PŘIHLÁŠKA VYNÁLEZU

zveřejněná podle § 31 zákona č. 527/1990 Sb.

(21) Číslo dokumentu:
2002-4116
(13) Druh dokumentu: **A3**
(51) Int. Cl.[7]:
H 04 L 9/00

(19)
ČESKÁ
REPUBLIKA

ÚŘAD
PRŮMYSLOVÉHO
VLASTNICTVÍ

(22) Přihlášeno: **16.12.2002**
(40) Datum zveřejnění přihlášky vynálezu:
(Věstník č. 8/2004)

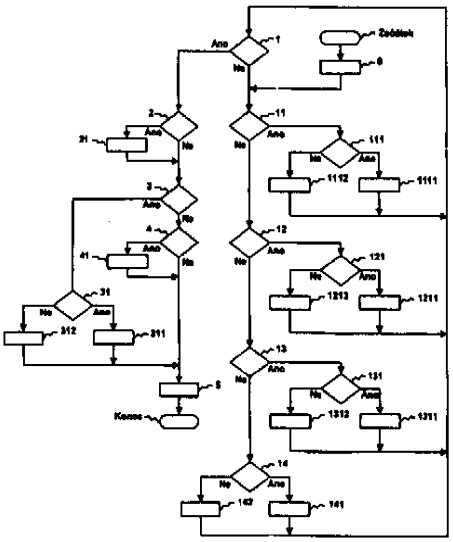
(71) Přihlašovatel:
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ FAKULTA ELEKTROTECHNICKÁ, Praha, CZ
LÓRENCZ Róbert Ing. CSc., Poděbrady, CZ

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

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

(54) Název přihlášky vynálezu:
Způsob generování multiplikativní inverze nad konečným tělesem GF /p/

(57) Anotace:
Podstata řešení spočívá v generování multiplikativní inverse nad konečným tělesem GF(p), kde p je prvočíslo, tj. generováním modulární inverse. Tento postup je odvozen z rozšířeného Euklidova algoritmu. Postup je upraven pro binární vykovávání operací v procesu generování modulární inverse a to vzhledem na co nejmenší počet operací sčítání, odečítání a posuvu. Navržený přístup odstraňuje redundantní operace pro konverzi lichých a záporných hodnot, které jsou prováděny u dosavadních postupů. k tomu využívá reprezentaci záporných čísel v doplňkovém kódu, posun hodnot doleva v řídící části rozšířeného Euklidova algoritmu a novou definici hlídacích a řídicích podmínek provádění postupu. Minimalizování počtu operací sčítání a odečítání je žádoucí v případě počítání s velkými čísly, které se vyskytují v kryptografii.



CZ 2002 - 4116 A3

Způsob generování multiplikativní inverze nad konečným tělesem GF(p)

Oblast techniky

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

Dosavadní stav techniky

Základní aritmetické 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 používaných v US Government Digital Signature Standard (NIST) a taktéž v současnosti často používané v kryptografii využívajících 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 definovaná jako celé číslo b z $\langle 1, p-1 \rangle$ takové, že platí $q.b \equiv 1 \pmod{p}$, často zapisované jako $b \equiv q^{-1} \pmod{p}$. 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 Inverse 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 přístupu umožňují jejích 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 důsledků odečítaní, je převedena na kladnou přičtením hodnoty p, co představuje operaci převodu záporného čísla na kladné modulo 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 vykonávaná půleni 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 do kladných hodnot a v případě lichých hodnot v procesu půlení je prováděn jejich převod na sudé hodnoty. Obě operace představují opět operaci sčítaní.

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 sudé při vykonávaní odloženého půlení ve druhé fázi generování a velký počet testů typů "větší/menší než" představujících operaci odečítaní.

Podstata vynálezu

Výše uvedené nedostatky odstraňuje způsob generování multiplikativní inverze $b \equiv q^{-1} \pmod{p}$ nad konečným tělesem GF(p) podle předkládaného vynálezu, kde q je celé kladné číslo a p je prvočíslo a platí pro ně podmínka, že $p > q > 1$, kde tento způsob generování využívá první až pátý n bitový registr R_u, R_v, R_r, R_s a R_m, kde u,v,r,s,m jsou n bitové proměnné, jejichž hodnoty jsou obsaženy v příslušných n bitových registrech, kde pro počet bitů n v registru platí, že $2^{n-1} > p$, a dále využívá první a druhý e bitový čítač C_u a C_v, kde $e = \lceil \log_2 n \rceil$, jejichž obsah reprezentují hodnoty e bitových proměnných cu a cv. Podstatou předkládaného vynálezu je, že sestává z následujících kroků. Nejprve jsou inicializované počáteční stavy prvního až pátého registru R_u, R_v, R_r, R_s a R_m a prvního a druhého čítače C_u, C_v pomocí příslušných 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 $D(x)$ představuje obraz 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. Po ukončení této inicializace se provede nejprve testování hodnot významných bitů prvního a druhého registru R_u a R_v, kdy se testují jednak stavy hodnot významných bitů každého registru zvlášť a jednak ve vzájemném vztahu mezi prvním a druhým registrem R_u a R_v. Pokud se při testování prvního registru R_u zjistí, že dva nejvíce významné bity tohoto prvního registru R_u jsou nulové nebo jsou nenulové a současně alespoň jeden ze zbývajících bitů je nenulový, provede se následně srovnání vzájemné velikosti hodnot e bitových proměnných cu a cv. Při zjištění, že $cu \geq cv$ se obsah prvního a třetího registru R_u a R_r posune o jeden bit doleva a

inkrementuje se obsah prvního čítače C_u. V případě, že cu < cv, posune se obsah prvního registru R_u o jeden bit doleva, dále se posune obsah čtvrtého registru R_s o jeden bit doprava a inkrementuje se obsah prvního čítače C_u. Obdobně se provede testování hodnot významných bitů druhého registru R_v. Pokud se zjistí, že dva nejvíce významné bity tohoto druhého registru R_v jsou nulové nebo jsou nenulové a současně alespoň jeden ze zbývajících bitů je nenulový, provede se srovnání vzájemné velikosti hodnot e bitových proměnných cv a cu. Při zjištění, že cv ≥ cu se obsahy druhého a čtvrtého registru R_v a R_s posunou o jeden bit doleva a inkrementuje se obsah druhého čítače C_v. V případě, že cv < cu, posune se obsah druhého registru R_v o jeden bit doleva, dále se posune obsah třetího registru R_r o jeden bit doprava a inkrementuje se obsah druhého čítače C_v. Následně se provede testování nejvíce významného bitu prvního registru R_u ve vztahu k nejvíce významnému bitu druhého registru R_v a dále se srovnává vzájemná velikost hodnot e bitových proměnných cv a cu. Pokud se dojde k závěru, že oba nejvíce významné bity mají stejnou hodnotu a dále platí, že cv ≥ cu, pak se velikost obsahu prvního registru R_u sníží o velikost obsahu druhého registru R_v a výsledek se uloží v doplňkovém kódu do prvního registru R_u. Zároveň se sníží obsah třetího registru R_r o obsah čtvrtého registru R_s a výsledek se uloží v doplňkovém kódu do třetího registru R_r. Pokud při rovnosti hodnot obou nejvíce významných bitů prvního registru R_u a a druhého registru R_v platí, že cu > cv pak se naopak sníží obsah druhého registru R_v o obsah prvního registru R_u a výsledek se uloží v doplňkovém kódu do druhého registru R_v. Zároveň se sníží obsah čtvrtého registru R_s o obsah třetího registru R_r a výsledek se uloží v doplňkovém kódu do čtvrtého registru R_s. Je-li však zjištěno, že nejvíce významný bit prvního registru R_u a nejvíce významný bit druhého registru R_v mají různou hodnotu a platí, že cv ≥ cu, pak se sečtou obsahy druhého a prvního registru R_v a R_u a zjištěná hodnota se uloží v doplňkovém kódu do prvního registru R_u. Zároveň se sečtou obsahy čtvrtého a třetího registru R_s a R_r a výsledná hodnota se uloží v doplňkovém kódu do třetího registru R_r. Pokud je cv < cu, pak se zjištěný součet registrů R_u a R_v respektive zjištěný součet registru R_r a R_s uloží v doplňkovém kódu do druhého registru R_v respektive do čtvrtého registru R_s.

Výsledkem opakovaného provádění souboru dosud uvedených kroků začínající po inicializaci je zajištění stavu, kdy některé z proměnných u nebo v přináleží hodnota 1 nebo –1 vyjádřená v doplňkovém kódu, posunutá o hodnotu cu míst doleva pro u nebo posunuta o hodnotu cv míst doleva pro v. Po dosažení tohoto stavu se zjišťuje, 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, uloží se obsah čtvrtého registru R_s do třetího registru R_r a hodnota nejvíce významného bitu druhého registru R_v se zapíše na místo hodnoty nejvíce významného bitu prvního registru R_u. Poté se zkontroluje hodnota nejvíce významného bitu prvního registru R_u a třetího registru R_r a jsou-li obě tyto hodnoty nenulové, zneguje se obsah třetího registru R_r a výsledek se uloží v doplňkovém kódu opět do registru R_r. Pokud je hodnota nejvíce významného bitu prvního registru R_u nenulová a hodnota nejvíce významného bitu třetího registru R_r nulová, pak se obsah pátého registru R_m zmenší o obsah třetího registru R_r a výsledek v doplňkovém kódu se uloží ve třetím registru R_r. Pokud je hodnota nejvíce významného bitu prvního registru R_u nulová a hodnota nejvíce významného bitu třetího registru R_r nenulová, pak se obsah pátého registru R_m zvětší o obsah třetího registru R_r a zjištěná hodnota se uloží v doplňkovém kódu do třetího registru R_r, přičemž hodnota zapsaná ve třetím registru R_r je hodnotou multiplikativní inverze $b \equiv q^{-1} \pmod{p}$ nad konečným tělesem GF(p).

Výhodou tohoto způsobu generování multiplikativní inverze je, že operace půlení hodnot v řídící části, kterou představují registry R_u a R_v, a řízené časti, kterou představují registry R_r a R_s, jsou zaměněny za operace násobení a tím je eliminován případ nutnosti přičítaní 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, které se nepřevádějí do kladných hodnot a tím eliminuji operace převodu reprezentující sčítaní. Další předností je zjednodušení provádění operace porovnání dvou čísel na operaci porovnání hodnot příslušných bitů daných hodnot. V průběhu generování multiplikativní inverze se porovnávají také vzájemné posuny hodnot v registrech řídící 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 sčítaní a odečítaní potřebných na generováni modulární inverze, u kterých časová náročnost jejích 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 inverse na konkrétním příkladě a na obr. 3 je schéma základního zapojení k provádění tohoto způsobu generováni modulární inverse.

Příklad provedení vynálezu

Způsob generování multiplikativní inverze nad konečným tělesem GF(p) 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

$b \equiv q^{-1} \pmod{p}$,

pro kterou platí

$qb \equiv 1 \pmod{p}$.

Nechť u, v, r, s, m jsou n bitové proměnné jejichž hodnoty v doplňkovém kódu jsou obsaženy v prvním 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 $e = \lceil \log_2 n \rceil$, a jejich obsah reprezentují 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. Pokud 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, kde se testuje hodnota nejvíce významného bitu prvního registru R_u a druhého registru R_v. Když má nejvíce významné bit prvního registru R_u a 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 významný 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 inverzi $b \equiv q^{-1} \pmod{p}$.

Tabulka na obr. 2 demonstruje generování modulární inverse 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, která 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 prováděné aritmetické operace. 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 generováni modulární inverze je uveden v poslední řádku s tím, že platí $b \equiv 10^{-1} \pmod{13} = 4$, nebo $4.10 \equiv 1 \pmod{13}$.

Výše uvedený způsob generování multiplikativní inverze lze obecně realizovat pomocí zařízení dle obr.3. Toto zařízení je tvořeno řídící jednotkou 600, která je propojena s prvním a druhým e bitovým čítačem 601 a 602, kde

$e = \lceil \log_2 n \rceil$, a dále je propojena s prvním až pátým n bitovým registrem 100, 200, 300, 400 a 500, kde pro počet bitů n v registru platí, že $2^{n-1} > p$, přičemž z důvodu spojitosti s popisem způsobu generování multiplikativní inverze nad konečným tělesem GF(p) jsou v jednotlivých blocích vyznačena označení C_u, C_v pro čítače a R_u, R_v, R_r, R_s a R_m pro registry. Registry 100, 200, 300, 400 a 500 jsou navzájem mezi sebou propojeny. První a druhý registr 100 a 200 jsou registry s posunem doleva a třetí a čtvrtý registr 300 a 400 jsou registry s posunem doleva i doprava, přičemž u všech je nejméně významný bit umístěn vpravo.

Průmyslová využitelnost

Multiplikativní inverze nad konečným GF(p) má zvláště důležitý význam ve výpočtech prováděných v kryptografii např. obzvlášť při operací s body na eliptických křivkách definovaných nad konečným tělesem GF(p) nebo při akceleraci exponenciálních operací. Kryptografie v současností při nebývalém rozvoji informačních technologii je v popředí zájmu v ekonomické oblasti a také zájmu národních a zejména mezinárodních institucí pro ochranu dat. Předmětem vynálezu je postup, který počítá efektivněji modulární inverzi jako doposud používané postupy. Uplatnění algoritmu může být využít zvláště v oblasti kryptografických hardwarových aplikacích a embedded systémech, např. v SMART kartách a samozřejmě všude, kde je potřebný rychlý a efektivní výpočet modulární inverze.

PATENTOVÉ NÁROKY

Způsob generování multiplikativní inverze nad konečným tělesem GF(p), pro kterou platí $b \equiv q^{-1} \pmod{p}$, kde (p) je prvočíslo větší než (q) a kde (q) je celé kladné číslo větší než jedna, kde tento způsob generování využívá první až pátý (n) bitový registr (R_u, R_v, R_r, R_s a R_m), kde (u,v,r,s,m) jsou n bitové proměnné, jejichž hodnoty jsou obsaženy v příslušných n bitových registrech, kde pro počet bitů (n) v registru platí, že $2^{n-1} > p$, a dále využívá první a druhý e bitový čítač (Cu) a (Cv), kde $e = \lceil \log_2 n \rceil$, jejichž obsah reprezentují hodnoty e bitových proměnných (cu) a (cv) **vyznačující se tím, že** jsou nejprve inicializované počáteční stavy prvního až pátého registru (R_u, R_v, R_r, R_s a R_m) a prvního a druhého čítače (C_u, C_v) pomocí příslušných proměnných (p) a (q) tak, že platí

u := *D*(p)

v := *D*(q)

r := *D*(0)

s := *D*(1)

m := *D*(p)

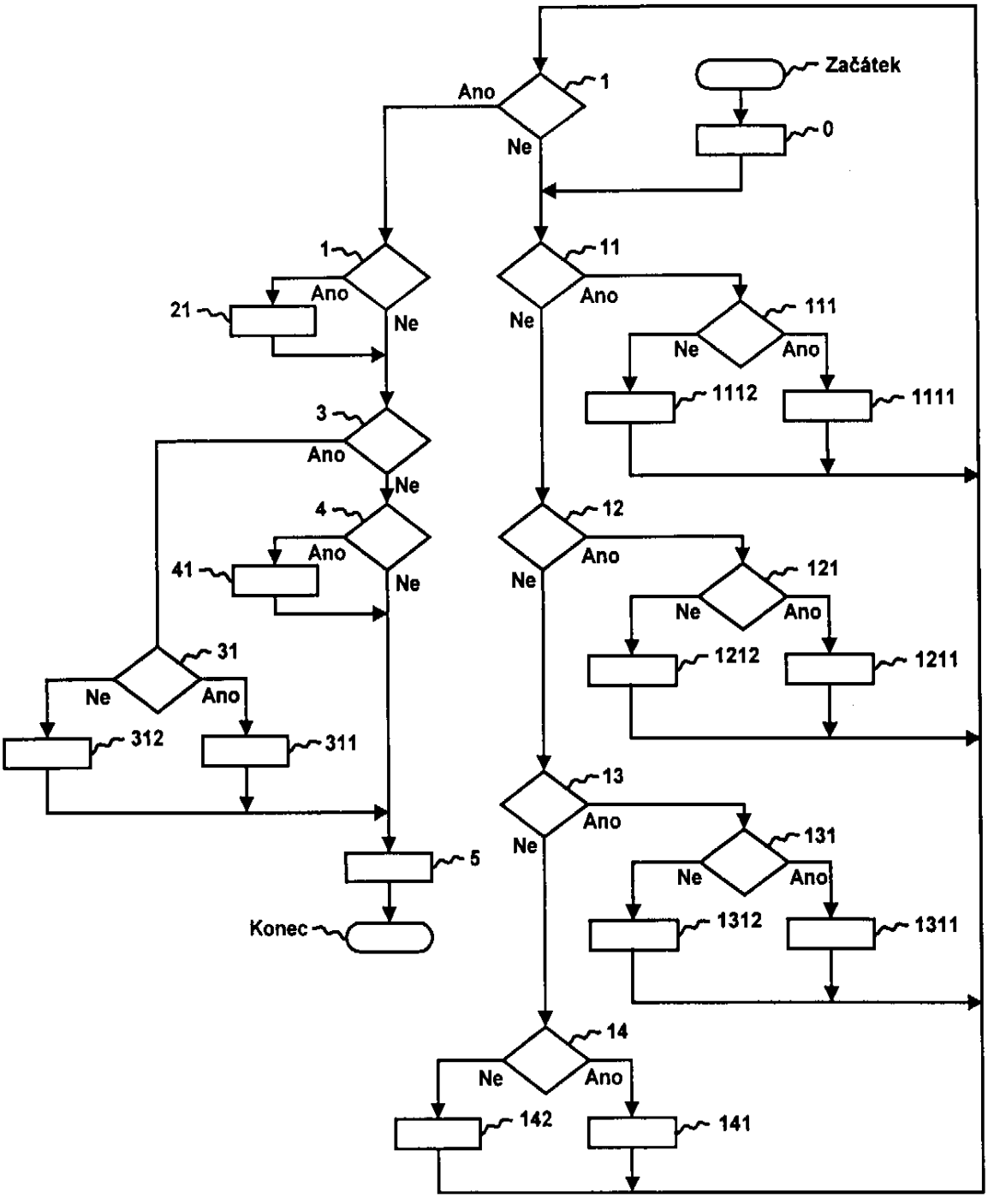
cu := 0

cv := 0,

kde *D*(x) představuje obraz čísla x v doplňkovém kódu tak, že v prvním až pátém registru (R_u, R_v, R_r R_s a R_m) je umístěn nejméně významný bit (LSB) vpravo, po ukončení této inicializace se provede nejprve testování hodnot významných bitů prvního a druhého registru (R_u) a (R_v), kdy se testují jednak stavy hodnot významných bitů každého registru zvlášť a jednak ve vzájemném vztahu mezi prvním a druhým registrem (R_u) a (R_v), přičemž pokud se při testování prvního registru (R_u) zjistí, že dva nejvíce významné bity tohoto prvního registru (R_u) jsou nulové nebo jsou nenulové a současně alespoň jeden ze zbývajících bitů je nenulový, provede se následně srovnání vzájemné velikosti e bitových proměnných (cu) a (cv) a při zjištění, že $cu \geq cv$ se obsah prvního a třetího registru (R_u) a (R_r) posune o jeden bit doleva a inkrementuje se obsah prvního čítače (C_u) a v případě, že $cu < cv$, posune se

obsah prvního registru (R_u) o jeden bit doleva, dále se posune obsah čtvrtého registru (R_s) o jeden bit doprava a inkrementuje se obsah prvního čítače (Cu), obdobně se provede srovnání hodnot významných bitů druhého registru (R_v) a pokud se zjistí, že dva nejvíce významné bity tohoto druhého registru (R_v) jsou nulové nebo jsou nenulové a současně alespoň jeden ze zbývajících bitů je nenulový, provede se srovnání vzájemné velikosti e bitových proměnných (cv) a (cu) a při zjištění, že cv ≥ cu se obsahy druhého a čtvrtého registru (R_v) a (R_s) posunou o jeden bit doleva a inkrementuje se obsah druhého čítače (C_v) a v případě, že cv < cu, posune se obsah druhého registru (R_v) o jeden bit doleva, dále se posune obsah třetího registru (R_r) o jeden bit doprava a inkrementuje se obsah druhého čítače (C_v), následně se provede testování nejvíce významného bitu prvního registru (R_u) ve vztahu k nejvíce významnému bitu druhého registru (R_v) při následném srovnání e bitových proměnných (cv) a (cu) a pokud se dojde k závěru, že oba nejvíce významné bity mají stejnou hodnotu a platí, že cv ≥ cu, pak se velikost obsahu prvního registru (R_u) sníží o velikost obsahu druhého registru (R_v) a výsledek se uloží v doplňkovém kódu do prvního registru (R_u) a zároveň se sníží obsah třetího registru (R_r) o obsah čtvrtého registru (R_s) a výsledek se uloží v doplňkovém kódu do třetího registru (R_r), pokud při rovnosti hodnot obou nejvíce významných bitů prvního a druhého registru (R_u) a (R_v) platí, že cu > cv pak se naopak sníží obsah druhého registru (R_v) o obsah prvního registru (R_u) a výsledek se uloží v doplňkovém kódu do druhého registru (R_v) a zároveň se sníží obsah čtvrtého registru (R_s) o obsah třetího registru (R_r) a výsledek se uloží v doplňkovém kódu do čtvrtého registru (R_s), je-li však zjištěno, že nejvíce významný bit prvního registru (R_u) a nejvíce významný bit druhého registru (R_v) mají různou hodnotu a zároveň cv ≥ cu, sečtou se obsahy druhého a prvního registru (R_v) a (R_u) a zjištěná hodnota se uloží v doplňkovém kódu do prvního registru (R_u) a zároveň se sečtou obsahy čtvrtého a třetího registru (R_s) a (R_r) a výsledná hodnota se uloží v doplňkovém kódu do třetího registru (R_r) a pokud je cv < cu, pak se zjištěný součet registrů (R_u) a (R_v) respektive zjištěný součet registrů (R_r) a (R_s) uloží v doplňkovém kódu do druhého registru (R_v) respektive do čtvrtého registru (R_s), přičemž výsledkem opakovaného provádění souboru dosud uvedených

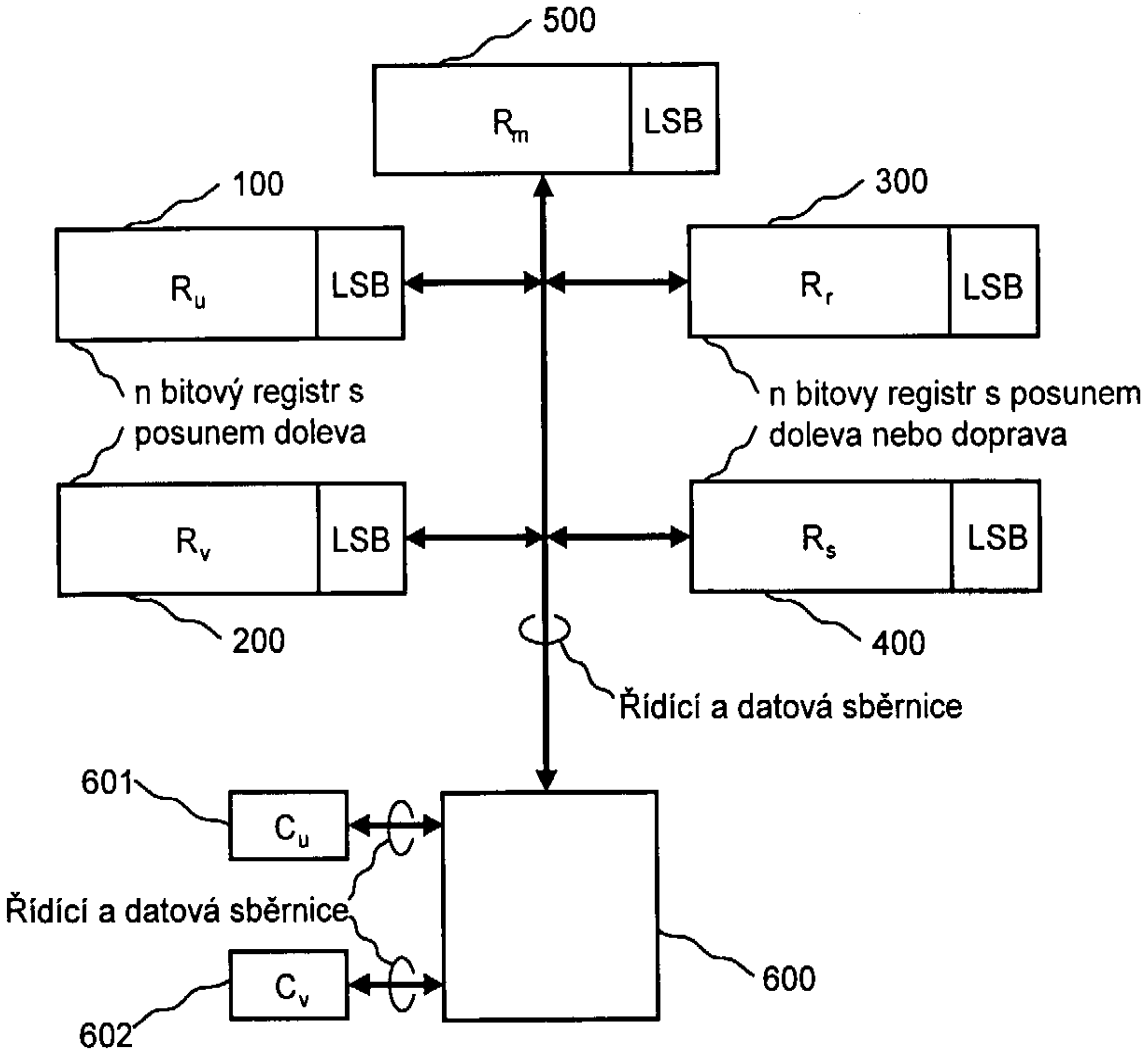
kroků začínající po inicializaci je zajištění stavu, kdy některé z proměnných (u) nebo (v) přináleží hodnota 1 nebo –1 vyjádřená v doplňkovém kódu, posunutá o hodnotu (cu) míst doleva pro (u) nebo posunuta o hodnotu (cv) míst doleva pro (v), načež po dosažení tohoto stavu se zjišťuje, zda poslední zápis byl proveden do druhého a čtvrtého registru (R_v) a (R_s), tedy zda hodnota proměnné (v) je rovná 1 nebo –1 vyjádřená v doplňkovém kódu posunuté o (cv) míst doleva a pokud ano, uloží se obsah čtvrtého registru (R_s) do třetího registru (R_r) a nejvíce významný bit druhého registru (R_v) se zapíše na místo nejvíce významného bitu prvního registru (R_u), poté se zkontroluje hodnota nejvíce významného bitu prvního registru (R_u) a třetího registru (R_r) a jsou-li obě tyto hodnoty nenulové, zneguje se obsah třetího registru (R_r) a výsledek se opět uloží v doplňkovém kódu do registru (R_r), pokud je hodnota nejvíce významného bitu prvního registru (R_u) nenulová a hodnota nejvíce významného bitu třetího registru (R_r) nulová, pak se obsah pátého registru (R_m) zmenší o obsah třetího registru (R_r) a výsledek se uloží v doplňkovém kódu ve třetím registru (R_r), pokud je hodnota nejvíce významného bitu prvního registru (R_u) nulová a hodnota nejvíce významného bitu třetího registru (R_r) nenulová pak se obsah pátého registru (R_m) zvětší o obsah třetího registru (R_r) a zjištěná hodnota se uloží v doplňkovém kódu do třetího registru (R_r), přičemž hodnota zapsaná ve třetím registru (R_r) je hodnotou multiplikativní inverze $b \equiv q^{-1} \pmod{p}$ nad konečným tělesem GF(p).


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

OBR. 1

Prováděná aritmetická operace a test dle označení z obr. 1	Pořadové číslo aritmetické operace	Hodnoty proměnných obsažené v registrech a čítačích po aritmetické operaci v reprezentaci: dekadické	binární	Provádění aritmetická operace
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	$u^{(2)} = 6$ $v^{(2)} = 10$ $r^{(2)} = -2$ $s^{(2)} = 1$ $m^{(2)} = 13$ $cu^{(2)} = 1$ $cv^{(2)} = 0$	$u^{(2)} = 00110$ $v^{(2)} = 01010$ $r^{(2)} = 11110$ $s^{(2)} = 00001$ $m^{(2)} = 01101$ $cu^{(2)} = 001$ $cv^{(2)} = 000$	$u^{(2)} := 2u^{(1)}$ $r^{(2)} := 2r^{(1)}$
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	$u^{(5)} = 12$ $v^{(5)} = -4$ $r^{(5)} = -2$ $s^{(5)} = 5$ $m^{(5)} = 13$ $cu^{(5)} = 2$ $cv^{(5)} = 1$	$u^{(5)} = 01100$ $v^{(5)} = 11100$ $r^{(5)} = 11110$ $s^{(5)} = 00101$ $m^{(5)} = 01101$ $cu^{(5)} = 010$ $cv^{(5)} = 001$	$v^{(5)} := 2v^{(4)}$ $r^{(5)} := r^{(4)}/2$
1, 11, 12, 121				
1212	6	$u^{(6)} = 12$ $v^{(6)} = -8$ $r^{(6)} = -1$ $s^{(6)} = 5$ $m^{(6)} = 13$ $cu^{(6)} = 2$ $cv^{(6)} = 2$	$u^{(6)} = 01100$ $v^{(6)} = 11000$ $r^{(6)} = 11111$ $s^{(6)} = 00101$ $m^{(6)} = 01101$ $cu^{(6)} = 010$ $cv^{(6)} = 010$	$v^{(6)} := 2v^{(5)}$ $r^{(6)} := r^{(5)}/2$
1, 11, 12, 13, 14				
141	7	$u^{(7)} = 4$ $v^{(7)} = -8$ $r^{(7)} = 4$ $s^{(7)} = 5$ $m^{(7)} = 13$ $cu^{(7)} = 2$ $cv^{(7)} = 2$	$u^{(7)} = 00100$ $v^{(7)} = 11000$ $r^{(7)} = 00100$ $s^{(7)} = 00101$ $m^{(7)} = 01101$ $cu^{(7)} = 010$ $cv^{(7)} = 010$	$u^{(7)} := u^{(6)} + v^{(6)}$ $r^{(7)} := r^{(6)} + s^{(6)}$
1, 2, 3, 4				
5		$b = r^{(7)} = 4$	$b = r^{(7)} = 00100$	

OBR. 2


500
R_m
LSB
100
R_u
LSB
300
R_r
LSB
n bitový registr s posunem doleva
n bitovy registr s posunem doleva nebo doprava
R_v
LSB
R_s
LSB
200
400
Řídící a datová sběrnice
601
C_u
Řídící a datová sběrnice
C_v
602
600

OBR. 3