SYNTHESIS OF THE COMBINED SELF-TESTING DEVICES
DISKRĒTO SISTĒMU KOMBINĒTO PAŠPĀRBAUDOŠO IERĪČU SINTĒZE

M. Mezītis, L. Sergejeva

Atslēgas vārdi: *pašpārbaudošs, testeru sintēze, kodu pārveidotājs*

Ievads

Pašpāprbaude, kā diskrēta aparatūras īpatnība tiek bieži lietota vadoša un skaitļošanas sistēmu izstrādē. Šī īpatnība ļauj efektīvi risināt problēmas kas saistīti ar drošības un kontroliespējas paaugstināšanu. Sistēma tiek saukta par pašpārbaudošas, ja tās funkcionēšanas laikā tiek nodrošināta iekšējo elementu bojājumu kontrole un kurā speciāla izeja formē bojājuma signālu, ja parādās bojājums. Tādā veidā sistēmai ir iekšējie papildresursi kuri kontrolē savu darbību. Šī īpatnība kopā ar dublēšanu dot iespēju sasniegt augstus drošības rādītājus.

Dotajā rakstā tiek apskatīta diskrēto ierīču uzbūves metode, kuras izmanto viensvara kodu nCm pie m=4. Pastāv universāla sintēzes metode m/n – pašpārbaudošo testeru, kas ļauj veidot testerus ar jebkādiem rādītājiem m un n. Universālo metožu trūkums ir tas, ka viņas pilnā mērā neievēro konkrēto kodu īpatnības. Tādu īpatnību izmantošana ļauj uzlabot testeru raksturojamus.

Testeri, pašpārbaudošo ierīču klasē izskatās kā kontroles shēmas. Tajā pat laikā, tas ir neatņemama daļa no šīm ierīcēm, un tāpēc izrāda ietekme uz tādiem svarīgiem diskrēto ierīču raksturojumiem, kā drošums, ātrdarbība un sarežģītība.

Pamata definīcijas

Pašpārbaudošs testeris pēc savas galvenās būtības ir kodu detektors (vai kodu dalījuma shēma)[1], kura uzdevums ir atšķirt koda vektorus, no visiem pārējiem iespējamiem vektoriem. Parastam detektoram ir viena izeja, kura tiek apraksta ar funkciju, kas noteikta pēc dizjunkcijas visiem koda vektoriem. Lai detektors iegūtu pašpārbaudes īpašības tam pievieno otru izeju. Tas ļauj konstruēt detektoru, kā kodu pārveidotāju, kura izejā formējās kaut kāda koda vārdi. Tad katrs vārds dotā koda transformējas detektora ieejā kādā citā izejoša koda vārdā, bet nekodēts vārds detektora ieejā par nekodētu vārdu izejā. Tādā veidā tiek realizēta ieejošā vektora kontroles īpašība. Pie tam rodas iespēja fiksēt detektora iekšējās struktūras bojājumu uzstādot uz izejas nekodētu vārdu.

Lai salīdzinātu testerus savā starpā tiek izmantoti sekojoši raksturojumi:

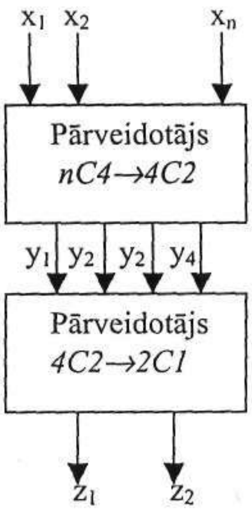
1) Sarežģījuma līmenis L - loģisko elementu kopējais ieeju skaits. Testeri tiek aprakstīti ar monotonu buļļu funkcijām, kurai uzbūvei tiek izmantoti elementi, kas realizē dizjunkcijas un konjunkcijas operācijas un pie kam elementu ieejas skaits netiek ierobežots.
2) Sarežģījuma līmenis N – loģisko elementu skaits pie augšminētiem nosacījumiem.
3) Pārbaudošo testu garums t – nCm koda vārdu skaits, kuru nokļūšana testera ieejā nodrošina visu viendabīgo kļūdu konstatāciju tā struktūrā, pamatojoties uz pašpārbaudes īpašību.
4) Posmu skaits r – maksimālo elementu skaits, caur kuru iet ceļš testera shēmā, kas savieno tā ieeju un izeju.

Literatūrā [2-6] tiek piedāvātas universālas *m/n* testeru sintēzes metodes, kas ļauj izveidot testeru ar jebkādiem rādītājiem *m* un *n*. Katra sintēzes metode dot savu testeru ar noteiktiem raksturojumiem.

Jebkurš *m/n* pašpārbaudošs testeris izskatās kā kodu pārveidotājs *nCm→2C1*. Lai sintezētu 4/n testeri tiek izstrādāti dažas pieejas. Tālāk tiek apskatīta pašpārbaudošo testeru sintēze, kas pamatota uz kodu pārveidojumu un ļauj ņemt vērā konkrēta koda īpatnības.

Pašpārbaudājamo testeru sintēze pie $9 \leq n \leq 16$

Šajā gadījumā testeri tiek sintezēti uz sekojošo kodu pārveidošanas pamata: *nC4→4C2→2C1*.Testera struktūra tiek parādīta att. 1



1. attēls. Testera struktūra

Ja testeris tiek realizēts kā shēma ar diviem blokiem, kas savā starpā saslēgti virknē, tad jāizpildās diviem nosacījumiem:

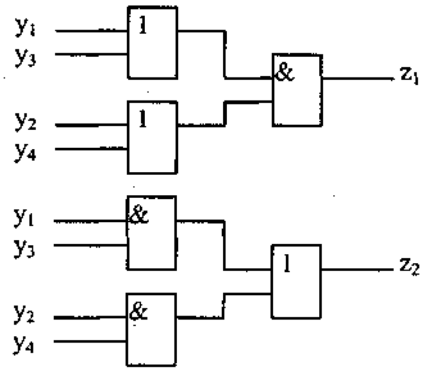
1) Pārveidotājiem *nC4→4C2* un *4C2→2C1* jābūt ienākoša vektora kontrole īpašībām un pašpārbaudes īpašībām. Ar šo mērķi *4C2→2C1* pārveidotāja vietā tiek izmantots 2/4-pašpārbaudājamais testeris, bet *nC4→4C2* pārveidotājs tiek izveidots tādā veidā, lai tā izejā paradās vektors kas nepieder *4C2* kodam ja ieejā padots vektors kas nepieder kodam *nC3*.

2) Pareizi funkcionējošā pārveidotāja *nC4→4C2* izejā ir jāformējas tādam daudzumam koda 4C2 vārdu, kas satur sevī pārbaudošo testu 2/4 - testerīm.

Pašpārbaudāmā 2/4 - testera vispiemērotākais shēmas variants būs divpakāpju shēma [1] kas paradīta att. 2.

1. tabula. Pārbaudes tests

y_1	y_2	y_3	y_4
0	1	0	1
1	0	0	1
0	1	1	0
1	0	1	0



2. attēls. 2/4 – pašpārbaudojošais testeris

Dotajam testerim ir vislabākie raksturojumi: L=12, N=6, r=2, t=4. Pārbaudes tests parādīts 1. tabulā. Sintezējot pārveidotāju $nC4 \rightarrow 4C2$ nepieciešams izmantot bāzes funkcijas sadalīšanas principu [4]:

$$f^m(x_1;x_2;..x_n) = \vee x_{i1}x_{i2}..x_{im} \qquad i_1;i_2;..i_m \in \{1,2,..n\} \tag{1}$$

kura tiek iegūta savienojot dizjunkcijas zīmi konjunkcijas ar rangu m:

$$\begin{aligned} f^m(x_1,..,x_k,..,x_n) = f^m(x_1,..,x_k) \vee f^m(x_{k+1},..,x_n) \vee f^{m-1}(x_1,..,x_k)f^1(x_{k+1},..,x_n) \vee \\ \vee f^{m-2}(x_1,..,x_k)f^2(x_{k+1},..,x_n) \vee .. \vee f^1(x_1,..,x_k)f^{m-1}(x_{k+1},..,x_n) \end{aligned} \tag{2}$$

Dotajā gadījumā:

$$\begin{aligned} f^4(x_1,..,x_k,..,x_n) = f^4(x_1,..,x_k) \vee f^3(x_1,..,x_k)f^1(x_{k+1},..,x_n) \vee \\ \vee f^2(x_1,..,x_k)f^2(x_{k+1},..,x_n) \vee f^1(x_{1,..,}x_k)f^3(x_{k+1},..,x_n) \vee f^4(x_{k+1},..,x_n) \end{aligned} \tag{3}$$

Pārveidošana (2) tiek saistīta ar daudzumu A={x_1, ..x_k,..,x_n} sadalīšanu uz diviem daudzumiem A_1={ x_1, ..x_k} un A_2= {x_k,..,x_n}. Testera īpašības ir atkarīgas no skaitļa k katrā dalīšanas etapā. Apskatāmā gadījumā k=]n/2[, kur]n/2[tuvākais pie n/2 pilns skaitlis.

Funkcijas y_1-y_4 kas apraksta pārveidotāju $nC4 \rightarrow 4C2$ tiek meklētas ar sekojošo algoritmu palīdzību:

1) Aprakstojoša funkcija $f^4(x_1,..,x_n)$ tiek sadalīta pēc (2).
2) Funkcijas $f^4(x_1,..,x_k)$ un $f^4(x_{k+1},..,x_n)$ kas ir dabūtas pēc pirmā punkta, arī tiek sadalītas pēc (2)

$$\begin{aligned} & f^4(x_1,..,x_l,..,x_k) = f^4(x_1,..,x_l) \vee f^3(x_1,..,x_l)f^1(x_{l+1},..,x_k) \vee f^2(x_1,..,x_l)f^2(x_{l+1},..,x_k) \vee \\ & f^1(x_1,..,x_l)f^3(x_{l+1},..,x_k) \vee f^4(x_{l+1},..,x_k) \\ & f^4(x_{k+1},..,x_d,..,x_n) = f^4(x_{k+1},..,x_d) \vee f^3(x_{k+1},..,x_d)f^1(x_{d+1},..,x_n) \vee f^2(x_{k+1},..,x_d)f^2(x_{d+1},..,x_n) \vee \\ & f^1(x_{k+1},..,x_d)f^2(x_{d+1},..,x_n) \vee f^4(x_{d+1},..,x_n) \end{aligned} \tag{4}$$

kur l=]k/2[
d=k+](n-2)/2[

3) Funkcijas y_1-y_4 tiek sastādītas no augstāk saņemtajiem elementiem pēc sekojošas formulas:

$$
\begin{aligned}
&y_1 = f^2(x_1,..,x_k) \vee f^3(x_{k+1},..,x_d)f^1(x_{d+1},..,x_n) \vee f^1(x_{k+1},..,x_d)f^3(x_{d+1},..,x_n) \\
&y_2 = f^2(x_{k+1},..,x_n) \vee f^3(x_1,..,x_l)f^1(x_{l+1},..,x_k) \vee f^1(x_{k+1},..,x_d)f^3(x_{l+1},..,x_k) \\
&y_3 = f^3(x_1,..,x_k)f^1(x_{k+1},..,x_n) \vee f^1(x_1,..,x_k)f^3(x_{k+1},..,x_n) \\
&y_4 = f^4(x_1,..,x_l) \vee f^2(x_1,..,x_l)f^2(x_{l+1},..,x_k) \vee f^4(x_{l+1},..,x_k) \vee f^4(x_{k+1},..,x_d) \vee \\
&\vee f^2(x_{k+1},..,x_d)f^2(x_{d+1},..,x_n) \vee f^4(x_{d+1},..,x_n)
\end{aligned}
\tag{5}
$$

Obligāti vajag atzīmēt to, ka tas nav vienīgais iespējamais risinājuma variants, lai atrast funkcijas y_1-y_4.

Piemēram, nepieciešams sintezēt testeri priekš koda *16C4*. Daudzums $A=\{x_1,x_2,x_3,..,x_{16}\}$ tiek sadalīts uz diviem: $A_1=\{x_1,..,x_8\}$ un $A_2=\{x_7,..,x_{16}\}$ saskaņā ar kuriem notiek sekojošā bāzes funkcijas sadalīšana:

$$
\begin{aligned}
&f^4(x_1,..,x_8,..,x_{16}) = f^4(x_1,..,x_8) \vee f^3(x_1,..,x_8)f^1(x_9,..,x_{16}) \vee \\
&\vee f^2(x_1,..,x_8)f^2(x_9,..,x_{16}) \vee f^1(x_1,..,x_8)f^3(x_9,..,x_{16}) \vee f^4(x_9,..,x_{16})
\end{aligned}
\tag{6}
$$

Katrs daudzums A_1 un A_2 arī tiek sadalīti: $A_{11}=\{x_1,..,x_4\}$; $A_{12}=\{x_5,..,x_8\}$; $A_{21}=\{x_9,..,x_{12}\}$; $A_{22}=\{x_{13},..,x_{16}\}$. Sakarā ar to:

$$
\begin{aligned}
&f^4(x_1,..,x_8) = f^4(x_1,..,x_4) \vee f^3(x_1,..,x_4)f^1(x_5,..,x_8) \vee \\
&\vee f^2(x_1,..,x_4)f^2(x_5,..,x_8) \vee f^1(x_1,..,x_4)f^3(x_5,..,x_8) \vee f^4(x_5,..,x_8) \\
&f^4(x_9,..,x_{16}) = f^4(x_9,..,x_{12}) \vee f^3(x_9,..,x_{12})f^1(x_{13},..,x_{16}) \vee \\
&\vee f^2(x_9,..,x_{12})f^2(x_{13},..,x_{16}) \vee f^1(x_9,..,x_{12})f^3(x_{13},..,x_{16}) \vee f^4(x_{13},..,x_{16})
\end{aligned}
\tag{7}
$$

Ar sistēmas (5) palīdzību var definēt funkcijas y_1-y_4, kas apraksta pārveidotāju *16C4→4C2*:

$$
\begin{aligned}
&y_1 = f^2(x_1,..,x_8) \vee f^3(x_9,..,x_{12})f^1(x_{13},..,x_{16}) \vee f^1(x_9,..,x_{12})f^3(x_{13},..,x_{16}) \\
&y_2 = f^2(x_9,..,x_{16}) \vee f^3(x_1,..,x_4)f^1(x_5,..,x_8) \vee f^1(x_1,..,x_4)f^3(x_5,..,x_8) \\
&y_3 = f^3(x_1,..,x_8)f^1(x_9,..,x_{16}) \vee f^1(x_1,..,x_8)f^3(x_9,..,x_{16}) \\
&y_4 = f^4(x_1,..,x_4) \vee f^2(x_1,..,x_4)f^2(x_5,..,x_8) \vee f^4(x_5,..,x_8) \vee f^4(x_9,..,x_{12}) \vee \\
&\vee f^2(x_9,..,x_{12})f^2(x_{13},..,x_{16})f^4(x_{13},..,x_{16})
\end{aligned}
\tag{8}
$$

Pārveidotāji *nC4→4C2* tiek izveidoti realizējot funkcijas y_1-y_4, kā saistības daudzpakāpju shēmas ar atzarojumiem. Tādu shēmu uzbūves procedūra tiek aprakstīta [4]. Pie tā funkcijas y_1-y_4 tiek piedāvātas kā strupceļa formulas.

Pašpārbaudājamo testeru sintēze pie $17 \le n \le 24$

Apskatīsim funkciju $f^4(x_1,x_2,..,x_n)$. Daudzumu $A=\{x_1,x_2,..,x_n\}$ sadalīsim uz *3* daudzumiem: $A_1=\{x_1,..,x_{k1}\}$, $A_2=\{x_{k1+1},..,x_{k2}\}$, $A_d=\{x_{kd-1+1},..,x_n\}$ tādā veida lai katrs no daudzumiem satur ne mazāk par 4 un ne vairāk par 8 elementiem. Ar tādu sadalījumu iegūsim bāzes funkcijas sadalīšanu:

$$
\begin{aligned}
&f^4(x_1,..,x_n) = f^4(x_1,..,x_{k_1}) \vee f^4(x_{k_1+1},..,x_{k_2}) \vee f^4(x_{k_2+1},..,x_n) \vee f^3(x_1,..,x_{k_1})f^1(x_{k_1+1},..,x_{k_2}) \vee \\
&\vee f^3(x_1,..,x_{k_1})f^1(x_{k_2+1},..,x_n) \vee f^2(x_1,..,x_{k_1})f^2(x_{k_1+1},..,x_{k_2}) \vee f^2(x_1,..,x_{k_1})f^2(x_{k_2+1},..,x_n) \vee \\
&\vee f^2(x_1,..,x_{k_1})f^2(x_{k_1+1},..,x_{k_2}) \vee f^2(x_1,..,x_{k_1})f^2(x_{k_2+1},..,x_n) \vee f^2(x_1,..,x_{k1})f^1(x_{k_1+1},..,x_{k_2}) * \\
&* f^1(x_{k_2+1},..,x_n) \vee f^1(x_1,..,x_{k_1})f^3(x_{k_1+1},..,x_{k_2}) \vee f^1(x_1,..,x_{k_1})f^3(x_{k_2+1},..,x_n) \vee \\
&\vee f^1(x_1,..,x_{k1})f^2(x_{k_1+1},..,x_{k_2})f^1(x_{k_2+1},..,x_n) \vee f^1(x_1,..,x_{k1})f^1(x_{k_1+1},..,x_{k_2})f^2(x_{k_2+1},..,x_n) \vee \\
&\vee f^3(x_{k_1+1},..,x_{k_2})f^1(x_{k_2+1},..,x_n) \vee f^2(x_{k_1+1},..,x_{k_2})f^2(x_{k_2+1},..,x_n) \vee f^1(x_{k_1+1},..,x_{k_2})f^3(x_{k_2+1},..,x_n)
\end{aligned}
\tag{9}
$$

Labajā daļā (9) atrodas funkcijas $F_1=f^4(x_1,..,x_{k1})$, $F_2=f^4(x_{k1+1},..,x_{k2})$,$F_3=f^4(x_{k2+1},..,x_n)$ pie kurām arī pielietosim sadalījumu (2). Piemēram:

$$
\begin{aligned}
F_1 &= f^4(x_1,..,x_p,..,x_{k_1}) = f^4(x_1,..,x_p) \vee f^3(x_1,..,x_p)f^1(x_{p+1},..,x_{k_1}) \vee \\
&\vee f^2(x_1,..,x_p)f^2(x_{p+1},..,x_{k_1}) \vee f^1(x_1,..,x_p)f^3(x_{p+1},..,x_{k_1}) \vee f^4(x_{p+1},..,x_{k_1})
\end{aligned}
\tag{10}
$$

kur $p=]k1/2[$

Jā pieņemt sekojošo apzīmējumu:

$$
\begin{aligned}
F_1^1 &= f^4(x_1,..,x_p) \vee f^2(x_1,..,x_p)f^2(x_{p+1},..,x_{k_1}) \vee f^4(x_{p+1},..,x_{k_1}) \\
F_1^2 &= f^3(x_1,..,x_p)f^1(x_{p+1},..,x_{k_1}) \vee f^1(x_1,..,x_p)f^3(x_{p+1},..,x_{k_1})
\end{aligned}
\tag{11}
$$

tad $F_1 = F_1^1 \vee F_1^2$ un kopējā gadījumā $F_i = F_i^1 \vee F_i^2$ ($i \in \{1,2,..,d\}$.

Pārveidotājs *nC4→6C3*, tiek aprakstīts ar sekojošo sistēmu:

$$
\begin{aligned}
y_1 &= f^1(x_1,..,x_{k_1}) \\
y_2 &= f^3(x_1,..,x_{k_1}) \vee f^2(x_{k_1+1},..,x_{k_2})f^2(x_{k_2+1},..,x_n) \vee F_2^1 \vee F_3^2 \\
y_3 &= f^1(x_{k_1+1},..,x_{k_2}) \\
y_4 &= f^3(x_{k_1+1},..,x_{k_2}) \vee f^2(x_1,..,x_{k_1})f^2(x_{k_2+1},..,x_n) \vee F_3^1 \vee F_1^2 \\
y_5 &= f^1(x_{k_2+1},..,x_{k_n}) \\
y_6 &= f^3(x_{k_2+1},..,x_{k_n}) \vee f^2(x_1,..,x_{k_1})f^2(x_{k_1+1},..,x_{k_2}) \vee F_1^1 \vee F_2^2
\end{aligned}
\tag{12}
$$

Piemēram priekš koda *24C4* daudzums $A=A_1 \cup A_2 \cup A_3$, kur $A_1=\{x_1,..,x_8\}$, $A_2=\{x_9,..,x_{12}\}$ un $A_3=\{x_{13},.., x_{16}\}$. Rezultātā iegūstam sadalījumu:

$$f^4(x_1,..,x_n)=f^4(x_1,..,x_8)\vee f^4(x_9,..,x_{16})\vee f^4(x_{17},..,x_{24})\vee f^3(x_1,..,x_8)f^1(x_9,..,x_{16})\vee$$
$$\vee f^3(x_1,..,x_8)f^1(x_{17},..,x_{24})\vee f^2(x_1,..,x_8)f^2(x_9,..,x_{16})\vee f^2(x_1,..,x_8)f^2(x_{17},..,x_{24})\vee$$
$$\vee f^2(x_1,..,x_8)f^2(x_9,..,x_{16})\vee f^2(x_1,..,x_8)f^2(x_{17},..,x_{24})\vee f^2(x_1,..,x_8)f^1(x_9,..,x_{16})*$$
$$*f^1(x_{17},..,x_{24})\vee f^1(x_1,..,x_8)f^3(x_9,..,x_{16})\vee f^1(x_1,..,x_8)f^3(x_{17},..,x_{24})\vee \qquad (13)$$
$$\vee f^1(x_1,..,x_8)f^2(x_9,..,x_{16})f^1(x_{17},..,x_{24})\vee f^1(x_1,..,x_8)f^1(x_9,..,x_{16})f^2(x_{17},..,x_{24})\vee$$
$$\vee f^3(x_9,..,x_{16})f^1(x_{17},..,x_{24})\vee f^2(x_9,..,x_{16})f^2(x_{17},..,x_{24})\vee f^1(x_9,..,x_{16})f^3(x_{17},..,x_{24})$$

Ar sistēmas (13) palīdzību atradām funkcijas, kas apraksta pārveidotāju 24C4→6C3:

$$y_1=f^1(x_1,..,x_8)$$
$$y_2=f^3(x_1,..,x_8)\vee f^2(x_9,..,x_{16})f^2(x_{17},..,x_{24})\vee F_2^1\vee F_3^2$$
$$y_3=f^1(x_9,..,x_{16})$$
$$y_4=f^3(x_9,..,x_{16})\vee f^2(x_1,..,x_8)f^2(x_{17},..,x_{24})\vee F_3^1\vee F_1^2 \qquad (14)$$
$$y_5=f^1(x_{17},..,x_{24})$$
$$y_6=f^3(x_{17},..,x_{24})\vee f^2(x_1,..,x_8)f^2(x_9,..,x_{16})\vee F_1^1\vee F_2^2$$

Noslēgums

Darbā tiek apskatīta iespēja sintezēt pašpārbaudošo testeri kodiem nC4 pie 9≤n≤24. Gadījums, kad n<9 netiek apskatīts, jo esošie sintēzes metodi dot vislabākās rezultātus. Pašpārbaudojoša testera raksturojumi ir atkarīgi no daudzuma A sadalīšanas metode un funkcijas y_1-y_4 meklēšanas algoritmu.

Sintezēs metode, kas tiek piedāvāta, dot arī iespēju sintezēt 4/n pašpārbaudojošo testeri no standarta blokiem – iespējams automatizēts process.

Literatūra

1.Сапожников В.В. Дискретные автоматы с обнаружением отказов. Л. Энергоатомиздат - 1984 – с 112.

2.Визирев И.С. Полностью самопроверяемые контрольные схемы с минимальном множеством тестов. //Автоматика и вычислительная техника – 1982 - №1 – с 43-49.

3.Gaitanis N. Halatsis C. A New Design Method for m-out-of-n TSC Checkers //IEEE Trans. Computer 1983-v32, № 3 – p.273-283.

4.Сапожников В.В. Сапожников Вл.В. Универсальный алгоритм синтеза самопроверяющихся тестеров для кодов с постоянным весом //Проблемы передачи информации – 1984 – т. 20 № 2 – с. 65-76.

5.Дербунович Л.В. Негивеев В.В. Проектирование полностью самопроверяемых схем контроля на программируемых логических матрицах. //Автоматика и телемеханика. – 1986 - № 4. – с. 149-156.

6.Сапожников В.В. Сапожников Вл.В. Цегловски Л. Синтез самопроверяющихся m/n – тестеров с максимальным быстродействием. // Автоматика и телемеханика. – 1988 - №10 с. 135-150.

7.Сапожников В.В. Рабара В.В. Универсальный алгоритм синтеза 1/n-тестеров. //Проблемы передачи информации. – 1982 – т18, № 3 - с.62-73.
8.Halatsis C. Gaitanis N. Sigala M. Fast and Efficient Totally Self-Checking Chekers for m-out-of-(2m±1) Codes //IEEE trans computer – 1983 – v32 № 5 – p. 507-511.
9.Piestrak S. Design Method of totally Self-Checking Checkers for M-out-of-N Codes //Proc. FTCS-13, 13th Annu. Int. Symp., Fault-Tolerant Computing – Milano, Italy –1983 – p.162-168.

Marek Mezitis, doktorands
Riga Technical University, Railway Transport Institute
Address: Indrika street No 8, LV 1004, Riga, Latvia
Phone: +371 9 169 240, Fax: +371 7 089 650
E-mail: marek@mail.lv

Ludmila Sergeyeva , Professor, Dr.sc.ing.
Riga Technical University, Railway Transport Institute
Address: Indrika street No 8, LV 1004, Riga, Latvia
Phone: +371 7 089 650, Fax: +371 7 089 650
E-mail: sla@latnet.lv

Mezītis M, Sergejeva L. Diskrēto sistēmu kombinēto pašpārbaudošo ierīču sintēze
Ir piedāvāta jauna pašpārbaudošos testeru izveidošanas metode kodiem ar pastāvīgo svaru m=4 Šī metode nodrošina testeru izveidi ar uzlabotiem raksturojumiem. Pirmajā daļā tiek aprakstīta sintēzes metode, ja ieejošā koda garums ir 9≤n≤16. Otrā daļā tiek apskatīta sintēzes metode, ja ieejošā koda garums ir vairāk nekā 17≤n≤24.

Mezitis M, Sergeyeva L. Synthesis of the combined self-testing devices
The new method of construction of self-checked testers for codes with constant weight m=4 is offered. The method provides construction of testers with the improved characteristics in comparison with known testers. In the first part of work the question of construction of self-checked testers if word length of an entrance vector is equal 9≤n≤16 is considered. In the second part of work the question of construction of self-checked testers if word length of an entrance vector more or is equal 17≤n≤24.

Мезитис М, Сергеева Л. Синтез самопроверяющихся устройств, учитывающий особенности конкретных кодов
Предложен новый метод построения самопроверяющихся тестеров для кодов с постоянным весом m=4. Метод обеспечивает построение тестеров с улучшенными характеристиками по сравнению с известными тестерами. В первой части работы рассматривается вопрос построения самопроверяющихся тестеров, если разрядность входного вектора равна 9≤n≤16. Во второй части работы рассматривается вопрос построения самопроверяющихся тестеров, если разрядность входного вектора больше или равна 17≤n≤24.