednotky ISU vzdálených uzlů 8 bez procesorů obsaženy v doméně přerušeni určeného pěstounského uzlu 8 a vnější přerušeni se spravují s využitím stejných typů transakcí přerušeni jaké se používají ke správě vnějších přerušeni generovaných v pěstounském uzlu 8. Využitím možností neomezujícího přenosu v uzlovém propojení 22 mohou výhodně existovat souběžné vztahy „pěstounský uzel“-„svěřený uzel“ bez porušení nezávislosti domén.

Zvláštní případ vedení přerušeni během zavedení systému se nazývá slévání přerušeni (interrupt funnelling). Při slévání přerušeni jsou všechna vnější přerušeni v počítačovém systému NUMA dočasně směřována ke hlavnímu procesoru, což je první, který bude nastaven. Poté, co byly nastaveny zbývající procesory a jsou tudíž schopné obsluhovat přerušeni, je uplatněno rozdělení do domén přerušeni.

2.2 Software přerušeni

Na obr. 7 je logický vývojový diagram vyšší úrovně, který znázorňuje část konfigurační rutiny pro konfiguraci zdrojů přerušeni podle tohoto vynálezu. Jak je znázorněno, část konfigurační rutiny zobrazená na obr. 7 začíná v bloku 300, přednostně poté, co proběhl počáteční test POST (z angl. Power On Self Test) a další kód hardwarové inicializace nižší úrovně, a pak pokračuje blokem 302. Blok 302 znázorňuje konfigurační rutinu, jak zjišťuje, které uzly 8 počítačového systému 6 NUMA obsahují zařízení schopná generovat vnější přerušeni. Dále, v bloku 304, konfigurační rutina prozkoumá každé zařízení schopné generovat vnější přerušeni, aby určila úroveň přerušeni, kterou každé takové zařízení bude chtít využívat. Konfigurační rutina vyřeší konflikty, pokud jsou, mezi zařízeními a přidělí úroveň každému z přerušeni daných zařízení. Proces pokračuje z bloku 304 do bloku 310, který znázorňuje konfigurační rutinu, jak pro každou příslušnou úroveň přerušeni vytváří v paměti pro všeobecné účely datovou strukturu, která uvádí všechna zařízení, která by mohla generovat vnější přerušeni na dané úrovni přerušeni, uzlový identifikátor každého zařízení a fyzickou adresu každého z registrů daného zařízení. V závislosti na implementačně specifikovaných podrobnostech lze další informace užitečné pro správu přerušeni také uložit v každé datové struktuře.

Konfigurační rutina pak nastaví hardware v každém uzlu 8, jak je uvedeno v blocích 312-334. Poté, co konfigurační rutina vybere uzel 8 v bloku 312, konfigurační rutina určí, zda vybraný uzel 8 obsahuje procesor 10. Pokud ne, konfigurační rutina implementuje vedení přerušeni tím, že vyřadí IDU 19 ve vybraném uzlu 8, jak je zobrazeno v bloku

330, a patřičně nastaví ISU 28 a uzlový řadič 20, např. zapsáním hodnot do paměťově mapovaných registrů. Jak je popsáno výše, konfigurace uzlového řadiče 20 zahrnuje nastavení bitu předávacího módu a specifikaci pěstounského uzlu 8 v předávacím registru režimu. Navíc konfigurační registr přednostně zapíše uzlový identifikátor vybraného uzlu 8 do uzlového identifikačního registru v uzlovém řadiči 20. Poté proces postoupí do bloku 334, který znázorňuje, jak konfigurační rutina určuje, zda zbývají nějaké další uzly k nastavení. Pokud ano, proces se vrací do bloku 312, ve kterém konfigurační registr vybere další uzel 8 ke zpracování.

Co se týče bloku 320, pokud konfigurační rutina určí, že uzel 8 vybraný v bloku 312 obsahuje procesor 10, postoupí proces do bloku 322. Blok 322 znázorňuje, jak konfigurační rutina nastavuje procesor(y) 10, jednotku IDU 19, jednotku/y ISU 28 a uzlový řadič 20 ve vybraném uzlu 8. Jak je uvedeno, konfigurace přednostně obsahuje zapsání uzlového identifikátoru do uzlového identifikačního registru v uzlovém řadiči 20 a zapsání vlastního identifikátoru každého procesoru do vnitřního identifikačního registru procesoru. Poté proces pokračuje do bloku 334 a pokud zbývají ke zpracování další uzly 8, pokračuje ostatními aktivitami nastavování a konfigurace v bloku 336.

Na obr. 8 je logický vývojový diagram vyšší úrovně, který znázorňuje způsob, jakým software správce přerušení první úrovně, zkr. FLIH (z angl. First Level Interrupt Handler) napomáhá obslužení přerušení, které bylo předáno procesoru 10 jednotkou IDU 19. Podle znázornění začíná proces v bloku 400 jako reakce na vznesení požadavku na spoj požadování přerušení jednotkou IDU 19, jak bylo pojednáno

výše v souvislosti s obr. 6. Jako reakci na vznesení požadavku na spoj požadování přerušeni obdrží procesor 10 výjimku a skočí na správce přerušeni první úrovně, který začíná v bloku 402. Blok 402 znázorňuje procesor 10, který pracuje pod řízením FLIH, jak vysílá transakci potvrzení (ACK) přerušeni jednotce IDU 19, aby získal úroveň přerušeni a vektor přerušeni, které má být obslouženo. FLIH také v bloku 403 určí, zda je přerušeni IPI nebo vnější přerušeni. Pokud je přerušeni IPI, proces postoupí do bloku 405, který znázorňuje, jak obsluhující procesor 10 čte z oblasti sdílené paměti dané IPI úrovně zprávu od přerušujícího procesoru 10. Poté proces postoupí do bloku 410, který je popsán níže.

Vrátíme-li se k bloku 403, jako reakce na určení, že přerušeni předané procesoru 10 je vnější přerušeni, postoupí proces do bloku 404. V bloku 404 zamaskuje FLIH přerušeni od jednotky IDU 19, pokud je to implementací vyžadováno, a obdrží softwarový zámek na jakémkoliv exkluzivní zdroje přerušeni potřebné k obsloužení daného přerušeni. Poté FLIH pošle úroveň přerušeni a ukazatel příslušnou na datovou strukturu dané úrovně přerušeni správci přerušeni druhé úrovně, zkr. SLIH (z angl. Second Level Interrupt Handler), jak je uvedeno v bloku 406.

Jak ocení odborníci, SLIH je rutina spravující přerušeni, která provádí operace potřebné k obsloužení přerušeni generovaného konkrétním zařízením. Protože mnoho zdrojů přerušeni může generovat stejnou úroveň přerušeni, jsou tyto rutiny SLIH typicky zřetězeny a tvoří dotazovací řetězec, takže pokud je zpracováván dotazovací řetězec SLIH, každá rutina SLIH se dotazuje svého příslušného (nebo svých příslušných) zařízení, aby určila, zda je zařízení zdrojem

přerušeni, a pokud ano, provede operace nutné k obsluze přerušeni. Tento vynález uznává, že latence správy přerušeni je vysoce závislá na délce dotazovacího řetězce, který je zase závislý na počtu úrovní externích přerušeni a počtu potenciálních zdrojů přerušeni v počítačovém systému NUMA. Pokud tedy počítačový systém 6 NUMA má pouze 16 úrovní externích přerušeni a počet potenciálních zdrojů přerušeni v počítačovém systému 6 NUMA je vysoký, bude vysoká latence správy přerušeni. Aby se docílilo lepší latence správy přerušeni, omezuje tento vynález počet SLIH v dotazovacím řetězci eliminací zdrojů v jednom či více uzlech jakožto kandidátů na zdroj přerušeni.

V prvním provedení je počet SLIH v dotazovacím řetězci omezen tím, že FLIH namapuje úroveň přerušeni jako úroveň přerušeni specifickou pro daný uzel (nebo nadřazenou úroveň), která je tvořena zřetězením (či jiným sloučením) uzlového identifikátoru, ve kterém se přerušeni vyskytlo a který je znám procesoru 10, jenž přerušeni obdržel, s konvenční úrovní přerušeni. Každá taková úroveň přerušeni specifická pro daný uzel by měla odpovídající datovou strukturu přerušeni, vytvořenou v paměti konfigurační rutinou, kde by datová struktura uváděla pouze ta zařízení v příslušném uzlu (tj. doméně přerušeni), která by mohla generovat vnější přerušeni dané úrovně. Úroveň přerušeni předaná první rutině SLIH v dotazovacím řetězci v bloku 406 by byla úroveň přerušeni specifická pro daný uzel, ukazatel předaný rutině SLIH v bloku 406 by ukazoval na datovou strukturu úrovně specifické pro daný uzel a dotazovací řetězec by zahrnoval SLIH příslušné zařízením, která jsou uvedena v datové struktuře přerušeni specifického pro daný uzel. Toto první provedení je výhodné v tom, že může být souběžně spuštěno několik správců přerušeni stejné úrovně na

procesorech 10 na různých uzlech 8, aniž by vznikaly konflikty (nebo by musely být uděleny zámky) ohledně zdrojů obsluhujících přerušeni, avšak vyžaduje, aby rutiny FLIH a SLIH rozeznaly úrovně přerušeni specifické pro dané uzly.

Počet SLIH v dotazovacím řetězci může být eventuálně omezen podle druhého provedení, ve kterém sama FLIH předává podmnožinu datové struktury přerušeni rutinně SLIH, kde tato podmnožina datové struktury přerušeni uvádí pouze zařízení, která mají stejný uzlový identifikátor jako procesor, jemuž je vnější přerušeni předáváno. Jelikož se neberou v úvahu zařízení v jiných uzlech, bude pravděpodobně dotazovací řetězec SLIH kratší. Jakékoliv z těchto provedení lze využít společně s vedením přerušeni popsaným výše, v tomto případě datová struktura sestavená konfigurační rutinou pro doménu přerušeni bude obsahovat zařízení v rámci pěstounského uzlu i svěřeného uzlu.

V každém případě jakmile bylo předáno řízení první SLIH v dotazovacím řetězci, čeká FLIH na dokončení obsluhy přerušeni, jak je uvedeno v bloku 408. Je důležité, že jakmile bylo přerušeni předáno dotazovacímu řetězci rutin SLIH, může operační systém rozvrhnout spuštění těchto SLIH na jakémkoliv procesoru 10 v počítačovém systému 6 NUMA a může vybrat jiný procesor 10 ke spuštění SLIH v reakci na vyvažování čtení, afinitu dat nebo jiná kritéria. Po skončení rutiny SLIH příslušné zdroji přerušeni je vrácena kontrola rutinně FLIH v procesoru 10, který původně přerušeni obdržel, a ten pošle jednotce IDU 19 EOI transakci, která udává úroveň obsluhovaného přerušeni, jak je uvedeno v bloku 410 a popsáno výše v souvislosti s blokem 220 na obr. 6. Poté FLIH skončí v bloku 412.

Jak bylo popsáno, tento vynález zajišťuje architekturu přerušení pro počítačový systém NUMA. Architektura přerušení, jež zahrnuje složky hardwaru i softwaru může být obecně popsána jako rozdělení počítačového systému NUMA do domén vnějších přerušení tak, že vnější přerušení je vždy předáváno procesoru v té doméně vnějšího přerušení, v níž se přerušení vyskytuje. I když každá taková doména vnějších přerušení typicky obsahuje pouze jediný uzel, lze implementovat vedení přerušení nebo slévání přerušení ke směrování vnějších přerušení přes hranice uzlů pro předání procesoru. Po předání procesoru se může spustit na jakémkoliv procesoru v systému software k obslužení přerušení. Architektura přerušení podle tohoto vynálezu výhodně umožňuje softwaru správy přerušení promptně obsloužit vnější přerušení tím, že se omezí velikost dotazovacího řetězce (stromu) správy přerušení v porovnání s metodami podle předchozího stavu techniky. Kromě vnějších přerušení podporuje architektura přerušení podle tohoto vynálezu meziprocessorová přerušení (IPI), jimiž může jakýkoliv procesor přerušit sám sebe nebo jeden či více procesorů v systému. Tento vynález využívá paměťově mapovaných registrů ke spouštění IPI, což usnadňuje přenos IPI přes hranice uzlů a umožňuje spustit všesměrová IPI prostým vysláním jedné transakce zápisu každému uzlu obsahujícímu procesor, který má být přerušen. Je důležité, že architekturu přerušení podle tohoto vynálezu lze škálovat od malých počítačových systémů NUMA obsahujících několik uzlů po velké systémy obsahující stovky uzlů. Hardware přerušení v každém uzlu je také rozmístěn za účelem škálovatelnosti, přičemž hardwarové součásti komunikují transakcemi přerušení, které jsou přenášeny sdílenými komunikačními cestami (tj. lokální sběrnice a propojení).

I když byl tento vynález popsán s ohledem na provedení vyhovující OpenPIC, je dobré si uvědomit, že tento vynález není omezen na systémy vyhovující OpenPIC. Navíc, přestože byly aspekty tohoto vynálezu popsány s ohledem na počítačový systém, který spouští software řídicí způsob tohoto vynálezu, je dobré si uvědomit, že tento vynález lze alternativně implementovat jako počítačový programový produkt pro využití v počítačovém systému. Programy, které určují funkce tohoto vynálezu, lze dodávat počítačovému systému celou řadou nosičů signálu, které zahrnují, bez omezení, nezapisovatelná paměťová média (např. CD-ROM), zapisovatelná paměťová média (např. floppy disky, harddisky, EEPROM) a komunikační média jako jsou např. počítačové a telefonní sítě. Je dobré si tudíž uvědomit, že takovéto nosiče signálu, pokud nesou či kódují instrukce čtené počítačem, které řídí funkce tohoto vynálezu, představují alternativní provedení tohoto vynálezu.

Zastupuje:

Dr. Petr Kalenský v.r.

Dr. Petr Kalenský
IČO 000000000
Číslo rejstříku

P A T E N T O V É N Á R O K Y

1. Systém zpracování dat, **vyznačující se tím**, že obsahuje množství domén přerušeni, z nichž každá obsahuje alespoň jeden z množství propojených zpracovávajících uzlů, kde každá doména přerušeni obsahuje alespoň jeden procesor schopný přijímat vnější přerušeni a alespoň jeden zdroj přerušeni schopný generovat vnější přerušeni, přičemž každé z množství domén přerušeni má příslušný hardware přerušeni, který přijímá vnější přerušeni generovaná alespoň jedním zdrojem přerušeni a předává vnější přerušeni alespoň jednomu procesoru, přičemž alespoň jeden procesor spouští software správy přerušeni, který může obsluhovat přerušeni předaná jednak procesoru ve stejné doméně přerušeni jako je alespoň tento jeden procesor jednak procesoru v jiné doméně přerušeni než je alespoň tento jeden procesor.

2. Systém zpracování dat podle nároku 1, **vyznačující se tím**, že hardware v každé z mnoha domén přerušeni obsahuje cílovou jednotku přerušeni, která předává přerušeni procesorům pouze ve své doméně přerušeni, a alespoň jednu zdrojovou jednotku přerušeni, která přijímá přerušeni od zdrojů přerušeni.

3. Systém zpracování dat podle nároku 2, **vyznačující se tím**, že cílová jednotka přerušeni a zdrojová jednotka přerušeni posílají informace o přerušeni sdíleným propojením.

4. Systém zpracování dat podle nároku 2,

vyznačující se tím, že alespoň pro jednu doménu přerušeni z množství domén přerušeni jsou alespoň cílová jednotka přerušeni a jedna zdrojová jednotka přerušeni umístěny v jiných zpracovávajících uzlech v množství propojených zpracovávajících uzlů.

5. Systém zpracování dat podle nároku 4, **vyznačující se tím**, že jeden z množství propojených zpracovávajících uzlů, obsahující alespoň jednu zdrojovou jednotku přerušeni, neobsahuje žádné procesory pro přijímání vnějších přerušeni.

6. Systém zpracování dat podle nároku 2, **vyznačující se tím**, že alespoň jedna z množství domén přerušeni obsahuje množství zdrojových jednotek přerušeni.

7. Systém zpracování dat podle nároku 1, **vyznačující se tím**, že hardware přerušeni v každé doméně přerušeni obsahuje globálně přístupný paměťově mapovaný registr, využívaný k přenosu přerušeni mezi doménami přerušeni.

8. Systém zpracování dat podle nároku 7, **vyznačující se tím**, že globálně přístupný paměťově mapovaný registr se využívá k přenosu meziprocesorových přerušeni.

9. Systém zpracování dat podle nároku 7, **vyznačující se tím**, že globálně přístupnému paměťově mapovanému registru každé z domén přerušeni je přidělena příslušná fyzická adresa a fyzická adresa globálně přístupného paměťově mapovaného registru každé z domén přerušeni má jednotný ofset od oblasti paměti přidělené

zpracovávajícímu uzlu, který obsahuje globálně přístupný paměťově mapovaný registr.

10. Způsob správy vnějšího přerušeni v systému zpracování dat, **vyznačující se tím**, že způsob obsahuje:

zřízení množství domén přerušeni z nichž každá obsahuje alespoň jedno z množství propojených zpracovávajících uzlů, kde každá doména přerušeni obsahuje alespoň jeden procesor schopný přijímat vnější přerušeni a alespoň jeden zdroj přerušeni schopný generovat vnější přerušeni, přičemž každá z množství domén přerušeni má příslušný hardware přerušeni;

v konkrétní doméně přerušeni z množství domén přerušeni přijetí vnějšího přerušeni generovaného alespoň jedním zdrojem přerušeni v hardwaru přerušeni a předání vnějšího přerušeni alespoň jednomu procesoru pomocí hardwaru přerušeni;

spuštění, alespoň jedním procesorem v konkrétní doméně přerušeni, softwaru správy přerušeni, který může obsloužit vnější přerušeni předané alespoň jednomu procesoru a vnější přerušeni předané procesoru v jiné doméně z množství domén přerušeni než v konkrétní doméně přerušeni.

11. Způsob podle nároku 10, **vyznačující se tím**, že hardware přerušeni v každé z mnoha domén přerušeni obsahuje cílovou jednotku přerušeni a alespoň jednu zdrojovou jednotku přerušeni, přičemž přijímání vnějšího přerušeni obsahuje přijímání vnějšího přerušeni alespoň jednou zdrojovou jednotkou přerušeni a předávání vnějšího přerušeni obsahuje předávání vnějšího přerušeni alespoň jednomu procesoru s využitím cílové jednotky přerušeni

12. Způsob podle nároku 11, **vyznačující se**

tím, že dále obsahuje předávání informací o přerušení mezi cílovou jednotkou přerušení a zdrojem přerušení sdíleným propojením.

13. Způsob podle nároku 12, **vyznačující se tím**, že alespoň pro jednu doménu přerušení z množství domén přerušení obsahuje předávání informace o přerušení sdíleným propojením, které propojuje alespoň dva z množství zpracovávajících uzlů.

14. Způsob podle nároku 13, **vyznačující se tím**, že zřizování množství domén přerušení obsahuje zřizování alespoň jedné domény přerušení, v níž jeden z množství propojených zpracovávajících uzlů obsahuje alespoň jednu zdrojovou jednotku přerušení a žádné procesory pro přijímání vnějších přerušení.

15. Způsob podle nároku 11, **vyznačující se tím**, že zřizování množství domén přerušení obsahuje zřízení alespoň jedné z množství domén přerušení obsahující množství zdrojových jednotek přerušení.

16. Způsob podle nároku 10, **vyznačující se tím**, že dále obsahuje přenos přerušení mezi doménami přerušení využitím globálně přístupného paměťově mapovaného registru v hardwaru přerušení.

17. Způsob podle nároku 16, **vyznačující se tím**, že přenos přerušení mezi doménami přerušení obsahuje přenos meziprocessorových přerušení mezi doménami přerušení.

18. Způsob podle nároku 16, **vyznačující se tím**, že dále obsahuje:

přidělení globálně přístupnému paměťově mapovanému registru každé domény přerušení příslušnou fyzickou adresu, přičemž fyzická adresa globálně přístupného paměťově mapovaného registru každé domény přerušení má jednotný offset od oblasti paměti přidělené zpracovávajícímu uzlu, který obsahuje globálně přístupný paměťově mapovaný registr.

19. Způsob zpracování přerušení v systému zpracování dat, který obsahuje množství propojených uzlů, kde každý z množství propojených uzlů obsahuje zařízení, které generuje přerušení, a zařízení ve více uzlech mohou generovat přerušení stejné úrovně, přičemž způsob **se vyznačuje tím**, že

odezvou na předání přerušení procesoru k obslužení, přičemž přerušení má nějakou úroveň, získání seznamu zařízení schopných generovat přerušení této úrovně;

dotázání se pouze zařízení ze seznamu, která jsou umístěna ve stejné doméně přerušení jako procesor, aby se zjistilo, které zařízení ze seznamu generovalo přerušení.

20. Způsob podle nároku 19, **vyznačující se tím**, že dále obsahuje poté spuštění správce přerušení spojeného s identifikovaným zařízením.

21. Způsob podle nároku 19, **vyznačující se tím**, že dále obsahuje:

před předáním přerušení vytvoření a uložení seznamu do prostoru globální paměti, přístupného všem z množství propojených uzlů.

22. Způsob podle nároku 21, **vyznačující se tím**, že seznam obsahuje pouze zařízení v jediné doméně přerušení.

23. Systém zpracování dat, **vyznačující se tím**, že obsahuje

množství propojených uzlů, přičemž každý z množství propojených uzlů obsahuje zařízení, které generuje přerušení, a zařízení na více uzlech mohou generovat přerušení stejné úrovně, přičemž alespoň jeden z množství propojených uzlů obsahuje procesor; a

software správy přerušení uložený v systému zpracování dat a spustitelný procesorem, přičemž software správy přerušení, odezvou na to, že je předáno procesoru přerušení mající nějakou úroveň, získá seznam zařízení schopných generovat přerušení dané úrovně a dotazuje se pouze zařízení v daném seznamu, která se nacházejí ve stejné doméně přerušení jako daný procesor, aby zjistil, které zařízení ze seznamu generovalo dané přerušení.

24. Systém zpracování dat podle nároku 23, **vyznačující se tím**, že software správy přerušení je správce přerušení první úrovně, přičemž systém zpracování dat dále obsahuje správce přerušení druhé úrovně uloženého v systému zpracování dat a spustitelného procesorem, kde správce přerušení druhé úrovně je spojen se zařízením a správce přerušení první úrovně volá správce přerušení druhé úrovně k obsluze identifikovaného zařízení.

25. Systém zpracování dat podle nároku 23, **vyznačující se tím**, že dále obsahuje prostor globální paměti přístupný všem z množství propojených uzlů, přičemž seznam je uložen v prostoru globální paměti před předáním přerušení.

26. Systém zpracování dat podle nároku 25,

vyznačující se tím, že seznam obsahuje pouze zařízení v jediné doméně přerušeni.

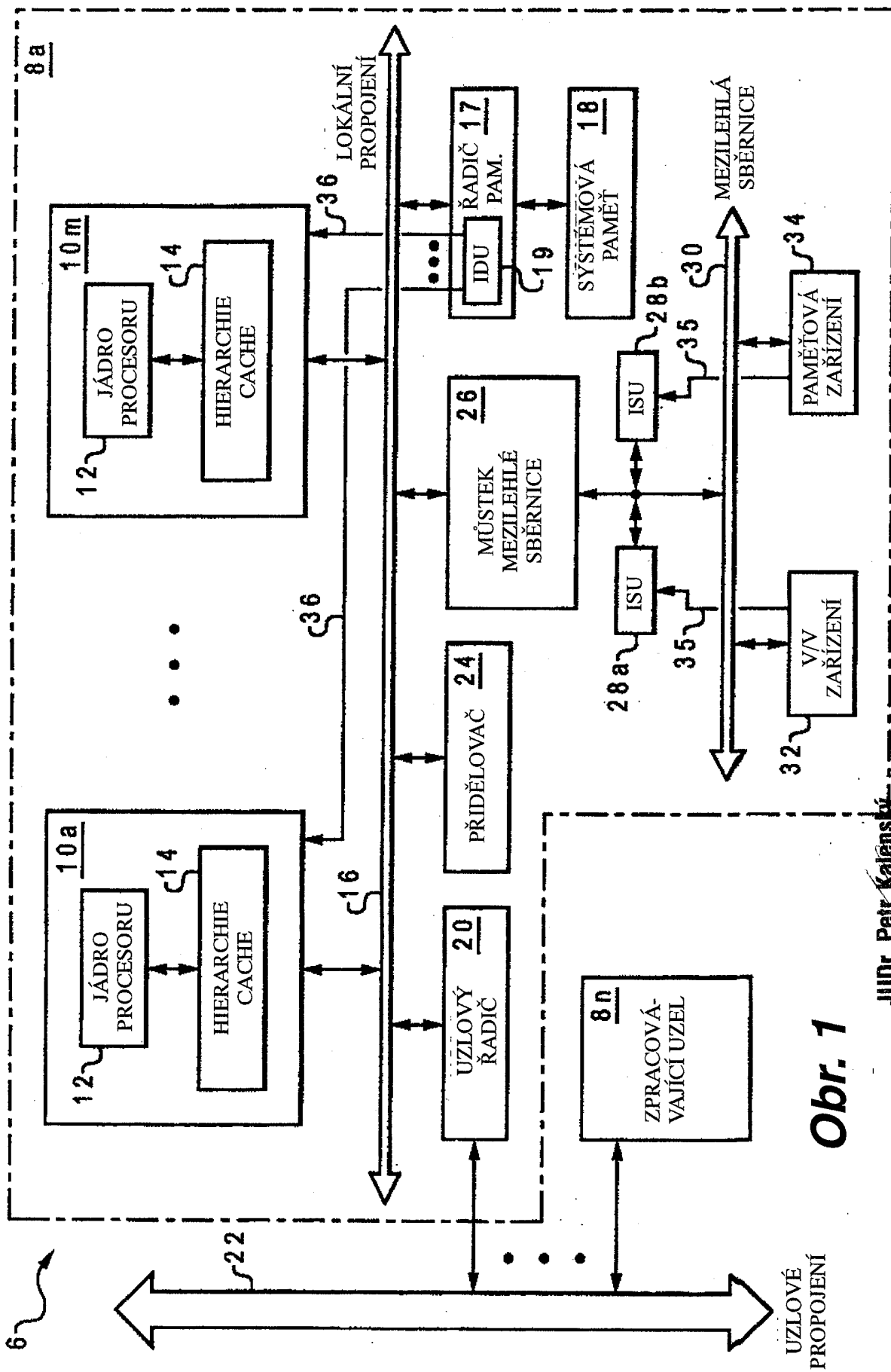
27. Programový produkt pro použití systémem zpracování dat, který obsahuje množství propojených uzlů, přičemž každý z množství propojených uzlů obsahuje zařízení generující přerušeni a zařízení ve více uzlech mohou generovat přerušeni stejné úrovně, kde alespoň jeden z mnoha propojených uzlů obsahuje procesor, přičemž programový produkt **se vyznačuje tím**, že obsahuje:

počítačem využitelné médium ; a

software správy přerušeni zakódovaný na počítačem využitelném médiu a spustitelný systémem zpracování dat, přičemž software správy přerušeni, odezvou na to, že je procesoru předáno přerušeni mající nějakou úroveň, získá seznam zařízení schopných generovat přerušeni dané úrovně a dotazuje se pouze zařízení v seznamu, umístěných ve stejné doméně přerušeni jako procesor, aby zjistil, které zařízení ze seznamu generovalo dané přerušeni.

28. Programový produkt podle nároku 27, **vyznačující se tím**, že software správy přerušeni je správce přerušeni první úrovně, přičemž programový produkt dále obsahuje správce přerušeni druhé úrovně zakódovaný na počítačem využitelném médiu, kde správce přerušeni druhé úrovně je spojen se zařízením, a správce přerušeni první úrovně volá správce přerušeni druhé úrovně k obsluze identifikovaného zařízení.

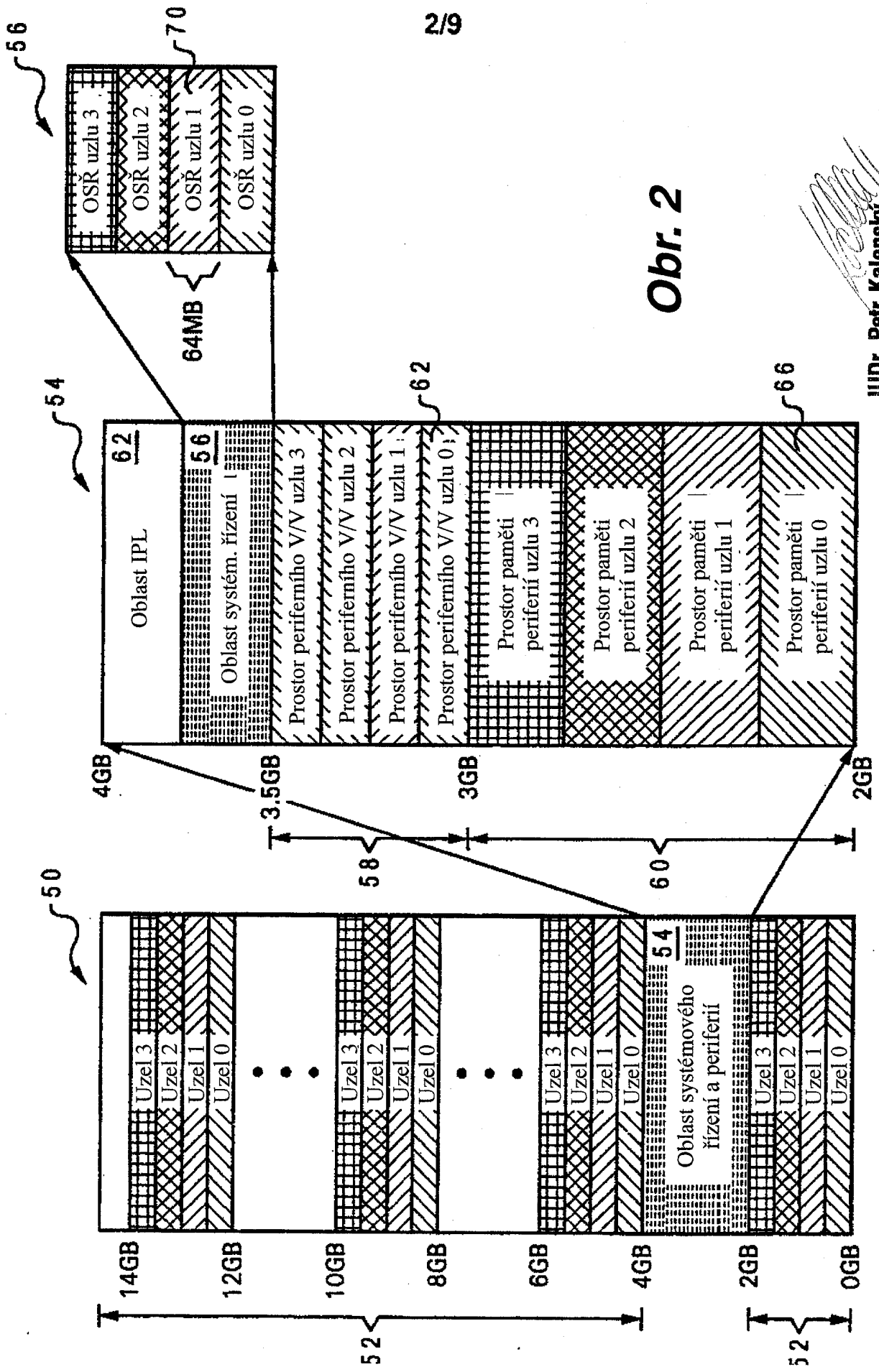
29. Programový produkt podle nároku 27, **vyznačující se tím**, že dále obsahuje konfigurační rutinu, zakódovanou na počítačem využitelném médiu, která vytváří seznam v prostoru globální paměti přístupném všem



Obr. 1

JUDr. Petr Kalenský

abovokar

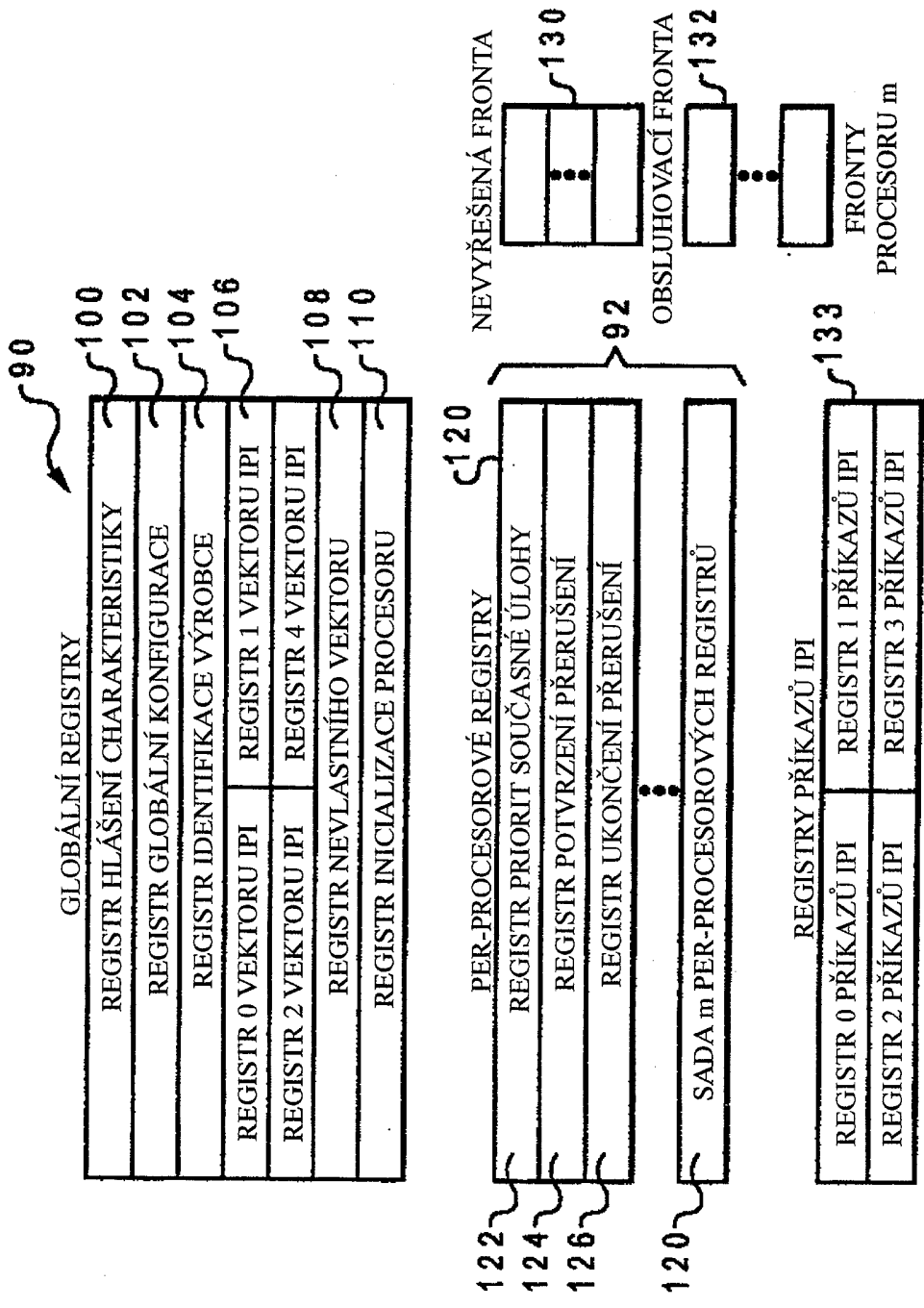


Obr. 2

[Handwritten signature]
 JUDr. Petr Kalenský
 advokát

CÍLOVÁ JEDNOTKA PŘERUŠENÍ (IDU)

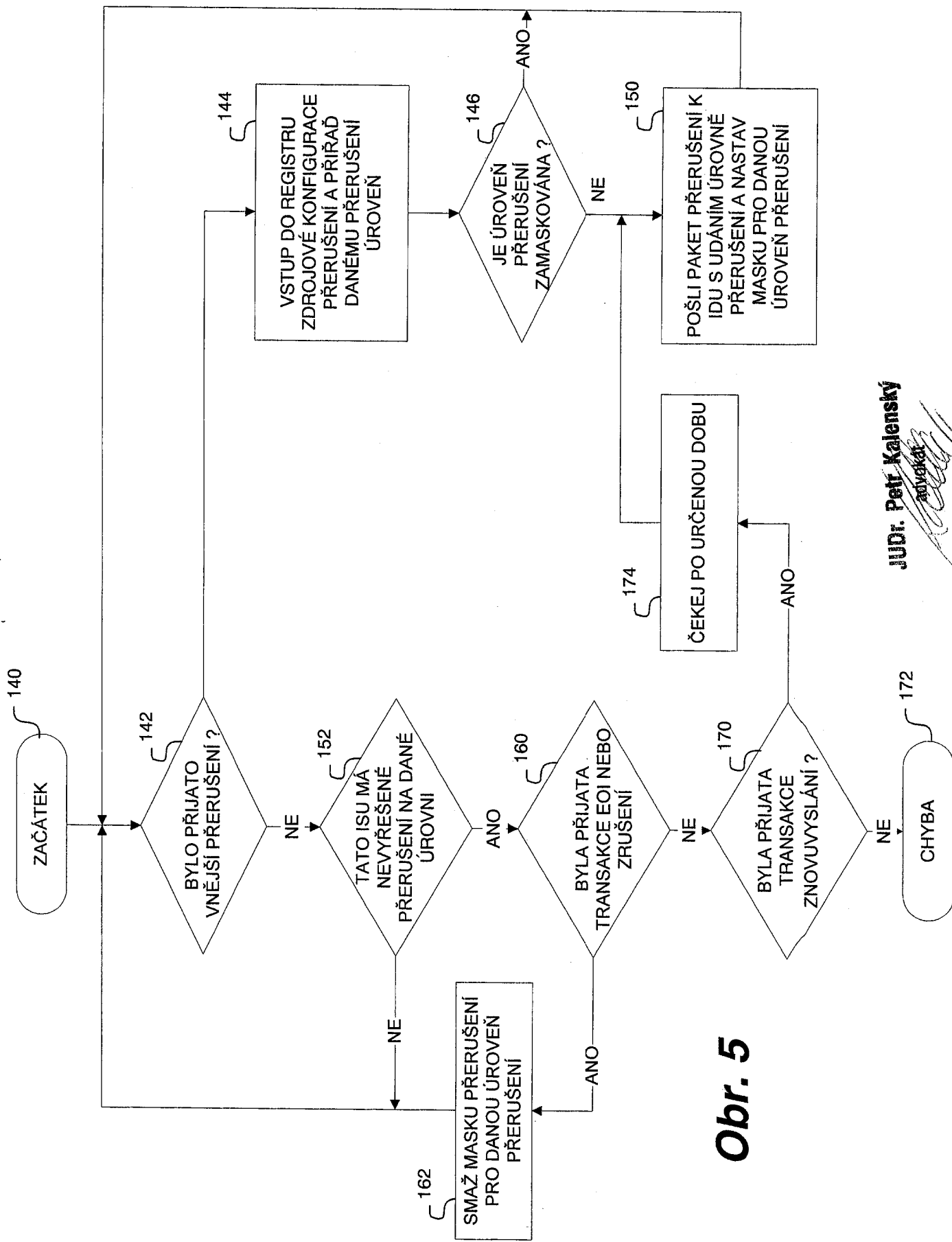
19



Obr. 4

JUDr. Petr Kalenský
advokát

4. 8. 01

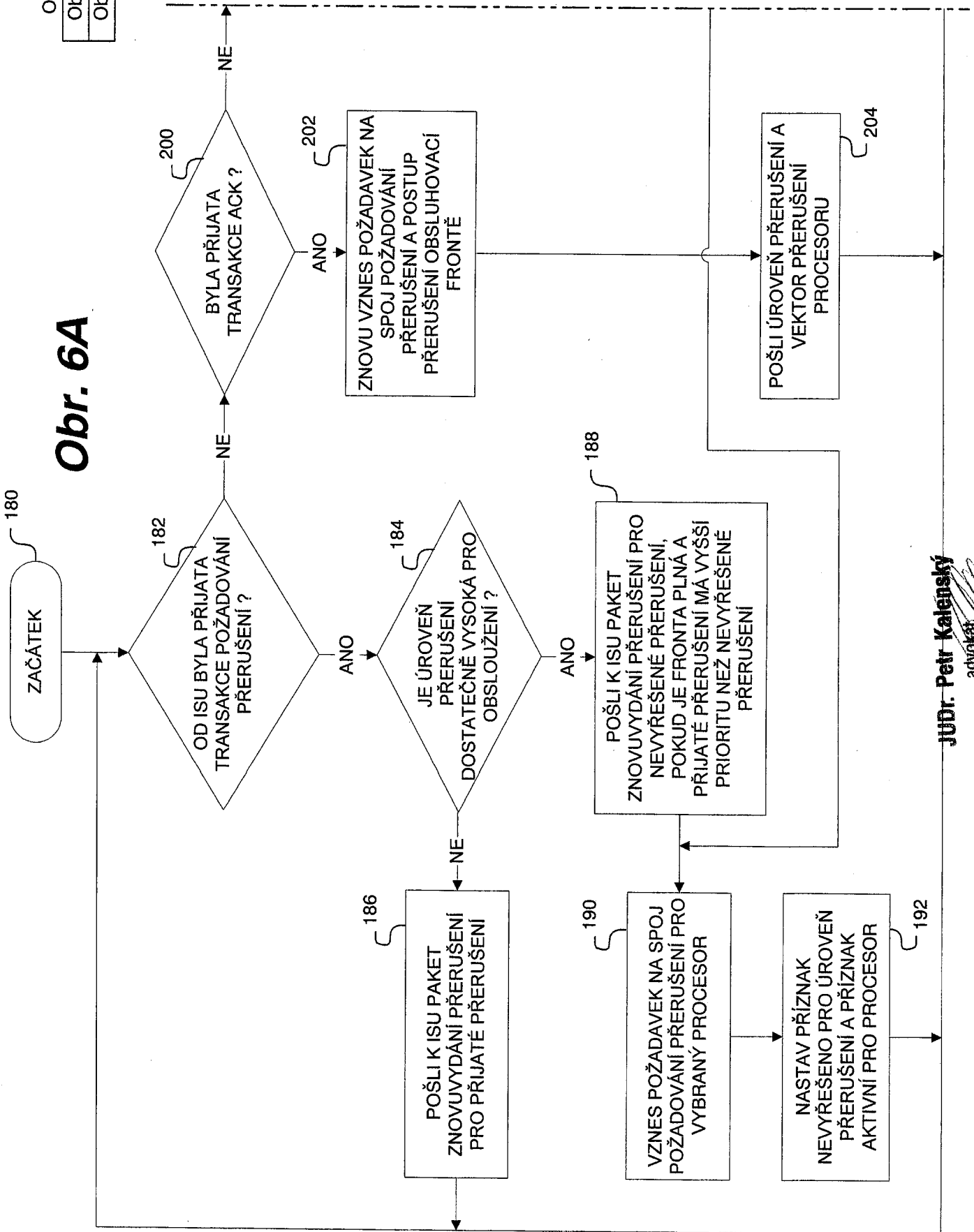


Obr. 5

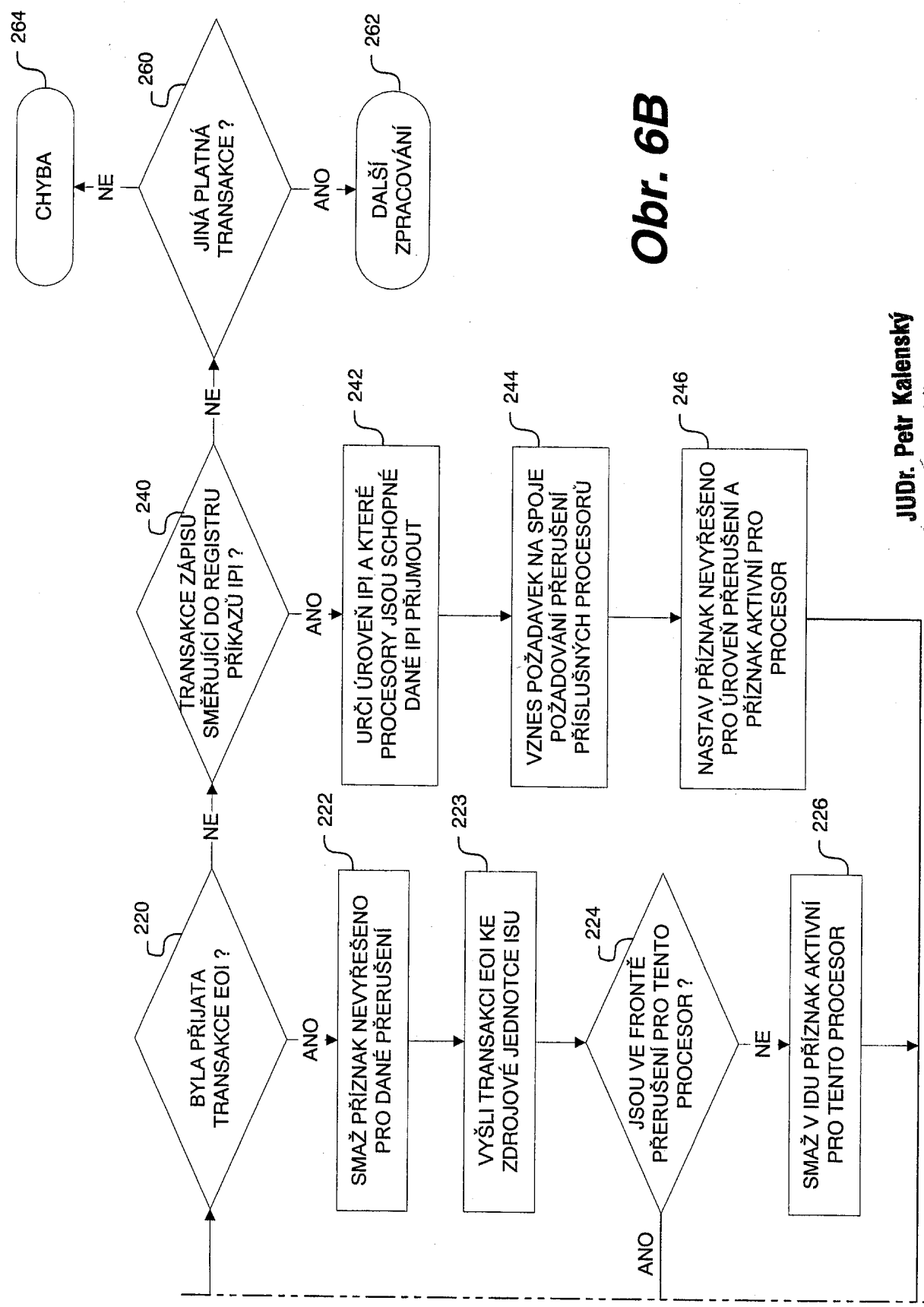
JUDr. Petr Kalenský
advokát

Obr. 6
Obr. 6A
Obr. 6B

Obr. 6A



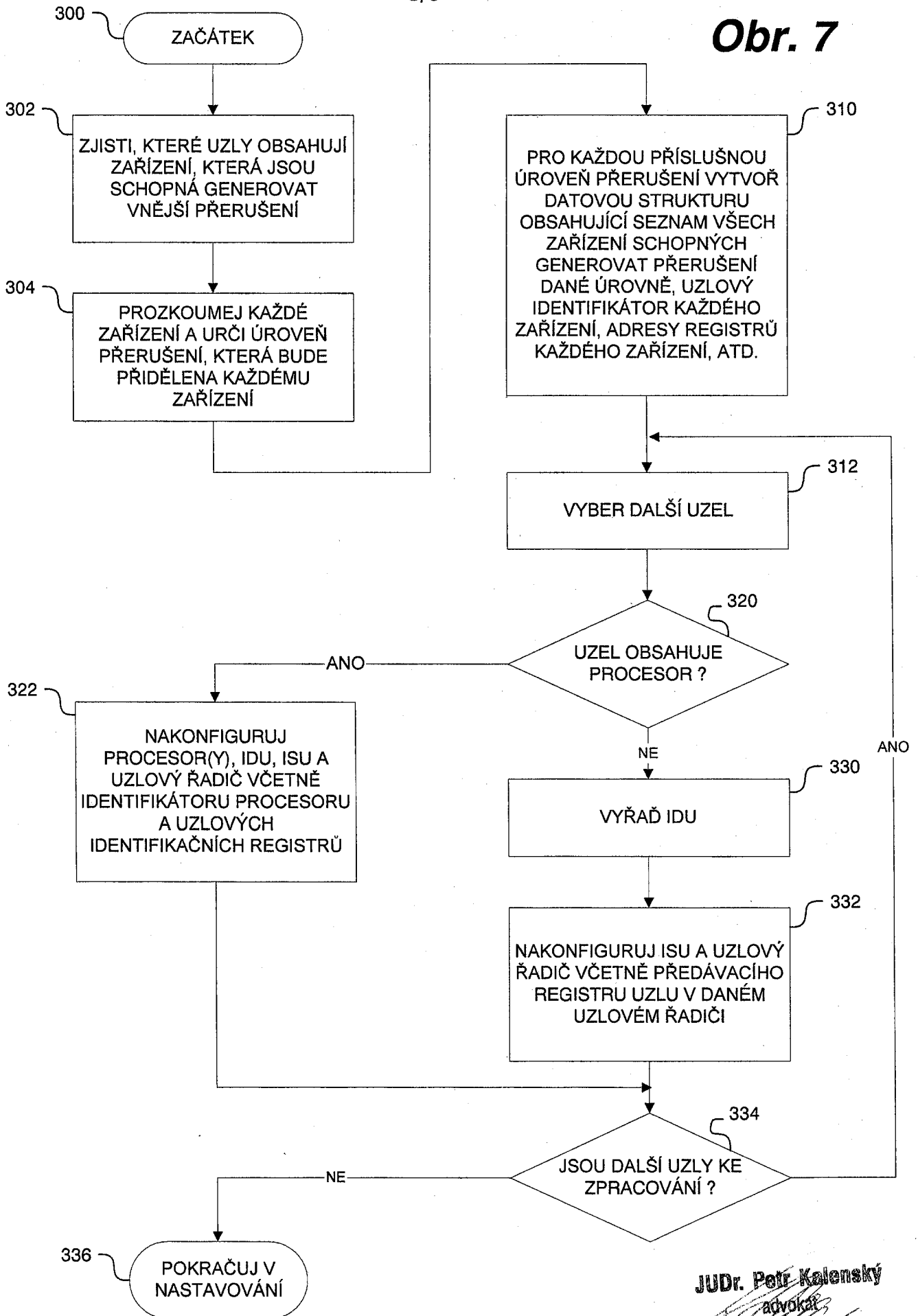
JUDr. Petr Kalenský
advokát

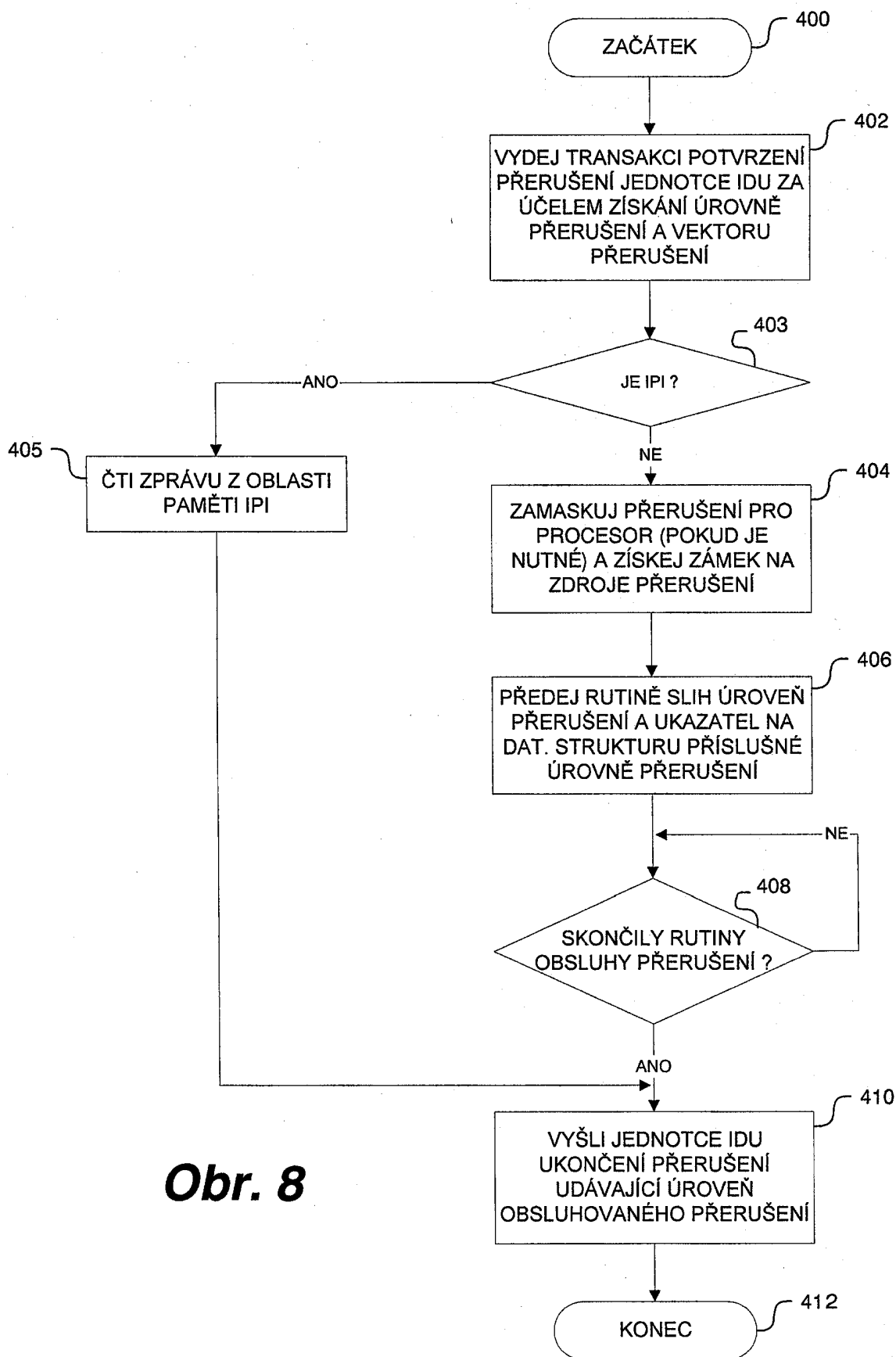


Obr. 6B

JUDr. Petr Kalenský
advokát







Obr. 8