PATENTOVÝ SPIS

(11) Číslo dokumentu:

294 898

(13) Druh dokumentu: **B6**

(51) Int. Cl. :[7]
H 04 L 9/00
H 04 L 9/28

(19)
ČESKÁ
REPUBLIKA

ÚŘAD
PRŮMYSLOVÉHO
VLASTNICTVÍ

(21) Číslo přihlášky: **2002-4116**
(22) Přihlášeno: **16.12.2002**
(40) Zveřejněno: **18.08.2004**
(Věstník č. 08/2004)
(47) Uděleno: **07.02.05**
(24) Oznámení o udělení ve Věstníku: **13.04.2005**
(Věstník č. 4/2005)

(73) Majitel patentu:
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ FAKULTA ELEKTROTECHNICKÁ, Praha, CZ
LÓRENCZ Róbert Ing. CSc., Poděbrady, CZ

(72) Původce:
Lórencz Róbert Ing. CSc., Poděbrady, CZ

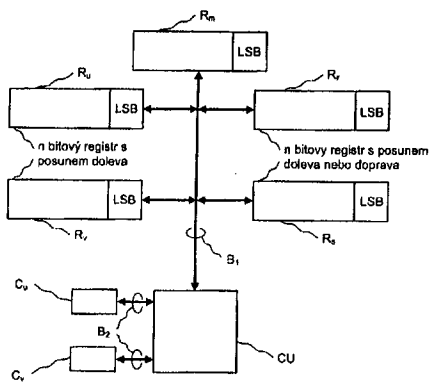
(74) Zástupce:
Ing. Hana Dušková, Travná 1285, Praha 14, 19800

(54) Název vynálezu:

Zapojení pro generování multiplikativní inverze nad konečným tělesem GF(p)

(57) Anotace:

Podstata vynálezu spočívá ve vytvoření zapojení pro efektivní generování multiplikativní inverze nad konečným tělesem GF(p), kde p je prvočíslo, tj. generováním modulární inverze. Zapojení je upraveno pro binární vykonávání operací v procesu generování modulární inverze, a to vzhledem k co nejmenšímu počtu operací sčítání, odečítání a posuvu. Postup realizovaný navrženým zapojením odstraňuje redundantní operace pro konverzi lichých a záporných hodnot, které jsou prováděny u dosavadních postupů. K tomu se využívá reprezentace záporných čísel v doplňkovém kódu, posun hodnot doleva v řídicí části rozšířeného Euklidova algoritmu a nová definice hlídacích a řídicích podmínek provádění postupu. Minimalizování počtu operací sčítání a odečítání je žádoucí v případě počítání s velkými čísly, která se vyskytují v kryptografii.



CZ 294898 B6

Zapojení pro generování multiplikativní inverze nad konečným tělesem GF(p)

Oblast techniky

Vynález se týká zapojení pro generování multiplikativní inverze nad konečným tělesem GF(p), kde p je prvočíslo, odvozené z rozšířeného Euklidova algoritmu pro výpočet největšího společného dělitele. Vynález má význam ve kryptografických výpočtech, zejména v kryptografických hardwarových aplikacích a embedded systémech, např. SMART kartách.

Dosavadní stav techniky

Základní aritmické operace modulární aritmetiky, tj. sčítání, odečítání, násobení a modulární inverze, kde modul je prvočíslo, jsou přirozenou a neoddělitelnou součástí kryptografických algoritmů jako například šifrovací operace v RSA algoritmech, kryptografických algoritmů požadovaných v US Government Digital Signature Standard (NIST) a taktéž v současnosti často používané kryptografii využívající eliptických křivek.

Multiplikativní inverze nad konečným tělesem GF(p), kde p je prvočíslo, má zvláště důležitý význam ve výpočtech operací s body na eliptických křivkách definovaných nad konečným tělesem GF(p) a při akceleraci exponenciálních operací.

Multiplikativní inverze nad konečným tělesem GF(p), tj. modulární inverze celého čísla q z $\langle 1, p-1 \rangle$ modulo p, kde p je prvočíslo, je definována jako celé číslo b z $\langle 1, p-1 \rangle$ takové, že platí $q.b \equiv 1 \pmod{p}$, často zapisované jako $b = q^{-1} \bmod p$. Nejpoužívanější přístupy pro generování modulární inverze je jednak tzv. klasická inverze podle Knutha a lze ji najít v publikaci D. E. Knuth: „The Art of Computer Programming 2, Seminumerical Algorithms, Addison–Wesley, Reading, Mass. Third edition (1998)“ a jednak způsob generování modulární inverze založený na tzv. Montgomeryho modulární inverzi, který lze najít v publikaci B. S. Kaliski Jr.: „The Montgomery Inverze and Its Application. IEEE Transaction on Computers 44 No. 8 (1995)“. Oba tyto přístupy vycházejí z rozšířeného Euklidova algoritmu. Využívají binárních operací sčítání, odečítání a dělení respektive násobení dvěma, přičemž operace dělení dvěma respektive násobení dvěma je operací posunu o jedno místo doprava respektive doleva pro binární reprezentaci dělence respektive činitele. Tyto vlastnosti u obou přístupů umožňují jejich snadnou hardwarovou implementaci.

Při generování tzv. klasické modulární inverze se průběžně podle algoritmu provádí půlení hodnot jejich posunem vpravo, a to jak sudých, tak lichých. Tato operace se provádí tak, že v případě liché hodnoty je tato hodnota nejdříve převedena na sudou přičtením hodnoty modulu p, která je prvočíselná a tudíž lichá a následně je proveden posun vpravo. Pokud se vyskytne v průběhu generování záporná hodnota v důsledku odečítání, je převedena na kladnou přičtením hodnoty p, což představuje operaci převodu záporného čísla na kladné modul p.

V případě generování modulární inverze s použitím Montgomeryho algoritmu je půlení, tj. dělení dvěma v průběhu vykonávání Euklidova algoritmu odložené do druhé fáze algoritmu. V této druhé fázi jsou vykonávána půlení modulo p, a to tak, že liché hodnoty jsou opět nejdříve převedeny na sudé přičtením hodnoty p.

Výše uvedené způsoby generování modulární inverze vykazují některé nevýhody. V případě tzv. klasického generování je to hlavně velký počet testů typů „větší/menší než“, které v podstatě představují operaci odečítání. V případě záporných hodnot je prováděn jejich převod na kladné hodnoty a v případě výskytu lichých hodnot v procesu působení je prováděn jejich převod na sudé hodnoty. Obě operace představují opět operaci sčítání.

V případě generování modulární inverze s využitím Montgomeryho metody jsou nevýhodami redundantní operace posuvu ve druhé fázi generování, operace sčítání v případě převodu lichých čísel na suché při vykonávání odloženého půlení ve druhé fázi generování a velký počet testů typů „větší/menší než" představujících operaci odečítání.

Podstata vynálezu

Výše uvedené nedostatky odstraňuje zapojení pro generování multiplikativní inverze $b = q^{-1} \bmod p$ nad konečným tělesem GP(p) podle předkládaného vynález, kde (p) je prvočíslo větší než (q) kde (q) je celé kladné číslo větší než jedna.

Podstatou předkládaného vynálezu je, že zapojení sestává z řídicí a výpočetní jednotky, která je propojena s prvním až pátým n–bitovým registrem pro datovou a řídicí sběrnici, která současně tyto registry propojuje navzájem. Dále je řídicí a výpočetní jednotka propojena druhou řídicí datovou sběrnici s prvním e–bitovým čítačem a s druhým e–bitovým čítačem, kde $e = [\log_2 n]$ a kde pro počet bitů n platí, že $2^{n-1} > p$. Dále platí, že n–bitové hodnoty proměnných obsažené v prvním až pátém registru v celém průběhu generování multiplikativní inverze jsou reprezentované v doplňkovém kódu a nejméně významový bit LSB je v nich umístěn při inicializaci úplně vpravo. Inicializační počáteční stavy prvního až pátého registru vyvolané pomocí příslušných proměnných (p) a (q) jsou nastaveny tak, že inicializační počáteční stav prvního registru je nastaven na hodnotu (p), druhého registru na hodnotu (q), třetího registru na hodnotu 0, čtvrtého registru na hodnotu 1 a pátého registru na hodnotu p. Inicializační stavy prvního a druhého čítače jsou nastaveny na hodnotu 0, přičemž první a druhý registr jsou upraveny jen pro posun doleva se zabezpečením proti přetečení jejich obsahu při posuvu. Při každém jejich posunu vlevo po inicializaci jsou první a druhý čítač upraveny pro inkrementování a srovnávání jejích obsahů, a to tak, že počet posunů vlevo obsahů prvního respektive druhého registru je obsažený v prvním respektive ve druhém čítači. Třetí registr je upraven pro současný posuv doleva s prvním registrem pro případ, kdy je hodnota obsažená v prvním čítači větší nebo rovná hodnotě obsažená ve druhém čítači. Pro ostatní případy hodnot těchto čítačů je při posunu prvního registru doleva upraven čtvrtý registr pro posun doprava. Zároveň na tento čtvrtý registr upraven pro současný posuv doleva s druhým registrem v případě, že hodnota obsažená v druhém čítači je větší nebo rovná než hodnota obsažená v prvním čítači. Pro ostatní případy hodnot těchto čítačů je při posunu druhého registru doleva upraven třetí registr pro posun doprava. Řídicí a výpočetní jednotka je v případě zastavení posuvu obsahů prvního a druhého registru doleva v důsledku zamezení přetečení těchto obsahů upravena při dosažení stejného významnějšího bitu prvního a druhého registru pro provedení odečtení obsahů prvního a druhého registru a třetího a čtvrtého registru a při dosažení různého nejvýznamnějšího bitu prvního a druhého registru je upravena pro provedení sčítání těchto obsahů, a to vždy tak, že prvním zdrojovým a zároveň cílovým registrem je vždy ten registr, jehož původní obsah je méně posunut doleva. V případě stejného počtu posunů je prvním zdrojovým a zároveň cílovým registrem první a třetí registr. Řídicí a výpočetní jednotka je upravena pro zastavení operací sčítání, odčítání a posuvu v prvním až pátém registru pokud se při těchto operacích v prvním nebo druhém registru objeví hodnota 1 nebo –1 reprezentovaná v doplňkovém kódu a posunutá doleva o hodnotu obsaženou v prvním nebo druhém čítači a pro indikování skutečnosti, že třetí respektive čtvrtý registr obsahuje hodnotu generované multiplikativní inverze $b = q^{-1} \bmod p$ ve tvaru b, –b, (b – p) nebo (p – b), a to v závislosti na znaménkách hodnot obsažených v prvním respektive druhém registru a ve třetím respektive čtvrtém registru.

Výhodou tohoto způsobu generování multiplikativní inverze je, že operace působení hodnot v řídicí části, kterou představují první a druhý registr, a řízené části, kterou představují třetí a čtvrtý registr, jsou zaměněny za operace násobení a tím je eliminován případ nutnosti přičítání modulu p v případě lichého čísla tak, jak je to u dosavadních metod. Výhodou oproti používaným postupům je také použití doplňkového kódu pro reprezentaci záporných čísel, která se nepřevádějí na kladná a tím se eliminují operace převodu reprezentující sčítání. Další předností je

zjednodušení provádění operace porovnávání dvou čísel na operaci porovnání hodnot příslušných bitů daných hodnot. V průběhu generování multiplikativní inverze se porovnávají také vzájemné posuny hodnot v registrech řídicí části a pomocí tohoto srovnání se řídí výběr registru pro zápis vypočtených hodnot v průběhu generování multiplikativní inverze. Vzhledem k výše uvedeným výhodám je předkládaný způsob podstatně rychlejší než dosud známé postupy generování multiplikativní inverze, a to hlavně v případě velkých čísel používaných v kryptografii. Efektivnost předkládaného postupu je založená na provádění co nejmenšího počtů operací sčítání a odečítání potřebných na generování modulární inverze, u kterých časová náročnost jejich provedení roste přibližně s logaritmem, o základu dva, počtu bitů potřebných pro binární reprezentaci prvočíselného modula p.

Přehled obrázků na výkresech

Příklad provedení vynálezu bude dále podrobněji popsán pomocí přiložených výkresů, kde na obr. 1 je vývojový diagram znázorňující postup generování multiplikativní inverze nad konečným tělesem GF(p), na obr. 2 je tabulka která demonstruje generování modulární inverze na konkrétním příkladě a na obr. 3 je schéma základního zapojení k provádění tohoto způsobu generování modulární inverze.

Příklad provedení vynálezu

Příklad zapojení pro generování multiplikativní inverze nad konečným tělesem GF(p), pro kterou platí $b = q^{-1} \bmod p$, kde (p) je prvočíslo větší než (q) a kde (q) je celé kladné číslo větší než jedna je uveden na obr. 3. Toto zapojení sestává z řídicí a výpočetní jednotky CU, která je propojena s prvním až pátým n–bitovým registrem R_u, R_v, R_r, R_s, R_m první datovou a řídicí sběrnici B_1. Řídicí sběrnice B_1 současně tyto registry propojuje navzájem. Řídicí a výpočetní jednotka CU je dále propojena druhou řídicí a datovou sběrnici B_2 s prvním e–bitovým čítačem C_u a s druhým e–bitovým čítačem C_v, kde $e = [\log_2 n]$ a kde počet bitů n platí, že $2^{n-1} > p$. Rovněž pak platí, že n–bitové hodnoty proměnných obsažené v prvním až pátém registru R_u, R_v, R_r, R_s, R_m v celém průběhu generování multiplikativní inverze jsou reprezentované v doplňkovém kódu a nejméně významový bit LSB je v nich umístěn při inicializaci úplně vpravo. Inicializační počáteční stavy prvního až pátého registru R_u, R_v, R_r, R_s, R_m vyvolané pomocí příslušných proměnných p a q jsou nastaveny tak, že inicializační počáteční stav prvního registru R_u je nastaven na hodnotu p, druhého registru R_v na hodnotu 1 a pátého registru R_m na hodnotu p. Inicializační stav prvního čítače C_u a druhého čítače C_v jsou nastaveny na hodnotu 0, přičemž první a druhý registr R_u, R_v, jsou upraveny jen pro posuv doleva se zabezpečením proti přetečení jejich obsahu při posuvu. Při každém jejich posuvu vlevo po inicializaci jsou první a druhý čítač C_u, C_v, upraveny pro inkrementování a srovnávání jejich obsahů, a to tak, že počet posunů vlevo obsahů prvního registru R_u respektive druhého registru R_v je obsažený v prvním čítači C_u respektive ve druhém čítači C_v. Třetí registr R_r je upraven pro současný posuv doleva s prvním registrem R_u pro případ, kdy je hodnota obsažená v prvním čítači C_u větší nebo rovná hodnotě obsažené ve druhém čítači C_v. Pro ostatní případy hodnot těchto čítačů je při posunu prvního registru R_u doleva upraven čtvrtý registr R_s pro posun doprava. Zároveň je tento čtvrtý registr R_s upraven pro současný posuv doleva s druhým registrem C_v v případě, že hodnota obsažená v druhém čítači C_v je větší nebo rovná než hodnota obsažená v prvním čítači C_u. Pro ostatní případy hodnot těchto čítačů je při posunu druhého registru R_v doleva upraven třetí registr R_r pro posun doprava. Řídicí a výpočetní jednotka CU je v případě zastavení posuvu obsahů prvního a druhého registru R_u, R_v doleva v důsledku zamezení přetečení těchto obsahů upravena při dosažení stejného nejvýznamnějšího bitu prvního a druhého registru R_u, R_v pro provedení odečtení obsahů prvního a druhého registru R_u, R_v a třetího a čtvrtého registru R_r, R_s a při dosažení různého nejvýznamnějšího bitu prvního a druhého registru R_u, R_v je řídicí a výpočetní jednotka CU upravena pro provedení sčítání těchto obsahů, a to vždy tak, že prvním zdrojovým a zároveň cílovým registrem je vždy ten registr, jehož původní obsah je méně posunut doleva a v případě stejného počtu posunů je prvním

zdrojovým a zároveň cílovým registrem první a třetí registr R_u, R_r. Řídicí výpočetní jednotka CU je ještě upravena pro zastavení operací sčítání, odčítání a posuvu v prvním až pátém registru R_u, R_v, R_r, R_s, R_m pokud se při těchto operacích v prvním nebo druhém registru R_u, R_v objeví hodnota 1 nebo –1 reprezentovaná v doplňkovém kódu a posunutá doleva o hodnotu obsaženou v prvním nebo druhém čítači C_u, C_v a pro indikování skutečnosti, že třetí registr R_r respektive čtvrtý registr R_s obsahuje hodnotu generované multiplikativní inverze $b = q^{-1} \bmod p$ ve tvaru b, –b, (b – p) nebo (p – b), a to v závislosti na znaménkách hodnot obsažených v prvním registru R_u respektive ve druhém registru, R_v a ve třetím registru R_r respektive ve čtvrtém registru R_s.

Způsob generování multiplikativní inverze nad konečným tělesem GF(p) prováděný zapojením podle předkládaného vynálezu bude dále popsán ve formě jednotlivých kroků podle vývojového diagramu z obr. 1. Vychází se z toho, že je dáno celé kladné číslo q větší než jedna a prvočíslo p větší než q. Potom k číslu q existuje multiplikativní inverze

$b = q^{-1} \bmod p$,
pro kterou platí
$q.b \equiv 1 \pmod p$.

Nechť u, v, r, s, m jsou n bitové proměnné jejichž hodnoty v doplňkovém kódu jsou obsaženy v první až pátém n bitovém registru R_u, R_v, R_r, R_v, R_m, kde pro počet bitů n platí vztah $2^{n-1} > p$. Dále nechť C_u a C_v jsou první a druhý e bitový čítač, kde $e = [\log_2 n]$, a jejich obsah reprezentující hodnoty e bitových proměnných cu a cv. Generování multiplikativní inverze b k číslu q modulo p lze vyjádřit následujícím postupem. Nejprve se inicializují v prvním až pátém registru R_u, R_v, R_r, R_v, R_m a prvním a druhém čítači C_u a C_v počáteční stavy pomocí příslušných n bitových proměnných p a q tak, že platí:

$$u: = D(p),\ v: = D(q),\ r: = D(0),\ s: = D(1),\ m: = D(p),\ cu: = 0,\ cv: = 0,$$

kde označení *D(x)* představuje obraz příslušného čísla *x* v doplňkovém kódu a platí, že v prvním až pátém registru R_u, R_v, R_r, R_s a R_m je umístěn nejméně významný bit LSB vpravo. Jakmile je ukončena tato inicializace, která představuje krok 0, postoupí se za krok 1 do kroku 11. Krok 1 představuje testování hodnot n bitových proměnných u a v prvního a druhého registru R_u a R_v. Pokud se hodnota proměnné u nebo hodnota proměnné v rovná 1 nebo –1, vyjádřená v doplňkovém kódu posunutá o hodnotu cu míst doleva pro u nebo posunutá o hodnotu cv míst doleva pro v proces pokročí do kroku 2, který bude popsán dále, jinak se pokračuje krokem 11. V kroku 11 se zjišťuje hodnota dvou nejvíce významných bitů prvního registru R_u. Jsou-li tyto dva nejvíce významné bity nulové nebo jsou nenulové a současně alespoň jeden ze zbývajících bitů je nenulový, potom se pokračuje krokem 111, jinak proces přejde do kroku 12. V kroku 111 se porovnávají vzájemné velikosti hodnot e bitových proměnných cu a cv. Pokud se zjistí, že mezi nimi platí vztah $cu \geq cv$, potom proces přechází do kroku 1111, jinak do kroku 1112. V kroku 1111 se provede jednobitový posun obsahu prvního a třetího registru R_u a R_r doleva, což znamená, že se hodnoty proměnných u a r zdvojnásobí, a současně se inkrementuje obsah prvního čítače C_u, což představuje zvětšení hodnoty proměnné cu o jednu. Po tomto kroku 1111 následuje návrat do kroku 1, čímž dochází k opětnému testování podmínky z kroku 1.

Pokud proces postoupil do kroku 1112, provede se posun obsahu prvního registru R_u o jeden bit doleva a zároveň se provede jednobitový posun čtvrtého registru R_s doprava, tedy zdvojnásobí se hodnota proměnné u a naopak hodnota proměnné s se sníží na polovinu. Zároveň se inkrementuje obsah prvního čítače C_u, tedy hodnota cu se zvětší o jednu. Následuje opět návrat do kroku 1, tedy dochází k opětnému testování podmínky z kroku 1.

Pokud proces postoupil z kroku 11 přímo do kroku 12, pak se v tomto kroku testuje obsah druhého registru R_v. Když jsou hodnoty dvou nejvíce významných bitů druhého registru R_v nulové, nebo jsou nenulové a současně alespoň jeden ze zbývajících bitů je nenulový, potom proces postoupí do kroku 121, jinak přejde do kroku 13. V kroku 121 se opět porovnávají

vzájemné velikosti hodnot e bitových proměnných cu a cv. Pokud se zjistí, že mezi nimi platí vztah cv ≥ cu, potom se přechází do kroku 1211, jinak proces pokračuje krokem 1212. V kroku 1211 se provede jednobitový posun obsahu druhého a čtvrtého registru R_v a R_s doleva, což znamená, že se hodnoty proměnných v a s zdvojnásobí, a současně se inkrementuje obsah druhého čítače C_v, což představuje zvětšení hodnoty proměnné cv o jednu. Po tomto kroku 1211 následuje návrat do kroku 1, za účelem opětného testování podmínky z kroku 1. Potom proces postoupil do kroku 1212, provede se posun obsahu druhého registru R_v o jeden bit doleva a zároveň se provede jednobitový posun třetího registru R_r doprava, tedy zdvojnásobí se hodnota proměnné v a naopak hodnota proměnné r se sníží na polovinu. Zároveň se inkrementuje obsah druhého čítače C_v, tedy hodnota cv se zvětší o jednu. Následuje opět návrat do kroku 1 za účelem testování podmínky z kroku 1.

Pokud proces v kroku 12 přešel přímo do kroku 13, pak se testuje hodnota nejvíce významného bitu prvního registru R_u a nejvíce významného bitu druhého registru R_v. Když mají nejvíce významný bit prvního registru R_u a nejvíce významných bit druhého registru R_v stejnou hodnotu, potom proces postoupí do kroku 131, jinak se pokračuje v kroku 14. V kroku 131 se porovnávají hodnoty e bitových proměnných cu a cv. Pokud se zjistí, že mezi nimi platí vztah cv ≥ cu, potom se přechází do kroku 1311, jinak proces pokračuje krokem 1312. V kroku 1311 se odečte obsah druhého registru R_v od obsahu prvního registru R_u a výsledek se uloží v doplňkovém kódu do prvního registru R_u. Zároveň se odečte obsah čtvrtého registru R_s od obsahu třetího registru R_r a výsledek se uloží v doplňkovém kódu do třetího registru R_r, načež nastává návrat do kroku 1 znamenající testování podmínky z kroku 1. Pokud se přešlo z kroku 131 do kroku 1312, pak se v tomto kroku 1312 odečte obsah prvního registru R_u od obsahu druhého registru R_v a výsledek se uloží v doplňkovém kódu do druhého registru R_v a také se odečte obsah třetího registru R_r od obsahu čtvrtého registru R_s a výsledek se uloží v doplňkovém kódu do čtvrtého registru R_s, načež následuje testování podmínek kroku 1 návratem do tohoto kroku 1.

Pokud proces postoupil z kroku 13 přímo do kroku 14, a bylo zde zjištěno, že mezi hodnotami e bitovými proměnnými cu a cv platí vztah cv ≥ cu, přejde proces do kroku 141, jinak pokračuje krokem 142. V kroku 141 se sečte obsah druhého registru R_v a obsah prvního registru R_u a výsledek se uloží v doplňkovém kódu do prvního registru R_u a také se sečte obsah čtvrtého registru R_s a obsah třetího registru R_r a výsledek se uloží v doplňkovém kódu do třetího registru R_r, načež nastane návrat do kroku 1. Pokud se postoupilo přímo do kroku 142, sečtou se obsahy prvního a druhého registru R_u a R_v a výsledek se uloží v doplňkovém kódu do druhého registru R_v a také se sečtou obsahy třetího a čtvrtého registru R_r a R_s a výsledek se uloží v doplňkovém kódu do čtvrtého registru R_s, načež se proces navrací do kroku 1.

Pokud se z kroku 1 postoupilo do kroku 2 testuje se nyní, zda poslední zápis byl proveden do druhého a čtvrtého registru R_v a R_s, tedy zda hodnota proměnné v je rovná 1 nebo –1, reprezentovaná v doplňkovém kódu, posunutá o cv míst doleva. Pokud ano, přejde proces do kroku 21, jinak se pokračuje krokem 3. V kroku 21 se provede zápis obsahu čtvrtého registru R_s do třetího registru R_r, a nejvíce významných bit druhého registru R_v se zapíše na místo nejvíce významného bitu prvního registru R_u, načež se pokračuje v kroku 3.

Když se v kroku 3 zjistí, že nejvíce významný bit prvního registru R_u je nenulový, přejde se do kroku 31, jinak se přejde do kroku 4. V kroku 31 se testuje hodnota nejvíce významného bitu třetího registru R_r a když se zjistí, že je nenulová, potom se pokračuje v kroku 311, jinak se pokračuje v kroku 312. V kroku 311 se zneguje obsah třetího registru R_r a výsledek se opět uloží v doplňkovém kódu do registru R_r, načež proces pokračuje krokem 5. Pokud se z kroku 31 pokračuje krokem 312, pak se odečte hodnota třetího registru R_r od hodnoty pátého registru R_m a výsledek se uloží v doplňkovém kódu do třetího registru R_r. Poté se přejde do kroku 5.

Je-li v kroku 4 nejvíce významný bit třetího registru R_r je nenulový, přechází proces do kroku 41, jinak přejde do kroku 5. V kroku 41 se přičte k obsahu třetího registru R_r hodnota obsahu pátého registru R_m a výsledek se uloží v doplňkovém kódu do třetího registru R_r. Nyní se přejde do

kroku 5, kde se zjistí obsah třetího registru R_r a zjištěná hodnota je multiplikativní inverze $b = q^{-1} \bmod p$.

Tabulka na obr. 2 demonstruje generování modulární inverze na konkrétním příkladě, kde inicializační hodnoty jsou p = 13, q = 10. Výpočet probíhá dle popsaného postupu. V prvním sloupci je uveden krok prováděné aritmetické operace a testu dle označení z obr. 1. Ve druhém sloupci je pořadové číslo aritmetické operace, které mění obsah registrů. V případě nulté operace jedná se jen o inicializaci tj. načtení daných hodnot. Ve třetím respektive ve čtvrtém sloupci jsou uvedené hodnoty proměnných obsažené v registrech a čítačích po aritmetické operaci v dekadické respektive binární reprezentaci, kde horní pravý index v závorce označuje pořadové číslo aritmetické operace. Poslední sloupec uvádí provádění aritmetické operace. V případě testů je v prvním sloupci uvedena posloupnost provedených testu. V případě testů je v prvním sloupci uvedena posloupnost provedených testu. Tyto testy nevykonávají žádnou aritmetickou operaci s následnou změnou obsahu registrů. Výsledek celého postupu generování modulární inverze je uveden v posledním řádku s tím, že platí $b = 10^{-1} \bmod 13 = 4$, nebo $4.10 \equiv 1 \pmod{13}$.

Průmyslová využitelnost

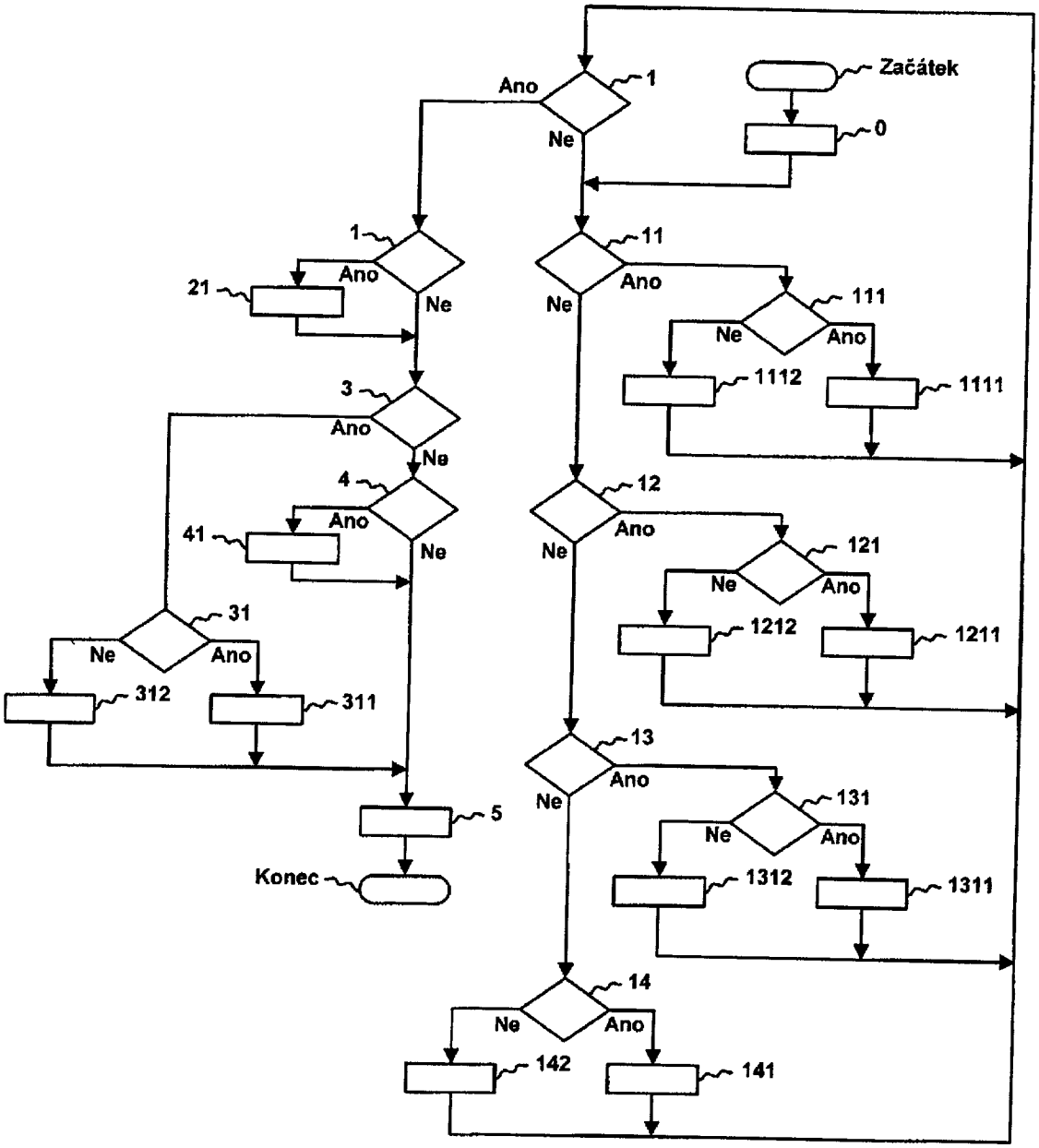
Multiplikativní inverze nad konečným GF(p) má zvláště důležitý význam ve výpočtech prováděných v kryptografii např. při operacích s body na eliptických křivkách definovaných nad konečným tělesem GF(p) nebo při akceleraci exponenciálních operací. Při současném nebývalém rozvoji informačních technologií je kryptografie v popředí zájmu v ekonomické oblasti a také v zájmu národních a zejména mezinárodních institucí pro ochranu dat. Předmětem vynálezu je zapojení, kterým lze realizovat generování modulární inverze efektivněji než doposud používanými způsoby. Zapojení může být využito v oblasti kryptografických hardwarových aplikací a embeddech systémech, například ve SMART kartách a samozřejmě všude, kde je potřebný rychlý a efektivný výpočet modulární inverze.

PATENTOVÉ NÁROKY

1. Zapojení pro generování multiplikativní inverze nad konečným tělesem GF(p), pro kterou platí $b = q^{-1} \bmod p$, kde (p) je prvočíslo větší než (q) a kde (q) je celé kladné číslo větší než jedna, **vyznačující se tím**, že sestává z řídicí a výpočetní jednotky (CU), která je propojena s prvním až pátým n–bitovým registrem (R_u, R_v, R_r, R_s, R_m) první datovou a řídicí sběrnicí (B_1), která současně tyto registry propojuje navzájem, a dále je řídicí a výpočetní jednotka (CU) propojena druhou řídicí a datovou sběrnicí (B_2) s prvním e-bitovým čítačem (C_u) a s druhým e-bitovým čítačem (C_v), kde $e = [\log_2 n]$ a kde pro počet bitů n platí, že $2^{n-1} > p$ a dále platí, že n–bitové hodnoty proměnných obsažené v prvním až pátém registru (R_u, R_v, R_r, R_s, R_m) v celém průběhu generování multiplikativní inverze jsou reprezentované v doplňkovém kódu a nejméně významný bit (LSB) je v nich umístěn při inicializaci úplně vpravo a inicializační počáteční stavy prvního až pátého registru (R_u, R_v, R_r, R_s, R_m) vyvolané pomocí příslušných proměnných (p) a (q) jsou nastaveny tak, že inicializační počáteční stav prvního registru (R_u) je nastaven na hodnotu (p), druhého registru (R_v) na hodnotu (q), třetího registru (R_r) na hodnotu 0, čtvrtého registru (R_s) na hodnotu 1 a pátého registru (R_m) na hodnotu p a inicializační stavy prvního čítače (C_u) a druhého čítače (C_v) jsou nastaveny na hodnotu 0, přičemž první a druhý registr (R_u, R_v) jsou upraveny jen pro posuv doleva se zabezpečením proti přetečení jejich obsahu při posuvu a při každém jejich posunu vlevo po inicializaci jsou první a druhý čítač (C_u, C_v) upraveny pro inkrementování a srovnávání jejích obsahů, a to tak, že počet posunů vlevo obsahů prvního registru (R_u) respektive druhého registru (R_v) je obsažený v prvním čítači (C_u) respektive ve druhém čítači (C_v) a třetí registr (R_r) je upraven pro současný posuv doleva s prvním registrem

(R_u) pro případ, kdy je hodnota obsažená v prvním čítači (C_u) větší nebo rovná hodnotě obsažené ve druhém čítači (C_v) a pro ostatní případy hodnot těchto čítačů je při posunu prvního registru (R_u) doleva upraven čtvrtý registr (R_s) pro posun doprava, a zároveň je tento čtvrtý registr (R_s) upraven pro současný posuv doleva a druhým registrem (C_v) v případě, že hodnota obsažená v druhém čítači (C_v) je větší nebo rovná než hodnota obsažená v prvním čítači (C_u) a pro ostatní případy hodnot těchto čítačů je při posunu druhého registru (R_v) doleva upraven třetí registr (R_r) pro posun doprava, přičemž řídicí a výpočetní jednotka (CU) je v případě zastavení posuvu obsahů prvního a druhého registru (R_u, R_v) doleva v důsledku zamezení přetečení těchto obsahů upravena při dosažení stejného nejvýznamnějšího bitu prvního a druhého registru (R_u, R_v) pro provedení odečtení obsahů prvního a druhého registru (R_u, R_v) a třetího a čtvrtého registru (R_r, R_s) a při dosažení různého významnějšího bitu prvního a druhého registru (R_u, R_v) je upravena pro provedení sčítání těchto obsahů, a to vždy tak, že prvním zdrojovým a zároveň cílovým registrem je vždy ten registr, jehož původní obsah je méně posunut doleva a v případě stejného počtu posunů je prvním zdrojovým a zároveň cílovým registrem první a třetí registr (R_u, R_r), přičemž a konečně je řídicí a výpočetní jednotka (CU) upravena pro zastavení operací sčítání a posuvu v prvním až pátém registru (R_u, R_v, R_r, R_s, R_m) pokud se při těchto operacích v prvním nebo druhém registru (R_u, R_v) objeví hodnota 1 nebo −1 reprezentovaná v doplňkovém kódu a posunutá doleva o hodnotu obsaženou v prvním nebo druhém čítači (C_u, C_v) a pro indikování skutečnosti, že třetí registr (R_r) respektive čtvrtý registr (R_s) obsahuje hodnotu generované multiplikativní inverze $b = q^{-1} \bmod p$ ve tvaru b, −b, (b − p) nebo (p − b), a to v závislosti na znaménkách hodnot obsažených v prvním respektive druhém registru (R_u, R_v) ve třetím respektive čtvrtém registru (R_r, R_s).

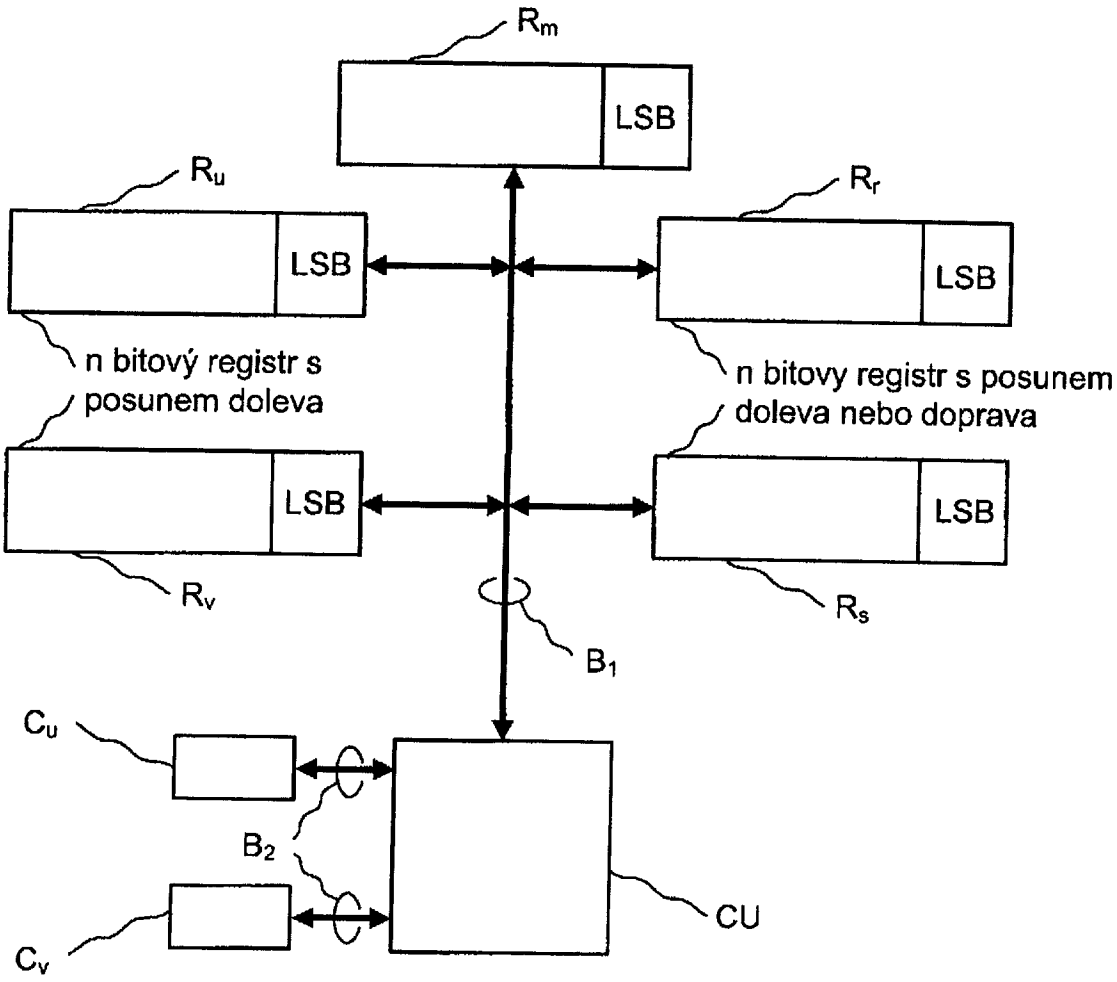
3 výkresy


Začátek
0
1
Ano
Ne
1
Ano
Ne
21
11
Ano
Ne
111
Ne
Ano
1112
1111
3
Ano
Ne
4
Ano
Ne
41
12
Ano
Ne
121
Ne
Ano
1212
1211
31
Ne
Ano
312
311
13
Ano
Ne
131
Ne
Ano
1312
1311
5
Konec
14
Ne
Ano
142
141

OBR. 1

Krok prováděné aritmetické operace a testu dle označení z obr. 1	Pořadové číslo aritmetické operace *i*	Hodnoty proměnných obsažené registrech a čítačích po aritmetické operaci *i* v reprezentaci		Provádění aritmetická operace
		dekadické	binární	
0	0	$u^{(0)} = 13$ $v^{(0)} = 10$ $r^{(0)} = 0$ $s^{(0)} = 1$ $m^{(0)} = 13$ $cu^{(0)} = 0$ $cv^{(0)} = 0$	$u^{(0)} = 01101$ $v^{(0)} = 01010$ $r^{(0)} = 00000$ $s^{(0)} = 00001$ $m^{(0)} = 01101$ $cu^{(0)} = 000$ $cv^{(0)} = 000$	$u^{(0)} := 13$ $v^{(0)} := 10$ $r^{(0)} := 0$ $s^{(0)} := 1$ $m^{(0)} := 13$ $cu^{(0)} := 0$ $cv^{(0)} := 0$
11, 12, 13 , 131				
1311	1	$u^{(1)} = 3$ $v^{(1)} = 10$ $r^{(1)} = -1$ $s^{(1)} = 1$ $m^{(1)} = 13$ $cu^{(1)} = 0$ $cv^{(1)} = 0$	$u^{(1)} = 00011$ $v^{(1)} = 01010$ $r^{(1)} = 11111$ $s^{(1)} = 00001$ $m^{(1)} = 01101$ $cu^{(1)} = 000$ $cv^{(1)} = 000$	$u^{(1)} := u^{(0)} - v^{(0)}$ $r^{(1)} := r^{(0)} - s^{(0)}$
1, 11, 111				
1111	2	$u^{(2)} = 6$ $v^{(2)} = 10$ $r^{(2)} = -2$ $s^{(2)} = 1$ $m^{(2)} = 13$ $cu^{(2)} = 2$ $cv^{(2)} = 0$	$u^{(2)} = 00110$ $v^{(2)} = 01010$ $r^{(2)} = 11110$ $s^{(2)} = 00001$ $m^{(2)} = 01101$ $cu^{(2)} = 010$ $cv^{(2)} = 000$	$u^{(2)} := 2u^{(1)}$ $r^{(2)} := 2r^{(1)}$
1, 11, 111				
1111	3	$u^{(3)} = 12$ $v^{(3)} = 10$ $r^{(3)} = -4$ $s^{(3)} = 1$ $m^{(3)} = 13$ $cu^{(3)} = 2$ $cv^{(3)} = 0$	$u^{(3)} = 01100$ $v^{(3)} = 01010$ $r^{(3)} = 11100$ $s^{(3)} = 00001$ $m^{(3)} = 01101$ $cu^{(3)} = 010$ $cv^{(3)} = 000$	$u^{(3)} := 2u^{(2)}$ $r^{(3)} := 2r^{(2)}$
1, 11, 12, 13, 131				
1312	4	$u^{(4)} = 12$ $v^{(4)} = -2$ $r^{(4)} = -4$ $s^{(4)} = 5$ $m^{(4)} = 13$ $cu^{(4)} = 2$ $cv^{(4)} = 0$	$u^{(4)} = 01100$ $v^{(4)} = 11110$ $r^{(4)} = 11100$ $s^{(4)} = 00101$ $m^{(4)} = 01101$ $cu^{(4)} = 010$ $cv^{(4)} = 000$	$v^{(4)} := v^{(3)} - u^{(3)}$ $s^{(4)} := s^{(3)} - r^{(3)}$
1, 11, 12, 121				
1212	5	$u^{(5)} = 12$ $v^{(5)} = -4$ $r^{(5)} = -2$ $s^{(5)} = 5$ $m^{(5)} = 13$ $cu^{(5)} = 2$ $cv^{(5)} = 2$	$u^{(5)} = 01100$ $v^{(5)} = 11100$ $r^{(5)} = 11110$ $s^{(5)} = 00001$ $m^{(5)} = 01101$ $cu^{(5)} = 010$ $cv^{(5)} = 010$	$v^{(5)} := 2v^{(4)}$ $r^{(5)} := r^{(4)}/2$
1, 11, 12, 121				
1212	6	$u^{(6)} = 12$ $v^{(6)} = -8$ $r^{(6)} = -1$ $s^{(6)} = 5$ $m^{(6)} = 13$ $cu^{(6)} = 2$ $cv^{(6)} = 2$	$u^{(6)} = 01100$ $v^{(6)} = 11000$ $r^{(6)} = 11111$ $s^{(6)} = 00001$ $m^{(6)} = 01101$ $cu^{(6)} = 010$ $cv^{(6)} = 010$	$v^{(6)} := 2v^{(5)}$ $r^{(6)} := r^{(5)}/2$
1, 11, 12, 13, 14				
141	7	$u^{(7)} = 4$ $v^{(7)} = -8$ $r^{(7)} = 4$ $s^{(7)} = 5$ $m^{(7)} = 13$ $cu^{(7)} = 2$ $cv^{(7)} = 2$	$u^{(7)} = 00100$ $v^{(7)} = 11000$ $r^{(7)} = 00100$ $s^{(7)} = 00001$ $m^{(7)} = 01101$ $cu^{(7)} = 010$ $cv^{(7)} = 010$	$u^{(7)} := u^{(6)} + v^{(6)}$ $r^{(7)} := r^{(5)} + s^{(5)}$
1, 2, 3, 4				
5		$b = r^{(7)} = 4$	$b = r^{(7)} = 00100$	

OBR. 2



OBR. 3

Konec dokumentu