ČESKOSLOVENSKÁ SOCIALISTICKÁ REPUBLIKA
(19)

ÚŘAD PRO VYNÁLEZY A OBJEVY

POPIS VYNÁLEZU K AUTORSKÉMU OSVĚDČENÍ

258 994

(11) (B1)

(61)
(23) Výstavní priorita
(22) Přihlášeno 25 06 86
(21) PV 4692-86.Z

(51) Int. Cl.[4]
G 06 F11/08,
G 06 F 11/00

(40) Zveřejněno 14 05 87
(45) Vydáno 1.11.1989

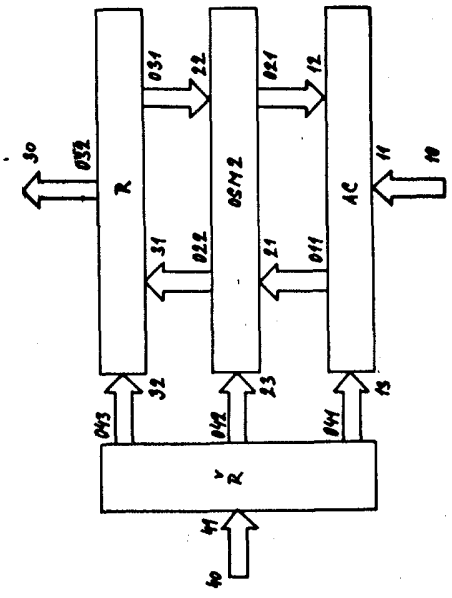
(75)
Autor vynálezu

KAŠKA ZDENĚK ing.,
BUREŠ JAROSLAV ing.,
VLAŠÍN JAROMÍR ing., BRNO

(54)

Zapojení na vytváření cyklického zabezpečovacího kódu ze šestnáctibitových slov

Zapojení řeší vytváření cyklického zabezpečovacího kodu ze šestnáctibitových slov, s generačním polynomem $x^{16} + x^{12} + x^5 + 1$ v čekacím čase přenosu mezi jednotlivými šestnáctibitovými slovy, přičemž rychlost je závislá pouze na cyklu procesoru a počtu použitých mikroinstrukcí uložených v řídicí paměti procesoru pomocí jednoduchého zapojení s registry a obvodem součtu modulu 2, řízenými řadičem. Řešení lze využít při přenosu dat, u kterých se generuje cyklický zabezpečovací kod.



258 994

Vynález se týká zapojení na vytváření cyklického zabezpečovacího kódu ze šestnáctibitových slov.

Každý přenos bloku dat může být ovlivněn poruchami. To platí i pro přenos dat mezi vnitřní pamětí výpočetního systému a jeho vnější pamětí. Přenos dat se běžně zabezpečuje cyklickým zabezpečovacím kódem, jehož generační polynom $x^{16} + x^{12} + x^5 + 1$ je definován ve standardech formátů pro diskové paměti a jenž je označován zkratkou CRC. Dosavadní běžná generace cyklického zabezpečovacího kódu se odvozuje ze sériového toku dat. Přenos toku dat probíhá obvykle tak, že data se zapisují do vnější paměti, kterou bývá zpravidla disková paměť a čtou se z tého sériově, kdežto ve vnitřní paměti výpočetního systému jsou ukládána paralelně. V přenosovém kanálu se proto provádí serializace a deserializace přenášených dat. Realizace generačního polynomu se převážně zajišťuje moderními monolitickými obvody velké integrace a méně často pak samostatnou obvodovou strukturou složenou z integrovaných obvodů malé integrace.

Nevýhodou sériové generace cyklického zabezpečovacího kódu je nezbytná přítomnost datových taktovacích hodin po celou dobu přenosu bloku dat, jakož i pomalá generace bit po bitu.

Uvedené nevýhody odstraňuje zapojení na vytváření cyklického zabezpečovacího kódu ze šestnáctibitových slov podle vynálezu, jehož podstatou je, že první skupina vstupů prvního registru tvoří současně skupinu vstupů zapojení, skupina výstupů prvního registru je připojena na první skupinu vstupů obvodu součtu modulo 2, jehož první skupina výstupů je připojena na druhou skupinu vstupů prvního registru, druhá skupina výstupů obvodu součtu modulo 2 je připojena na první skupinu vstupů druhého registru, jehož první skupina výstupů je připojena na druhou skupinu vstupů obvodu součtu modulo 2, druhá skupina výstupů druhého registru tvoří současně skupinu výstupů zapojení, první sku-

pina výstupů řadiče je připojena na skupinu řídicích vstupů prvního registru, druhá skupina výstupů řadiče je připojena na skupinu řídicích vstupů obvodu součtu modulo 2, třetí skupina výstupů řadiče je připojena na skupinu řídicích vstupů druhého registru, skupina ovládacích vstupů řadiče tvoří současně skupinu ovládacích vstupů zapojení. Za výstupem jednoho krajního bitu prvního registru je zapojen paměťový prvek, jehož výstup je připojen ke vstupu druhého krajního bitu prvního registru.

Výhodou zapojení na vytváření cyklického zabezpečovacího kódu ze šestnáctibitových slov podle vynálezu je, že po dokončení serializace nebo deserializace přenášených dat se paralelně přenese šestnáctibitové datové slovo z datové sběrnice do registru a následující vytváření mezivýsledku cyklického zabezpečovacího kódu již nezávisí na datových taktovacích hodinách, nýbrž je řízeno pouze vnitřními hodinami systému. Vytváření cyklického zabezpečovacího kódu se uskutečňuje v čekacím čase procesoru vymezeném přenosem mezi jednotlivými šestnáctibitovými slovy, přičemž rychlost jeho vytváření je závislá pouze na cyklu procesoru a počtu použitých mikroinstrukcí uložených v řídicí paměti procesoru. Zapojení nevyžaduje žádné přídavné obvody, neboť využívá s výhodou vnitřních registrů procesoru, dále mikroinstrukcí, uložených v řídicí paměti procesoru, které jsou součástí firmware celého systému.

Příklad zapojení na vytváření cyklického zabezpečovacího kódu ze šestnáctibitových slov podle vynálezu je znázorněn na připojených výkresech, na nichž

obr. 1 představuje blokové schéma zapojení,

obr. 2 přehled vytváření cyklického zabezpečovacího kódu ze šestnáctibitových slov v obecné formě,

obr. 3 přehled vytváření konkrétního cyklického zabezpečovacího kódu.

První skupina vstupů 11 prvního posuvného registru AC pro bit 0 až 15 datové informace tvoří současně skupinu vstupů 10 zapojení pro bit 0 až 15 datové informace pro připojení na neznázorněnou systémovou sběrnici dat. Skupina výstupů 011 prvního posuvného registru AC pro bit 0 až 15 je připojena na první skupinu vstupů 21 obvodu OSM2 součtu modulo 2, jehož první skupina výstupů 021 pro bit 0 až 15 je připojena na druhou skupinu vstupů 12 prvního posuvného registru AC pro bit 0 až 15. Druhá skupina výstupů 022 obvodu OSM2 součtu modulo 2 pro bit 0 až 15

je připojena na první skupinu vstupů 31 druhého posuvného registru R, jehož první skupina výstupů 031 pro bit 0 až 15 je připojena na druhou skupinu vstupů 22 obvodu OSM2 součtu modulo 2 pro bit 0 až 15. Druhá skupina výstupů 032 druhého posuvného registru R pro bit 0 až 15 tvoří současně skupinu výstupů 30 zapojení pro bit 0 až 15 cyklického zabezpečovacího kódu. První skupina výstupů 041 řadiče Ř je připojena na skupinu řídicích vstupů 13 prvního posuvného registru AC. Druhá skupina výstupů 042 řadiče Ř je připojena na skupinu řídicích vstupů 23 obvodu OSM2 součtu modulo 2. Třetí skupina výstupů 043 řadiče Ř je připojena na skupinu řídicích vstupů 32 druhého posuvného registru R. Skupina ovládacích vstupů 41 Ř tvoří současně skupinu ovládacích vstupů 40 zapojení pro připojení na neznázorněné obvody určující zahájení vytváření cyklického zabezpečovacího kódu. Místo posuvných registrů lze použít i jiných vhodných pamětí typu RAM nebo jiných vhodných registrů.

Vytváření cyklického zabezpečovacího kódu je řízeno řadičem Ř a probíhá v těchto krocích:

1. Signálem, přicházejícím z řadiče Ř, se druhý posuvný registr R nastaví do počátečního stavu, kterým může být i předem známý mezivýsledek cyklického zabezpečovacího kódu, odpovídající příslušným vstupním datům.

2. Ovládacími signály řadiče Ř se do prvního posuvného registru AC uloží první nebo příslušné datové slovo přivedené paralelně ze systémové sběrnice na první skupinu vstupů 11.

3. Ovládacími signály řadiče Ř, přivedenými na skupiny řídicích vstupů 13, 23, 32 se obsahy prvního a druhého posuvného registru AC a R sečtou modulo 2 mezi stejnolehlými bity a výsledek se uloží do prvního posuvného registru AC. V tomto okamžiku se obsah prvního registru AC označí jako "původní obsah" a hodnota těchto bitů se zapisuje AC(x).

4. Druhý posuvný registr R se vynuluje signály, přicházejícími na skupinu řídicích vstupů 32 z řadiče Ř.

5. Signálem na skupině řídicích vstupů 13 se provede rotace obsahu prvního posuvného registru AC, až jeho původní bit AC(0) se nastaví proti bitu R0 druhého posuvného registru R. Pokud je tento krok prvním rotačním krokem při vytváření cyklického zabezpečovacího kódu, pak tato rotace je nulová.

6. Pomocí signálů na skupinách řídicích vstupů 13, 23, 32 se k

obsahu druhého posuvného registru R přičtou modulo 2 bity prvního posuvného registru AC, a sice bity AC0 až AC11 s původním obsahem, zbývající bity, to je AC12 až AC15 se přičtou s nulovou hodnotou bez ohledu na jejich původní stav.

7. Signálem na skupině řídicích vstupů 13 se provede rotace obsahu prvního posuvného registru AC, až jeho původní bit AC(4) se nastaví proti bitu R5 druhého posuvného registru R.

8. Pomocí signálů na skupinách řídicích vstupů 13, 23, 32 se k obsahu druhého posuvného registru R přičtou modulo 2 bity AC5 až AC11 prvního posuvného registru AC s původním obsahem, zbývající bity, to je AC0 až AC4 a AC12 až AC15 se přičtou s nulovou hodnotou bez ohledu na jejich původní stav.

9. Pomocí signálů na skupině řídicích vstupů 13 se provede rotace prvního posuvného registru AC, až jeho původní bit AC(8) se nastaví proti bitu R12 druhého posuvného registru R.

10. Pomocí signálů přivedených na skupiny řídicích vstupů 13, 23 a 32 se k obsahu druhého posuvného registru R přičtou modulo 2 bity AC12 až AC15 prvního posuvného registru AC s původním obsahem, zbývající bity, to je bity AC0 až AC11 se přičtou s nulovou hodnotou, bez ohledu na jejich původní stav.

11. Signálem na skupině řídicích vstupů 13 se provede rotace obsahu prvního posuvného registru AC, až jeho původní bit AC(0) se nastaví proti bitu R5 druhého posuvného registru R.

12. Pomocí signálů, přivedených na skupiny řídicích vstupů 13, 23 a 32 se k obsahu druhého posuvného registru R přičtou modulo 2 bity AC5 až AC15 prvního posuvného registru AC s původním obsahem, zbývající bity, to je bity AC0 až AC4 se přičtou s nulovou hodnotou bez ohledu na jejich původní stav.

13. Signálem na skupině řídicích vstupů 13 se provede rotace obsahu prvního posuvného registru AC, až jeho původní bit AC(12) se nastaví proti bitu R0 druhého posuvného registru R.

14. Pomocí signálů, přivedených na skupiny řídicích vstupů 13, 23 a 32, se k obsahu druhého posuvného registru R přičtou modulo 2 bity AC0 až AC3 prvního posuvného registru AC s původním obsahem, zbývající bity, to je AC4 až AC15 se přičtou s nulovou hodnotou bez ohledu na jejich původní stav.

15. Pomocí signálů na skupině řídicích vstupů 13 se provede rotace obsahu prvního posuvného registru AC, až jeho původní bit

AC(11) se nastaví proti bitu R0 druhého posuvného registru R.

16. Signály na skupinách řídicích vstupů 13, 23, 32 se k obsahu druhého posuvného registru R přičtou modulo 2 bity AC0 až AC4 prvního posuvného registru AC s původním obsahem, zbývající bity, to je AC5 až AC15 se přičtou s nulovou hodnotou bez ohledu na jejich původní stav.

17. Pomocí signálů na skupině řídicích vstupů 13 se provede rotace obsahu prvního posuvného registru AC, až jeho původní bit AC(4) se nastaví proti bitu R12 druhého posuvného registru R.

18. Pomocí signálů na skupinách řídicích vstupů 13, 23, 32 se k obsahu druhého posuvného registru R přičtou modulo 2 bity AC12 až AC15 prvního posuvného registru AC s původním obsahem, zbývající bity, to je AC0 až AC11 se přičtou s nulovou hodnotou bez ohledu na jejich původní stav.

19. Pomocí signálů na skupině řídicích vstupů 13 se provede rotace obsahu prvního posuvného registru AC, až jeho původní bit AC(8) se nastaví proti bitu R0 druhého posuvného registru R.

20. Pomocí signálů na skupinách řídicích vstupů 13, 23, 32 se k obsahu druhého posuvného registru R přičtou modulo 2 bity AC0 až AC7 prvního posuvného registru AC s původním obsahem, zbývající bity, to je AC8 až AC15 se přičtou s nulovou hodnotou bez ohledu na jejich původní stav.

21. Pomocí signálů na skupině řídicích vstupů 13 se provede rotace obsahu prvního posuvného registru AC, až jeho původní bit AC(12) se nastaví proti bitu R5 druhého posuvného registru R.

22. Pomocí signálů na skupinách řídicích vstupů 13, 23, 32 se k obsahu druhého posuvného registru R přičtou modulo 2 bity AC5 až AC8 prvního posuvného registru AC s původním obsahem, zbývající bity, to je AC0 až AC4 a AC9 až AC15 se přičtou s nulovou hodnotou bez ohledu na jejich původní stav.

23. Pomocí signálů na skupině řídicích vstupů 13 se provede rotace obsahu prvního posuvného registru AC, až jeho původní bit AC(11) se nastaví proti bitu R5 druhého posuvného registru R.

24. Pomocí signálů na skupinách řídicích vstupů 13, 23, 32 se k obsahu druhého posuvného registru R přičtou modulo 2 bity AC5 až AC9 prvního posuvného registru AC s původním obsahem, zbývající bity, to je AC0 až AC4 a AC10 až AC15 se přičtou s nulovou hodnotou bez ohledu na jejich původní stav.

25. Pomocí signálů na skupině řídicích vstupů 13 se provede rotace obsahu prvního posuvného registru AC, až jeho původní bit AC(0) se nastaví proti bitu R12 druhého posuvného registru R.

26. Pomocí signálů na skupinách řídicích vstupů 13, 23, 32 se k obsahu druhého posuvného registru R přičtou modulo 2 bity AC12 až AC15 prvního posuvného registru AC s původním obsahem, zbývající bity, to je AC0 až AC11 se přičtou s nulovou hodnotou bez ohledu na jejich původní stav.

27. Pomocí signálů na skupině řídicích vstupů 13 se provede rotace obsahu prvního posuvného registru AC, až jeho původní bit AC(4) se nastaví proti bitu R0 druhého posuvného registru R.

28. Pomocí signálů na skupinách řídicích vstupů 13, 23, 32 se k obsahu druhého posuvného registru R přičtou modulo 2 bity AC0 až AC11 prvního posuvného registru AC s původním obsahem, zbývající bity prvního posuvného registru AC, to je AC12 až AC15 se přičtou s nulovou hodnotou bez ohledu na jejich původní stav.

29. Pomocí signálů na skupině řídicích vstupů 13 se provede rotace obsahu prvního posuvného registru AC tak, až jeho původní bit AC(8) se nastaví proti bitu R5 druhého posuvného registru R.

30. Pomocí signálů na skupinách řídicích vstupů 13, 23, 32 se k obsahu druhého posuvného registru R přičtou modulo 2 bity AC5 až AC12 prvního posuvného registru AC s původním obsahem, zbývající bity, to je AC0 až AC4 a AC13 až AC15 se přičtou s nulovou hodnotou bez ohledu na jejich původní stav.

Popsaný postup probíhá po každém převzetí šestnáctibitového slova na první skupinu vstupů 11 prvního posuvného registru AC, a sice od druhého kroku, a to tak dlouho, až přijde poslední datové slovo z přenášeného toku dat. Po ukončení přenosu bloku dat je ve druhém posuvném registru R konečná hodnota cyklického zabezpečovacího kódu, z jehož druhé skupiny výstupů 032 se data přivádějí k dalšímu zpracování. Pokud přenos dat není ukončen, pak se v druhém posuvném registru R nachází mezivýsledek cyklického zabezpečovacího kódu. Postup je dále zřejmý z obr. 2 a 3.

Při alternativním postupu vytváření cyklického zabezpečovacího kódu lze využít vazby mezi bitem s nejnižší hodnotou a bitem s nejvyšší hodnotou prvního posuvného registru AC. V tom případě lze spojit desátý krok se čtrnáctým krokem, přičemž třináctý krok odpadne, to znamená, že se přičtou bity AC0 až AC3 a AC12 až AC15 s původním obsahem, zbývající bity, to je AC4 až AC11 se přičtou

s nulovou hodnotou bez ohledu na jejich původní stav. Dále lze spojit dvanáctý krok se šestnáctým krokem, přičemž patnáctý krok odpadne, to znamená, že se přičtou bity AC0 až AC15, zbývající bity nejsou. Konečně lze spojit dvacátý šestý krok s dvacátým osmým krokem, přičemž dvacátý sedmý krok odpadne, to znamená, že se přičtou bity AC0 až AC15, zbývající bity nejsou. Při tomto postupu nezáleží na směru rotace obsahu prvního posuvného registru AC. Postup je zřejmý z obr. 2, na němž jsou uvedeny pouze sudé, to znamená sečítací kroky.

V případě, že bit s nejnižší hodnotou nenavazuje přímo na bit s nejvyšší hodnotou, jako tomu je u dvanáctého a čtrnáctého kroku, pak se první posuvný registr AC prodlouží o jeden bit, a sice zařazením neznázorněného pomocného klopného obvodu nebo podobného paměťového prvku. Tím vzniknou dva případy, závisející na směru rotace obsahu prvního registru AC. V prvním případě směřuje rotace obsahu posuvného registru AC směrem k bitu s nižší hodnotou. Pak se pomocný klopný obvod zařazuje za výstup bitu AC0 s nejnižší hodnotou prvního posuvného registru AC a jeho výstup se pak propojí se vstupem bitu AC15 s nejvyšší hodnotou. V druhém případě směřuje rotace obsahu prvního posuvného registru AC směrem k bitu AC15 s nejvyšší hodnotou. Pomocný klopný obvod se pak zařazuje před bit AC15 s nejvyšší hodnotou prvního posuvného registru AC a jeho výstup se spojí se vstupem bitu AC0 s nejnižší hodnotou. V obou případech lze spojit dvanáctý a čtrnáctý krok, přičemž se přičtou bity AC0 až AC3 a AC5 až AC15, zbývající bit AC4 se přičte s nulovou hodnotou bez ohledu na jejich původní stav. Pořadí dvojic kroků, sestávajících vždy z lichého, to je rotačního kroku a jemu bezprostředně následujícího sudého, to je sečítacího kroku, lze měnit. Na příklad po první dvojici kroků, sestávající z pátého a šestého kroku, může následovat dvojice kroků, sestávající z devátého a desátého kroku. Zapojení lze realizovat též dynamicky vnitřními registry mikroprocesoru spolu s jeho řídicími obvody jako řadičem pouze na dobu potřebnou pro vytvoření částečných mezivýsledků cyklického zabezpečovacího kódu v čekacím čase serializace a deserializace šestnáctibitových slov.

Vynálezu lze využít při přenosu dat, u nichž se generuje cyklický zabezpečovací kód.

P Ř E D M Ě T V Y N Á L E Z U

258 994

1. Zapojení na vytváření cyklického zabezpečovacího kódu ze šestnáctibitových slov s registry a obvodem součtu modulo 2, vyznačené tím, že první skupina vstupů (11) prvního registru (AC) tvoří současně skupinu vstupů (10) zapojení, skupina výstupů (011) prvního registru (AC) je připojena na první skupinu vstupů (21) obvodu (OSM2) součtu modulo 2, jehož první skupina výstupů (021) je připojena na druhou skupinu vstupů (12) prvního registru (AC), druhá skupina výstupů (022) obvodu (OSM2) součtu modulo 2 je připojena na první skupinu vstupů (31) druhého registru (R), jehož první skupina výstupů (031) je připojena na druhou skupinu vstupů (22) obvodu (OSM2) součtu modulo 2, druhá skupina výstupů (032) druhého registru (R) tvoří současně skupinu výstupů (30) zapojení, první skupina výstupů (041) řadiče (Ř) je připojena na skupinu řídicích vstupů (13) prvního registru (AC), druhá skupina výstupů (042) řadiče (Ř) je připojena na skupinu řídicích vstupů (23) obvodu (OSM2) součtu modulo 2, třetí skupina výstupů (043) řadiče (Ř) je připojena na skupinu řídicích vstupů (32) druhého registru (R), skupina ovládacích vstupů (41) řadiče (Ř) tvoří současně skupinu ovládacích vstupů (40) zapojení.

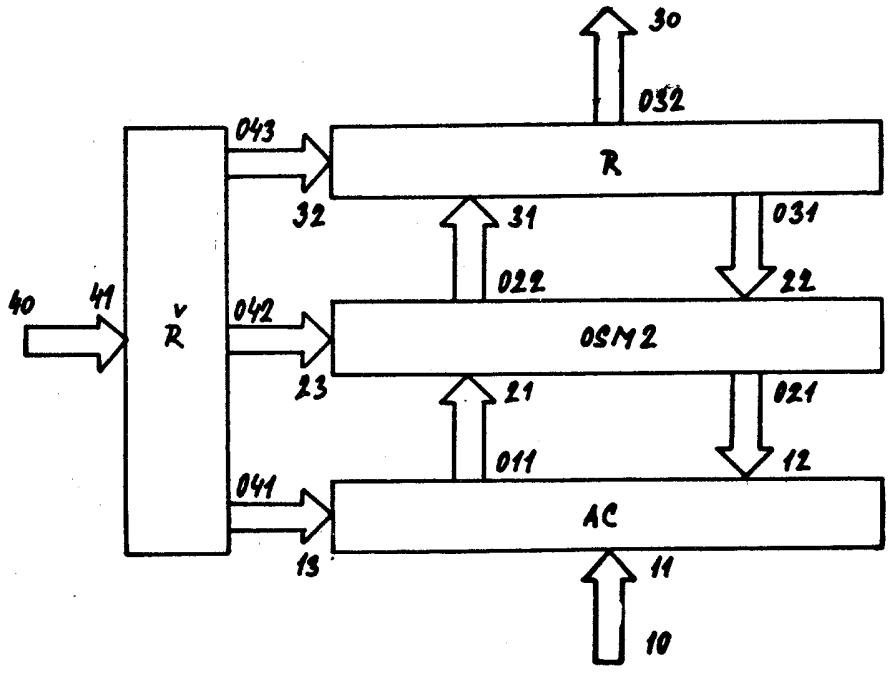
2. Zapojení podle bodu 1, vyznačené tím, že za výstupem jednoho krajního bitu prvního registru (AC) je zapojen paměťový prvek, jehož výstup je připojen ke vstupu druhého krajního bitu prvního registru (AC).

2 výkresy

R

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KROK:																
6.					11	10	9	8	7	6	5	4	3	2	1	0
8.					10	9	8	7	6	5	4					
10.	11	10	9	8												
12.	10	9	8	7	6	5	4	3	2	1	0					
14.													15	14	13	12
16.												15	14	13	12	11
18.	7	6	5	4												
20.									15	14	13	12	11	10	9	8
22.								15	14	13	12					
24.							15	14	13	12	11					
26.	3	2	1	0												
28.					15	14	13	12	11	10	9	8	7	6	5	4
30.				15	14	13	12	11	10	9	8					

Obr. 2



Obr. 1

KROK	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BIT
1.	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	R F F F F
2.	1	0	1	0	0	0	0	1	1	0	1	0	0	0	0	1	DATA AC A 1 A 1
3.	0	1	0	1	1	1	1	0	0	1	0	1	1	1	1	0	$R_{0-15} \oplus AC_{0-15} \rightarrow AC$
4.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$0 \rightarrow R$
5.	0	0	0	0	1	1	1	0	0	1	0	1	1	1	1	0	
6.	0	0	0	0	1	1	1	0	0	1	0	1	1	1	1	0	$R \oplus AC_{0-11} \rightarrow R$
7.	0	0	0	0	1	1	0	0	1	0	1	0	0	0	0	0	
8.	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	0	$R \oplus AC_{5-11} \rightarrow R$
9.	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
10.	1	1	1	0	0	0	1	0	1	1	1	1	1	1	1	0	$R \oplus AC_{12-15} \rightarrow R$
11.	1	1	0	0	1	0	1	1	1	1	0	0	0	0	0	0	
12.	0	0	1	0	1	0	0	1	0	0	1	1	1	1	1	0	$R \oplus AC_{5-15} \rightarrow R$
13.	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	
14.	0	0	1	0	1	0	0	1	0	0	1	1	1	0	1	1	$R \oplus AC_{0-3} \rightarrow R$
15.	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	
16.	0	0	1	0	1	0	0	1	0	0	1	1	0	0	0	0	$R \oplus AC_{0-4} \rightarrow R$
17.	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
18.	0	1	1	1	1	0	0	1	0	0	1	1	0	0	0	0	$R \oplus AC_{12-15} \rightarrow R$
19.	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	
20.	0	1	1	1	1	0	0	1	0	1	1	0	1	1	1	0	$R \oplus AC_{0-7} \rightarrow R$
21.	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	
22.	0	1	1	1	1	0	0	1	1	1	0	0	1	1	1	0	$R \oplus AC_{5-8} \rightarrow R$
23.	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	
24.	0	1	1	1	1	0	0	0	1	0	1	0	1	1	1	0	$R \oplus AC_{5-9} \rightarrow R$
25.	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
26.	1	0	0	1	1	0	0	0	1	0	1	0	1	1	1	0	$R \oplus AC_{12-15} \rightarrow R$
27.	0	0	0	0	0	1	0	1	1	1	1	0	0	1	0	1	
28.	1	0	0	1	1	1	0	1	0	1	0	0	1	0	1	1	$R \oplus AC_{0-11} \rightarrow R$
29.	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	
CRC 30.	1	0	0	1	0	1	1	0	1	0	0	0	1	0	1	1	$R \oplus AC_{5-12} \rightarrow R = 9683$

Obr. 3

Vytiskly Moravské tiskařské závody,
střed. 11 100, tř.Lidových milicí 3, Olomouc Cena: 2,40 Kčs