Cellulaire logica

Steltenpool, H.J.

Award date:
1972

Dit verslag vormt de afsluiting van een afstudeeropdracht welke door Prof. ir. A Heetman werd verleend. De opdracht werd, onderleiding van ir. C.P.J. Schnabel, uitgevoerd in de groep Telecommunicatie - B van de Technische Hogeschool te Eindhoven gedurende de periode: augustus 1971 - mei 1972.

Eindhoven, 25 mei 1972.

H.J. Steltenpool.

- 0 -

Inhoudsopgave.

0. Samenvatting.

De eerste opdracht is een literatuur studie over het onderwerp micro cellulaire logica. De eerste 6 hoofdstukken geven een overzicht van deze literatuur studie. De hoofdstukken 2 tot en met 4 behandelen cellulaire netwerken die een willekeurige boolse funktie kunnen realiseren. Hoofdstuk 5 geeft een overzicht van cell. netwerken, die rekenkundige operaties kunnen uitvoeren.(Delen, vermenigvuldigen, kwadrateren, worteltrekken enz.)
Hoofdstuk 6 behandelt in het kort "special purpose " netwerken, zoals sequentiële machines, codeernetwerken enz. Ook wordt in hoofdstuk 6 een kort overzicht gegeven van het onderwerp macrocellulaire logica.

Omdat de opbrengst van het fabrikage proces van cell. netwerken gering is, luidde de tweede opdracht: maak het cell. netwerk redundant, zodanig dat er een opbrengst verbetering optreedt.
In hoofdstuk 7 worden voternetwerken toegepast om het cell. redundant te maken. Er treedt dan alleen een opbrengst verbetering op voor cell. netwerken met een zeer gering aantal outputs.
In hoofdstuk 8 en 9 worden lineaire kodes toegepast. Hiermee kan slechts één enkel type cell. netwerk redundant gemaakt worden.
De toepassing van single error correcting codes geeft ook hier een opbrengst verbetering voor cell. netwerken met een gering aantal outputs.
Bij double error correcting codes is de dekodeerschakeling, uitgevoerd met cellulaire netwerken, veel komplexer dan het cell. netwerk, dat redundant gemaakt moet worden.

Hoofdstuk I

INTRODUKTIE CELLULAIRE LOGICA.

Cellulaire logica is een tak van de schakeltechniek, die de analyse en synthese van cellulaire netwerken behandelt.
De cellulaire netwerken bestaan uit gelijkvormige of identieke cellen, die met een regelmatig verbindingspatroon met elkaar doorverbonden zijn. Bij deze cellulaire-netwerken wordt een groot aantal gelijke cellen toegepast, zodat deze methode geschikt is voor large scale integration (LSI). Een voorbeeld van de fabrikage van een cellulair netwerk laat fig. Ia zien.

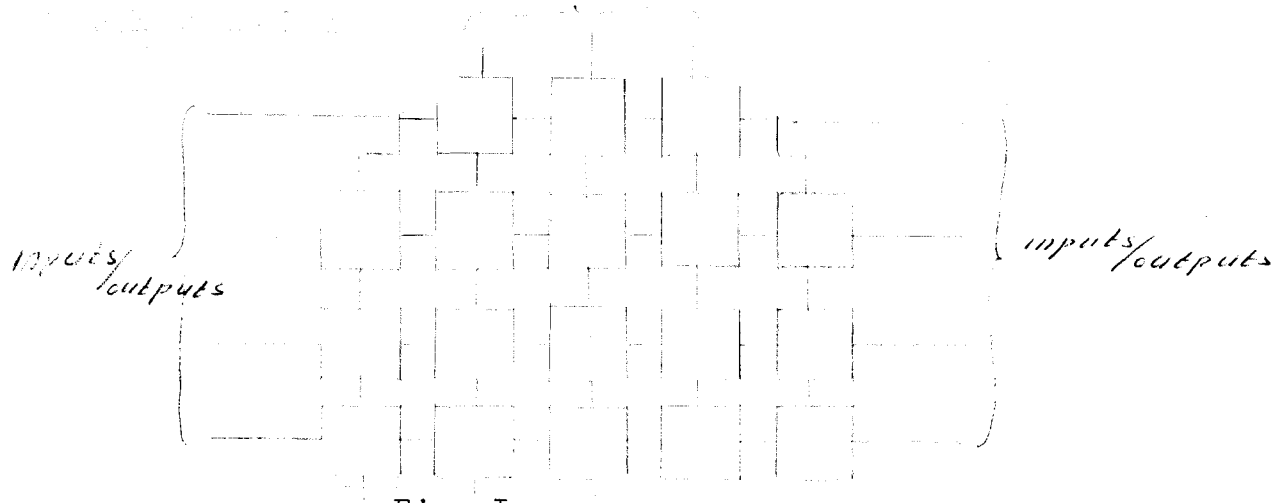


Fig. Ia

De cellen en hun onderlinge verbindingen worden door middel van monolitische techniek op een plak van b.v. 3 cm diameter in een bewerkingsproces tegelijk aangebracht. Het geheel wordt netwerk genoemd. Het netwerk wordt in een omhulsel geplaatst met in en uitgangspennen.

In fig. Ib wordt hetzelfde netwerk gemaakt m.b.v. geintegreerde circuits (IC's). Eerst worden op een plak vele cellen aangebracht. Daarna wordt de plak gebroken en iedere cel wordt in een omhulsel geplaatst (IC). Met de nu verkregen IC's wordt hetzelfde netwerk van fig. Ia opgebouwd. De IC's worden dan m.b.v. uitwendige verbindingen met elkaar doorverbonden.

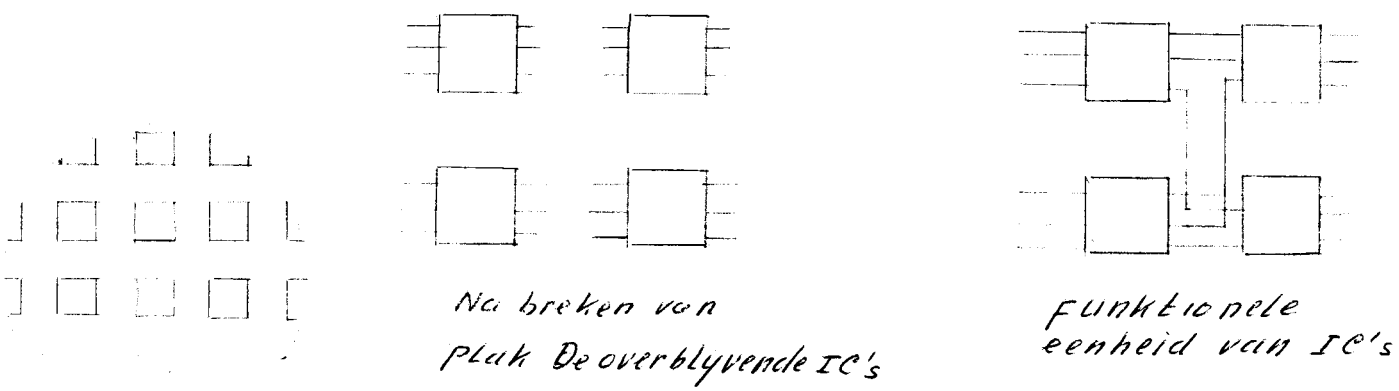


Fig. Ib

Uit bovenstaande voorbeelden is gemakkelijk in te zien welke voordelen cellulaire logica heeft t.o.v. de samenvoeging van IC's tot een funktioneel geheel.

De voordelen van cellulaire logica zijn:

a/ plak hoeft niet gebroken te worden.

b/ snellere schakeling (verbindingen compacter).

c/ De miniaturisering is groter.
Bij cellulaire logica worden echter veel meer cellen gebruikt dan bij de andere methode, maar daar iedere cel vele malen kleiner is dan een IC met omhulsel+pennen+bedrading, is cellulaire logica qua ruimte en gewicht voordeliger.

d/ door de geringe uitwendige bedrading is de betrouwbaarheid groter.

e/ Cellulaire logica netwerken kunnen goedkoper zijn dan netwerken opgebouwd uit IC's, omdat de bedrading, het aantal pennen en omhulsels geringer is. (arbeids-besparend)

De nadelen zijn:

Door het gebruik van grotere complexe monolitische geintegreerde circuits, wordt de kans op defekte cellen (componenten) in het circuit vergroot, zodat de opbrengst van het fabrikage proces gering blijft. De systeem ontwerper dient de aanwezigheid van defekte componenten te accepteren. Er wordt daarom gestreefd naar logische ontwerp algorithmes, die deze fouten in rekening brengen. (zie hfd. 4,7,8 en9) De systeem ontwerper moet fout-tolerante circuits trachten te ontwerpen.
Een ander probleem is het onderhoud. Het opsporen van defekte cellen geeft grote problemen. Het aantal beschikbare testpunten (in- en outputs) is erg klein in verhouding tot systemen opgebouwd uit afzonderlijke elementen. Hoe komplexer de cellen worden, hoe meer testen men moet toepassen aan het relatief kleiner wordende aantal testpunten.
Standaardisatie probleem.
De technologie stelt ons in staat zeer komplexe cellen te produceren, waarbij het aantal logische funkties van een cel zeer snel toeneemt bij toenemende in- en outputs. Men weet echter niet hoe men deze cellen efficiënt in logische ontwerpen moet gebruiken. Hierbij komt de volgende vraag naar voren: welke cel types moet men gebruiken zodat deze algemeen toegepast kunnen worden in digitale schakelingen ?
Een oplossing oplossing is iedere cel als een universele logische module (ULM) uit te voeren. Nadeel hiervan is de zeer snelle groei van het aantal inputs. Behandeling van de ULM zie lit 3.

LSI is nog in ontwikkeling en zal in de volgende computergeneratie zijn invloed doen gelden op het systeem ontwerp, zoals decentralisatie van geheugens, microgeprogrammeerde R.O.S., en toepassing van parallel logica. Cellulaire logica zal een belangrijke rol spelen in de uitvoering van bovengenoemde onderwerpen.
Om de flexibiliteit van de cellen te verhogen worden celparameters en/of verbindingsparameters ingebouwd. Doormiddel van de verbindingsparameters kan men de onderlinge verbinging tussen de cellen wijzigen. Doormiddel van celparameters kan men, veelal tijdens het laatste gedeelte van het produktie proces, de funkties van de cellen instellen. Door bovengenoemde methode toe te passen kan men met cellulaire netwerken van dezelfde struktuur verschillende funkties creëren overeenkomstig de wensen van de klant.
Ook worden netwerken toegepast waarbij de celparameters elektrisch ingesteld kunnen worden, zodat door middel van regelsignalen de funktie van het netwerk veranderd kan worden. Deze netwerken worden programmeerbare netwerken genoemd.

Cellulaire logica wordt vaak in twee hoofd groepen onderverdeeld :

a/ microcellulaire logica. Wordt omvat door hfd. 2 t/m 5

b/ macrocellulaire logica. Wordt omvat door hfd. 6

Bij microcellulaire logica zijn de gebruikte cellen eenvoudig van struktuur. Iedere cel bestaat uit een gering aantal poorten.

Bij macrocellulaire logica worden zeer complexe cellen gebruikt. Iedere cel bestaat uit een zeer groot aantal poorten.

Literatuur overzicht;

1 Hambata k. introduktion large scale integration.

2 Minnick R.C. A survey of microcellular Research
J.A.C.M. vol. I4 No.2, April pp. 203-24I

3 H. v.d. Weg Aantekeningen voorjaarscollege.
Beschikbaar bij afdelingbibl. van prof. Heetman

Hoofdstuk 2

Eendimensionale netwerken.

Samenvatting.

In sektie 2.1 wordt de theorie van een algemene enkelvoudige rail cascade behandeld. Dit is de eenvoudigste vorm van een cell. netwerk. Deze cascades worden toegepast om boolse funkties te realiseren. Zij zijn echter niet instaat iedere willekeurige boolse funktie te realiseren.

Sektie 2.2 behandelt de multi-rail cascade. Om een uitgebreide behandeling van dit onderwerp te voorkomen worden alleen de resultaten van diverse theorieën opgesomd. Vooral wordt gekeken naar twee rail cascades. De behandeling van deze enkel en multi-rail cascades kan het inzicht van twee dimensionale cellulaire netwerken vergemakkelijken.

Sektie 2.3 behandelt uni- en bilaterale iteratieve netwerken.

2.I. Enkelvoudige rail cascade.

2.I.I Introduktie.

De cel welke in deze cascades gebruikt wordt is de maitra cel. Deze cel heeft twee inputs en een output. De celfunktie kan alle I6 funkties van twee variabelen zijn.
Er zijn twee soorten enkelvoudige rail cascades ;

a/ Niet redundante cascade. Deze cascade bestaat uit m cellen zie fig 2.2. Er zijn m+I inputs. Op ieder cel input wordt een andere variabele aangesloten. Er moeten dus m+I verschillende variabelen aanwezig zijn.

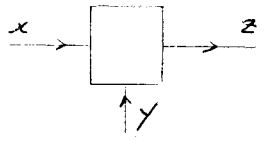


Fig. 2.1

Fig. 2.2

b/ Redundante cascade. In deze cascade kunnen twee verschillende cellen dezelfde inputs hebben. zie fig. 2.3

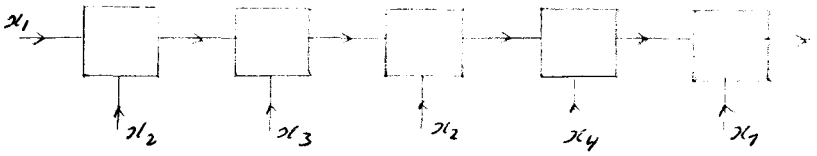


Fig. 2.3

Maitra I962 (lit. I3) ontwikkelde een algorithme, dat aangaf welke funkties gerealiseerd kunnen worden in een niet redundante cascade. Hierbij werd de maitra cel gebruikt. De synthese van een realiseerbare funktie in cascade wordt daar ook in behandeld.
In het college voortgezette schakeltechniek (zie lit I9) wordt een samenvatting gegeven van hetdoor maitra ontwikkelde algorithme.

Sklansky 1963 (lit. 16) toonde aan dat een funktie gerealiseerd door een cascade van maitra cellen, ook gerealiseerd kan worden door een cascade van dezelfde lengte, waarbij 3funktie-cellen worden gebruikt. Deze funkties zijn de AND, OR en EXCLUSIVE OR. Met deze drie funkties kan men alle 16 funkties van twee variabelen realiseren door het al of niet inverteren van de ingangsvariabelen. Ashenhurst 1959(lit.1) toonde aan dat m. b. v. deze 3 funkties alle bestaande ontledingen van funkties in combinaties van 2 argument-funkties te realiseren zijn. Tenslotte wordt nog een tabel gegeven van het aantal realiseerbare funkties door middel van de maitra methode en m.b.v. Stone H.S. en Korenjak A. 1965 (litI8)

n	aantal mogelyke funkties	realiseerbare funkties m.b.v. Maitra theorie	aantal realiseerbare funkties m.b.v. methode Stone en Korenjak. niet redundante case.	redundante case.
1	4	4	4	4
2	16	16	16	16
3	256	80	152	242
4	65.536	520	2368	27.216

Tabel I

Het grote nadeel vanenkelvoudige rail cascades is, dat niet alle logische funkties gerealiseerd kunnen worden (zie tabel). Voordeel is echter dat men een keuze heeft uit een beperkt aantal funkties.

2.I.2 Canonische vorm van een redundante cascade

In deze sektie wordt een algorithme ontwikkeld dat test of de funktie realiseerbaar is m.b.v. een redundante cascade en geeft tevens de synthese van deze cascade.
De onderstaande theorie is in grote lijnen afkomstig van Stone en Korenjak (lit. 18).
Maitra 1962 (lit.13), Sklansky 1963 (lit.16), Levy 1964 (lit.12) Minnick 1964 (lit.14), en Weiss 1969 (lit 20) hebben ook hun bijdrage geleverd aan de synthese en de voorwaarde voor realiseerbaarheid van funkties in cascade vorm. De methode van aanpak is voor iedere schrijver enigszins verschillend. Echter zij gebruikten allen stelling één in onderstaande tekst. Fantauzzi 1968 (lit. 5 en 6) presenteerde de theorie van redundante cascades anders dan men tot dan toe gewend was, echter zijn resultaten komen overeen met die van Stone en Korenjak.

Stone en korenjak (lit.I8) bewijzen dat de meest algemene vorm van een redundante cascade is die schakeling, die in fig. 2.4 wordt aangegeven. De gebruikte celfunkties zijn AND, OR en EXCLUSIVE OR.

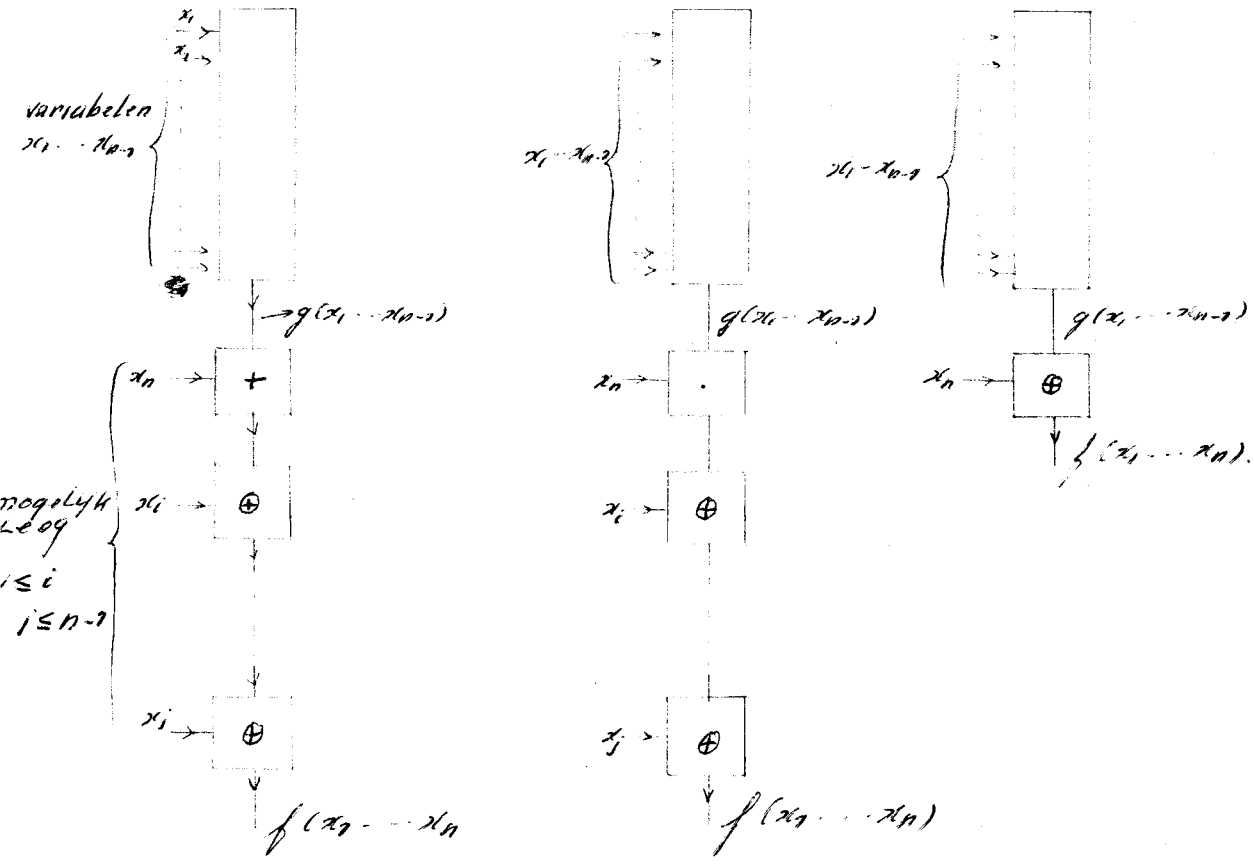


fig. 2.4

Deze cascade bestaat uit een subcascade (redundant of niet redundant), die de funktie $g(x_1, x_2 \cdots x_{n-1})$ produceert, gevolgd door een cel , waarvan de input is aangesloten op x_n . Is deze laatst genoemde cel een AND of OR dan kunnen de daarop eventueel nog volgende cellen alleen maar exclusive-or's zijn, die als input de variabele x_1 t/m x_{n-1} kunnen hebben en geen van deze cellen mag dezelfde input variabele hebben. Is echter de cel waarop x_n wordt aangesloten een EXCLUSIVE-OR, dan zijn er geen volgende cellen meer aanwezig.

2.I.3 De ontwikkeling van een algorithme voor het testen van cascade realiseerbaarheid en synthese van een boolse funktie.

Bij deze methode wordt eerst de onderste cel of cellen van de cascade bepaald. Zijn deze cellen bepaald, dan houdt men een restfunktie over, die aangesloten is op de input van de zojuist bepaalde laatste cel. Is de restfunktie niet in cascade realiseerbaar dan is de gehele funktie niet realiseerbaar. Tegelijkertijd wordt met deze methode ook de synthese van de funktie bepaald.

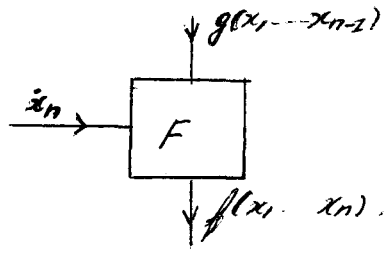
Definitie; Het gewicht van de funktie is het aantal combinaties van zijn argumenten, waarvoor de funktie de waarde één aanneemt. Dit is het aantal enen in de funktie-tabel. De funktie heeft een neutraal gewicht als het aantal eenen gelijk is aan het aantal nullen in de funktie-tabel. Dit komt dus overeen met 2^{n-1} enen. n is het aantal variabelen (argumenten).

Stelling I Laat $f(x_1, x_2 \cdots x_n)$ en $g(x_1 \cdots x_{n-1})$ niet triviale boolse funkties zijn. Dan

1) $f = x_n g$ als $\bar{x}_n f = 0$ en $g = f(x_1, x_2 \cdots x_{n-1}, 1)$.
   $f = \bar{x}_n g$ als $x_n f = 0$ en $g = f(x_1, x_2, \ldots x_{n-1}, 0)$
2) $f = x_n + g$ als $x_n f = 0$ en $g = f(x_1, x_2 \cdots x_{n-1}, 0)$.
   $f = \bar{x}_n + g$ als $\bar{x}_n f = 0$ en $g = f(x_1, x_2 \cdots x_{n-1}, 1)$.
3) $f = x_n \oplus g$ als $g = f(x_1, x_2 \cdots x_{n-1}, 0) = f(x_1, x_2 \cdots x_{n-1}, 1)$

In geval I moet het gewicht van de funktie $f(x_1 \cdots x_n)$ minder dan neutraal zijn, in geval 2 meer dan neutraal en in geval 3 gelijk aan neutraal.

Geval I en 2 ontleedt de funktie op onderstaande manier (zie fig. 2.5).



x_n is of x_n of $\bar{x}_n$
F is of and of or

Fig. 2.5

Geval 3 ontleedt de funktie op de manier zoals die in fig.2.6 is weergegeven.

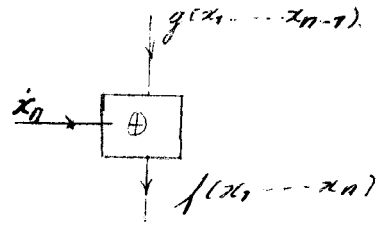


Fig. 2.6

<u>Stelling</u> 2 Laat $f(x_1, x_2 \cdots x_n)$ een n variabele funktie zijn, zodat of $f(x_1 \cdots x_{n-1}, 0)$ of $f(x_1 \cdots x_{n-1}, 1)$ een deelfunktie is van een of meer variabelen. Als p deze deelfunktie is, dan heeft f een ontleding van de vorm $f = p \oplus \dot{x}_n h = \bar{p} \oplus (\bar{x}_n + \bar{h})$ waarbij $h = f(x_1 \cdots x_{n-1}, 0) \oplus f(x_1 \cdots x_{n-1}, 1)$. ——→ . h is een funktie van n-I variablen.

als $p = f(x_1 \cdots x_{n-1}, 0)$ dan $\dot{x}_n = x_n$

" $p = f(x_1 \cdots x_{n-1}, 1)$ " $\dot{x}_n = \bar{x}_n$

Bewijs; ontleedt de funktie in

$f = \bar{x}_n g_0 + x_n g_1$ stel $p = g_0$ wordt als deelfunktie gekozen.

$f = \bar{x}_n p + x_n g_1 = \bar{x}_n p \oplus x_n g_1$ $\quad p \oplus x_n (p \oplus g_1) = \bar{x}_n p \oplus x_n g_1$

$f = p \oplus x_n . (p \oplus g_1) = p \oplus (x_n . h)$ waarbij $h = p \oplus g_1$

$h = f(x_1 \cdots x_{n-1}, 0) \oplus f(x_1 \cdots 1)$

$f = \bar{p} \oplus (\bar{x}_n + \bar{h})$.

Voldoet de funktie niet aan een van de gevallen, die in stelling I zijn genoemd, dan kan stelling 2 worden toegepast. Wil men de funktie in cascade realiseren vlg. stelling 2 dan moet p echter te ontleden zijn in een cascade van cellen, waarbij de celfunkties alleen maar exclusive-or's mogen zijn. (zie fig. 2.7) h
de funktie h dient cascade-realiseerbaar te zijn.

Is de funktie p niet teschrijven als $p = x_i \oplus \ldots \oplus x_j$ dan moet men een andere deelfunktie zoeken waarbij dit wel lukt. Is deze echter niet te vinden, dan is de funktie niet in cascade realiseerbaar.

fig 2.7

Stone en korenjak (lit. I8) hebben bewezen dat de cascade realiseerbaarheid niet gevoelig is voor de volgorde waarin de variabelen van de funktie afgesplitst kunnen worden. Dit resultaat is belangrijk voor het hierna ontwikkelde algorithme.

Allereerst specificeren we enkele tests.

"AND"cel test; Als $\bar{x}_i . f = 0$, splits dan x_i af van de funktie doormiddel van de AND-cel. De rest funktie is $f(x_i = 1)$ De Andcel heeft als inputs x_i en $f(x_i = 1)$. (toepassing stelling I)

Als $x_i f = 0$ dan splits $\bar{x}_i$ af m.b.v. een AND-cel, die als inputs heeft $\bar{x}_i$ en de rest-funktie $f(x_i = 0)$.

"OR"cel test; Als $x_i f = 0$ dan splits x_i af m.b.v. een

OR-cel, die als inputs heeft x_i en de restfunktie $f(x_i=0)$. Als $\bar{x}_i f=0$, dan splits $\bar{x}_i$ af m.b.v. een OR-cel, die als inputs heeft $\bar{x}_i$ en de restfunktie $f(x_i=1)$.

EXCLUSIVE-OR cel test; Als $f(x_i=0)=\bar{f}(x_i=1)$, dan splits de variabele x_i van de funktie af doormiddel van een EXCL.OR-cel met als input x_i en de restfunktie $f(x_i=0)$.

Hoofd algorithme.

Stap 1 De funktie wordt opgegeven met zijn argumenten.
Stap 2 Het gewicht van de funktie wordt berekend. Is het gewicht minder dan neutraal, dan wordt de "AND"cel test toegepast voor ieder argument x_i. Is het gewicht groter dan neutraal, dan wordt de "OR"cel test toegepast voor iedere variabele x_i. Is het gewicht neutraal dan wordt de "EXCL.OR"cel test toegepast voor iedere variabele x_i. (stelling I)
Stap 3 Als de "AND" resp. "OR" cel test lukt, dan wordt de input variabele x_i van de lijst geschrapt. Stap 2 wordt herhaald voor de restfunktie. De restfunktie bevat een variabele minder. De onderste cel is dan tevens bepaald. Als een van de testen in stap 2 faalt, dan gaat men verder met stap 5.
Stap 4 Als bij de "EXCL.OR"cel test de restfunktie een konstante waarde 0 of 1 heeft, dan is de ontleding volledig. Is de restfunktie een konstante 1, dan wordt de laatste "EXCL.OR"cel geschrapt en zijn input variabele x_i wordt geinverteerd en aangesloten op de input van de dan overblijvende bovenste cel. Is de restfunktie een konstante nul dan wordt de laatste cel geschrapt en vervangen door zijn input variabele x_i.
Stap 5 Bereken voor elk argument van de funktie de restfunkties $f(x_i=0)$ en $f(x_i=1)$. Als een van deze restfunkties een neutraal gewicht heeft, dan moet de "EXCL.OR" cel test worden toegepast. Deze test moet dan iteratief worden toegepast totdat de funktie geheel ontleed is. Is de restfunktie te ontleden door steeds de "EXCL.OR" cel test toe te passen, dan wordt deze restfunktie deelfunktie genoemd. (i.v.m. stelling 2) Nu wordt verwezen naar stap 6. Bestaat er geen restfunktie, die bovengenoemde eigenschap van een deelfunktie heeft dan is de oorspronkelijke funktie niet in cascade realiseerbaar.

Stap 6 Splits, x_i en de redundante EXCL.OR-cellen waaruit p (is bovengenoemde deelfunktie) bestaat, van de funktie af, m.b.v. de relatie $f = p \oplus \dot{x}_i g$.
Ter ondersteuning laat fig.2.8 zien hoe dit laatste geschiedt.

$g = f(x_i = 0) \oplus f(x_i = 1)$
en $\dot{x}_i = x_i$ als $f(x_i = 0)$ as de deelfunktie p
$\dot{x}_i = \bar{x}_i$ " $f(x_i = 1)$ " " "
$p = x_\nu \oplus \cdots \oplus x_j$ $\quad 1 \leq \nu$, $j \leq n-1$

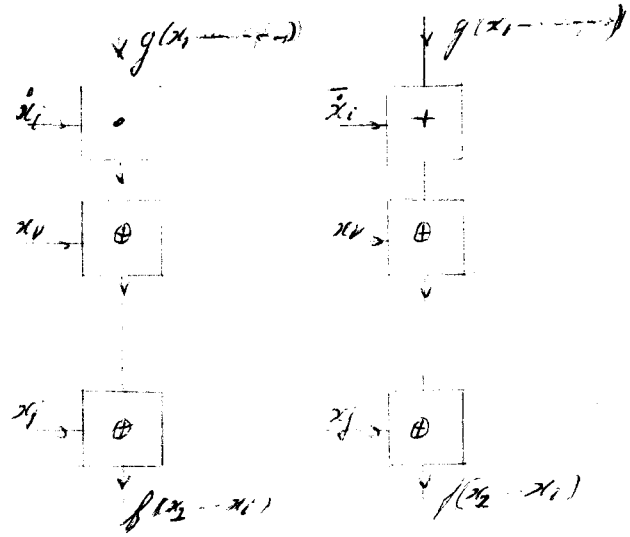


Fig 2.8

Voor de realisatie van g of $\bar{g}$ wordt verwezen naar stap3.

2.1.4 Bepaling van de bovengrens van het aantal benodigde cellen.

Het maximaal aantal cellen waaruit een cascade kan bestaan wordt C(n) genoemd. n is het aantal variabelen. De toename van C(n) voor zeer grote n wordt vaak growth rate genoemd
De meest algemene vorm van een redundante cascade (zie fig. 2.4.) bestaat uit 2 subcascades. De bovenste cascade kan uit maximaal C(n-1) cellen bestaan. De onderste bestaat uit maximaal n cellen. Dus $C(n) \leq C(n-1)+n$ waarbij $C(2)=1$.
Daaruit volgt; $C(n) = 1 + \sum_{i=3}^{n} i = (n^2 + n - 4)/2$
Growth rate is $(n^2+n)/2$.

2.1.5 Het aantal realiseerbare funkties.

Het aantal realiseerbare funktie is volgens stone en Korenjak (lit.18) gelijk aan T(n), waarbij

$T(n) < 3n!\,2^n\,2^{(n+1)(n-1)/2}$
Het totaal aantal funkties van n variabelen is 2^{2^n}.
De verhouding $T(n)/2^{2^n}$ wordt zeer klein bij toenemende n.
In tabel 2 is het werkelijke aantal realiseerbare funkties van 2, 3, 4 variabelen getabelleerd

Tabel 2

	niet redundante cascade	redundante cascade	totaal aantal funkties
n=2	16	16	16
n=3	152	242	256
n=4	2368	27.216	65.536

2.2 Multi-rail cascades.

Een uitbreiding van enkel-rail cascades is een verhoging van de externe inputs en slagaderlijke verbindingen. (fig. 2.IO)

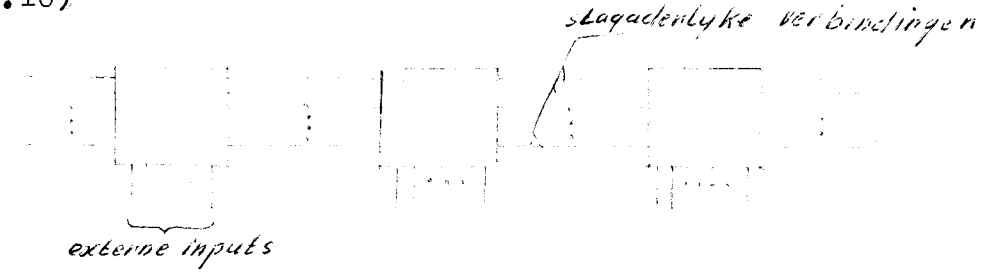


Fig. 2.IO

Stel iedere cel heeft p slagaderlijke verbindingen en q externe inputs. Dus p+q=m is het totaal aantal inputs en p het totaal aantal uotputs van iedere cel.
Het aantal mogelijke celfunkties is 2^{p2^m} .
Voor p=2 en q=I (twee rail-cascade) is het aantal mogelijke funkties $2^{2\cdot 2^3} = 2^{16} = 65.536$. Men begrijpt nu dat de volgende problemen een rol gaan spelen bij de sythese van multi-rail cascades.

a/ Het aantal benodigde type celfunkties voor realisering van een universele cascade.
b/ Lengte van de cascade. (Growth rate)
c/ De komplexiteit van de cel.

Lendaris en Stanley I963 (lit.I0) hebben cascades onderzocht die één enkele slagaderlijke verbinding hebben, maar waarbij het aantal externe inputs tot meer dan één worden uitgebreid. (zie fig. 2.II)

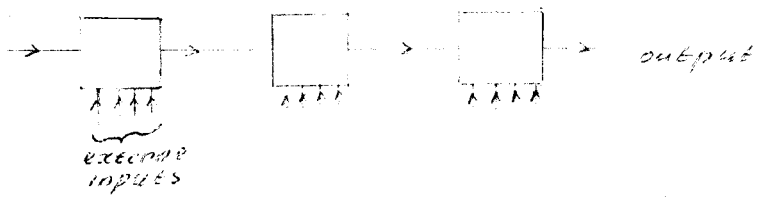


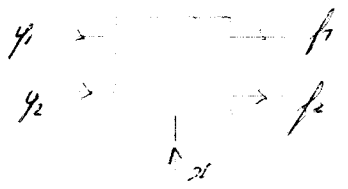
Fig. 2.II

Deze methode heeft hetzelfde nadeel als een enkele rail-cascade nl.; de cascade is logisch niet kompleet d.w.z. niet alle mogelijke funkties van nvariabelen zijn met deze cascade realiseerbaar.

Een andere uitbreiding van de enkel rail-cascade is:
Het laten toenemen van het aantal slagaderlijke verbindingen tussen de cellen onderling. Short I965 (lit.I7) heeft de twee rail-cascade onderzocht. Deze cascade is logisch kompleet. Alle willekeurige funktie van n variabelen zijn te realiseren.

Onderstaande tekst geeft het bewijs, dat de twee rail-cascade logisch kompleet is.

De cel en celfunkties, die worden toegepast, zijn weergegeven in fig. 2.I2.



$f_1 = x y_2$ of $\bar{x} y_1$ of ?
$f_2 = x y_1 + y_2$ of $\bar{x} y_1 + y_2$ of y_2
(er worden dus 4 verschillende cel types toegepast)

Fig 2.12.

Een voorbeeld van een universele cascade zie fig. 2.13

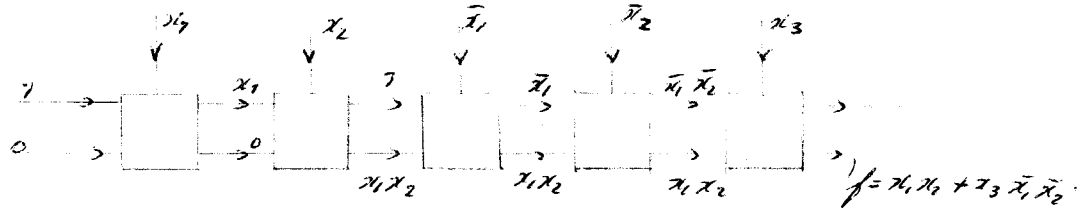


Fig 2.13

Iedere willekeurige produkt term (of priemprodukt) kan op de bovenste rail gevormd worden. Tegelijkertijd kan deze produkt term worden toegevoegd aan de som van produkt termen op de onderste rail. De maximale lengte van deze cascade is $n2^n$. (voor iedere minterm n cellen nodig en er zijn 2^n mintermen)
Is echter een komplementaire output aanwezig dan is de maximale lengte $n2^{n-1}$.

Short (lit.17) behandelde een cascade met een growth rate van $1/8 n2^n$. Deze cascade is logisch kompleet. Het aantal toegepaste cel types is veel groter dan in het vorige geval. De toegepaste cel types zijn:

a) $f_1 = x$ $\quad f_2 = y_2 \oplus x y_1$

b) $f_1 = y_1$ $\quad f_2 = x y_2$.

c) $f_1 = x y_1$ $\quad f_2 = y_2$.

d) $f_1 = h(x,y_1)$ $f_2 = y_2$

e) $f_1 = h_1(x,y_1)$ $f_2 = y_2 \oplus h_2(x,y_1)$.

a,b enc zijn eenvoudig van struktuur.
d en e vertegenwoordigen meerdere celtypes
d bestaat uit I6 types.
e bestaat uit 256 types.

a,c,en d kan men beschouwen als vereenvoudigde versies van celtype e. Er zijn dan uiteindelijk twee soorten basis-types nl. b en e.
Bij de synthese van deze cascade werd gebruik gemaakt van de Reed ontwikkeling (exclusive-or ontwikkeling). Zie lit. I7.

Yoeli I965 (lit.2I) heeft,m.b.v. de groep theorie, graph theorie en de theorie van homorphisme of direct graphs, de multi rail-cascade behandeld.

fig, 2.I4 multi rail-cascade

De resultaten van zijn theorie konden op de twee rail-cascade worden toegepast. De boolse funkties die daarmee gerealiseerd kunnen worden, moesten echter aan bepaalde beperkende voorwaardens voldoen. [illegible]
In I966 behandelde Yoeli en Turner een krachtigere theorie voor twee rail cascades die;
a/ voldeed aan alle willekeurige funkties. En alle resultaten van voorgaande theorieën omvatte.
b/ En aantoonde dat de twee rail-cascade funktioneel kompleet is. D.w.z. Dat met deze cascade ieder willekeurig paar van boolse funkties van willekeurig aantal variabelen te realiseren is.(Bij Short werd slechts een van de twee outputs van de cascade benut)
c/ Een bovengrens van het aantal cellen aangaf nl.

YT(n)= $3\,2^{n-1}-n-8$
Dus een growth tate van $3\,2^{n-1}$!!!!
YT(n) is onafhankelijk van het aantal slagaderlijke verbindingen, Hier is een netwerk gecreëerd met een growth rate anders dan $c\,n\,2^n$.
Daar de cascade funktioneel kompleet is, kan men door het toevoegen van een cel meer aan de cascade een willekeurige funktie van n+I variabelen creëren. Zie fig. 2.I5.

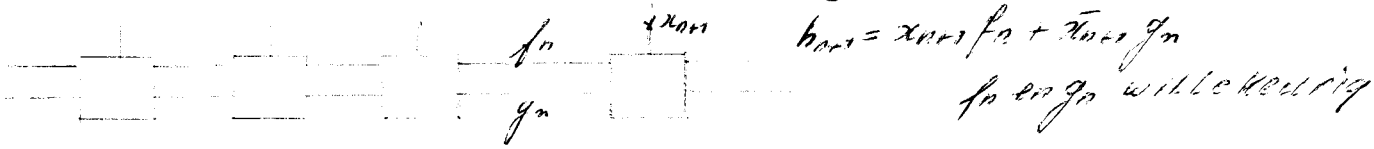


Fig. 2.I5

Yoeli-Turner maakt bij de ontleding van de funktie in een twee rail cascade gebruik van algemene funktiecellen en permutatie cellen.
De algemenefunktie cellen zijn nodig voor een "Ashenhurst ontleding" I959 (lit.I) van de twee rail funktie. Deze cellen moeten in staat zijn een willekeurig paar van 3 x variabele funkties te produceren. (Keuzen uit 65.536 mogelijkheden). De lage growth rate gaat hier ten koste van de cel komplexiteit.
De permutatie cellen komen qua komplexiteit overeen met de cellen van Short (lit.I7) De permutatie cellen bestaan uit 8 individuele celtypes.
Echter is het ook mogelijk universele cascades te produceren, waarbij alleen permutatiecellen worden toegepast. (lit. 22en) De rail inputs van de eerste cel moeten als vaste parameters van de cascade worden gezien. Deze cascade is ook funktioneel kompleet.
De bovengrens van het aantal cellen in deze cascade is;
P(n)= $3.2^{n-1}-2$
Growth rate 3.2^{n-1}
Deze cascade heeft dezelfde growth rate als de vorige cascade. Hier wordt echter een veel minder komplexe cel gebruikt. (Keuze uit 8 types)
Een beknopte beschrijving van het bovengenoemde werk van Yoeli-Turner wordt gegeven in lit. I5 blz3I5-367.

Vergelijking van het maximale aantal cellen van de cascades van Short en Yoeli-turner.

aantal var. n	3	4	5	6	7	8	9	10
Short cascade S(n)	7	2	10	30	78	190	508^{16}	1210^{16}
YT(n-1)+1	7	2	13	36	83	178	369	752
P(n-1)+1	5	11	23	47	95	191	383	767

$YT(n) + n + 6 = P(n)$

Bij n≤7 Bevat de cascade van Short minder cellen. Bij n≥8 bevatten de cascades van Yoeli-Turner minder cellen.

2.3 Uni- en bilaterale iteratieve netwerken.

2.3.1 Introduktie.

Een netwerk wordt iteratief genoemd als alle cellen uit dezelfde circuits bestaan en als ze op dezelfde wijze worden door verbonden.
Een voorbeeld van een unilateraal netwerk laat fig. 2.16 zien.

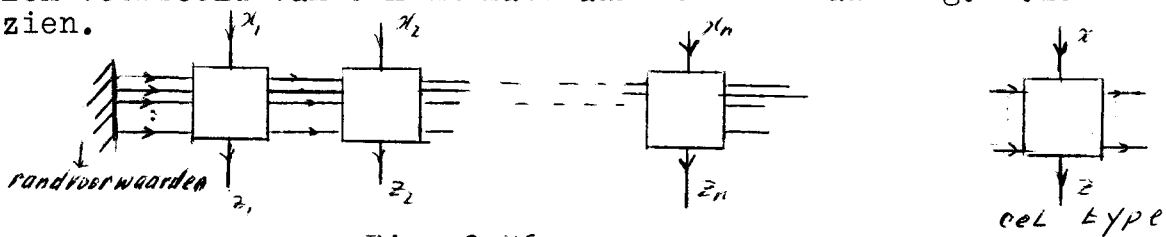


Fig. 2.16

Iedere cel heeft; a/ een primaire in- en output nl. x en z
b/ een set van rechterlaterale inputs.
Dit zijn de rechter inputs waarmee de cellen onderling zijn doorverbonden.
De laterale inputs van de meest linkse cel worden ~~rechter~~ linker randvoorwaarden genoemd. De data doorloopt de cel in een richting.
Fig. 2.17 laat een voorbeeld van een bilateraal netwerk zien.

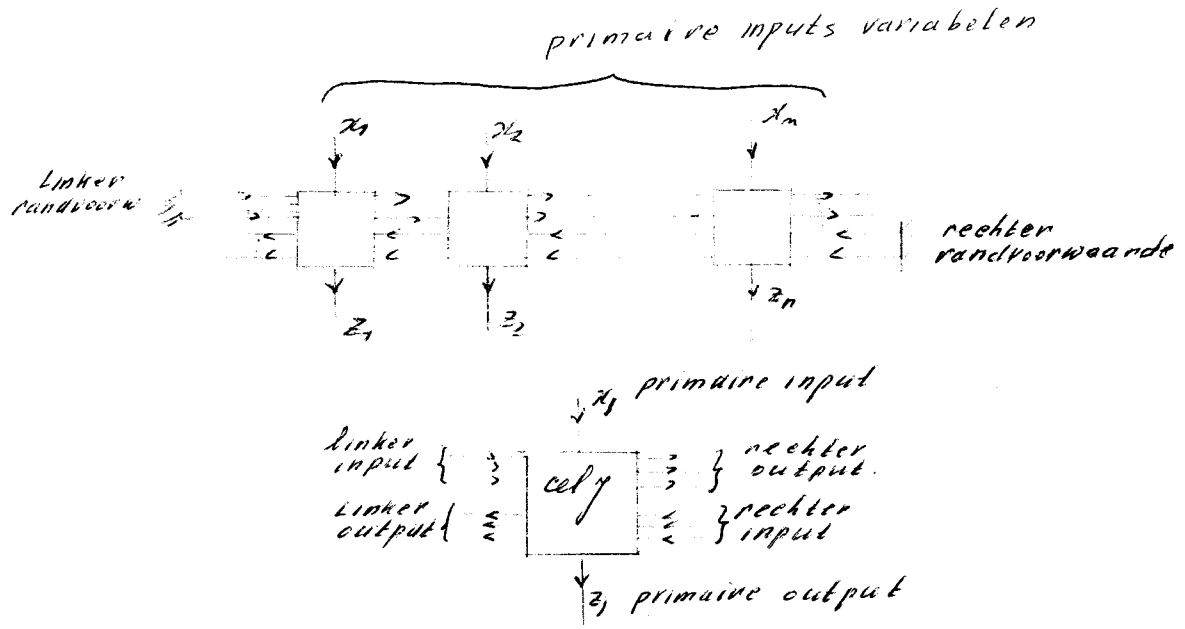
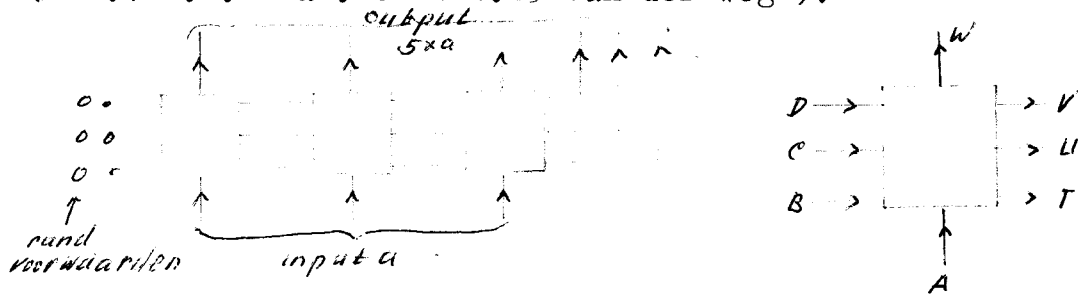


Fig. 2.17

Deze cellen werken in heen en terugwaartse richting.
De cellen kunnen ~~xixxxx~~ combinatorische circuits of sequentiële circuits bevatten.
Hennie 1959 (lit.7) behandelt netwerken waarvan de cellen uit combinatorische circuits bestaan. Deze cellen bezitten dan geen geheugen. Dit betekent voor unilaterale netwerken dat bij iedere kombinatie van de input variabelen een evenwichtstoestand te bereiken is, waarbij een bepaalde kombinatie van output variabelen hoort.
Een elementaire behandeling van unilaterale iteratieve netwerken wordt gegeven door Caldwell (lit.2) blz.419. Hij brengt deze netwerken alleen in verband met de kontakt theorie. Mc. Cluskey 1958 geeft een meer algemene behandeling van iterative netwerken. Verder zal in verband hiermee nog genoemd worden; Keister (lit.9).
Deze unilaterale netwerken kunnen een set van boolse funkties realiseren.
Een voorbeeld van een unilateraal netwerk, zie fig.2.18 (Lit.4 K.J. Dean. en lit.19 van der Weg).



parallel vermenigvuldiger
input a wordt met 5 vermenigvuldigd

$T = ACD + AB$
$U = A\bar{B}\bar{C} + A\bar{B}\bar{D} + \bar{A}B\bar{C}$
$V = C\bar{D} + \bar{A}C + A\bar{C}D$
$W = A\bar{D} + \bar{A}D$

Fig. 2.18.

Ook een parallel deler wordt door K.J. Dean (lit. 4) m.b.v. een unilateraal netwerk behandeld. In sektie 2.3.2 wordt een korte beschrijving gegeven van bilaterale netwerken.
In het boek van Hennie (lit. 8 1968) worden een-dimensionale sequentiële iterative netwerken beschreven. De cellen bestaan uit sequentiële circuits, die worden gesynchroniseerd door dezelfde klok. Deze netwerken zijn in staat om sequentiële transformaties uit te voeren. De belangrijkste toepassing van deze netwerken in grotere systemen, is het uitvoeren van rekenkundige operaties. (twee dimensionaal) Ook kunnen boolse funkties daarmee gerealiseerd worden.

2.3.2 Bilateraal iteratief netwerk.

Als aangenomen wordt dat het bilateraal netwerk precies één evenwichtstoestand heeft bij ieder combinatie van de input variabelen, dan kan dit netwerk volgens Hennie (lit. 7 en8) vervangen worden door twee unilaterale netwerken die in tegengestelde richting werken. Zie fig. 2.19.
Beide unilaterale netwerken hebben bij iedere input combinatie een evenwichtstoestand. Het subnetwerk, dat de output z produceert, bezit geen geheugen. Daardoor heeft het bilateraal netwerk van fig. 2.19 bij iedere combinatie van input variabelen een evenwichtstoestand die binnen eindige tijd bereikt wordt.

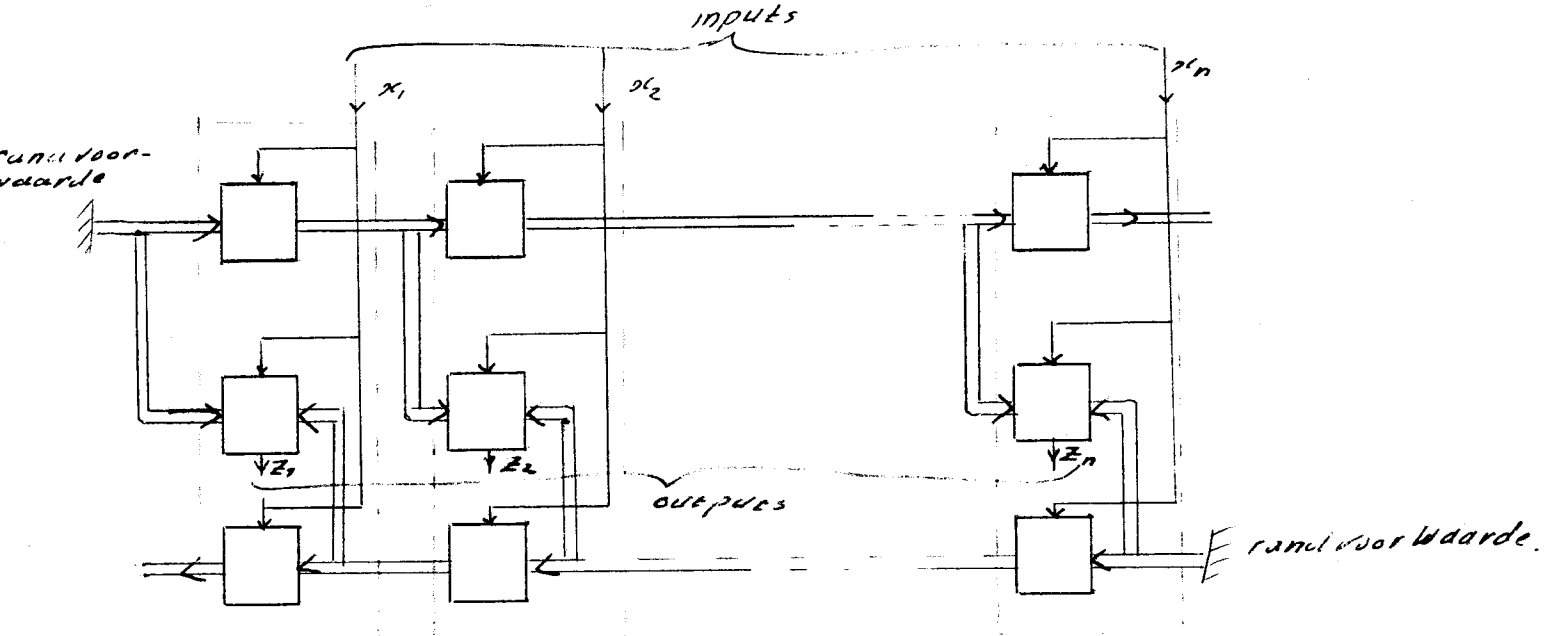


Fig. 2.19

Literatuur opgave.

I Aschenhurst R. (I959). The decomposition of switching funktions. Am. Comput. Lab. Havard. Univ., 29 No 30 Havard. Univ. Press. Cambridge, Massachusetts.

2 Caldwell S.H. Switching circuits and logical design. (I960) Wiley, New York.

3 MC. Cluskey E.J. Iterative Combinational Switching Networks-General Design Considerations. IRE Trans. Elec. Comput. EC-7 285-29I (Dec I958)

4 Dean K.J. Logical circuits for use in iterative array. Electronic Letters 4 maart I968.

5 Fantauzzi I968 A semigroup theory fot the Maitra cascade. In Proceedings IFIP Congress. I Supplement Book I, pp. 54-6I.

6 Fantauzzi Application Karnaugh maps to maitra cascades In AFIPS Conf. Proc. 32 SJCC, Thomson, 29I-296.

7 Hennie F.C. Analysis of bilateral iterative networks. IRE Trans. on circuit theory I959 blz 35 maart.

8 Hennie F.C. I968 Finite State Models for logical Machines Wiley New York.

9 Keister W.S.A. Ritchie and S. Washburn. The design of Switching Circuits,(I95I)D. vam Nostrand, New York.

I0 Lendaris G.G., and Stanley G.L. (I963) On the structure - Dependent properties of adaptive logic networks. GM Defense Res. Labs.,

II Lendaris G.G. Mathematics and Evaluation studies Dept., Santa Barbara, California, TR 63-2I9, July

I2 Levy S.Y. Winder R.O. and Mott T.H.J.R. A note on tributary switching networks. IEEE Trans. Elec. Comput. EC-I3 I48-I5I

I3 Maitra K.K. I962 Cascaded switching networks of two inputs flexible cells. IRE Trans. Elec. Comput. EC-II I36-I43.

I4 Minnick R.C. I964 Cutpoint cellular logic. IEEE Trans. Elec. Comput. EC-I3 685-698.

I5 Mukhopadhyay A. Recent Developments in switching theory . Academic Press. INC., New York.

I6 Sklansky J. I963 General synthesis of tributary switching networks. IEEE Trans. El. Comput EC-I2 464-

I7 Short R.A. I965 Two rail cellular arrays. In AFIPS Conf. Proc. 27, Pt. I, Wasington D.C. Spartan 355-69

I8 Stone H.S. and Korenjak A. I965 Canonical form and synthesis of cellular cascades. IEEE Trans. Electr. Comput EC-I4 852-862.

I9 Weg v.d. H. Voortgezette schakeltechniek. Aantekeningen voorjaars college x THE.

20 Weiss I969 The characterization and properties of cascade realizable switching functions. IEE+E Trans. Comput C-I8 624-633.

2I Yoeli M. I965 A group theoretic approach to two rail cascades. IEEE Trans. Electron. Comput. EC-I4 8I5-822.

22 Yoeli M. and Turner J.B. I966 Decompositions of group functions with applications to two rail cascades Sci. Rep 2, Contract AF I9(628)-5828, SRI Project 5876, AFCRL 66-472. Stanford Res. Inst., Menlo Park, California.

Hoofdstuk 3

Twee dimensionale cellulaire netwerken.

Samenvatting.

Sektie 3.1 behandelt cellulaire netwerken, waarvan de funkties van de cellen vastliggen. De celfunkties kunnen zijn: NOR's, NAND's en drempel elementen.
In sektie 3.2 en 3.3 zijn de celfunkties en hun onderlinge verbindingen variabel. Voorbeelden zijn: Cutpoint netwerken, Cobweb netwerken, q-funktie netwerken en geprogrammeerde netwerken.

Introduktie.

Fig. 3.1 laat een twee dimensionaal cellulair netwerk zien. Elke cel en zijn verbindingen met de buren zijn gelijk.
De celfunktie en de verbindingstruktuur tussen de cellen onderling kunnen met behulp van celparameters ingesteld worden.

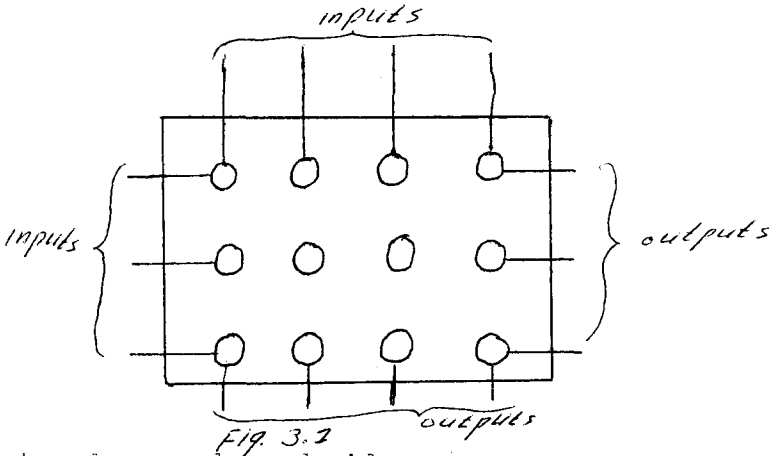


Fig. 3.1

Men kan 3 soorten cellulaire netwerken onderscheiden:
a/ Netwerken met vaste celfunkties en variabele verbindingstruktuur. Sektie 3.1.
b/ Netwerken met vaste verbindingstruktuur en variabele celfunkties. Sektie 3.2.
c/ netwerken met variabele verbindingstruktuur en variabele celfunkties. Sektie 3.2 (Cobweb netwerken, Diamond 2 netwerken, 2 way en 4 way netwerken en toepassing van de permutatie cel).
Door toepassing van de celparameters heeft men één type cel nodig om het netwerk te realiseren. (Geschikt voor LSI). De cel wordt echter komplexer.

De celparameters kunnen tijdens het laatste gedeelte van het fabrikage proces ingesteld worden. Dit gebeurt met behulp van"cutpoints". Deze "cutpoints bestaan uit kleine fotocellen of schakelaars of punten,waarbij de geleiders mechanisch verbroken kunnen worden.
De celparameters kunnen ook elektrisch ingesteld worden. De celparameters zijn dan als flipfloppen uitgevoerd. Deze parameters kunnen elektrisch,van uit de randen van het netwerk,ingesteld worden.
De voordelen van deze programmeerbare netwerken zijn;

a/ Netwerk is flexibeler. Hierdoor wordt het aantal netwerken in een systeem gereduceerd.

b/ Betere fout korrectie. Defekte cellen in het netwerk kunnen opgeheven of omzeild worden,doordat het netwerk op meer dan een manier geprogrammeerd kan worden voor het realiseren van dezelfde funktie. Een gehele rij of kolom waarin de defekte cel zit kan uitgeschakeld worden.

c/ De mogelijkheid om vele kleinere subnetwerken met elkaar door te verbinden tot een groter netwerk. Produceren de de subnetwerken funktics van n variabelen, dan is men instaat om funktics van meer dan n variabelen te realiseren, door meerdere subnetwerken toete passen.

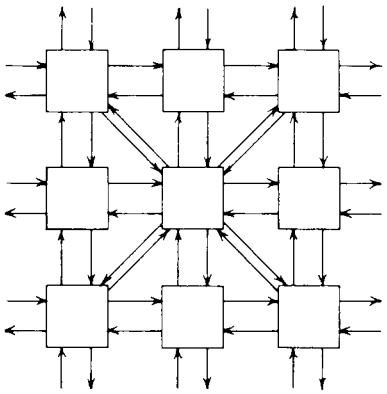
d/ Grotere logische prestatie per chip. Het netwerk kan in een systeem zeer veel verschillende funkties uitvoeren.

3.I Twee dimensionale netwerken met vaste celfunkties.

Bij deze netwerken zijn de cellen eenvoudig van struktuur. de celparameters zorgen voor het verbreken of intakt blijven van de verschillende verbindingen tussen de cellen onderling. De celfunktie ligt vast.

3.1.A NAND's en NOR's

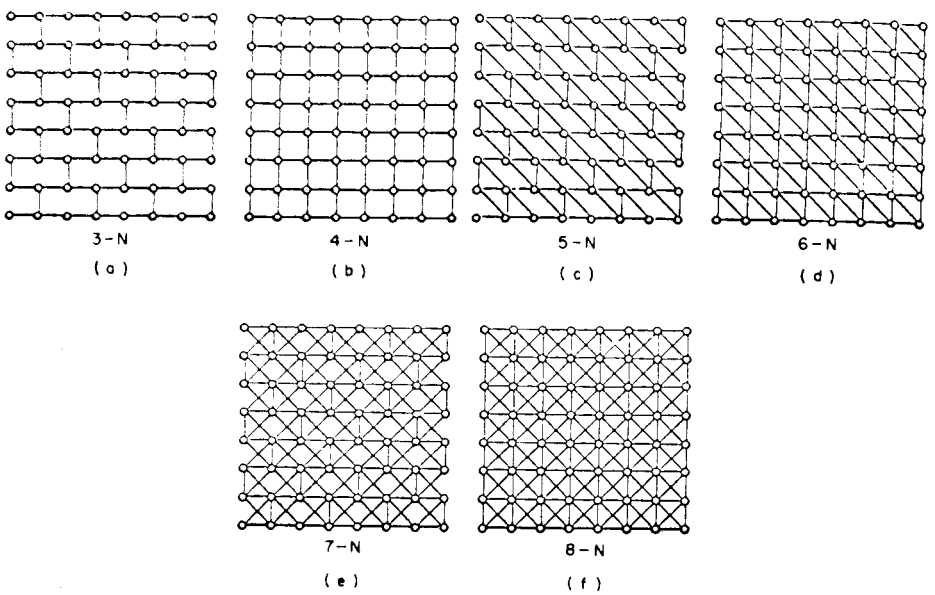
Brooking heeft netwerken bestudeerd opgebouwd uit cellen, die de NOR als celfunktie hebben. (lit 2) In fig. 3.2 wordt weergegeven hoe de cel met zijn 8 buren wordt doorverbonden.



The 8-neighbor NOR array.

Fig. 3.2

Iedere subset van 8 inputs kan worden geselekteerd door een set van 8 parameters. Door een juiste keuze van deze parameters zijn zeer veel verbindingstrukturen mogelijk. (zie fig. 3.3)

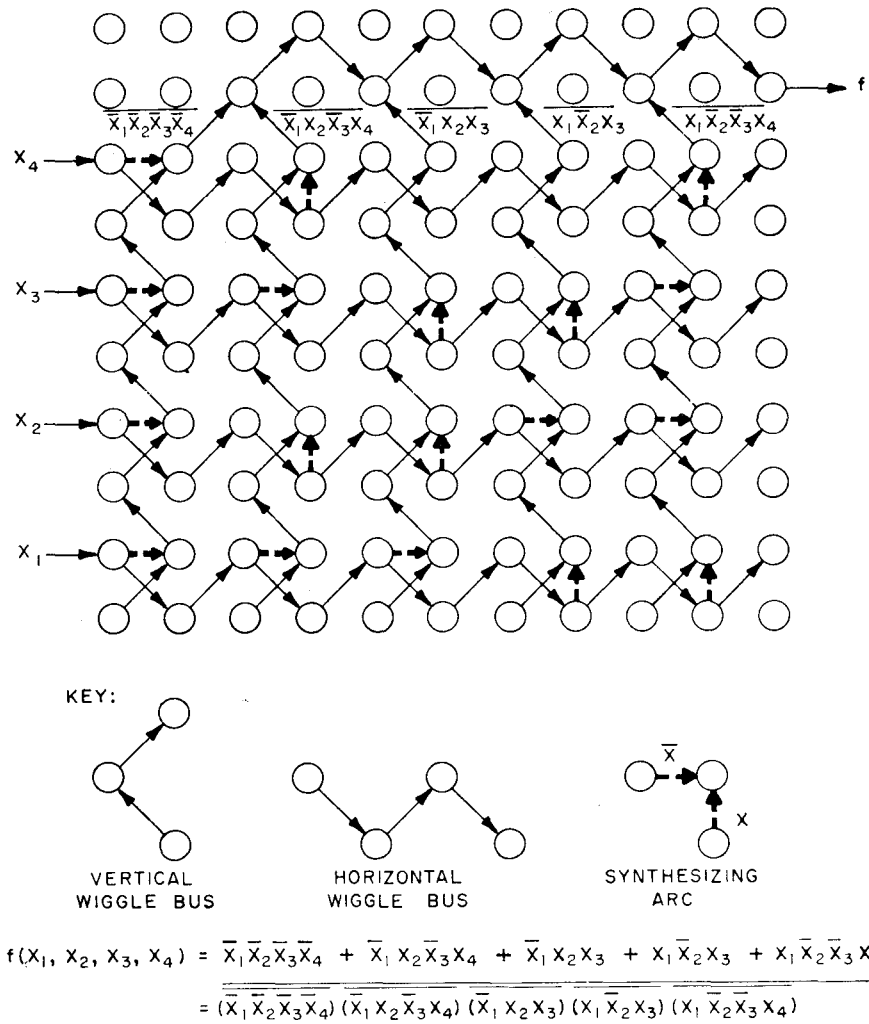


Several possible eight-neighbor interconnection graphs (From [9], with permission)

Fig. 3.3

De lijnen(in fig 3.3)tussen de cellen onderling stellen een twee richtingsinformatie lijnen voor.
Brooking heeft vele types van deze subgraphs bestudeerd, en vond dat deze subgraphs hetzelfde gedrag konden vertonen als die van een sequentiële machine (binaire teller enz.).
King (lit8) toonde aan dat iedere willekeurige autonome sequentiële machine kan worden opgebouwd uit deze cellen. Echter op een onefficiënte manier. Hij representeerde de toestand van alle cellen door een toestandsvektor S_t . De onderlinge verbindingen tussen de cellen door de matrix $[A]$. De overgang van iedere vorige toestandsvektor S_t in de volgende toestandsvektor S_{t+1} wordt voorgesteld door $S_{t+1} = [A]\, S_t$.
Hieruit ontstaat een toestandsdiagram van een autonome sequentiële machine. De noodzakelijke en voldoende voorwaarden voor de matrix A, om iedere klasse toestandsdiagrammen te genereren, zijn door King bepaald.

Spandorfer en Murphy (lit I7 I963) ontwikkelden logische ontwerp methodes voor de realisatie van kombinatorische funkties, waarbij de NAND celfunktie wordt toegepast. Het cellulaire netwerk bestaat uit vertikale en horizontale op en neer gaande bussen. Deze bussen dienen om de input variabelen zowel in vertikale als in horizontale richting te verplaatsen. (zie fig 3.4)
De cellen hebben maximaal 2 inputs en 2 outputs. De funktie van het netwerk wordt bepaald door het al of niet plaatsen van bogen (synthesizing arcs), zoals in fig. 3.4 wordt weergegeven.
De funktie van n variabelen wordt eerst omgevormd in een som van logische produkten. Het aantal logische produkten wordt geminimaliseerd volgens de conventionele methodes.
Daarna wordt het aantal literals geminimaliseerd.
Neemt men een worst case funktie (d.i. een funktie met een maximum aantal priemprodukten b.v. een wisselende symetrische funktie), dus een funktie met 2^{n-1} termen, dan heeft men $2(n+1)(2^n+2)$ cellen nodig. Voor een toenemend aantal variabelen n, nadert het aantal cellen tot $n\,2^{n+1}$.
(Growth rate).
Minnick 1968 (lit II) ontwikkelde eenzelfde ontwerp-methode met de NOR als celfunktie.



$$f(X_1, X_2, X_3, X_4) = \bar{X}_1\bar{X}_2\bar{X}_3\bar{X}_4 + \bar{X}_1X_2\bar{X}_3X_4 + \bar{X}_1X_2X_3 + X_1\bar{X}_2X_3 + X_1\bar{X}_2\bar{X}_3X_4$$

$$= \overline{(\bar{X}_1\bar{X}_2\bar{X}_3\bar{X}_4)(\bar{X}_1X_2\bar{X}_3X_4)(\bar{X}_1X_2X_3)(X_1\bar{X}_2X_3)(X_1\bar{X}_2\bar{X}_3X_4)}$$

Spandorfer's eight-neighbor-array synthesis method (From [105], with permission)

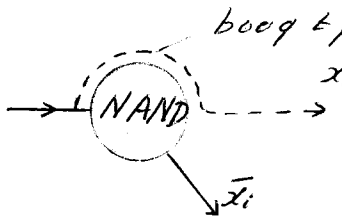
Fig. 3.4

Een ander type netwerk, ook ontwikkeld door Spandorfer en Murphy, gebruikt cellen met maximaal 3 inputs en 2 outputs. De celfunktie is weer een NAND.
Door middel van celparameters kan iedere subset van inputs en outputs gerealiseerd worden. Dit netwerk wordt Diamond I netwerk genoemd. Geinteresseerde lezers worden verwezen naar het artikel van Minnick 1967 (lit 10) blz 216 , waar dit netwerk wordt behandeld.

De growth rate van het Diamond 1 netwerk is $n2^n$. Dit is gunstiger dan het netwerk met vertikale en horizontale bussen. Het Diamond 2 netwerk (ook ontwikkeld door Spandorfer en Murphy) gebruikt een cel met maximaal 3 inputs en 3 outputs. De celfunktie is ook hier weer en NAND. Er is een extra parameter aan de cel toegevoegd, die de celfunktie enigszins verandert. (zie fig. 3.5).

Diamond II

ieder subset van in- en out-puts kan m.b.v. cel-parameters uitgeselecteerd worden.



boog type II

Door extra parameter boog type II mogelijk

Fig. 3.5

De cel is komplexer geworden. De growth rate is $3/4.2^n.n$.
Voor gedetailleerde informatie over dit netwerk zie lit. 10.

3.1.B Drempelelementen

Er bestaan cell. netwerken die opgebouwd zijn uit drempel elementen. Deze netwerken hebben een vaste celfunktie en een "semi"-vaste verbindingstruktuur. In onderstaande tekst wordt een methode behandeld, om een willekeurige funktie te realiseren m.b.v. drempelelementen (majority gates) van 3 variabelen.

De logische funktie van een drempel element ziet er als volgt uit: $f(a,b,c) = a.b + b.c + c.a = a \# b \# c$

Als $b.c = 0$ dan $a \# b \# c = a(b + c)$
Als $b + c = 1$ dan $a \# b \# c = a + b.c$

Beschouw nu de volgende twee gevallen:

A/ Als $F = D.E$ en F, D en E zijn funktie van n variabelen.

Dan $E(x_1 \cdots x_n) = x_i.E(x_1 \cdots 1 \cdots x_n) + \bar{x}_i.E(x_1 \cdots 0 \cdots x_n)$.

Stel $G = x_i.E(x_1 \cdots 1 \cdots x_n)$ en $H = \bar{x}_i.E(x_1 \cdots 0 \cdots x_n)$.

Dan $E = G + H$ en $G.H = 0$.

Daaruit volgt $F = D.(G + H) = D \# G \# H$.

B/ Als $F = K + L$ en F,K en L zijn funkties van n variabelen.

Dan $L(x_1 \cdots x_n) = \{x_i + L(x_1 \cdots 0 \cdots x_n)\} \cdot \{\bar{x}_i + L(x_1 \cdots 1 \cdots x_n)\}$.

Stel $M = x_i + L(x_1 \cdots 0 \cdots x_n)$ en $N = \bar{x}_i + L(x_1 \cdots 1 \cdots x_n)$

Dan $F = K + M.N$ en $M + N = 1$ $F = K \# M \# N$.

De ontleding van een funktie in drempel elementen geschiedt als volgt:

$$F(x,y,\ldots) = x.F_1 + \bar{x}.F_0 \qquad \begin{matrix} F_1 = F(1,y,z\cdots) \\ F_0 = F(0,y,z\cdots) \end{matrix}$$

Pas geval B toe

$$\left.\begin{matrix} \text{stel} \quad \bar{x}.F_0 = \bar{x}.F_0 .1 \\ 1 + \bar{x}.F_0 = 1 \end{matrix}\right\} \Longrightarrow F = 1 \# x.F_1 \# \bar{x}.F_0$$

$x.F_1 = x.(y.F_{11} + \bar{y}.F_{10})$ pas hierop geval A toe en

$x.F_1 = x \# y.F_{11} \# \bar{y}.F_{10}$

$\bar{x}.F_0 = x.(y.F_{01} + \bar{y}.F_{00})$ pas ook hierop geval A toe en

$\bar{x}.F_0 = x \# y.F_{01} \# \bar{y}.F_{00}$

Door geval A herhaald toe te passen krijgt men de volgende ontleding van $F(x,y,z)$

$F = 1 \# x.F_1 \# \bar{x}.F_0$

$x.F_1 = x \# y.F_{11} \# \bar{y}.F_{10}$

$y.F_{11} = y \# z.F_{111} \# \bar{z}.F_{110}$

-
-
-
-

enz.

$\bar{x}.F_0 = x \# y.F_{01} \# \bar{y}.F_{00}$

$y.F_{01} = y \# z.F_{011} \# \bar{z}.F_{010}$

-
-
-
-
-

enz.

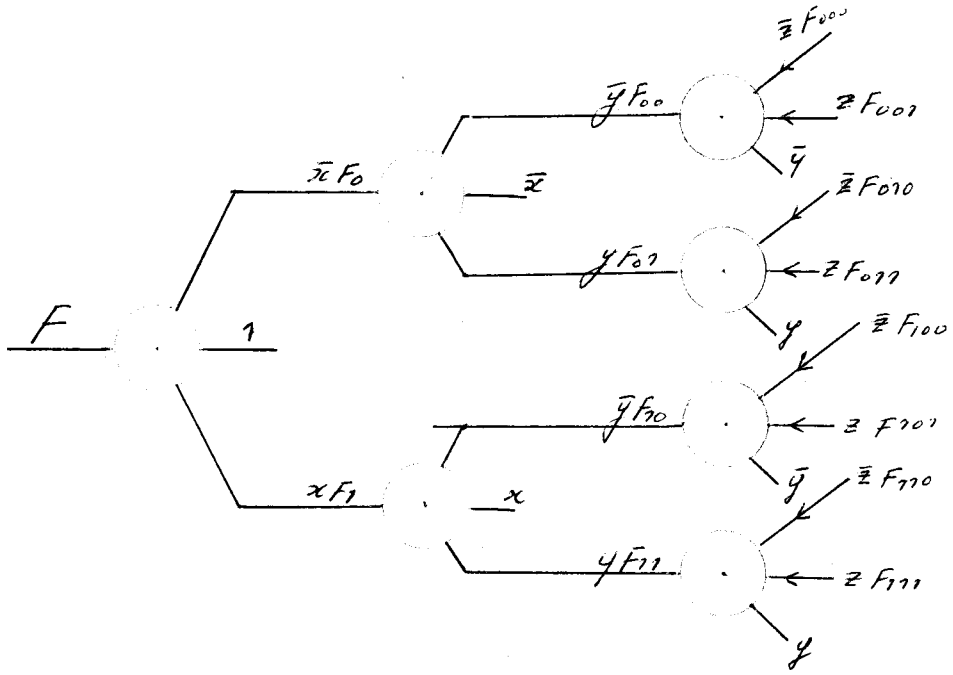


Fig. 3.6

Fig. 3.6. laat de realisatie zien van de funktie F m.b.v. drempel elementen.
Fig. 3.7 geeft dezelfde funktie weer. De boom is hier ingebed in een rechthoekig netwerk bestaande uit identieke cellen. De inputs van de cellen worden óf aan een van de variabelen x,y en z gelegd óf aan een konstante waarde nul of één gelegd. Het aantal cellen is $3n.2^n + 1 - 3.2^{n-1} - 2n$, growth rate is $3n2^n$. De variabelen worden hier niet via de randen van het netwerk toegevoerd. Ze zijn enigszins willekeurig over het netwerk verdeeld. De celparameters zijn verdwenen. Daar voor in de plaats komt nu het aan de waarde nul of een leggen van de diverse inputs.
F. Miyata (lit. 12 a) gaat dieper in op de ontleding van willekeurige funkties in drempel elementen.

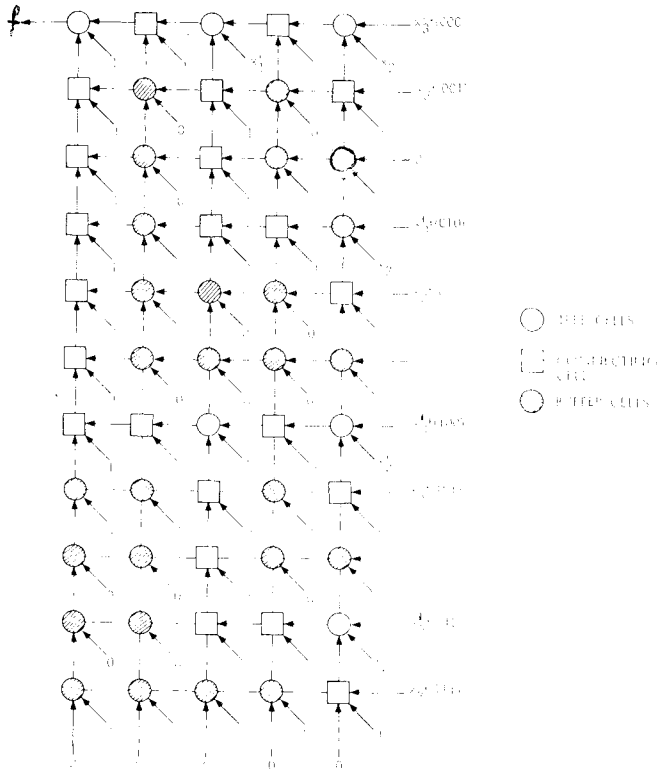
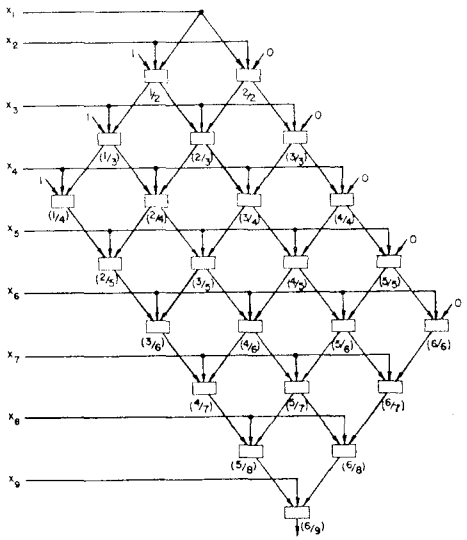


FIG. 19. Tree-based majority-gate cellular array: rectangular form (From [84]).

Fig. 3.7

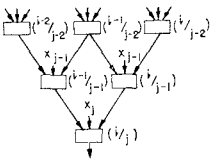
Canaday ontwikkelde in 1964 verschillende methodes voor de synthese van willekeurige kombinatorische funkties, waarbij de 3 inputx drempel elementen gebruikt worden. (lit. 2a) Canaday's eerste synthese algorithme behandelde de konstruktie van een intersektie matrix. Doormiddel van deze intersektie matrix kan men onmiddellijk een majority- gate netwerk afleiden voor de sythese van de vereiste funktie. Canaday heeft ook verschillende methodes ontwikkeld zowel rekenkundige als heuristische, voor de reduktie van de afmeting van de intersektie matrix. Hierdoor namen de afmetingen van het netwerk ook af.

Amarel 1964 (lit.1) en zijn kollega's hebben ook de synthese van cellulaire netwerken, bestaande uit 3 input drempel elementen, bestudeerd. Deze netwerken worden echter gebruikt voor de realisatie van k uit n funkties (k < n).
Een k uit n funktie heeft n variabelen. De funktie heeft de waarde één als k of meer variabelen de waarde één hebben. In het vervolg wordt de funktie aangeduid door (k/n).
Fig. 3.8 toont een cell. netwerk, dat een (6/9) funktie realiseert. De cellen bestaan uit 3 input drempel elementen. Amarel bewijst dat een netwerk, opgebouwd overeenkomstig het schema van fig. 3.8 a enb , iedere (a/b) funktie realiseert.



(6/9) function realized by 3-input majority gates according to the network scheme $N_3(2)$.

a



b

Fig. 3.8

Het netwerk van fig. 3.8 kan alleen (k/n) funkties realiseren. Het aantal cellen dat daarbij nodig is, is gelijk aan k(m-k+1)-1. Nadeel van dit netwerk is: bij iedere (k/n) funktie behoort een bepaalde topologische struktuur. Het netwerk is daardoor niet universeel(kan niet met cel parameters in gesteld worden).

De theorie van de synthese van (m/n) funkties kan met sukses worden toegepast op de synthese van symetrische funkties.
Een symetrische funktie kan als volgt gedefinieerd worden:
Stel $S_{A,B,C\ldots}(x_1, x_2 \ldots x_n)$ is een symetrische funktie

A, B, C enz. zijn gehele getallen. $S_{A,B,C\ldots}(x_1, \ldots x_n)$ neemt de waarde één aan dan en slechts dan, als het aantal variabelen, die de waarde één hebben, gelijk is aan A, B, C ... enz...
Ook komt men de volgende schrijfwijze tegen,

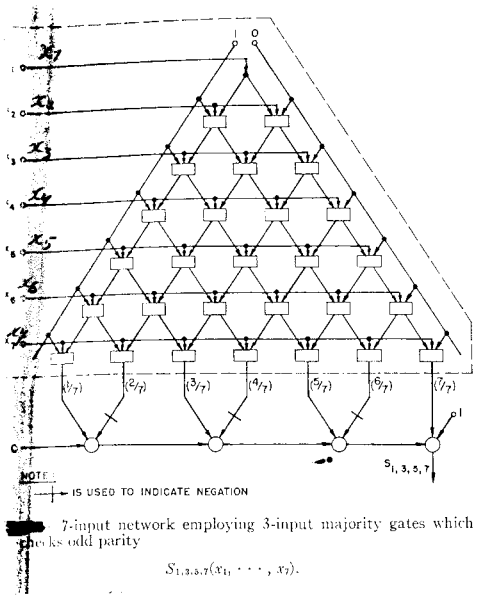
$S_{a<t<a'\ldots\ldots l<t<l'}(x_1, x_2 \ldots x_n)$

t is het aantal variabelen met de waarde één. Als t voldoet aan een van de ongelijkheden dan heeft S de waarde één.

Als $a<a'\ldots\ldots$ enz. dan wordt

$S_{a\leq t<a'\ldots\ldots l\leq t<l'}(x_1, \ldots x_n) = (a/n)\overline{(a'/n)} + \ldots\ldots \overline{(l/n)}(l'/n).$

In fig 3.9 wordt de realisatie van de funktie $S_{1,3,5,7}$ weergegeven.



7-input network employing 3-input majority gates which checks odd parity

$S_{1,3,5,7}(x_1, \ldots, x_7)$.

Fig. 3.9

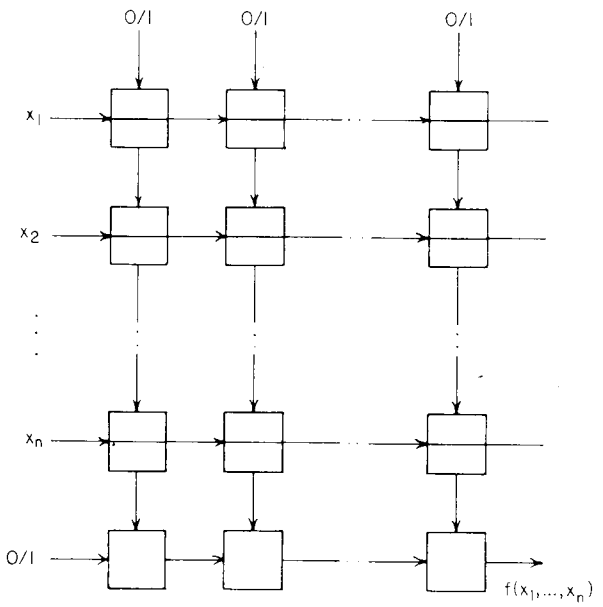
Zschirnt (lit 21) 1966 heeft ook cellulaire netwerken bestaande uit drempel elementen bestudeerd. Hij bewees dat een boom van drempel elementen ingebed kan worden in een cellulair netwerk van NOR poorten.

3.2 Twee dimensionale netwerken met variabele celfunkties.

De celparameters, die de celfunktie veranderen, kunnen als"cutpoints" worden uitgevoerd. De celparameters worden dan tijdens het laatste gedeelte van het fabrikage proces ingesteld. Door b.v. fotocellen en de mogelijkheid om op bepaalde plaatsen verbindingen te verbreken.
De celparameters kunnen ook als flipfloppen worden uitgevoerd, zodat deze parameters elektrisch ingesteld kunnen worden.

3.2.1 Randgevoed horizontaal bus type.

Een voorbeeld van een rand gevoed bus type laat fig 3.10 zien.



Edge-fed horizontal bus type array with 2-input, 1-output cells.

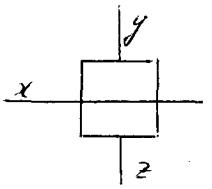
Fig. 3.10

De cellen bestaan uit 2 inputs en een output. Het bovenste gedeelte van het netwerk wordt hoofdnetwerk of minterm netwerk genoemd. Het onderste gedeelte wordt kollektor netwerk genoemd. De bovenste inputs van het hoofdnetwerk worden

voorzien van konstante nullen of enen. Het hoofdnetwerk produceert een aantal funkties, die in het kollektornetwerk worden samengevoegd tot een output funktie. (Ook meerdere outputs zijn mogelijk zie fig. 3.I3b blz 41)
Een eenvoudige ontwerp methode is, door de kolommen van het hoofdnetwerk de mintermen van de funktie te laten produceren. Het kollektornetwerk sommeert dan deze mintermen. De celfunkties kunnen dan b.v. zijn;

$z = x.y$ $\quad z = x+y$

$z = \bar{x}.y$ $\quad z = \overline{x+y}$



Deze funkties kunnen m.b.v. celparameters ingesteld worden.
Growth rate is $(n + I)2^{n-1}$.

3.2.I.A Cutpoint netwerken.

De cutpoint netwerken zijn gebaseerd op de maitra cascades. (zie hfd. 2) Elke kolom van het hoofdnetwerk is in staat om elke niet redundante cascade realiseerbare funkties te produceren. In het vervolg worden deze niet redundante cascade realiseerbare funkties maitra funkties genoemd.
Minnick 1964 bewijst dat een van de twee onderstaande series van 6 celfunkties voldoende is om een maitra funktie te realiseren door een niet redundante cascade. Minnick 1964 (lit.9)

$\bar{x}.y$	$\bar{x}\,y$
$x.\bar{y}$	$x\,y$
$\bar{y}$	y
$x \oplus y$	$x \oplus \bar{y}$
$\bar{x} + \bar{y}$	$\bar{x} + y$
$x + \bar{y}$	$x + y$

De zes celfunkties worden met 3 bits ingesteld. Hierdoor zijn 8 celfunkties mogelijk.
Minnick gebruikte de celfunkties van fig. 3.I2.

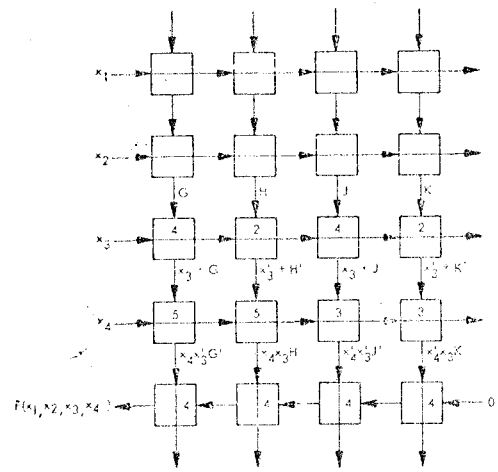
INDEX	a	b	c	d	z
0	0	0	0	0	1
1	0	0	0	1	y'
2	0	0	1	0	x'+y'
3	0	0	1	1	x'y'
4	0	1	0	0	x+y
5	0	1	0	1	xy'
6	0	1	1	0	x⊕y
7	0	1	1	1	0
F	1	1	0	1	x=S, y=R

fig 3.12

Fig. 3.I2

De funktie f=0 en f=I zijn er bij gekomen, omdat deze funkties reeds in de cel aanwezig zijn, zonder toevoeging van extra logica. Minnick voert tevens nog een extra vierde bit in om de cel als flipflop te laten werken. Er hoeft in het laatste geval maar zeer weinig logica aan de cel worden toegevoegd om de flipflop werking te realiseren.
Iedere willekeurige funktie van twee variabelen kan, door het in cascade schakelen van twee van deze cellen, gerealiseerd worden.
Een funktie van 4 variabelen kan op de volgende manier worden ontleed, $F(x_4,x_3,x_2,x_1) = x_4\bar{x}_3\,\bar{g}(x_2,x_1) + x_4 x_3 H(x_2,x_1) + \bar{x}_4\bar{x}_3 J(x_2,x_1) + \bar{x}_4 x_3 K(x_2,x_1)$

Realisatie van deze funktie zie fig 3.I3a



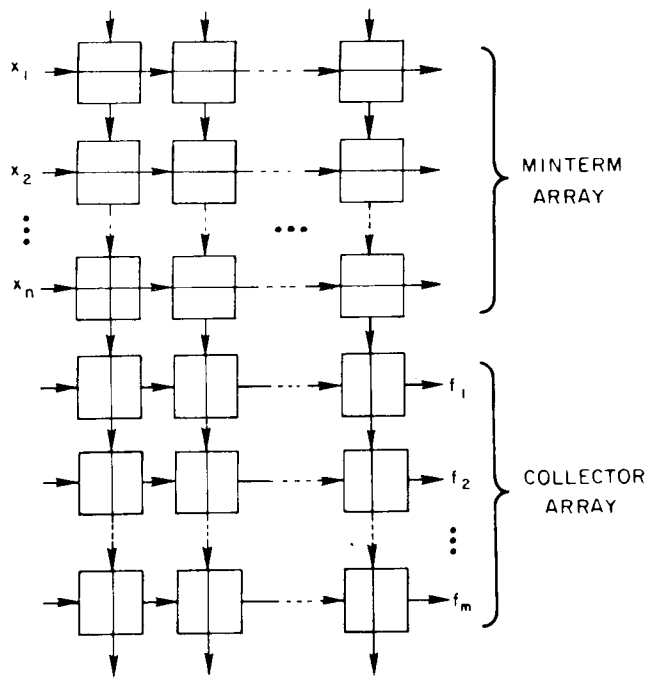
-Cutpoint array for an arbitrary function of four variables.

Fig. 3.I3a

Het aantal produkttermen bij een funktie van n variabelen, wanneer deze volgens bovenstaande methode wordt ontleed, is $(n+I)2^{n-2}$. Growth rate $n2^{n-2}$.

De laatst genoemde ontwerp methode garandeert geen minimum aantal cellen in de cutpoint netwerken. Minnick past in zijn artikel wel enige reduktie technieken toe, maar die zijn sterk afhankelijk van de vindingrijkheid van de systeemontwerper. (Geen methodische reduktie techniek)
Er bestaan minimalisatie algorithmes, die de synthese van bepaalde vormen van cutpoint netwerken aanzienlijk verbeteren. (Mukhopadhyay en Weiss zie sektie 3.2.1.C)

Fig. 3.13b toont een uitgebreider type cutpoint netwerk.



Augmentation of the row-bussed array for the simultaneous realization of many functions.

Fig. 3.13b

Het kollektor netwerk is tot meerdere rijen uitgebreid. Het netwerk heeft nu meerdere outputs.
De celfunktie van het hoofdnetwerk kan een van de funkties van de cutpoint cel van Minnick zijn. De funktie wordt aangeduid door Z_c. Zie fig 3.13c.

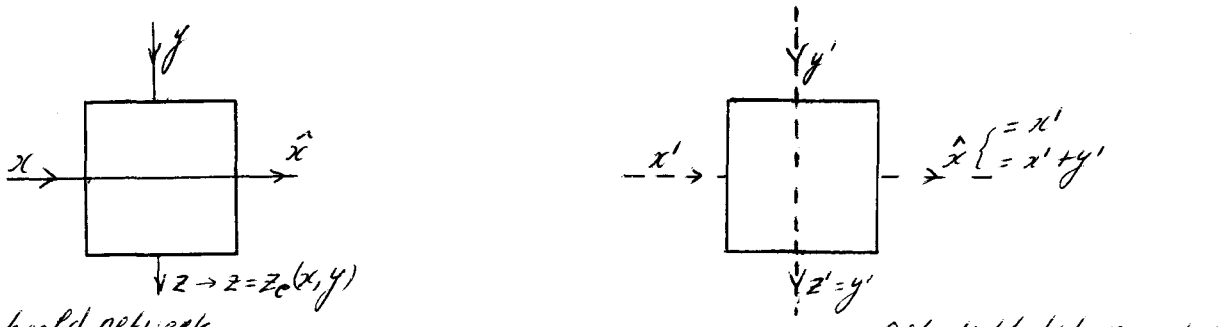


Fig 3.13c

Ter onderscheiding van de cel van het hoofdnetwerk, zijn de verbindingen van de cel van het kollektor netwerk gestreept weergegeven.
Beide type cellen kan men in een cel onderbrengen. Hiertoe worden de outputs van het hoofdnetwerk via een bus in vertikale richting omhoog geleid, waarbij de cel in fig. 3.13d wordt toegepast.

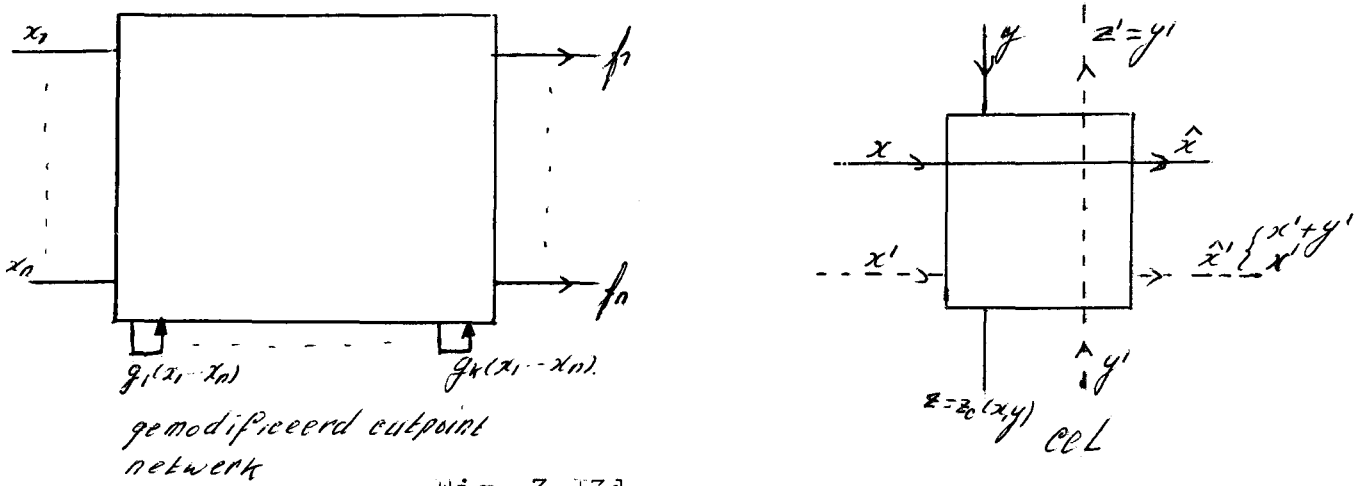


Fig. 3.13d

Deze cel bezit meer celparameters dan de cutpoint cel van Minnick. Door verschillende funkties van $Z_c(x,y)$ weg te laten of te laten samenvallen met een van de twee funkties van $\hat{x}'(x',y')$, kan men het aantal parameters reduceren. Hierdoor zijn diverse types netwerken mogelijk.

Minnick beschrijft ook de synthese van enige komputer elementen zoals schuifregister, vermenigvuldiger, code omzetter enz... m.b.v. cutpoint netwerken. Toepassing van de cutpoint cel op bovengenoemde elementen, blijkt inefficiënt te zijn. Minnick heeft daarom in 1965 (lit 9a) een ander type cel ontwikkeld, dat efficiënter toegepast kan worden op netwerken, die parallele rekenoperaties realiseren.

3.2.1.B Cobweb netwerken.

Een efficiënter cellulair netwerk dan het cutpoint netwerk is het cobweb netwerk. De gebruikte cel heeft ongeveer dezelfde logica als die van het cutpoint netwerk. De informatie stroom stroomt hier niet alleen van links naar rechts en van boven naar beneden, maar ook van rechts naar links en van boven naar beneden in diagonale richting. Daaruit volgt dat een cascade realiseerbare funktie niet gerealiseerd hoeft te worden door een vertikale cascade, maar ook gerealiseerd kan worden door een zig-zag pad in het netwerk.
De cobweb netwerken zijn meer bedoeld voor het ontwerpen van rekenkundige operaties dan het realiseren van boolse funkties. Er zijn reeds speciale cellen ontwikkeld voor vermenigvuldigers, deelers enz.. , die sneller en veel efficiënter werken dan deze cobweb cellen. Om deze reden wordt het principe van de cel in het kort behandeld.
Fig 3.14 geeft de cobweb cel weer.

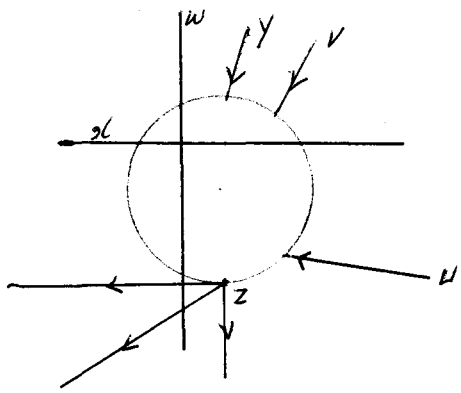


Fig 3.14

De celfunktie van deze cel is gelijk aan die van de cutpoint cel. Het nummer van de cel funktie wordt in de cel geschreven. Dit nummer komt overeen met het nummer in de tabel van fig. 3.12. Normaal zijn de inputs van de cel niet doorverbonden met het inwendige van de cel. Wordt een van de inputs door verbonden met het inwendige van de cel,dan wordt dit aangegeven met een cirkeltje of driehoek om de input. Zie fig. 3.15.

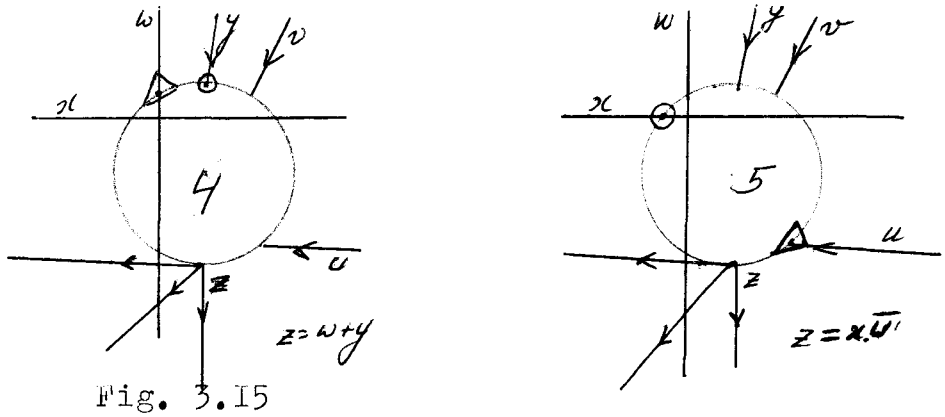


Fig. 3.15

Voor het realiseren van de boolse funkties met de nummers 0,1,2,3,4,5,6,7 worden 2 van de 5 inputs met het inwendige van de cel doorverbonden. De input die omcirkeld is, komt overeen met de x input van de tabel van fig. 3.12.
De input van de cobweb cel,die aangeduid wordt met een driehoek,komt overeen met de y input van de tabel van fig. 3.12.
De cobweb cel is ook instaat om de informatie stroom te wijzigen. De cel met deze eigenschap wordt springcel genoemd. Deze cel is instaat twee of meerdere in en outputs met elkaar door te verbinden. De inwendige celfunktie wordt hierbij uit geschakeld. De sprongcel wordt aangeduid met een J in de cirkel. De in- en outputs, die met een cirkel of driehoek zijn voorzien, zijn met elkaar doorverbonden. Wil men nog een extra doorverbinding maken van in- en outputs, die geisoleerd is t.o.v. de eerste doorverbinding, dan wordt dit aangeduid met driehoekjes om de betreffende in- en outputs. Zie fig.3.16.

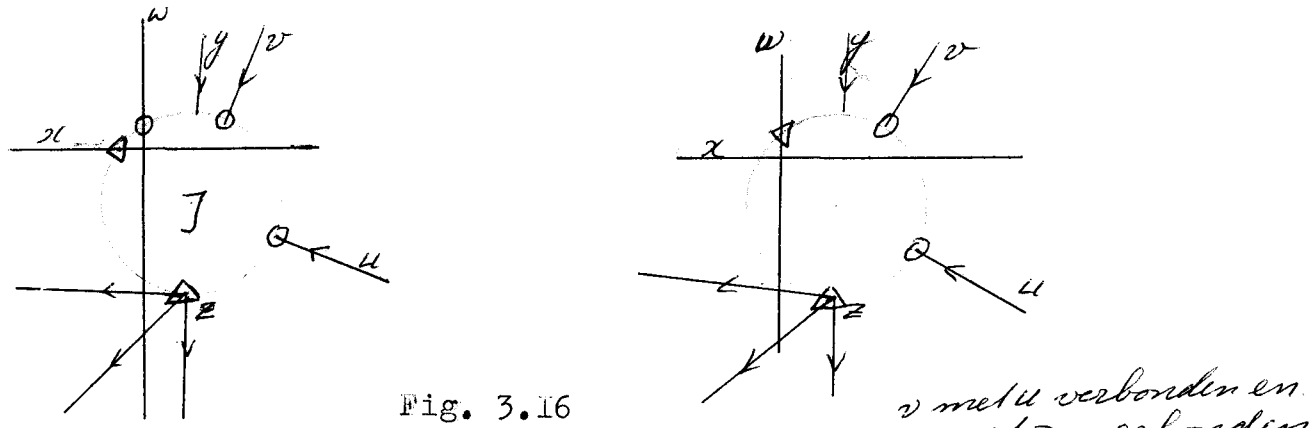


Fig. 3.16

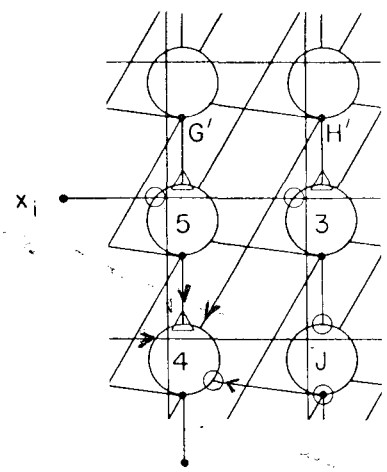
v met u verbonden en w met z verbonden

v, w en u zijn met elkaar doorverbonden
x en z " " " "

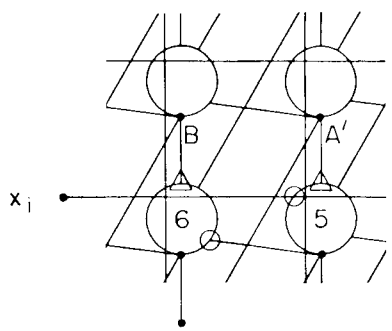
Fig. 3.I7 a en b laten twee ontledingen van een funktie $E(x_1, x_2 \ldots\ldots x_n)$ zien.

Fig. 3.I7a	$E=Gx_i+H\bar{x}_i$	Shannon ontleding.
Fig. 3.I7b	$E=Ax_i \oplus B$	Reed ontleding

G,H,A en B zijn funkties van n-I variabelen. De variabele x_i komt daarin niet meer voor. Aangenomen wordt dat $\bar{G}, \bar{H}, \bar{A}$ en $\bar{B}$ cascade realiseerbaar zijn.



(a) SHANNON



(b) REED

Fig. 3.I7

Konklusie

Het essentiële verschil tussen het cutpoint en cobweb netwerk is de gekompliceerdere en flexibelere onderlingen verbindingstruktuur van het laatst genoemde netwerk. Deze flexibiliteit geeft de ontwerper meer vrijheid op topologisch gebied. Voor cobweb netwerken bestaan echter geen algorithmes of systematische methodes voor het ontwerpen van boolse funkties.

3.2.1.C Minimalisatie algorithmes.

Uit het voorgaande blijkt, dat er behoefte bestaat aan algorithmes of systematische methodes om de cellen in een cellulair netwerk zo efficiënt mogelijk te gebruiken. Onder "zo efficiënt mogelijk" wordt verstaan, het optimaliseren van diverse funkties en parameters waaruit het cellulaire netwerk bestaat. Zoals het aantal cutpoints per cel, de koplexiteit van de cel, de onderlinge verbindingstruktuur, de vertraging en het totaal aantal cellen enz.... Vele van deze parameters zijn onderling afhankelijk van elkaar. Elspas 1964 (lit. 3) heeft de afhankelijkheid van enige topologische parameters bestudeerd, zoals het totaal aantal verbindingen die in een cel plaats vinden, de maximale en gemiddelde cel afstand, de afmetingen en de onderlinge verbindingstruktuur van de netwerken.
De cutpoint en cobweb netwerken produceren een boolse funktie door de outputs van meerdere maitra cascades te verzamelen in een kollektornetwerk. Daardoor bestaat er eenbehoefte om een funktie uit te drukken in een minimaal aantal maitrafunkties. Er is nog geen algorithme bekend, die een willekeurige funktie kan ontleden in een minimaal aantal maitra funkties. De variabelen van de maitra cascades moeten zodanig geordend zijn, dat deze gerealiseerd kunnen worden m.b.v. een randgevoed bus type cell. netwerk.
Er bestaan wel enige minimalisatie algorithmes, die toegepast kunnen worden op enkele speciale vormen van cutpoint netwerken.

Mukhopadhyay 1969 (lit. 13) ontwikkelde een minimalisatie algorithme voor een unate cellulair netwerk. Een unate cellulair netwerk is van het randgevoed bus type, waarvan de unate cel op slechts één uitzondering na geheel gelijk is aan de cutpoint cel van Minnick. De unate cel is namelijk niet instaat om de exclusive-or of equivalentie funktie te genereren. Een cascade, opgebouwd uit deze unate cel, wordt unate cascade genoemd. De unate cascade is niet redundant. Het kollektornetwerk bestaat ook uit een unate cascade, in zijn artikel bewijst Mukhopadhyay dat:

a/ De output van een unate cascade een unate funktie* is.
b/ Iedere unate funktie is m.b.v. een unate cascade realiseerbaar.
c/ De prime implicants van een unate funktie zijn essentiële prime-implicants.

* Def. unate funktie:
Iedere funktie, waarvan de variabelen steeds in dezelfde polariteit in zijn gereduceerde vorm (som van prime-implicants)voorkomen, worden unate funkties genoemd.

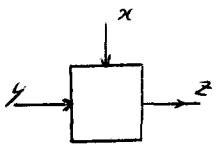
Het minimalisatie probleem waar Mukhopadhyay (lit. 13) zich mee bezig hield, was als volgt gedefinieerd:

a/ Vind een set van unate funkties $u_1, u_2, \ldots\ldots$ zodanig dat $f = \sum_{i=1}^{k} u_i$ of $f = \prod_{i=1}^{k} u_i$

b/ Elke unate funktie is unate cascade realiseerbaar, met dezelfde ordening van input variabelen. (randgevoed-bus type)

c/ K is minimaal.

C.D. Weiss (lit. 19 en 20) ontwikkelde in 1969 een algemener minimalisatie algorithme. Hij ontleedde de funktie in een minimaal aantal cascades (redundant of niet redundant). De cel, die daarbij werd toegepast, is weergegeven in fig. 3.18.



$z = x + y$

$z = \bar{x} + y$

$z = \bar{x} . y$

$z = x . y$

$z = \bar{x} \oplus y$

Fig. 3.18

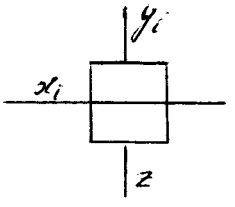
De input variabelen van iedere cascade hebben niet dezelfde ordening. Het kollektornetwerk kan alleen bestaan uit AND en OR cellen. Door laatst genoemde beperking, geeft dit algorithme niet de garantie de funktie in een minimum aantal cascade realiseerbare funkties te ontleden. (Exclusive-or is niet in het kollektornetwerk aanwezig).
In juli 1969 (lit. 19) introduceerde Weiss een speciaal type boolse funktie (standaard cascade vorm) met de eigenschap dat alle maitra funkties in deze vorm geschreven kunnen worden. De introduktie van deze standaard cascade vorm leidt tot een sterke konditie voor niet redundante cascade realiseerbare funkties nl.: De prime- implicants van deze funkties zijn essentiële prime-

implicants. Dit leidt tot een nieuwe test procedure om een willekeurige funktie te testen voor niet redundante cascade realiseerbaarheid.
In september 1969 (lit. 20) wordt de standaard cascade vorm uitgebreid voor redundante cascades. Tevens wordt aangetoond dat alleen cascade realiseerbare funkties in deze standaard cascade vorm geschreven kunnen worden.
Van een gegeven funktie worden alle prime cascade realiseerbare funkties, die de gegeven funktie bedekken, bepaald. Daarna wordt een minimale set van prime cascade realiseerbare funkties gevormd, die de gegeven funktie geheel bedekken.
Weiss geeft verder aan, hoe het algorithme kan worden veranderd om optimale netwerken te vinden voor een set van funktie die dan't care kondities bezitten.

3.2.I.D q-funktie of adder netwerken.

Dit netwerk wordt in het artikel van Minnick (lit. I2) door Kautz en Myhill geintroduceerd. Dit artikel is niet op korte termijn verkrijgbaar. In onderstaande tekst zal een korte samenvatting gegeven worden van dit netwerk. (Gegevens zijn afkomstig van lit. I5).
Dit netwerk is van het randgevoed bus type(zie fig. 3.I9).
De cel van het hoofdnetwerk bezit één specifikatie bit (celparameter b).

$$z = x_i.y_i+b.y_i+b.x_i$$



Is b = 0 dan celfunktie is een AND.
Is b = I " " " " " OR.

Het kollektor netwerk bestaat uit exclusive-or cellen. De output van de cascade van het hoofd netwerk wordt q-funktie genoemd. De cel parameters ($b_n,.....b_2,b_1$) bepalen welke q-funktie de cascade realiseert. Er bestaan, bij n variabelen, 2^n verschillende q-funkties.
In fig. 3.I9a wordt de specifikatie bit b als input aande cel toegevoerd. De cascade kan als de carry keten van een full-adder beschouwd worden, waarbij de binaire getallen ($b_n,.... b_2,b_1$) en ($x_n,...x_2,x_1$) bij elkaar worden opgeteld.

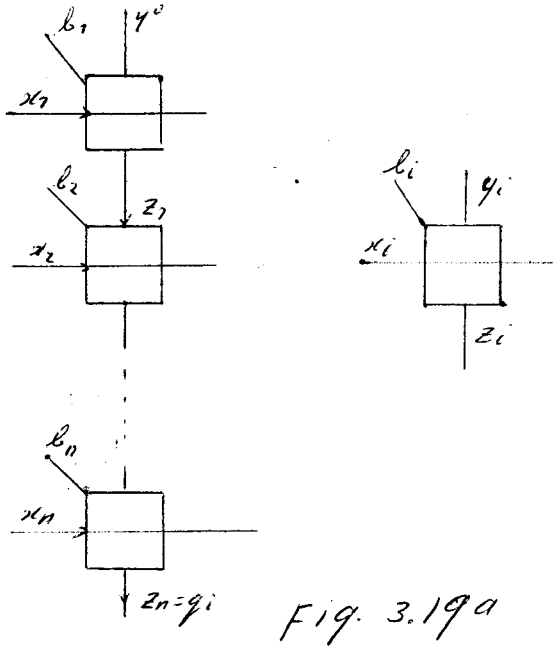


Fig. 3.19a

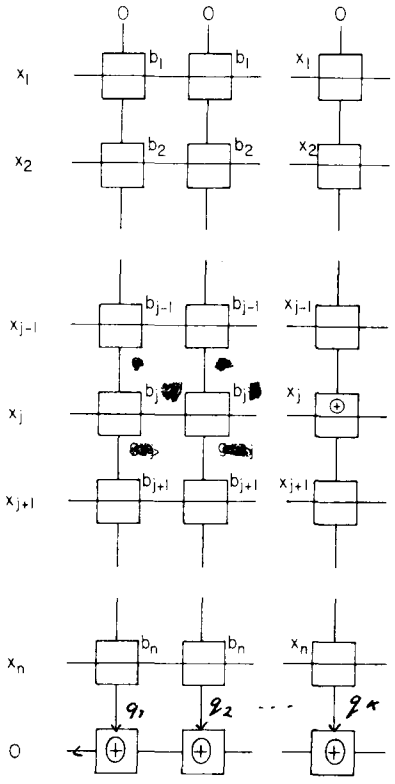


FIG. [illegible]. Simplification of q-function arrays.

Fig. 3.I9

De output z_i fungeert als carry out en y_i als carry in. Er treedt een overflow op als $X+B \geqslant N$
X is het binaire getal $x_n, \ldots\ldots x_2, x_I$.

B is het binaire getal $b_n, \ldots\ldots b_2, b_I$.

$N = 2^n$.

Stel dat B = (00I0I), dan N-B = (II0II), er treedt dan een overflow op bij $X \geqslant$ II0II. De funktie tabel voor z_n is dan:

mintermen

C_i	x_5	x_4	x_3	x_2	x_1	$z_n = q_i$
C_0	0	0	0	0	0	0
C_1	0	0	0	0	1	0
⋮			⋮			⋮
						0
						0
C_{27}	1	1	0	1	1	1
C_{28}	1	1	1	0	0	1
C_{29}	1	1	1	0	1	1
C_{30}	1	1	1	1	0	1
C_{31}	1	1	1	1	1	1

Beschouwt men een 2^n dimensionale vektorruimte, met C_0 t/m C_{2^n-1} als basis vektoren, dan kan iedere funktie door een karakteristieke vektor worden voorgesteld. De karakteristieke vektor van $z_n=q_i$ in laatste voorbeeld is: $q_i=z_n=$ (IIIII000.....000)

Het getal i stelt het aantal enen voor in de kar. vektor. In dit geval is i=5.

Het is nu gemakkelijk in te zien dat de kombinatie $(b_n,....b_2,b_1)$ een binaire representatie is van het decimale getal i.

Tabel 3 in fig. 3.20 laat alle q-funkties zien voor n=2.

TABLE III
q-Functions For $n = 2$

Characteristic vector b_2b_1	$c_3c_2c_1c_0$	q-Function
00	0 0 0 0	$q_0 = 0$
01	1 0 0 0	$q_1 = x_1x_2$
10	1 1 0 0	$q_2 = x_2$
11	1 1 1 0	$q_3 = x_1 + x_2$

fig. 3.20

De volgende stelling is belangrijk voor q-funktie netwerken.

Stelling I

De minterm $C_i = q_{2^n-i} \otimes q_{2^n-i-1}$.

waarbij wordt aangenomen dat $q_{2^n}=I$.

Bewijs:

Volgens het bovenstaande is $q_\nu = \sum_{j=2^n-\nu}^{2^n-1} c_j$

$$q_{2^n-i} \otimes q_{2^n-i-1} = \sum_{j=i}^{2^n-1} c_j \otimes \sum_{m=i+1}^{2^n-1} c_m =$$

$$= \left(\sum_{j=i+1}^{2^n-1} c_j + c_i\right) \otimes \sum_{m=i+1}^{2^n-1} c_m =$$

$$= \left(\sum_{j=i+1}^{2^n-1} c_j + c_i\right) \cdot \overline{\sum_{m=i+1}^{2^n-1} c_m} + \sum_{n=i+1}^{2^n-1} c_n \cdot \overline{\sum_{j=i+1}^{2^n-1} c_j} \cdot \overline{c_i}$$

$$= c_i \cdot \overline{\sum_{n=i+1}^{2^n-1} c_n} = c_i .$$

Een willekeurige funktie kan als volgt geschreven worden

$$f = c_i + c_j + \ldots\ldots c_k$$

$$f = c_i \oplus c_j \oplus \ldots\ldots c_k$$

$$f = q_{2^{n-i}} \oplus q_{2^{n-i-1}} \oplus q_{2^{n-j}} \oplus q_{2^{n-j-2}} \oplus \ldots\ldots q_{2^{n-k}} \oplus q_{2^{n-k-1}}$$

Een willekeurige funktie kan dus worden opgebouwd m.b.v. een kollektor_netwerk, die alleen maar uit exclusive-or cellen bestaat, met de q-funktie als input.
Er geldt

$$c_i + c_{i+1} = q_{2^{n-i}} \oplus q_{2^{n-i-2}} \oplus q_{2^{n-i-1}} \oplus q_{2^{n-i-2}}$$

$$= q_{2^{n-i}} \oplus q_{2^{n-i-2}}$$

In het algemeen is $c_i + c_{i+1} + c_{i+2} + \ldots c_{i+k} = q_{2^{n-i}} \oplus q_{2^{n-(i+k)-1}}$

mits $c_i, \ldots\ldots c_{i+k}$ een aaneengesloten serie mintermen is.
Dit komt in de karakteristieke vektor van een funktie f overeen met een aaneengesloten rij enen.
Als het aantal rijen enen en nullen tesamen gelijk is aan r, dan zijn er maximaal r-1 kolommen (cascades) nodig voor de realisatie van de funktie.
Voorbeeld.

$$f(x_1, x_2, x_3) = (1\,0\,0\,1\,1\,1\,1\,0) = c_7 + c_4 + c_3 + c_2 + c_1$$

$$c_7 = q_1 \qquad c_4 + c_3 + c_2 + c_1 = q_7 \oplus q_3$$

$$f(x_1, x_2, x_3) = q_1 \oplus q_3 \oplus q_7$$

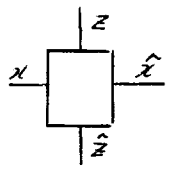
De specifikatie bits voor q_1, q_3 en q_7 zijn resp.
$(b_3 = 0, b_2 = 0, b_1 = 0)$ $(b_3 = 0, b_2 = 1, b_1 = 1)$ en $(b_3 = 1, b_2 = 1, b_1 = 1)$.
De growth rate van het q-funktie netwerk is afhankelijk van het maximaal aantal rijen enen en nullen in de karakteristieke vektor van de betreffende funktie. Dit aantal rijen nullen en enen is afhankelijk van de ordening en polariteit van de input variabelen. Er bestaat nog geen algorithme die m.b.v. permutatie en komplementatie van de input variabelen het aantal rijen nullen en enen minimaliseert.
Elspas en Short (1964 lit.3) hebben de bovengrens van dit aantal rijen onderzocht, waarbij permutatie en komplementatie van de variabelen is toegestaan. Zij kwamen op een maximum aantal rijen nullen en enen van ;

$(2^{n+1} + 2)/3$ voor even n

$(2^{n+1} + 2)/3$ voor oneven n

De growth rate is in dit geval $\frac{1}{3} n 2^n$

Men kan de cellen van het hoofd_netwerk en van het kollektor-netwerk in een cel onderbrengen.
De cel ziet er dan als volgt uit;



$\hat{z} = z.x$ $\quad \hat{z} = z$ $\quad \hat{z} = z + x$

$\hat{x} = x$ $\quad \hat{x} = x \oplus z$ $\quad \hat{x} = x$

Deze celfunkties kunnen met 2 bits ingesteld worden. Hierdoor is nog een vierde funktie mogelijk. Hiervoor wordt gekozen $\hat{z} = z$
$\hat{x} = x$
De kolommen kunnen nu iedere willekeurige pos. unate funktie produceren.
Er dient te worden opgemerkt, dat de q-funktie ook een unate funktie is.

Elspas et. al. 1967 (lit4) introduceerde een cel met de volgende celfunkties

$\hat{z} = z$ $\quad \hat{z} = zx$ $\quad \hat{z} = z$ $\quad \hat{z} = x$

$\hat{x} = x$ $\quad \hat{x} = x$ $\quad \hat{x} = x \oplus z$ $\quad \hat{x} = z$

↑ permutatie

Doormiddel van de permutatie cel bezit dit netwerk veel meer mogelijkheden dan de voorafgaande netwerken. Alleen het cobweb netwerk bezit ook deze eigenschap.
Het is moeilijk om hier een systematische methode te vinden, voor de synthese van logische funkties. Lit 15 geeft blz384 een kort overzicht van dit type netwerk.

3.2.I.E Exclusive or of equivalentie minimalisatie algorithme.

Het is reeds lang de ervaring van logische ontwerpers dat de realisatie van een schakelfunktie, als een modulo 2 of komplementaire modulo 2 som van produkten, ekonomischer is dan de konventionele normaal vorm realisaties. Hieruit is het volgende minimalisatie probleem ontstaan;
vind een systematische procedure om een willekeurige

gegeven funktie uit te drukken in een minimale exclusive-or som van cascade realiseerbare funkties. Dit probleem schijnt zeer moeilijk oplosbaar te zijn. De oplossing is tot nu toe niet bekend. Zelfs niet in het simpelste geval, wanneer men een funktie ontleedt in een minimale exclusive-or som van willekeurige produkten van variabelen.
D.E. Muller (lit. I5a) en S.Even, I. Kohavi en A. Paz (lit. 4a) hebben bovenstaand probleem onderzocht. Er zijn daaruit geen bevredigende resultaten gekomen.
Mukhopadhyay 1970 (lit. I4) heeft een algorithme ontwikkeld voor toekenning van de polariteiten aan de input variabelen, zodanig dat een maximaal aantal konstanten van de REED-MULLER ontleding nul worden.

Een willekeurige funktie van n variabelen kan in de volgende vorm geschreven worden:

$$f(x_1 \dots x_n) = a_0 \oplus a_1 x_1 \oplus a_2 x_2 \dots \oplus a_{n+1} x_1 x_2 \dots \oplus a_{2^n-1}\, x_1 \cdot x_2 x_3 \dots x_n \quad \text{(I)}$$

De konstante a_i heeft de waarde nul of een.
Deze ontleding wordt REED-MULLER ontleding of "komplement ring sum" ontleding genoemd.

Bewijs: $f = p_i + p_j \dots p_k$ som van mintermen

er geldt $p_i \cdot p_j = 0$ dan $f = p_i \oplus p_j \oplus \dots \oplus p_k$ (2)
We splitsen de literals behorende bij de minterm p_i in twee groepen. De niet gekomplementeerde en de gekomplementeerde literals.
Dus bijvoorbeeld:

$$p_i = x_1 \dots x_s \bar{x}_{s+1} \dots \bar{x}_n$$

$$p_i = x_1 \dots x_s (1 \oplus x_{s+1}) \oplus \dots (1 \oplus x_n)$$

$$p_i = (x_1 x_2 \dots x_s) \oplus (x_1 x_2 \dots x_s x_{s+1}) \oplus \dots (1 \oplus x_n)$$

Door de faktoren $(1 \oplus x_i)$ te elimineren en dan p_i in formule 2 te substitueren verkrijgt men formule I.

Er dient te worden opgemerkt dat het algorithme van Mukhopadhyay geen minimale exclusive-or ontleding garandeert voor een willekeurige funktie.

3.2.2 2^n cel netwerken

Deze netwerken bezitten 2^n cellen. Voor iedere willekeurige funktie zijn alle 2^n cellen nodig. Daarom zijn deze netwerken geschikt voor het realiseren van funkties, die m.b.v. andere bekende standaard methoden moeilijk te realiseren zijn.
Kautz (lit 12 1966) behandelt twee types van deze netwerken.
a/ Minterm netwerken
b/ Symetrische funktie netwerken.

3.2.2.A Minterm netwerken.

Lit. 15 blz 293 geeft een korte samenvatting van deze netwerken. De cel bestaat uit 3 lagen en is zeer komplex.

3.2.2.B Symmetrische funktie netwerken.

Ook van dit netwerk geeft lit 15 blz 293 een korte samenvatting. Caldwell realiseert een symetrische funktie m.b.v. een kontakten-netwerk. (lit 2b) Volgens dat principe zijn de symetrische funktie netwerken opgebouwd.
Hierbij wordt ook gebruik gemaakt van de volgende ontleding:
Iedere willekeurige funktie $f(x_1, x_2, \dots x_n)_n$ kan geschreven worden als een symetrische funktie van $2^n - 1$ variabelen

$$f(x_1, x_2 \dots x_n) = S_{a_1 \dots a_s}(x_1, x_2, x_3, \dots x_n)$$

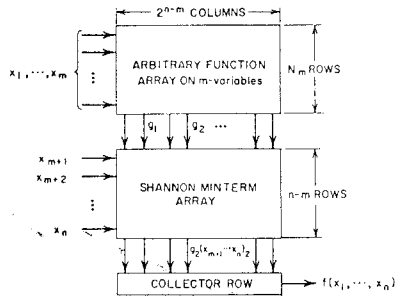
Waarbij x_i, 2^{i-1} keer in de funktie voorkomt. De index a_i komt overeen met het nummer van de minterm waaruit de funktie bestaat. Zie lit 7a .

3.2.3. Twee dimensionale netwerken m.b.v. twee rail cascades.

Short 1965 heeft een twee rail cascade behandeld (zie hfd 2 sektie 2.2), die hij toepast op een twee dimensionaal netwerk. Hij ontleedt een willekeurige funktie in een Shannon ontleding.

$$f(x_1 \dots x_n) = \sum_i g_i(x_1 \dots x_m) \cdot (x_{m+1} \dots x_n)_i$$

$f(x_1 \dots x_n)$ wordt uitgedrukt in een som van mintermen, bestaande uit de variabelen x_{m+1} t/m x_n . Elke minterm bezit een coëfficiënt $g_i(x_1 \dots x_m)$, die een willekeurige funktie van de variabelen $(x_1, x_2, \dots x_m)$ is.
De funkties $g_i(x_1 \dots x_m)$ worden door een twee rail cascade (Short type) gerealiseerd. De mintermen worden door een minterm netwerk gerealiseerd. Zie fig. 2.23



Construction of rectangular array.

Fig. 2.23

De kollektor rij sommeert de outputs van het"Shannon minterm" netwerk, zodat $f(x_1, \ldots x_n) = \sum_i g_i(x_1 \cdots x_m).(x_{m+1} \cdots x_n)_i$

Kollektor-

Deze cellen bezitten parameters om de gewenste g_i's uit te selekteren.

Het aantal cellen van het twee rail cascade netwerk bestaat uit minder dan $(m+2^m-1)2^{n-m}$ cellen

Het aantal cellen van "Shannon minterm"netwerk is, $(n-m)2^{n-m}$

Het aantal cellen van de kollektor rij is 2^{n-m}

Het totaal aantal cellen is $\leq m2^n+(n-m)2^{n-m}$

Men kan m=F(n) zodanig kiezen dat het aantal cellen minimaal is.

Als bijvoorbeeld m=logn dan is de growth rate $(\log n)2^n$. Dit is een verbetering t.o.v. het rand gevoed horizontaal bus type.

Verdere toepassing van twee rail cascades (Yoeli-Turner zie hfd.2 sektie 2.2.) in twee dimensionale netwerken ben ik nog niet tegen gekomen.

3.3 Programmeerbare cellulaire netwerken.

Bij deze netwerken kunnen de cel parameters uitwendig door elektische signalen ingesteld worden. De funktie van het netwerk kan dus elektisch ingesteld worden. Het netwerk is universeler dan al zijn voorgangers.

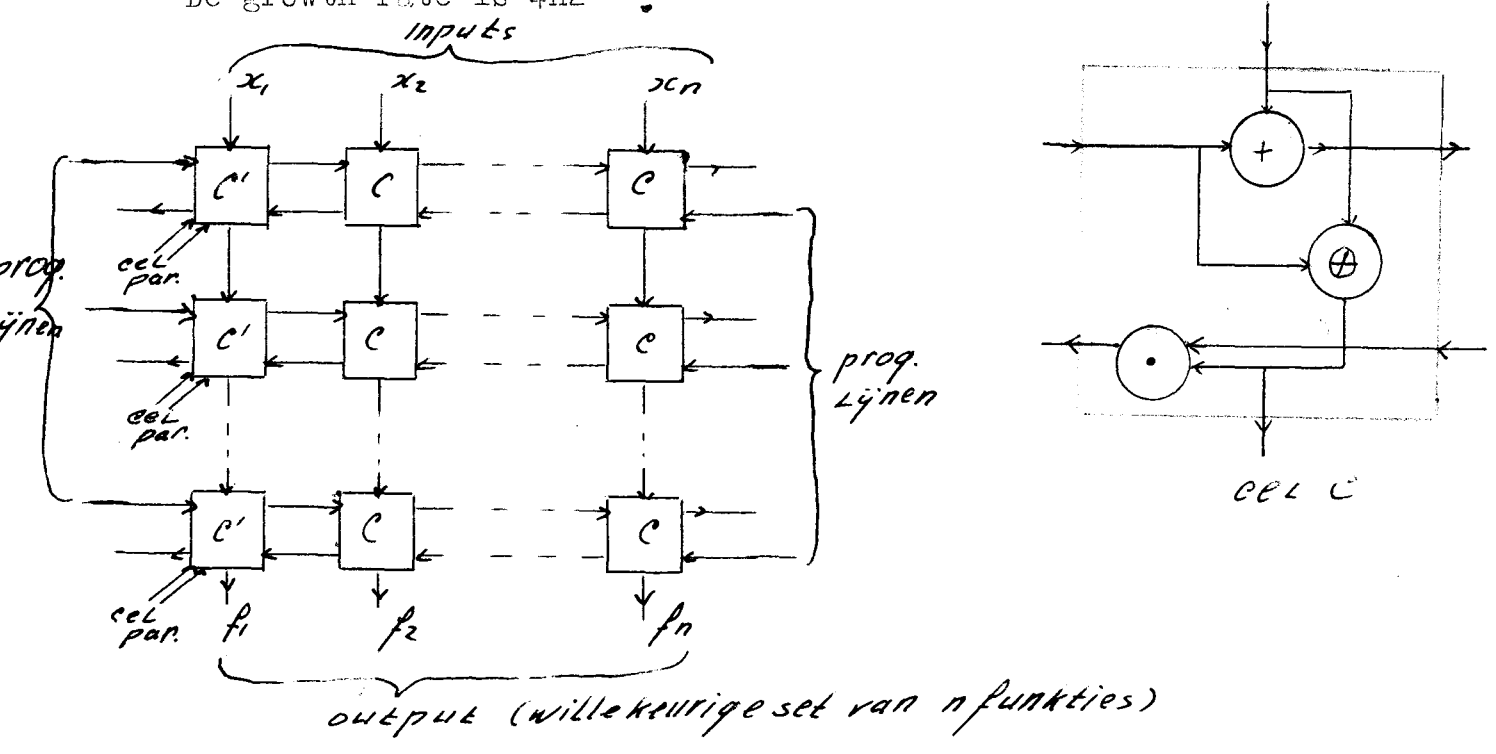
De voordelen die programmeerbare netwerken bieden, zijn in de introduktie van dit hoofdstuk genoemd.

3.3.I

De cutpoint cellen zijn de voorlopers geweest van de geprogrammeerde cellen. Door de cutpoints te vervangen door flipflops (mogelijk schuifregisters) kan men de cel-parameters elektrisch instellen. Van alle in sektie 3.I behandelde netwerken kunnen de celparameters m.b.v. flipflops ingesteld worden. De cel wordt dan in vele gevallen komplexer. Ook de verbindingstruktuur wordt ingewikkelder. Iedere cel afzonderlijk moet geprogrammeerd worden, Hiervoor zijn extra programmeerlijnen voor nodig. De programmering via een sequentiële machine geschieden (schuifregisters), die in het netwerk is ingebouwd en waarbij de programmeer data sequentieel aan de machine via één input kan worden toegevoerd.
Literatuur over deze netwerken zie:
Wahlstrom S.E. I967 (lit.8)
Minnick R.C. I967 (lit. I0)

3.3.2 Universeel cellulair netwerk.*

Dit netwerk is volgens een geheel andere methode opgebouwd dan de netwerken uit sektie 3.3.I.
Dit netwerk kan iedere willekeurige funktie realiseren door alleen de cellen aan de randen van het netwerk te programmeren.
Dit levert een vereenvoudiging van de verbindingstruktuur en cel komplexiteit op.
Het netwerk bestaat uit cellen met 3 inputs en 3 outputs. (zie fig. 2.34) De cellen zijn in een rechthoek gerangschikt.
De signaal stroom is in vertikale richting unilateraal en in horizontale richting bilateraal. Alleen de meest linkse cellen bezitten celparameters.
De growth rate is $4n2^{2n}$.



* Literatuur: Huang J.C. 1970 en 1971 (Lit. 6 en 7).

3.4 Algemene nabeschouwing.

Twee dimensionale netwerken met vaste celfunkties en variabele verbindingstruktuur hebben zeer eenvoudige cellen met weinig logica. De growth rate is $c_o n 2^n$. De coëfficiënt c_o is onafhankelijk van het aantal input variabelen n.

In dit hoofdstuk zijn ook voorbeelden gegeven van netwerken, die bestaan uit drempel elementen. Deze netwerken kunnen op efficiënte wijze de (m/n) funkties en alle andere symmetrische funkties produceren.

In twee dimensionale netwerken met variabele celfunktie wordt vaak de eendim. maitra-cascade of een vereenvoudiging daarvan toegepast.
Voorbeelden: Cutpoint en q-funktie netwerken.
Van deze type netwerken bestaan diverse minimalisatie algorithmen. Echter een algorithme, die een minimale oplossing in het algemene geval garandeert, is nog niet bekend.

De meest flexibele netwerken zijn programmeerbare netwerken. Hieronder vallen alle reeds bekende netwerken, waarbij de celparameter elektrisch ingesteld kan worden. Huang J.C. publiceerde een nieuw type netwerk, die geheel afwijkt van het bekende patroon. Hij maakt gebruik van de theorie der sequentiële machines.

Literatuur opgave

I Amarel S. et al. Majority gate networks. IEEE trans. elec. comp. febr. 1964 p.5-13 .

2 Brooking M.E.,King W.F. 3 , and Giusti A. 1965 . A survey of in-house work on cellular logic. Unpublished memorandum, AFCRL

2a Canaday R.H. Two-dimensional iterative logic . Proceedings Fall Joint Computer Conferance 1965.

2b Caldwell S.H. Switching circuits and logical design. 1960 Wiley, New York.

3 Elspas B. et al.(1964) Investigation of propagation limited computer networks. Stanford Res. Inst. Project 4523.

4 Elspas B. (1967), Goldberg J., Jackson, Kautz W.H., and Stone H.S. Properties of cellular arrays for logic and storage. Sci. Rep. 3, Project 5876, AFCRL Report 67-0463, pp. 103-107. Stanford Res. Inst., Menlo Park, California.

4a Even S.,Kohavi I.,and Paz A. . On minimal modulo 2 sums of products for switching functions. IEEE Trans. Elec. Comp. vol. EC-16 pp. 671-674 October 1967.

4b Haring D.R. Sequential-circuit synthesis; State assignment aspects. Cambridge, Mass.: M.I.T. Press 1966.

5 Hill-Peterson Introduction to switching and logical design 1968. John Wiley and Sons Inc. New York.

6 Huang J.C. Synthesis of combinational cellular arrays. SWIEECO Rekord 1970.

7 Huang J.C. A universal cellular array. IEEE Trans. on Comp. march 1971.

7a Kautz W.H. 1961 The realization of symmetric switching functions with linear input logical element. IRE Trans. on Elec. Comp. EC-10 p.371-378 September 1961.

8 King W.F. 3 , 1965 State logic relations in an iterative structure for autonomous sequential machines. AFCRL Rep. 65-439.

9 Minnick R.C. Cutpoint cellular logic. IEEE Trans. Elec. Comp. December 1964.

9a Minnick R.C. Cobwebs cellular arrays. 1965 Proceedings Fall Joint Computer Conferance.

10 Minnick R. C. A survey of microcellular research. Journal of the assocation for computing machinery. Vol. 14 no 2 April 1967 pp. 203-241.

11 Minnick 1968 Cellular networks. Academic Press. New York.

12 Minnick R.C. , Short R.A., Goldger J. et al. 1966 Cellular arrays for logic and storage. Stanford Res. Inst., Project 5087, Contract AF 19(628)-42333; AFCRL.

12a Miyata F. Realization of arbitrary logical functions using majority element IEEE Trans. Elec. Comp. EC-12 June 1963, 183-191.

13 Mukhopadhyay A. Unate cellular logic. IEEE Trans. on Comput. 1969 Blz. 114 .

14 Mukhopadhyay A. 1970 Minimization of exclusive or and logical equivalence switching circuits. IEEE Trans. on Comput. vol. C-19 No. 2 february 1970.

15 Mukhopadhyay A. 1971 Recent developments in switc-hing theory. Academic Press. INC. 111 fifth Avenue, New York.

15a Muller D.E. Application of boolean algebra to switching circuit design and to error detection. IRE Trans. Elec. Comput. vol. EC-3 pp. 6-12, September 1954

16 Short R.A. 1965 Two rail cellular arrays. AFIPS Conf. Proc. 27 pti , Wasington D.C. Spartan 355-69.

17 Spandorfer and Murphy 1963 Synthesis of logical functions on an array of integrated circuits. Sci. Rep. No 1 and 2. Univac Project 4645, AFCRL Contracts 63-528, Contract AF 19(628)2907, Sperry Rand. Corp. Bluebell, Pennsylvania.

18 Wahlstrom S.E. 1967 Programmable arrays and networks. Electron. 40, 91-95, 1967.

19 Weiss C.D. 1969 The characterization and properties of cascade realizable switching functions. IEEE C-18 July 1969 p. 624-633.

20 Weiss C.D. 1969 Optimal synthesis of arbitrary switching functions with regular arrays of 2 input 1 output switching elements. IEEE Trans. on Comp. C-18, September 1969 p. 839-856.

21 Zschirnt H.H. 1966 Miscellaneous notes on cellular logic structure and their synthesis in eight-neighbour nets. Unpublished memoramdum.

Hoofdstuk 4

Fouten in cellulaire netwerken.

Samenvatting.

In dit hoofdstuk wordt een téchniek behandeld, die fouten in een cutpoint netwerk detekteert en deze fouten herstelt. Tevens worden enige samenvattingen gegeven van de literatuur op dit gebied.

Introduktie.

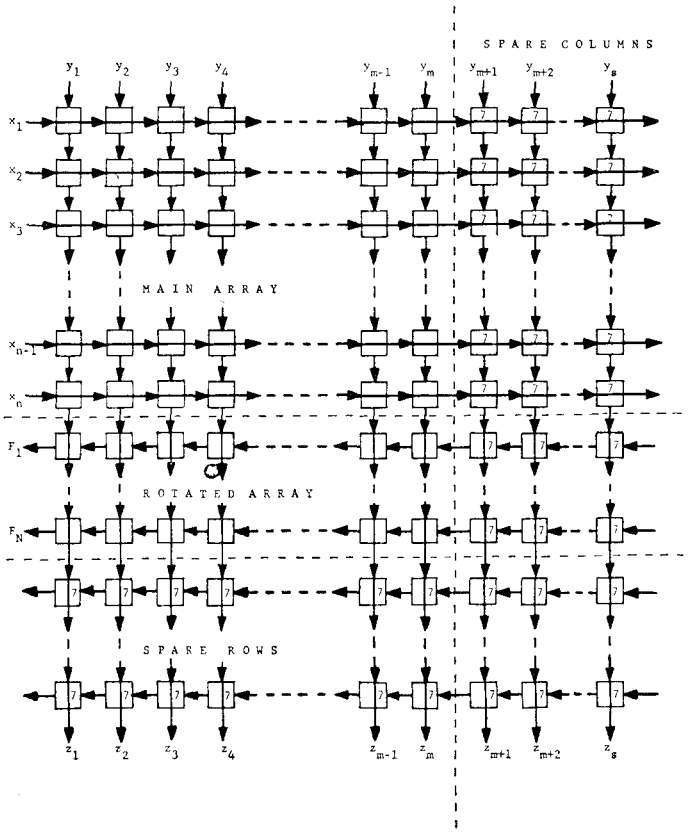
Het opsporen van defekte cellen en het herstellen van deze fouten kan bij niet geprogrammeerde netwerken alleen plaats vinden tijdens het laatste gedeelte van het produktieproces. Men is dan nog instaat om diverse parameters van de verschillende cellen in te stellen.
Bij geprogrammeerde cellen kan men de detektie en reparatie van de defekte cellen na het produktie proces uitvoeren. Er bestaan enige technieken om de plaats van de defekte cellen te kunnen bepalen. Deze technieken kunnen worden toegepast op cell. netwerken van het randgevoed-bus type.

4.1 Fout detektie en reparatie.

Fig. 4.1 laat een cutpoint netwerk zien plus een tabel van celfunkties die toegepast kunnen worden. De reserve kolommen dienen om een defekte kolom in het hoofdnetwerk te vervangen. De reserve rijen dienen om een defekte rij van het kollektornetwerk (rotated array) te vervangen.
Celfunktie 7 wordt alleen in de reserve-rijen en kolommen toegepast.
Fig. 4.2 laat zien op welke wijze een defekte rij of kolom hersteld wordt. De tweede kolom van het hoofdnetwerk is daar defekt en wordt vervangen door een reserve kolom. De verbinding tussen de output van de tweede kolom van het hoofdnetwerk en het kollektornetwerk wordt verbroken. (Zie kruisje).

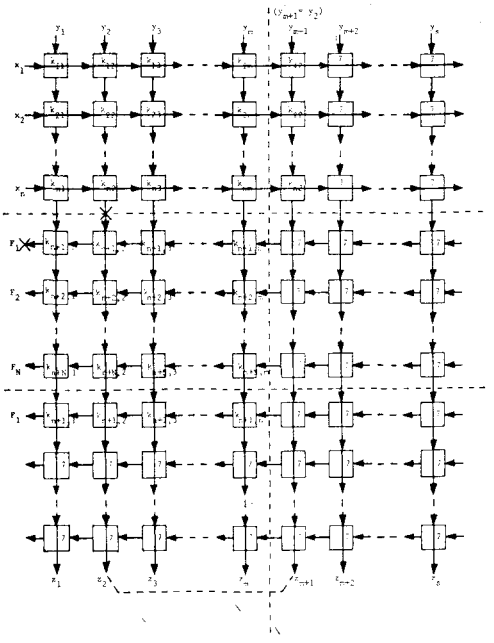
TABLE I

Index k	Cell Output $f(x, y)$
1	y
2	$\bar{x}+y$
3	$\bar{x}y$
4	$x+y$
5	xy
6	$x \oplus y$
7	0



A cutpoint cellular array with spare columns and spare rows.

Fig. 4.1



Connections for replacing the second column in the main array by the first spare column and the first row in the rotated array by the first spare row.

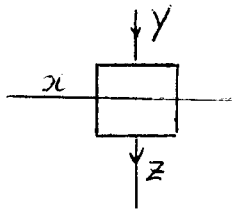
Fig. 4.2

Y_{m+I} wordt Y_2 en de celfunkties van kolom m+I worden gelijk gemaakt aan die van kolom 2. De Z_{m+I} uitgang wordt met Z_2 door verbonden. (zie fig.4.2)
In fig.4.2 wordt ook aangegeven hoe een defekte rij (F_I met kruisje) vervangen wordt door een reserve rij.

Om defekte cellen in cutpoint netwerken op te sporen, is het voldoende om de cascade waarin deze cel zich bevindt te detekteren.
Er wordt hier een procedure behandeld, die de defekte cascade in een hoofd netwerk opspoort. Deze procedure kan men ook toepassen op het kollektor_netwerk, omdat dit netwerk identiek is met een 90° gedraaid hoofd_netwerk.(

Er wordt aangenomen dat de volgende fouten op kunnen treden: SA-1 (stuck at 1) fouten en SA-0 (stuck at 0) fouten. Deze fouten worden veroorzaakt doordat één of beide inputs en/of de output van de cel op de waarde nul of één blijven staan.
Het principe van de procedure (afkomstig van S.S.YAU lit.6) zal hier aan de hand van een voorbeeld worden behandeld.
De cel kan ingesteld worden in 7 verschillende celfunkties, die in tabel 1 van fig. 4.1 worden weergegeven.

Zijn de celfunkties $\bar{x}y$, $x+y$, $x \oplus y$ (overeenkomend met index 3,4en 6) en heeft de x input een konstante waarde 0,dan is in alle 3 gevallen de output Z gelijk aan Z=y



Zijn de celfunktie echter $\bar{x}+y$, xy (index 2 en5) en heeft de x input de waarde 1 dan is Z=y

Bij de cel met index 1 is de output Z onafhankelijk van de x input en ook hier is Z=y

Stel dat een cascade van het hoofd netwerk er alsvolgt uitziet:

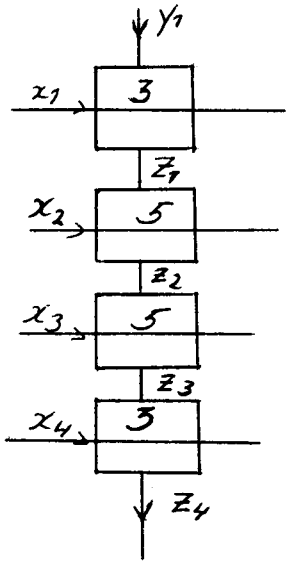
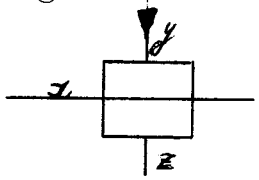


Fig. 4.3

Verondersteld wordt dat de output Z voor meting toegankelijk is.
Geeft men de inputs van de cascade de volgende binaire waarden $(x_1,x_2,x_3,x_4)=(0,1,1,0)$ dan is $Z_4=y_1$, Zijn er geen defekten aanwezig bij de y inputs en z outputs van de cellen, dan moet Z_4 van waarde veranderen als men y_1 van waarde laat veranderen.
Verandert Z_4 hierbij echter niet van waarde dan is de cascade defekt.
Om de x inputs van de cellen te kontroleren gaat men als volgt te werk;

als de celfunktie index 2 of 4 heeft en y=0 dan $Z=x$ of $\bar{x}$



als de celfunktie index 3,5 of 6 heeft en y=1 dan Z= x of $\bar{x}$

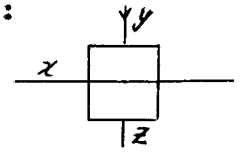
De inputs van de cascade hebben ook hier de binaire waarden $(x_1,x_2,x_3,x_4)=(0,1,1,-)$. Hierdoor is $Z_1=Z_2=Z_3=y_1$ Wordt x_4 als variabele beschouwd en $Z_3=y_1=1$ dan $Z_4=\bar{x}_4$
Door x_4 van waarde te laten veranderen kan men kontroleren of er een SA-1 of SA-0 fout bij de x_4 input is opgetreden. (Geen fout als Z_4 ook van waarde verandert)

Door de inputs van de cascade de binaire waarden $(x_1,x_2,x_3,x_4)=(0,1,-,0)$ te geven en $y_1=Z_2=1$, is $Z_4=x_3$
Laat men de waarde van x_3 varieren dan kan men kontoleren of er een fout bij de x_3 input is opgetreden door de binaire waarden van Z_4 te meten.

Dezelfde procedure kan men herhalen voor de x_1 en x_2 inputs.

Thurber K.J. 1969 (lit.4) geeft een methode aan om de plaats van een defekte cel in een cascade te bepalen. De cascade bestaat hier uit cellen, waarvan de celfunkties alle funkties van twee variabelen kunnen zijn, met uitzondering van de volgende 4 celfunkties:

Z=0
Z=1
Z=x en $Z=\bar{x}$



Zijn methode dedekteert slechts één enkele cel in een cascade.

Kautz W.H. 1970 (lit.2) geeft een methode aan om defekte cellen in een cascade uit te schakelen. Treedt er in een cascade een defekte cel op, dan wordt deze cel van de cascade afgeschakeld en via een speciaal reserve netwerk wordt de informatie stroom langs de cel om geschakeld. De cascade mag bestaan uit unilaterale of bilaterale netwerken, full adder enz..

Urbano R.H. (lit. 7,8 en 9) geeft een methode aan hoe men de betrouwbaarheid van systemen willekeurig kan opvoeren m.b.v. speciale iteratieve netwerken.

Verdere literatuur op dit gebied:

Minnick R.C. Lit. 3
Kautz W.H. Lit. I

Literatuur opgave.

1 Kautz W.H. Testing for faults in combinational cellular logic array. In Proc. 8 th annual symp. on Switching and Automata theory. pp. I6i-I74 oktober I967.

2 Kautz W.H. Bypass switching for cellular cascades. IEEE Trans. on Comp. I970.

3 Minnick R.C. Cutpoint cellular logic. IEEE Trans. Elect. Comp. December I964.

4 Thurber K.J. Fault location in cellular arrays. Fall Joint Computer Conference I969. blz 8I.

5 Weg v.d. H. Voortgezette schakeltechniek. Diktaat ligt ter inzage bij afdelingsbibliotheek LCB.

6 Yau S.S. Fault diagnosis and repair of cutpoint cellular array. IEEE Trans. on Comp. March I970.

7 Urbano R.H. Matrix criteria for arbitrary reliability in iterated neural nets. IEEE Trans. on Elect. Comp. August I965 pp. 627.

8 Urbano R.H. On the convergence and ultimate reliability of iterated neural nets. IEEE Trans. Elect. Comp. June I964 pp. 204.

9 Urbano R.H. Some new results on the convergence, oscillation, and reliability of polyfunctional nets. IEEE Trans. on Elect. Comp. December I965 pp. 769.

Hoofdstuk 5

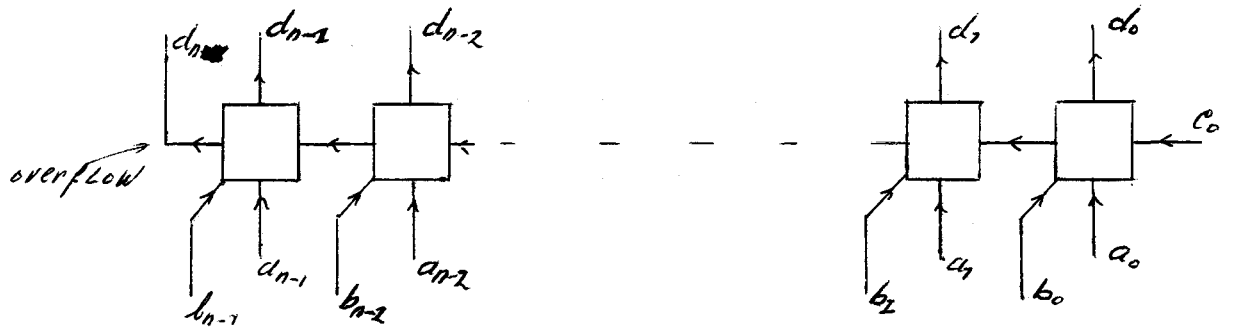
Rekenkundige elementen.

Samenvatting.

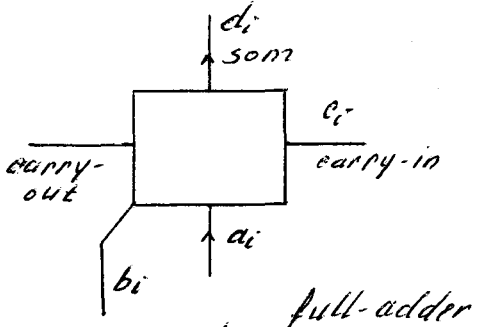
In dit hoofdstuk worden cellulaire netwerken behandeld, die in staat zijn om rekenkundige operaties uit te voeren. Na 1967 hebben deze netwerken een stormachtige ontwikkeling doorgemaakt. In sektie 5.1 wordt de optteller en aftrekker behandeld. Sektie 5.2 en 5.3 gaan respektievelijk over het vermenigvuldigen en het kwadrateren. Sektie 5.4 behandelt de deling en sektie 5.5 behandelt het worteltrekken (vierkantswortel). In sektie 5.6 worden netwerken behandeld, die de logarithme en de antilogarithme van een bepaald getal kunnen berekenen. In sektie 5.7 wordt de toepassing van drempelelementen in dit type cell. netwerken besproken.

5.1 Optellen en aftrekken.

Optellen: Een netwerk, dat een optelling realiseert, wordt in onderstaande figuur weergegeven.



Optteller



full-adder

cel

volgens boolse schrijfwijze

$$d_i = a_i \oplus b_i \oplus c_i$$

$$p_i = a_i b_i + c_i b_i + a_i c_i$$

$$a = a_{n-1}2^{n-1} \quad \ldots \quad a_0 2^0$$

$$b = b_{n-1}2^{n-1} \quad \ldots \quad b_0 2^0$$

$$d = a + b = d_{n-1}2^{n-1} \quad \ldots \quad d_0 2^0$$

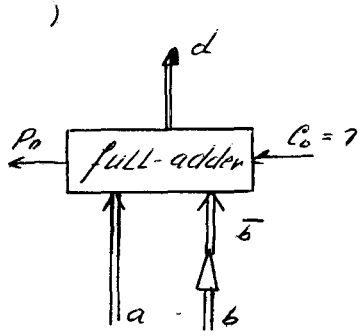
Aftrekken;

Stel d = a-b

Voor de zojuist behandelde opteller geldt $a+2^n = a$

$d = a+(2^n-b) = a+(2^n-1-b)+1$

2^n-1-b wordt verkregen door alle b_i's te inverteren. De aftrekker wordt dus verkregen door alle b ingangen van de opteller te inverteren en de carry-in op de waarde één te zetten. (zie onderstaande fig.)



$a < b$ dan $p_n = 0$ (leenbit)

$a > b$ dan $p_n = 1$ (geen leenbit)

Literatuur:

Hennie (lit. 6a)

White G. 1969 (lit 13) Ontwerp van een parallel B.C.D. opteller/aftrekker.

5.2 Vermenigvuldigen.

Stel $a = a_n 2^n \cdots a_0 2^0$ en $b = b_n 2^n \cdots b_0 2^0$

$$c = a \times b = a_n b_n 2^{2n} + \cdots \left(\sum_{L+K=i} a_L b_K\right) 2^i \cdots + a_0 b_0 2^0$$

Het netwerk van fig. 5.1 is een hard-ware uitvoering van een vermenigvuldiging. Zie Hoffmann J.C. (lit 7) 1968.

$$c_i = \sum_{L+K=i} a_L b_K$$

$L = 0, 1 \cdots n$

$K = 0, 1 \cdots n$

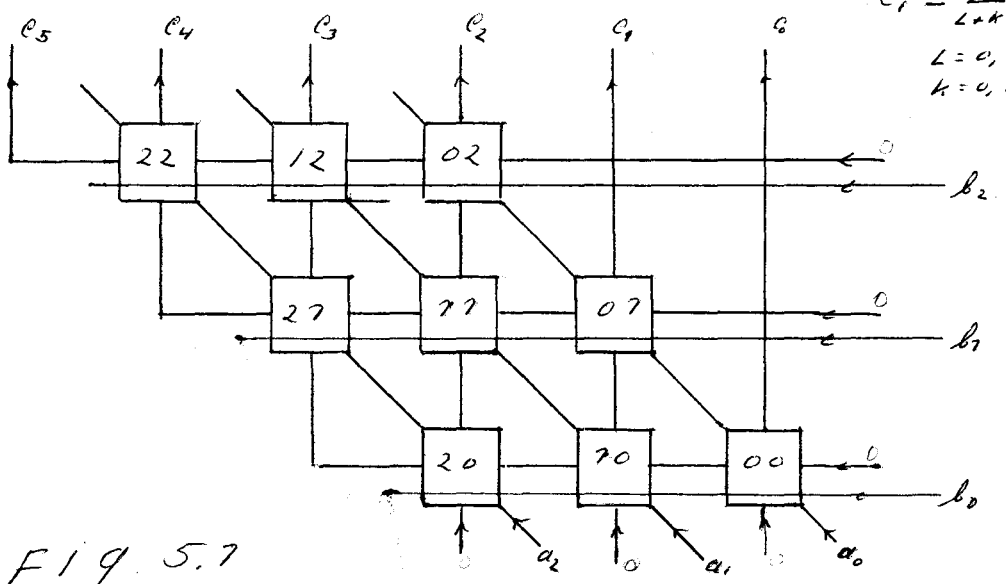
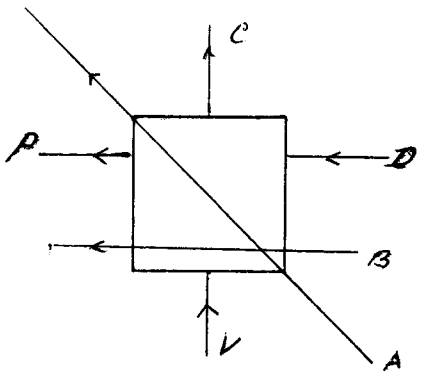
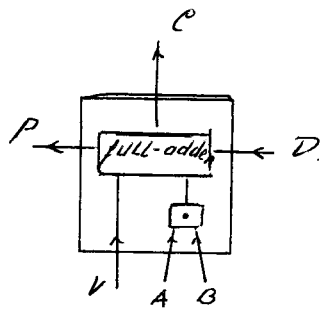


Fig 5.1



a



b

vlg. boolse schrijf wijze $\begin{cases} C = A.B \oplus V \oplus D \\ P = A.B.V + A.B.D + V.D \end{cases}$

Fig 5.2.

Fig. 5.2a geeft de cel met de betreffende cel-funkties weer. Fig. 5.2.b geeft dezelfde cel in enigszins gewijzigde vorm weer.
Bij de "soft-ware" uitvoering van een vermenigvuldiging, wordt de inhoud van de accumulator steeds een plaats opgeschoven en wordt er al of niet het vermenigvuldigtal (getal a) bij de inhoud opgeteld. Deze optelling is afhankelijk van de waarde van de betreffende bit van de vermenigvuldiger. Dit "schuif optel principe" , dat bij de soft-ware uitvoering wordt gebruikt, is in fig. 5.1 geheel in hard-ware uitgevoerd.
Het totaal aantal benodigde cellen is $n \times m$
a bestaat uit n bits
b bestaat uit m bits
De growth rate is n^2 voor m=n.
De max. vertraging is $(3n-2)\tau$ sec. voor n=m.
τ is de max. vertraging per cel.

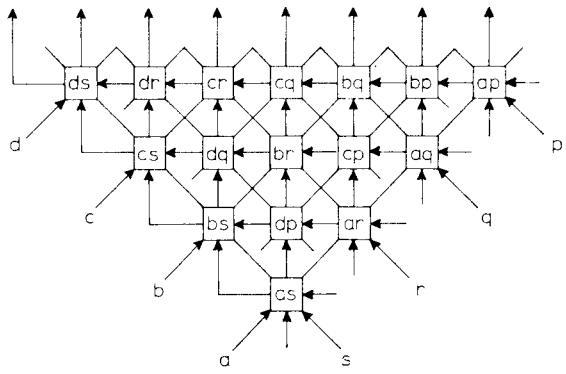
Guild H.H.1969 (lit. 6) geeft een ander ontwerp van een "full iterative multiplier" voor binaire vermenigvuldiging. "Full multiplier" betekent, dat de vermenigvuldiger ook als opteller gebruikt kan worden.

Stel $M = d2^3 + c2^2 + b2^1 + a2^0$

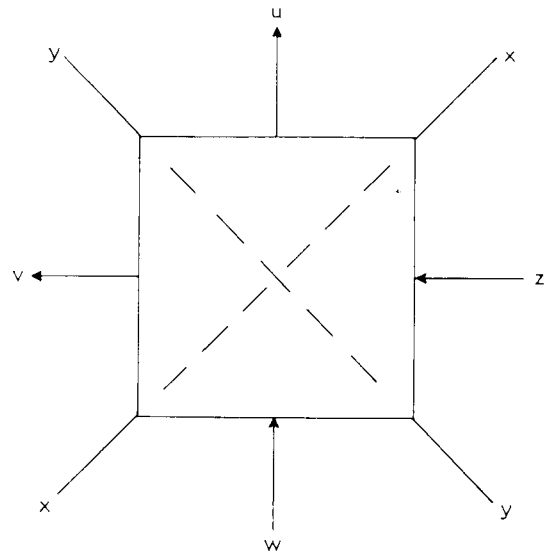
$N = s2^3 + r2^2 + q2^1 + p2^0$

Het produkt $M \times N = d.s\,2^6 + (dr + cs)2^5 + (dq + cr + bs)2^4 + (dp + cq + br + as)2^3 + (cp + bq + ar)2^2 + (bp + aq)2^2 + ap\,2^0$

Fig. 5.5 toont het cellulaire netwerk plus de cel, die daarin wordt toegepast.



4-bit fully iterative multiplying array
The partial product determined in each cell is shown



Basic cell for use in array

Fig. 5.5

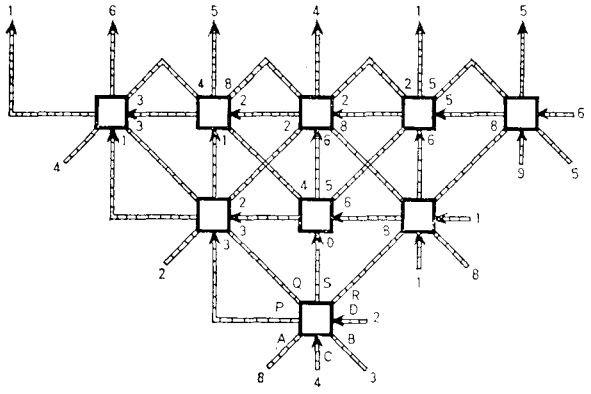
In de cellen zijn de deelprodukten neergeschreven, die door de cel geproduceerd worden.

De celfunkties zijn volgens boolse schrijfwijze $\begin{cases} u = xy \oplus w \oplus z \\ v = xyw + xyz + wz \end{cases}$ | z is carry-in, v is carry-out

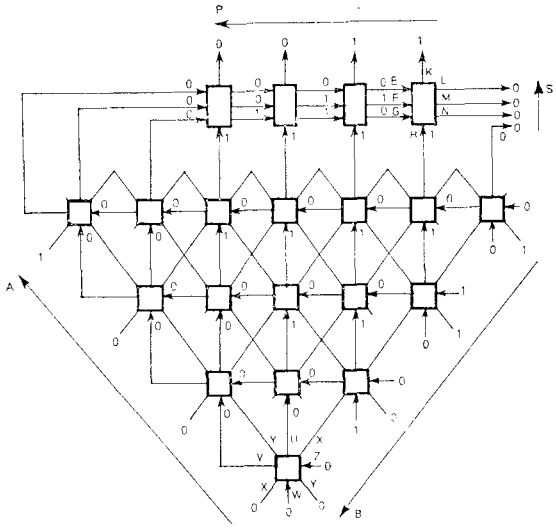
Deze cel heeft dezelfde struktuur als de cel in fig. 5.2b
Deze vermenigvuldiger is sneller dan, die in fig. 5.1 [redacted]. De max. vertraging is $(2n-1)\tau$ *sec.*
τ is de max. vetraging van een cel.

De Mori R. (lit. 10) heeft een vermenigvuldiger ontworpen, die dezelfde vertraging heeft als de vermenigvuldiger van H.H. Guild (fig.5.5). Zijn netwerk is niet volledig iteratief. Dit netwerk is een gewijzigde versie van de vermenigvuldiger van J.C. Hoffmann (zie fig. 5.1).

White G. 1970 (lit. 14) behandelt een B.C.D. vermenigvuldiger. Dit netwerk bestaat uit macro-cellen. (macro cell. logica) Fig. 5.6 laat deze vermenigvuldiger zien plus een detail van de inwendige macro-cel struktuur. De macro-cel bestaat uit een vermenigvuldiger, die identiek is aan die in fig. 5.5.



Cellular array for b.c.d. multiplication.



Cellular structure of macro-cell.

Fig. 5.6

p is de carry-out van de cel. Het produkt $A \times B$ wordt in de macro-cel door 10 gedeeld m.b.v. het unilaterale netwerk (zie hfst,2). Hieruit wordt de carry out afgeleid. De rest van deze deling verschijnt op de output S.

Frécon L. 1970 (lit. 6) behandelt de "floating - point multiplier". Hierin worden alle reeds bekende technieken over vermenigvuldigen en optellen toegepast. De schakeling levert geen nieuwe gezichts-punten op.

Dean K.J. 1971 (lit. 4) beschrijft een methode om een parallel vermenigvuldiger onder te verdelen in kleinere subnetwerken. Tevens ontwikkelde hij een algorithme, dat de propagatie tijd van zo'n netwerk, dat uit meerdere subnetwerken bestaat, berekent.
Uit de resultaten bleek dat de propagatie tijd zeer klein gehouden kon worden en in wezen niet veel verschilt van de hiervoor reeds behandelde netwerken.

5.3 Kwadrateren.

Speciaal voor kwadrateren kunnen nog eenvoudigere schakelingen ontworpen worden. Stel $a = a_3 2^3 + a_2 2^2 + a_1 2 + a_0 2^0$

$a^2 = a_3^2 2^6 + 2 a_3 a_2 2^5 + (2 a_3 a_1 + a_2^2) 2^4 + (2 a_3 a_0 + 2 a_2 a_1) 2^3 +$
$+ (2 a_2 a_0 + a_1^2) 2^2 + 2 a_1 a_0 2^1 + a_0^2 2^0$

Na herschrijving van het kwadraat:

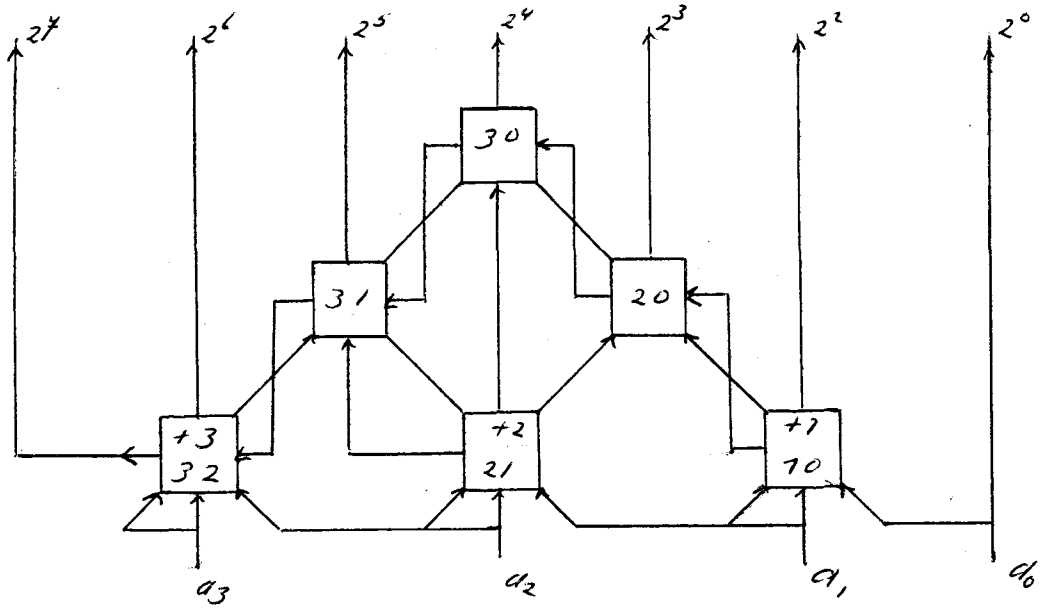
$a^2 = (a_3^2 + a_3 a_2) 2^6 + a_3 a_1 2^5 + (a_3 a_0 + a_2 a_1 + a_2^2) 2^4 + a_2 a_0 2^3 + (a_1^2 + a_1 a_0) 2^2 + a_0 2^0$

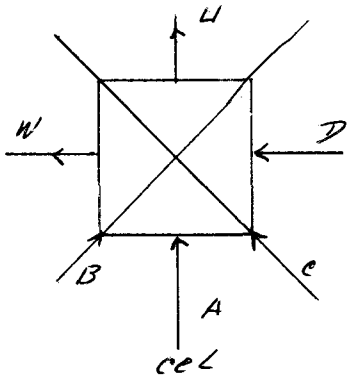
algemeen geldt dat de coëfficiënt voor 2^k gelijk is aan

a/ $\frac{1}{2} \left(\sum_{i+j=k-1} a_i a_j + 2 a_{k/2} \right)$ voor k is even en $k \geq 2$.

b/ $\frac{1}{2} \sum_{i+j=k-1} a_i a_j$ voor k is oneven en $k \geq 2$ en $i \neq j$.

Door deze rangschikking krijgt men het volgende cell. netwerk.

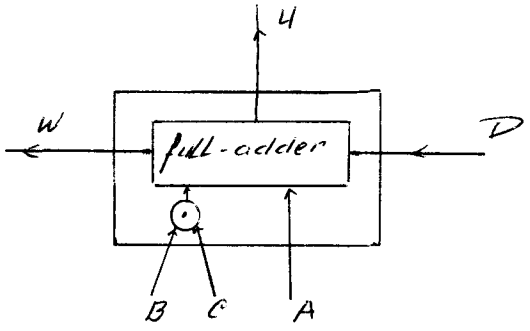




Volgens boolse schrijfwijze $U = A \otimes B C \otimes D$

$W = A B C + B C D + A D$

D is carry-in
W is carry-out



enigszins gewijzigde cel structuur
Celfuncties zijn hetzelfde gebleven.

Fig. 5.7

De verbindingstruktuur is niet regelmatig. (niet volledig iteratief cell. netwerk)
Het aantal benodigde cellen is $\frac{1}{2}n(n-1)$.
Max. vertraging $(n-1)\tau$ sec.

Literatuur
Weg v/d H. zie lit. 12

5.4 Delen.

Er wordt onderscheid gemaakt tussen
a/ herstellende deling
b/ niet herstellende deling.

a) herstellende deling.

voorbeeld: Stel a= I,00 } binaire getallen
b= I,01 }

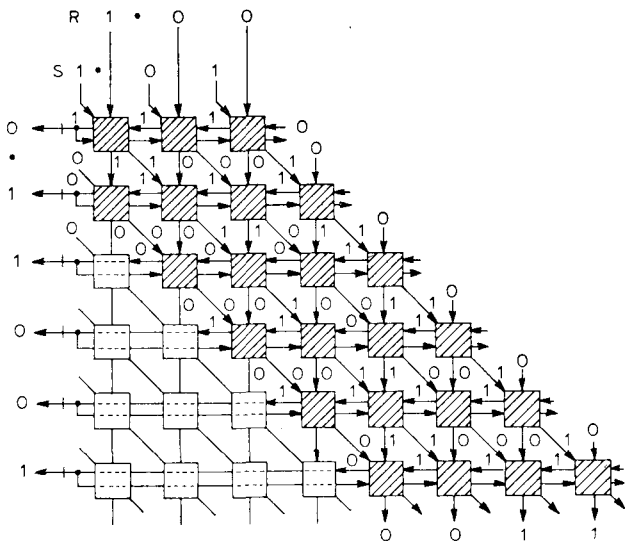
bereken a:b a wordt deeltal genoemd
b wordt deler genoemd

quotient

↓ 1.000000
0. - 1,01 ← ⟶ rest negatief aftrekking niet uitvoeren
1,000
1 - 101
00110
1 - 101
000010
0 101 ⟶ rest neg aftrekking niet uitvoeren
0,000010

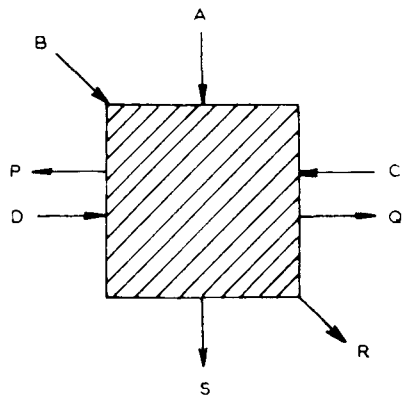
resultaat quotient $0,110 = \frac{3}{6}$
rest 0,000010

Uit bovenstaand voorbeeld blijkt dat eerst wordt nagegaan of de aftrekking uitgevoerd kan worden, daarna wordt de deler van de rest afgetrokken.
Fig. 5.8 laat een herstellende deler zien, afkomstig van K.J. Dean 1968 (lit. I).



Iterative array of cells connected as a divider

Note that the negation of the output from each row is required to obtain the quotient



Basic cell of an array for carrying out binary division

Boolse schrijfwijze

$$\begin{cases} P = \bar{A}B + \bar{A}C + BC \\ Q = D \\ R = B \\ S = AD + A\bar{B}\bar{C} + A\,BC + \bar{A}\,B\bar{C}\bar{D} + \bar{A}\,\bar{B}\,C\bar{D} \end{cases}$$

Fig 5.8

p is leen bit van de aftrekker.

De lijn D heeft de volgende funktie:
Als D = 0 dan S = A-(B+C) aftrekking wordt uitgevoerd
Als D = I dan S = A aftrekking wordt niet uitgevoerd
De max. vertraging van dit cell. netwerk is $\nu(n+I)\tau$ sec.
ν= aantal bits van het quotiënt
n= aantal bits van het deeltal

Het aantal benodigde cellen is $(n+I)\nu-I$

b) Niet herstellende deling.

Voorbeeld: a:b a= I,00
b= I,0I

quotient

```
                1,0000
              - 1,01
0,    leenbit 1 1,110   → rest neg
              + 101
1     carry out 1 0,0110  →  rest pos
              -  101
1     —           0,00010 → rest pos
              -   101
0     —  leenbit 1 1,111010 → rest neg.
              +    101
0     —           1,11111110 → rest neg
              +     101
1     —  carry-out 1 0,00000011
```

resultaat quotient 0,11001
rest 0,00000011

Bij niet hetstellende deling wordt altijd in de eerste deelcyclus afgetrokken. In bovenstaande voorbeeld wordt de deler van het deeltal afgetrokken. Is de rest positief, dan quotiënt bit wordt één. Is de rest negatief dan quotiënt bit wordt nul.
Als de rest pos. is, dan wordt bij de volgende cyclus weer afgetrokken.
Als de rest daarentegen negatief is, dan wordt bij de volgende deelcyclus een optelling uitgevoerd(zie bovenstaand voorbeeld).
Dit principe is op de volgende eigenschap gebaseerd:

Stel $A/B = a_4 2^4 + a_3 2^3 + a_2 2^2 + a_1 2 + a_0$ a_i is nul of één

dus $A = a_4 B 2^4 + a_3 B 2^3 \ldots a_0 B$ A is deeltal, B is deler

Bij de eerste deelcyclus wordt de rest $A - B2^4$ (Aftrekking uitgevoerd)

a/ als $A - B2^4 > 0$ dan $a_4 = 1$
Bij de volgende cyclus wordt $B2^3$ van de rest ($A - B2^4$) afgetrokken.

b/ Als $A - B2^4 < 0$ dan $a_4 = 0$
Bij de volgende cyclus wordt $B2^3$ bij de rest ($A - B2^4$) opgeteld. Dus $A - B2^4 + B2^3 = A - B2^3$

Men ziet dat door deze optelling de vorige aftrekking $A - B2^4$ geëlimineerd wordt.

Voor uitgebreidere informatie over de deling wordt verwezen naar het boek van Flores I. (lit. 5)
Fig. 5.9 laat een cell. netwerk zien plus een detail van de cel, die een niet herstellende deling uitvoert.
(Afkomstig van Majithia 1970 Lit.9).

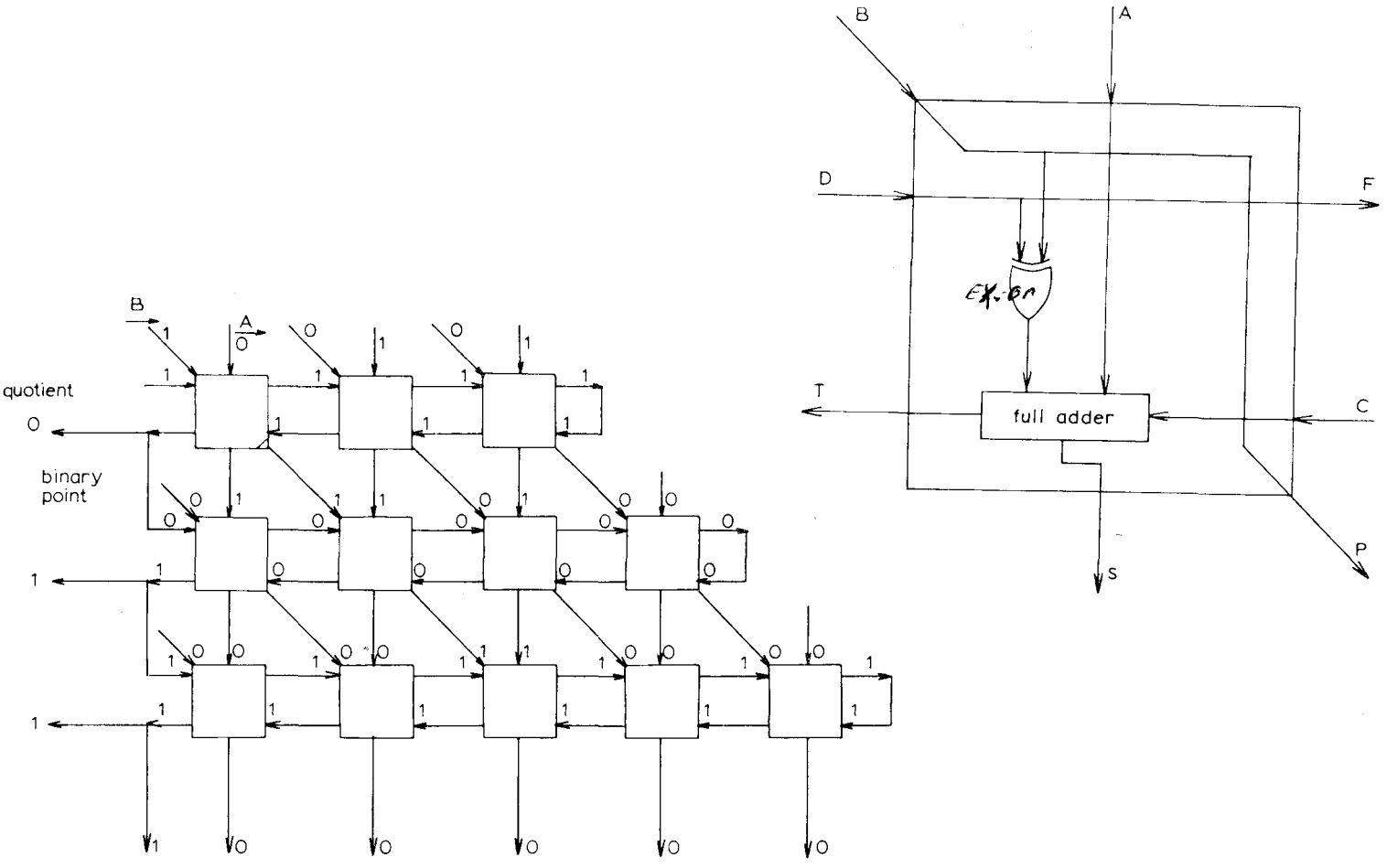


Fig. 5.9

Als de D input van de cel de waarde nul heeft, dan werkt de cel als opteller.
Als de D input van de cel de waarde één heeft, dan werkt hij als aftrekker, Deze aftrekker werkt volgens het principe behandeld in sektie 5.I.
Het netwerk en de cel zijn eenvoudig van struktuur.
z_a is de verzorging van de full-adder
z_y " " " " exclusive or

De totale vertraging van dit netwerk is $\frac{p(p+2n-3)}{2}t_a + pt_g$ sec.
p= aantal bits van het quotiënt , n= aantal bits van de deler.

P.A.W. Walker 1970 (lit.II) behandelt een asynchrone deler.
De gemiddelde vertraging van deze deler is zeer klein.
~~XXXXXXXXXX~~ Deze deler wordt gerealiseerd door een cell. netwerk, dat niet volledig iteratief is.

5.5 Vierkantswortel

Het bepalen van de vierkantswortel van een getal, is een iteratief proces. Dit proces bestaat uit al of niet herhaald optellen of aftrekken.
Flores I. (lit.5) geeft een uitgebreide verhandeling over een herstellende en niet herstellende algorithme van een vierkantswortel trekking.
H.H. Guild1970 (lit 6c) behandelt dat algorithme zeer kort en maakt het geschikt voor toepassing in cell. netwerken.
De cel, die H.H.Guild toepast in zijn cell. netwerk,is in fig. 5.10 afgebeeld.

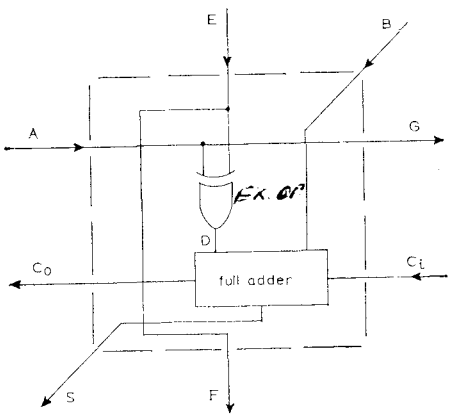


Fig. 10 *Basic cell*
Possible implementation of the required cell function is shown

Fig. 5.10

De cel heeft dezelfde struktuur als die van de deling.
Het cell. netwerk, bestaande uit deze cel is verreweg het eenvoudigste netwerk van wat tot nu toe bekend is. Het cell. netwerk is echter niet volledig iteratief.

De andere twee types cell. netwerken, die de vierkantswortel realiseren, zijn afkomstig van:

J.K. Dean 1968 (lit. 3a) Dit netwerk is niet volledig iteratief.

Devries R.C. en Chao M.H. (lit. 4a) 1970. Dit netwerk is volledig iteratief. De cel is komplex.

In fig. 5.II is een cel getekend, die kan worden toegepast in cell. netwerken, die de volgende rekenkundige operaties kunnen realiseren:
Optellen en aftrekken
vermenigvuldigen
delen
kwadrateren
worteltrekken

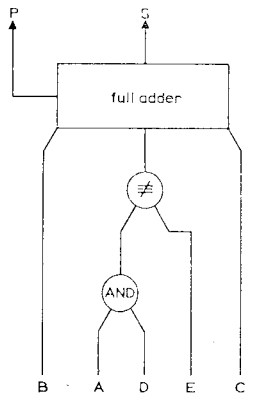


Fig. 5.II

Als vermenigvuldig- en kwadrateer-cel moet E=0 zijn. (zie sektie 5.2 en 5.3)
Als deelcel dan moet A=D zijn. (zie sektie 5.4)
Als vierkantswortelcel dan moet A=D zijn. (zie sektie 5.5)

5.6 Logarithme

De toepassing van logarithmische berekeningen is bij uitstek geschikt voor het berekenen van exponentiële funkties.
K.J. Dean 1969 (lit. 3) behandelt een algorithme, die antilog. en de logarithme van een getal berekent.
Daarin wordt $\log x$ berekent door een iteratief proces, dat bestaat uit herhaald kwadrateren.
Om de antilogarithme 2^y te berekenen wordt ook een iteratief proces uitgevoerd, waarin herhaaldelijk de bewerking worteltrekken voorkomt.

5.7 Toepassing drempelelementen.

S.L.Hurst beschrijft in zijn artikel enige toepassingen van drempelelementen in cell. netwerken. Hij beschrijft een ontwerp van een vermenigvuldig- en deel-cel, bestaande uit drempelelementen.
Daaruit bleek, dat de propagatie tijd van zo'n cel teruggebracht kan worden tot± 50% van de waarde, die optreedt bij cellen, die de nand of nor als bouwsteen hebben.

Konklusie.

De tot nu toe bekende cellulaire netwerken voor rekenkundige operaties zijn afgeleid van algorithmes, die via een iteratief proces (meestal optellen of aftrekken) deze rekenkundige bewerking realiseren.
Voor de berekening van de logarithme en antilogarithme bestaat dit proces uit resp. kwadrateren en worteltrekken.
De growth rate van deze netwerken is laag t.o.v. de cell. netwerken behandeld in hfd.3.
De cellen bestaan meestal uit een full/adder en/of exclusive-or en/of and-poort.

Literatuur opgave.

1 Dean K.J. Binary division using a data dependent iterative array. Electronics letters July 1968 vol.4 No 14 pp. 283-284.

2 Dean K. J. Design for a multiplier. Proc. IEE 1968, 115 pp. 1592 - 1594 .

3 Dean K.J. A fresh approach to logarithmic computation. Electronic Engineering April 1969 pp. 489.

3a Dean K. J. Cellular logical array for extracting square roots. 1968 Electronics letters July pp. 314-315.

4 Dean K.J. Cellular multiplier subarrays: A critical path approach to propagation times. Electronics letters February 1971 vol.7 No.3.

4a Devries R.C. and Chao M.H. Fully iterative array for extracting square roots.

5 Flores I. The logi of computer arithmetic. Prentice - Hall, INC. 1963.

6 Frécon L. Multiplicateur cellulaire parallèle de nombres en virgule flottante. Electronics letters April 1970 vol.6 pp.227.

6a Hennie F.C. Finite-State models for logical machines. New York: Wiley, 1968.

6b Guild H.H. Fully iterative fast array for binary multiplication and addition. Electronics letters June 1969 No.12.

6c Guild H.H. Cellular logical array for nonrestoring square-root extraction. Electronics letters Febr. 1970 pp.66-67.

7 Hoffmann J.C., Lacaze B. and Csillag P. Multiplieur parallèle à circuits logiques itératifs: Electronics letters 1968 p.178.

8 Hurst S.L. Realisation of cellular arithmetic cells by threshold logic assemblies. Electronics letters August 1970 pp.501-503.

9 Majithia J.C. Nonrestoring binary division using a cellular array. Electronic letters 1970 May pp.303

10 Mori de R. Suggestion for an i.c. fast parallel multiplier. Electronics letters 1969 ,5, pp.50-51.

11 Walker P.A.W. A synchronous binary dividing array. Electronics letters August 1970 vol.6 pp.515-517.

12 Weg v/d H. Voortgezette Schakeltechniek. College diktaat - Ter inzage bij afdelingsbibl. (Prof. Heetman)

13 White G. Design of a parallel b.c.d. adder/subtractor. Electronic Engineering 41, pp.229-230. February 1969.

14 White G. A cellular 8.4.2.1. b.c.d. multiplier. The radio and electronic engineer vol. 40,no.6 December 1970.

Hoofdstuk 6

"Special purpose" cellulaire netwerken en macro-cellulaire logica.

Special purpose cellulaire netwerken.

Sequentiële machines, codeernetwerken, schuifregister, geheugens, en "interconnection arrays, die m.b.v. cell. netwerken gerealiseerd worden, heb ik samengebracht onder de naam "special purpose" cell. netwerken.

De sequentiele machines kunnen met de reeds behandelde netwerken in hfd 3 gerealiseerd worden. Het aantal cellen waaruit de sequentiële machine bestaat is niet alleen afhankelijk van het aantal toestanden, in- en outputs van de machine, maar ook van de toestand toewijzing.
Literatuur over dit onderwerp:

Ferrari D. and Grasselli A. 1969 A cellular structure for sequential networks. IEEE Trans. Comput C-18, 947-956

Arnold T.F. Tan C.J. and Newborn M.N. 1970 Iteratively realized sequential circuits IEEE Trans. Comput C-19,54-66

A. Mukhopadhyay geeft in zijn boek een korte samenvatting van verschillende types "special purpose" cellulaire netwerken. Titel boek: Recent developments in switching theory
Uitgever : Academic press New York.

Macro cellulaire logica.

Ik beperkte mij tijdens de literatuur studie alleen tot het onderwerp micro-cellulaire logica, maar ben daarbij herhaaldelijk gestuit op onderwerpen die betrekking hebben op macro-cellulaire logica.
In onderstaande wil ik een kleine literatuur opgave geven over dit onderwerp.
Het zojuist genoemde boek van Mukhopadhyay geeft een korte samenvatting van de werken van S.H. Unger(lit. 4,5en6) en J.H. Holland (lit.1). Beide schrijvers houden zich bezig met de opbouw van een computer bestaande uit macro cellen.

J. von Neumann (lit. 3) introduceerde een macro cell, die in 29 toestanden gebracht kan worden. Met een groot aantal van deze cellen wordt een turing machine opgebouwd.

R.C. Minnick pp. 206 (lit 2) geeft een uitgebreide literatuurlijst over dit onderwerp.

Literatuur opgave:

I Holland J.H. 1959 A universal computer capable of executing an arbitrary number of subprograms simultaneously. In Proceedings Eastern Joint Computer Conference. pp. 113-118.

2 Minnick R.C. A survey of microcellular research. J.A.C.M. vol. 14 No.2 April 1967 pp.203-241.

3 Von Neumann J. 1966 "Theory of self-reproducing Automata" (Edited and completed by A.W. Burks) Univ. of Illinois Press., Urbana Illinois.

4 Unger S.H. 1958. A computer oriented towards spatial problems. Proc. IRE 46, 1744 - 1750

5 Unger S.H. 1959 Pattern detection and recognition. Proc. IRE 47, 1737 - 1752.

6 Unger S.H. 1962 Pattern recognition using two dimensional bilateral iterative combinational switching circuits. In Proceedings symposium on mathematical theory of automata. Polytechnic Institute of Brooklyn, New York.

Hoofdstuk 7

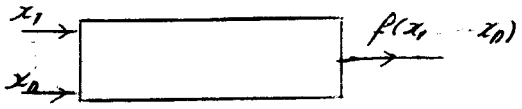
REDUNDANTIE m.b.v. VOTERNETWERKEN.

Samenvatting.

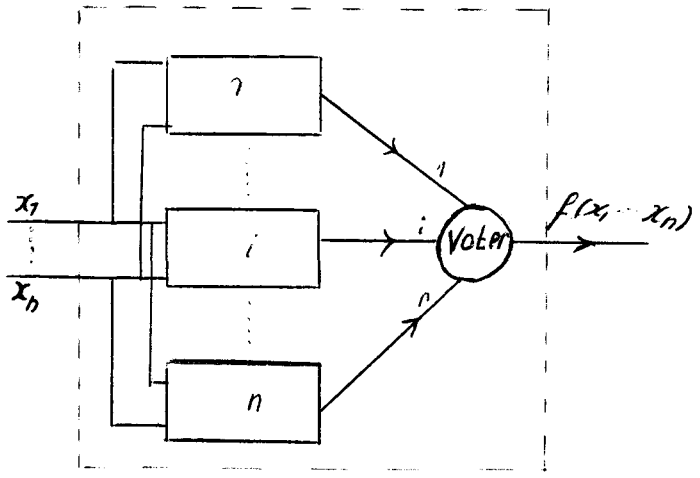
De opbrengst van het fabrikage proces van cellulaire netwerken is klein (b.v. 5%, dit betekent dat 5 van de 100 geproduceerde netwerken goed zijn). Om die reden wordt in hoofdstuk 7,8 en 9 geprobeerd cellulaire netwerken van het randgevoed-bus type (sektie 5.1) redundant te maken, zodanig dat de opbrengst van het fabrikage proces groter wordt.
Hierbij is gedacht aan aktieve redundantie d.w.z. er worden meer cellen in het netwerk aangebracht dan strikt noodzakelijk is voor de juiste werking van het netwerk.
Deze extra cellen zijn tegelijk met de niet redundante cellen in bedrijf (aktief) en worden zodanig aangebracht, dat bij het optreden van een beperkt aantal defekte cellen hun invloed "automatisch" opgeheven wordt en dat deze extra cellen de taak van de defekte cellen overnemen.
De meest bekende vorm van aktieve redundantie is de toepassing van voternetwerken (ook wel majority gates genoemd). Sektie 7.1 en 7.2 behandelen de toepassing van voternetwerken in cellulaire circuits.
In sektie 7.1 is getracht het randgevoed-bus type redundant te maken m.b.v. voternetwerken.
In sektie 7.2 wordt het hoofd- of kollektor netwerk van het randgevoed-bus type m.b.v. voternetwerken redundant gemaakt.

7.1 Redundantie van het randgevoed-bus type.

De voternetwerken zijn identiek aan de drempel elementen behandeld in sektie 5.1.B. De output van een voternetwerk is één resp. nul als de meerderheid van de inputs één resp. nul is.
Fig. 7.1 laat zien hoe een netwerk met één output redundant gemaakt kan worden. Het netwerk wordt n maal herhaald en aangesloten op een voternetwerk met n inputs.



niet redundant netwerk



redundant netwerk

Fig. 7.1

Definities:

Pc is de kans dat het niet redundante netwerk goed werkt na het produktie proces.

Pred is de kans dat het redundante netwerk goed werkt na het produktie proces.

Pvoter is de kans dat het voternetwerk goed werkt na het produktie proces.

Pred is de kans dat het redundante netwerk goed werkt na het produktie proces als het voternetwerk ideaal verondersteld wordt. (Pvoter=1)

Wat betreft fig. 7.1 is

$$Pred = \sum_{i=0}^{\frac{n-1}{2}} \binom{n}{i} (1 - Pc)^i . Pc^{n-i} \qquad n \text{ is oneven}$$

In nevenstaande grafiek wordt Pred als funktie van Pc weergegeven.
Men ziet dat er een opbrengst verbetering (Pred > Pc) kan optreden als Pc > 0.5. (Voter netwerk ideaal verondersteld)

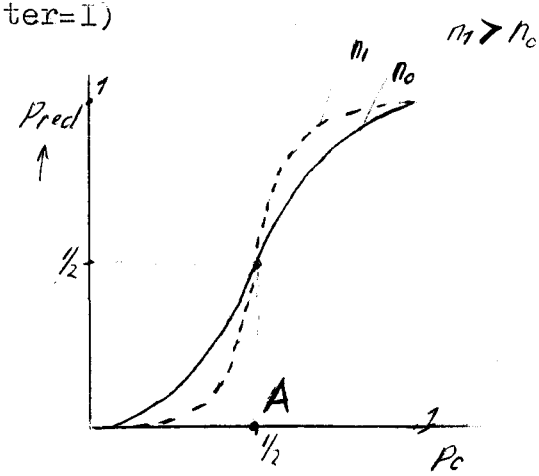


Fig. 7.2

Worden cellulaire netwerken met meerdere outputs m.b.v. de voternetwerken redundant gemaakt, dan verkrijgt men de schakeling in fig. 7.3

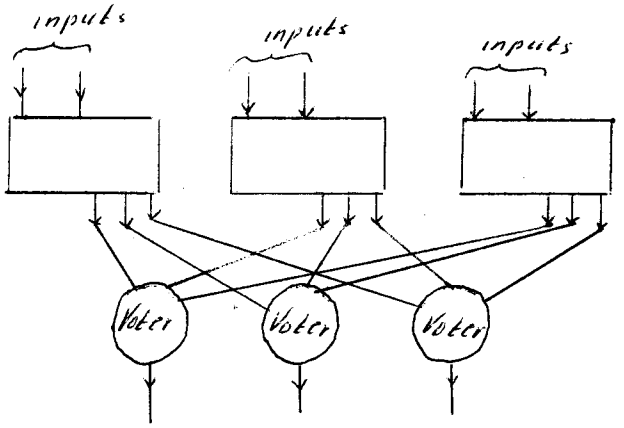


Fig. 7.3

Het cellulair netwerk wordt hier drie maal herhaald. Als eis wordt gesteld dat het voternetwerk uit dezelfde cellen bestaat als het cellulaire netwerk.
In appendix 10.I wordt weer gegeven hoe deze voternetwerken opgebouwd worden uit cellen van het randgevoed-bus type.
Fig. 10.3 geeft in detail het gehele cellulaire redundante netwerk weer.
In appendix 10.2 wordt Pred berekend van deze schakeling.

Er worden nog enige definities gegeven die straks gebruikt worden:

p is de kans dat elke cel goed werkt na het produktie proces.
n is het aantal inputs van het cell. netwerk.
m is het aantal cascades van het hoofdnetwerk.
N is het aantal outputs van het cell. netwerk.

Met behulp van een komputer programma wordt, door de grootheden m, n en N te varieren, nagegaan of er een opbrengstverbetering mogelijk is. Zie bijlage 1 en 2.

Voor p zijn twee verschillende waarden gekozen;

$p = p_1$ waarbij $p_1^{300} = 5\%$ (overeenkomend met een cell. netwerk bestaande uit 300 cellen en een opbrengst van 5%).

$p = p_2$ waarbij $p_2^{600} = 5\%$ (overeenkomend met een cell. netwerk van 600 cellen en een opbrengst van 5%).

Uit de resultaten van het komputer programma blijkt dat voor

$1 < N \leq 10$ en $m \leq 30$ geen opbrengst verbetering kan optreden.

Omdat bij toenemende N en m de verhouding $\frac{Pred}{Pc}$ volgens het komputer programma zeer snel afneemt en omdat

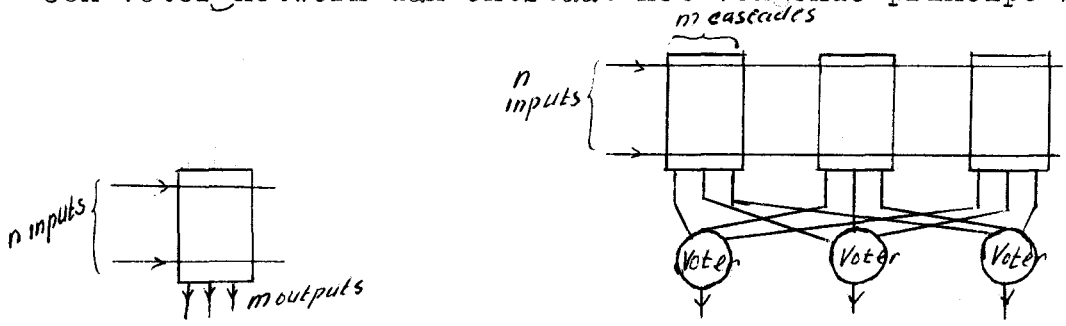
$\lim_{m \to \infty} \frac{pred}{pc} = 0$ voor ieder N en n

$\lim_{N \to \infty} \frac{pred}{pc} = 0$ voor iedere m en n

mag men verwachten dat voor $N > 10$ en voor $m > 30$ ook geen opbrengst verbetering zal optreden.

7.2 Voternetwerken toegepast op kollektor- of hoofd-netwerk van het randgevoed-bus type.

Wordt het kollektor- of hoofd-netwerk redundant gemaakt m.b.v. een voter netwerk dan ontstaat het volgende principe schema.



symbool hoofd netwerk redundant hoofd netwerk.

Fig. 7.4

Nu is volgens appendix 10.3

$$Pred = p^{2nm}\left(3 - 2p^{n}\right)^{m} \cdot p^{8m^2}$$

$$Pc = p^{nm}$$

met behulp van een komputer programma is ook hier, door de grootheden m en n te varieren, nagegaan of er een opbrengst verbetering mogelijk is bij twee verschillende waarden van p nl. p_I en p_2. (zie vorige bladzijde).

Bijlage 3 bevat het komputer programma met de reken resultaten. Het programma laat m varieren van I tot I0 en n wordt zo ver opgehoogd totdat punt A in fig. 7.2 is bereikt.
Voor $m < 3$ is er een opbrengst verbetering mogelijk. Voor $3 \leq m \leq 10$ is ook hier geen opbrengst verbetering mogelijk.
Omdat de komputer berekeningen aantonen dat $\frac{Pred}{Pc}$ zeer snel daalt bij toenemende m, en omdat

$\lim_{m \to \infty} \frac{pred}{Pc} = 0$ voor iedere positieve n

$\lim_{n \to \infty} \frac{pred}{Pc} = 0$ voor iedere positieve m

mag men ook hier verwachten dat voor m > 10 de verhouding

$\frac{P_{red}}{P_c}$ blijft dalen en dus ook geen opbrengst verbetering mogelijk is.

Bijlage 4 geeft Pred en Pc weer als fumktie van n bij m=4 en m=I als $p = p_1$.

Konklusie.

Tengevolge van de eis, dat het voternetwerk uitgevoerd moet worden m.b.v. dezelfde cellen en dezelfde verbindingstruktuur als het niet redundante cellulaire netwerk, wordt het voternetwerk op onefficiënte wijze opgebouwd uit deze cellen. Worst-case berekeningen tonen aan dat bij het redundant maken van het gehele cellulaire netwerk m.b.v. voter netwerken geen opbrengst verbetering op treedt voor netwerken met meer dan één output

Exakte berekeningen tonen aan dat het redundant maken van alleen het kollektor- of hoofd-netwerk m.b.v. voternetwerk geen opbrengst verbetering geeft voor netwerken met meer dan twee ouputs.

Literatuur opgave

I Shooman M.I. Probabalistic Reliability: An engineering approach. Mc. Graw-Hill, New York. 1968.

2 Wood P.E. Switching Theory. Lincoln Laboratory Publications. 1968.

3 Winograd S. Cowan J.D. Reliable Computation in the presence of noise. The M.I.T. Press, Massachusetts Institute of Technology Cambridge, Massachusetts. 1963.

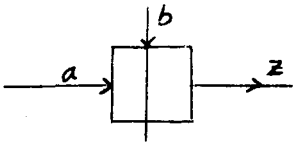
4 Wilcox and Mann Redundancy Techniques for computing systems 1962

Hoofdstuk 8

Single error correcting code en cell. netwerken.

Samenvatting.

Om cellulaire netwerken redundant te maken kan men lineaire kodes toepassen. Sektie 8.1 behandelt de single error correcting code met de daarbij behorende kodeer en dekodeer schakeling uitgevoerd in cellulaire logica.
Om het koderen mogelijk te maken kunnen alleen de kollektor netwerken van het q-funktie netwerk redundant gemaakt worden.
Er is gestreefd naar een kodeer en dekodeer schakeling, die dezelfde cellen en dezelfde struktuur heeft als het niet redundante netwerk.
Om de dekodeer schakeling te kunnen realiseren moesten cellen met $z = a.b$ en $z = a.\bar{b}$ als celfunktie toegelaten worden.
De regelmatige verbindingstruktuur bleef gehandhaafd.



8.1 De single error correcting code, kodeer en dekodeer schakeling.

Het koderen wordt veel toegepast in de telekommunikatie. De zender zendt gekodeerde berichten uit van m+k bits. Deze gekodeerde berichten worden kodewoorden genoemd. Een dergelijk kodewoord bestaat uit m informatie bits en k check bits. Deze check bits voegen een dusdanige hoeveelheid redundante informatie aan het bericht toe, dat bij het optreden van storingen, tijdens de transmissie weg tussen zender en ontvanger, het juiste bericht bij de ontvanger uit het ontvangen woord gedekodeerd kan worden.
De berichten, die hier worden beschouwd, bestaan uit een rij nullen en enen. De lengte van ieder bericht is m+k.
Er kunnen 2^{m+k} verschillende woorden ontvangen worden. Daarvan zijn 2^m woorden kodewoorden. Deze woorden worden ook wel vektoren genoemd.
Bij lineaire kode is de som van twee kode-vektoren weer een kode vektor. (Onder "som" wordt verstaan de modulo 2 som)

Voorbeeld	11001I	kodewoord 1
	$\oplus$ 010101	kodewoord 2
	100110	kodewoord 3

De hamming afstand tussen twee kodewoorden is het aantal posities waarin de twee kodewoorden verschillen.
De kodevektoren bevinden zich in een m+k dimensionale vektorruimte. Slaat men om iedere kodevektor een bol, waarvan de schil bestaat uit vektoren die een hamming afstand één verschillen met de kodevektor, dan mogen deze bollen elkaar niet overlappen bij een single error correcting code. Daarom moet de minimum hamming afstand tussen twee vektoren, die in twee verschillende bolschillen liggen, één zijn. Daaruit volgt dat de minimum hamming afstand tussen twee kodevektoren 3 moet zijn.
Er zijn 2^m kodevektoren en dus ook 2^m bollen. Iedere bol bevat m+k vektoren, die hamming afstand één verschillen met de kodevektor, plus de kodevektor zelf. Het totaal aantal vektoren nodig voor deze bolstapeling is $(m+k+1)x2^m$.
Het aantal beschikbare vektoren is 2^{m+k}.

Hieruit volgt $(m+k+1)x2^m \leq 2^{m+k}$

dus $m \leq 2^k - k - 1$.

Bij hamming code is $m = 2^k - k - 1$. Het aantal redundante bits in een kodewoord is dan minimaal.

Lineaire kodes zijn met behulp van een parity matrix te beschrijven. De theoretische achtergronden en de konstruktie van de parity matrix voor diverse soorten lineaire kodes zal hier niet behandeld worden, daarvoor wordt verwezen naar lit. 1 en 2 .
Aan de hand van een voorbeeld zal de konstruktie van een parity matrix van een single error correcting code beschreven worden.

Voorbeeld:
Stel het kodewoord bestaat uit 6 bits. Drie informatie bits en drie check bits.
De kodevektor ziet er dan als volgt uit ($\underbrace{M_1, M_2, M_3}_{\text{inf. bits}}, \underbrace{M_4, M_5, M_6}_{\text{check bits}}$)

We voegen de drie check bits zodanig toe dat

$M_4 = M_1 \oplus M_2$ (1)
$M_5 = M_1 \oplus M_3$ (2)
$M_6 = M_2 \oplus M_3$ (3)

Deze 3 vergelijkingen kan men ook schrijven als

$$
\begin{array}{lllll}
M_I \oplus M_2 & & M_4 & & =0 \quad (I)\\
M_I & \oplus M_3 & \oplus M_5 & & =0 \quad (2)\\
& M_2 \oplus M_3 & & \oplus M_6 & =0 \quad (3)
\end{array}
$$

of in matrix notatie

$$
\begin{bmatrix} 1 & I & 0 & 1 & 0 & 0 \\ 1 & 0 & I & 0 & 1 & 0 \\ 0 & 1 & I & 0 & 0 & 1 \end{bmatrix} \underline{M}^t = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}
$$

De vektor $\underline{M}$ stelt het kodewoord voor $\underline{M} = (M_I, M_2, M_3, M_4, M_5, M_6)$.

$$
\underline{M}^t = \begin{bmatrix} M_I \\ M_2 \\ M_3 \\ M_4 \\ M_5 \\ M_6 \end{bmatrix}
$$

De matrix wordt parity matrix H genoemd.
Een korte schrijfwijze van deze vergelijkingen is: $H\underline{M}^t = \underline{0}$

Treedt er nu tijdens de transmissie één enkele fout op (b.v. 3-de bit van $\underline{M}$ wisselt van waarde), dan is het ontvangen woord

$\underline{R} = \underline{M} \oplus \underline{E}$ de foutvektor $\underline{E}$ ziet er als volgt uit: $\underline{E}^t = \begin{bmatrix} 0 \\ 0 \\ I \\ 0 \\ 0 \\ 0 \end{bmatrix}$

Nu is $H\underline{R}^t = H(\underline{M}^t \oplus \underline{E}^t) = H\underline{M}^t \oplus H\underline{E}^t = \underline{0} \oplus H\underline{E}^t = H\underline{E}^t = S^t = \begin{bmatrix} 0 \\ I \\ 1 \end{bmatrix}$

$\underline{S} = (S_I, S_2, S_3)$ en wordt syndroom genoemd.
$\underline{S}$ is gelijk aan de derde kolom vektor van de matrix H. Het syndroom geeft dus de plaats van de fout in het ontvangen woord aan.
Is eenmaal de plaats van de fout bekend, dan kan men de opgetreden fout herstellen.

Het bovenstaande voorbeeld geeft een parity matrix van een single error correcting code met woord lengte 6.
Nu geldt voor iedere parity matrix van deze kode dat:

$H\underline{R}^t = H\underline{E}^t = \underline{S}^t$

$\underline{R} = \underline{M} \oplus \underline{E}$
$\underline{R}$ is het ontvangen woord.
$\underline{E}$ is de fout vektor.

Als $\underline{R} = \underline{M}$ dan is het syndroom nul. Er heeft dan geen fout plaats gevonden.
Als $\underline{R} \neq \underline{M}$ dan is er een fout opgetreden op b.v. de i-de plaats van het kodewoord. Het syndroom is nu gelijk aan de i-de kolom van de parity matrix.
Bij het optreden van twee of meerdere fouten is geen juiste dekodering mogelijk.

het aantal rijen van de parity matrix is gelijk aan het aantal kontrole bits k .
Het aantal kolommen van deze matrix is gelijk aan het aantal bits van het kodewoord (m + k kolommen) ,
De kolommen van de parity matrix moeten verschillend zijn en niet gelijk aan de nul vektor, wil men één fout kunnen korrigeren.
Het schoon vegen van de kolommen van de parity matrix is toegestaan.*

Het koderen van de check bits geschiedt alsvolgt:
Gegeven de parity matrix H
Door vegen kan men de matrix H in de volgende vorm krijgen

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} & 1 & & & \\ \vdots & & & & & 1 & & \\ a_{i1} & & & a_{im} & & & 1 & \\ \vdots & & & & & & & 1 \\ a_{m1} & & & a_{mm} & & & & \end{bmatrix} = H$$

Kodewoord
$(M_1, M_2, \ldots M_m, \ldots M_{m+k})$
inf. bits — check bits

Uit $H\underline{M}^t = \underline{0}$ blijkt dat $M_{m+i} = \sum_{j=1}^{m} \oplus a_{ij}M_j = a_{i1}M_1 \oplus \ldots\ldots \oplus a_{im}M_m$.

De hamming kode is niet toepasbaar op het gehele randgevoed bus type, omdat daar de kans op twee of meer fouten groter is dan de kans op een fout. Voor hoofd- resp. kollektor-netwerken geldt het omgekeerde, daar is de kans op één fout groter dan op meerdere fouten.
Omdat de check bit een exclusive or som is van de informatie bits is het niet mogelijk deze check bit met één enkele cascade te realiseren wanneer het hoofd- of kollektor-netwerk cellen bevat, die een andere funktie hebben dan de exclusive or of doorverbinding funktie. Ieder check bit moet dan m.b.v. een hoofd én kollektor netwerk bestaande uit meerdere cascades gerealiseerd worden, wat de regelmatige struktuur niet ten goede komt en tevens is de kans op meerdere fouten weer groter dan de kans op een fout.
We beperken ons daarom alleen tot een netwerk, dat samengesteld is uit cascades waarvan de cellen de exclusive or of een doorverbinding als celfunktie hebben. Het kollektor netwerk van het q-funktie netwerk voldoet er aan. (zie sektie 3.2.1).

* De rijvektoren van de parity matrix vormen een deelruimte van dim. k, die loodrecht staat op de deelruimte van kodevektoren van dim. m .

volgens laatste voorbeeld is $M_4 = M_1 \oplus M_2$
$M_5 = M_1 \oplus M_3$
$M_6 = M_2 \oplus M_3$

Het kodeer scheme ziet er als volgt uit :
(Verondersteld wordt dat $M_1 = x_1 \oplus x_2$
$M_2 = x_2 \oplus x_4$
$M_3 = x_1 \oplus x_3$)

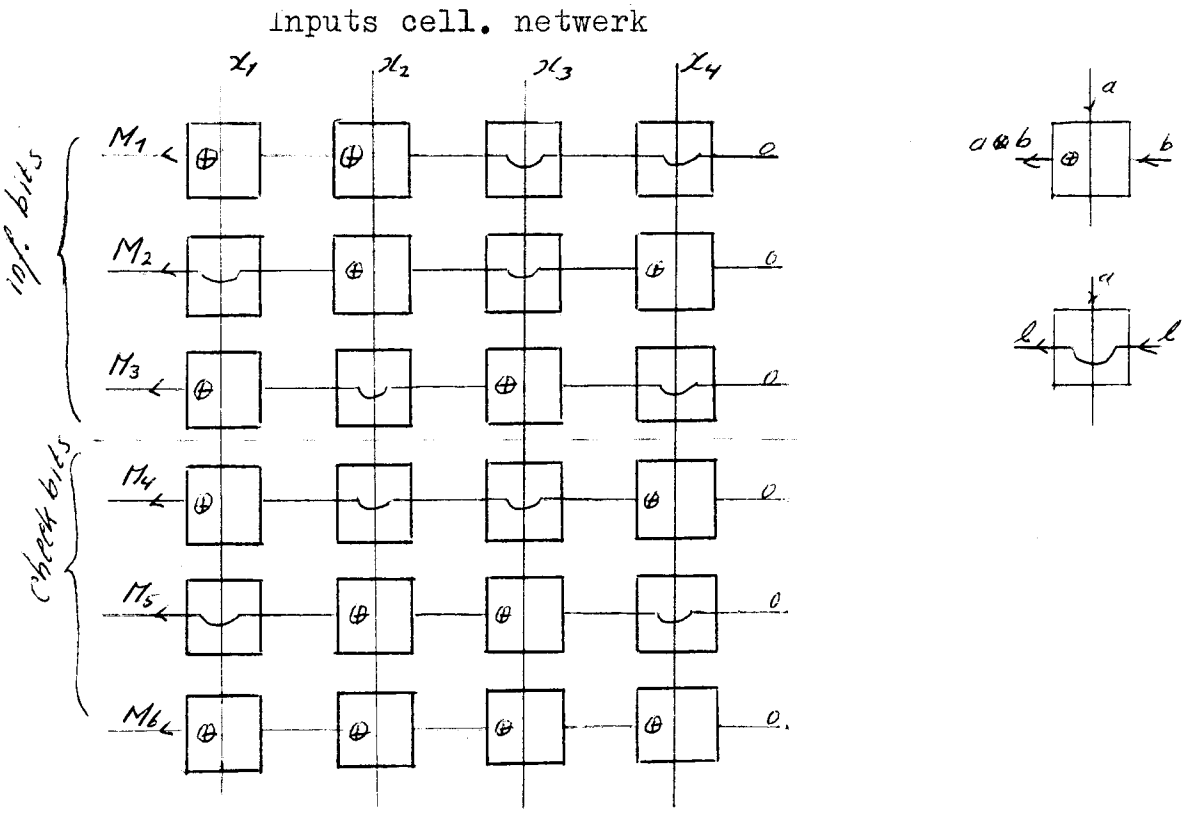
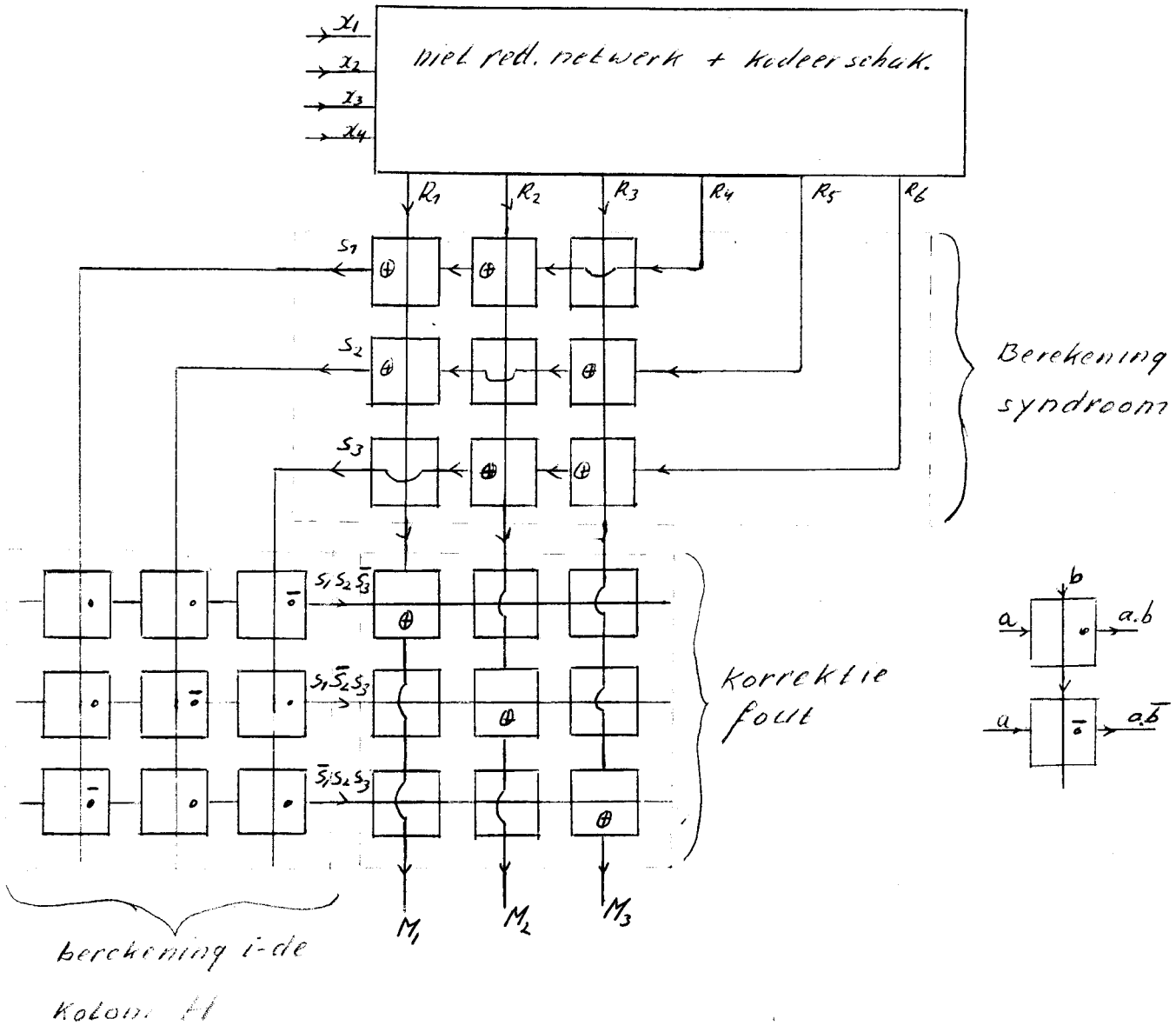


Fig. 8.1

Na het fabrikage proces kan één van de outputs defekt zijn. Door een juiste dekodering kan één fout in de 6 outputs gecorrigeerd worden.

Dekodeer schakeling.
Het syndroom $\underline{S}$ wordt in de dekodeer schakeling berekend en daaruit kan men het juiste bericht dekoderen.
Als voorbeeld is de parity check matrix van blz 89 genomen.
In fig. 8.2 wordt het gehele redundante netwerk weergegeven.



Redundant netwerk.(toepassing hamming kode)

Fig. 8.2

8.2 Berekening opbrengst verbetering.

Verondersteld wordt dat alle cellen toegepast in fig. 8.2 een zelfde fout kans bezitten, die gelijk is aan(1 - p). Uit appendix 10.4 volgt dat voor een redundant netwerk met m informatie bits en k check bits geldt:

$$Pi = p^n$$

$$Pred = \left((m + k)(1 - Pi)Pi^{m+k-1} + Pi^{m+k}\right) Pdek .$$

$$Pdek = p^{2mk+m^2}.$$

$$Pc = p^{nm}.$$

Pred is de kans dat het redundante netwerk van fig. 8.2 goed werkt.
Pdek is de kans dat de dekodeerschakeling goed werkt.
Pc is de kans dat het niet redundante netwerk goed werkt.
Pi is de kans dat de i-de output van het gekodeerde kollektor netwerk goed werkt.
n is het aantal inputs van het niet redundante netwerk.

Met behulp van een komputer programma wordt nagegaan of er een opbrengst verbetering mogelijk is, door m, n en k te varieren. (Bylage 5).
Er wordt steeds voldaan aan de vergelijking $m \leq 2^k - k - 1$.
Voor p worden twee waarden genomen nl:

$p = p_1$ EN voor p_1 geldt $p_1^{300} = 5\%$

$p = p_2$ en voor p_2 geldt $p_2^{600} = 5\%$

zie blz

Bijlage 6 geeft Pred en Pc in een grafiek weer, waarbij m= 3 en 4 genomen wordt.

Het blijkt dat voor $m \leq 4$ een opbrengst verbetering aanwezig is. Bij voternetwerken moest m kleiner of gelijk aan 2 zijn om een opbrengst verbetering mogelijk te maken.

Konklusie.

Door toepassing van single error correcting codes is een op brengst verbetering aanwezig voor cell. netwerken met minder dan 5 outputs.
Voor m=2 en $p=p_1$ geeft de toepassing van voternetwerken een grotere opbrengst verbetering.
Voor m=1 en p is resp. p_1 en p_2 geeft de toepassing van single error correcting codes een grotere opbrengst verbetering. De schakeling is in het laatste geval onregelmatiger dan bij voter-netwerken.

Literatuur opgave.

1 Peterson W.W. Error correcting codes. Cambridge, Mass., M.I.T. press. 1961.

2 Berlekamp E.R. Algebraic coding theory. Mc. Graw-Hill, Inc. New York. 1968.

3 Boddeke R.J.J. Foutendetektie en foutenkorrektie met behulp van cyclische kodes. Afstudeer verslag ECB. 1966.

Hoofdstuk 9.

DOUBLE ERROR CORRECTING BCH CODES EN CELLULAIRE NETWERKEN.

Samenvatting.

De BCH codes zijn lineaire kodes, die een generalisatie zijn van de hamming code. De woord lengte van deze BCH codes is gelijk aan 2^k-1. k is een geheel getal. Deze kode kan t fouten korrigeren. Er zijn dan k.t check bits in ieder kodewoord vereist. De BCH code is evenals de hamming code m.b.v. een parity matrix te beschrijven. In sektie 9.1 wordt de konstruktie van het dekodeer algorithme voor BCH code aan de hand van een voorbeeld behandeld. De behandeling van de theoretische achtergrond zou te ver voeren en daarvoor wordt verwezen naar lit. 1 en 2 . Het ontwerp van de detail schakelingen voor de dekodering geschiedt in sektie 9.2.

9.1 Double error correcting BCH codes.

Het koderen gebeurt op dezelfde wijze als bij de hamming code. Het dekoderen van deze BCH codes is echter ingewikkelder.

Alvorens met de behandeling van double error correcting codes te beginnen wordt een andere schrijfwijze van de parity matrix van de hamming code behandeld.
De kolommen van de parity matrix kunnen worden beschouwd als de niet nul elementen van een galois lichaam $GF(2^k)$, omdat de betreffende kolommen van de k matrix een k dimensionale vektorruimte vormen, bestaande uit 2^k-1 vektoren ongelijk nul, die modulo 2 opgeteld kunnen worden. (k= aantal check bits bij hamming code).
Bijlage 7 geeft de representatie van de multiplicatieve groep vzn een $GF(2^4)$ als machten van α weer en tevens ook de additieve groep, die wordt voorgesteld door polynomen en vektoren over de 4 dim. vektorruimte over GF(2).
Als voorbeeld wordt een hamming code genomen met woord lengte 15. De parity matrix ziet er dan als volgt uit:

$$H = \begin{bmatrix} 0&0&0&1&0&0&1&1&0&1&0&1&1&1&1 \\ 0&0&1&0&0&1&1&0&1&0&1&1&1&1&0 \\ 0&1&0&0&1&1&0&1&0&1&1&1&1&0&0 \\ 1&0&0&0&1&0&0&1&1&0&1&0&1&1&1 \end{bmatrix}$$

en $H\underline{R}^t = \underline{S}^t$ $\qquad \underline{S} = (s_1, s_2, s_3, s_4)$ $\qquad \underline{R}$ is het ontvangen woord.

De voorstelling van de kolommen van deze matrix H als elementen van $GF(2^4)$ ziet er als volgt uit:

$$H = \left[1 \mid \alpha \mid \alpha^2 \mid \alpha^3 \mid \alpha^4 \mid \alpha^5 \mid \alpha^6 \mid \alpha^7 \mid \alpha^8 \mid \alpha^9 \mid \alpha^{10} \mid \alpha^{11} \mid \alpha^{12} \mid \alpha^{13} \mid \alpha^{14} \right]$$

$H\underline{R}^t = \sum_{i=0}^{14} R_i \alpha^i = S(\alpha)$

α is een primitief element in $GF(2^4)$.
Daardoor is α een wortel van het
* primitieve polynoom $x^4 + x + 1$

Dus $\sum_{i=0}^{14} R_i \alpha^i = g(\alpha)(\alpha^4 + \alpha + 1) + S(\alpha)$

$= g(\alpha) \cdot 0 + S(\alpha) = S(\alpha)$

$g(\alpha)$ is een willekeurige polynoom.
Graad van $S(\alpha)$ is kleiner of gelijk 3.
$S(\alpha) \equiv \sum_{i=0}^{14} R_i \alpha^i \bmod (\alpha^4 + \alpha + 1)$.
$\underline{S}$ is de vektor notatie van polynoom $S(\alpha)$

Voorbeeld:

Stel $\underline{R}$ = (000000000100000)

In vektor notatie is $H\underline{R}^t = \underline{S}^t = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \Longrightarrow$ 10-de plaats is fout.

In $GF(2^4)$ voorstelling is

$H\underline{R}^t = \sum_{i=0}^{14} R_i \alpha^i = \alpha^9 = \alpha^3 + \alpha = S(\alpha) = (1010) \Longrightarrow$ 10-de plaats fout.

Bij double error correcting codes moet het aantal check bits worden uitgebreid. Bij woord lengte $2^k - 1$ is het aantal check bits 2k. Dit betekent dat bij woord lengte 15 het aantal check bits 8 en het aantal informatie bits 7 is.
De matrix H' ziet er dan als volgt uit:

$$H' = \begin{pmatrix} H'_1 \\ H'_3 \end{pmatrix} = \left[\begin{array}{c|c|c|c|c|c|c} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \cdots & \alpha^{13} \;\; \alpha^{14} \\ f(1) & f(\alpha) & f(\alpha^2) & f(\alpha^3) & f(\alpha^4) & \cdots & f(\alpha^{13}) \;\; f(\alpha^{14}) \end{array}\right]$$

H'_1 is de parity matrix van de hamming code.

H'_3 bestaat uit k rijen, die extra aan de matrix H' zijn toegevoegd.

Voor $f(\alpha^i)$ wordt genomen $f(\alpha^i) = (\alpha^i)^3$

* In dit hoofdstuk betekent een + teken een modulo 2 optelling, tenzij het uitdrukkelijk anders wordt vermeld.

De matrix H' wordt dan

$$H' = \begin{bmatrix} H'_1 \\ H'_3 \end{bmatrix} = \left[\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c|c|c|c} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 & \alpha^8 & \alpha^9 & \alpha^{10} & \alpha^{11} & \alpha^{12} & \alpha^{13} & \alpha^{14} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} \end{array}\right]$$

of

$$H' = \begin{bmatrix} H'_1 \\ H'_3 \end{bmatrix} = \left[\begin{array}{ccccccccccccccc} 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ \hline 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

Stel het ontvangen woord $\underline{R}$ bevat twee fouten. De bits op de v-de en j-de plaats zijn verwisseld.* $\underline{R} = \underline{M} + \underline{E}$ De fout vektor $\underline{E}$ heeft overal nullen behalve op de v-de en j-de plaats.

Dan geldt $H'\underline{R}^t = H'(\underline{M}^t + \underline{E}^t) = H'\underline{E}^t = \underline{S}^t = \begin{bmatrix} \underline{S}_1^t \\ \underline{S}_3^t \end{bmatrix}$

$\underline{S}_1^t = H'_1 \underline{E}^t$

$\underline{S}_3^t = H'_3 \underline{E}^t$

$\underline{S}_1 = (S_{11}, S_{12}, S_{13}, S_{14})$

$\underline{S}_3 = (S_{31}, S_{32}, S_{33}, S_{34})$

Volgens galois lichaam voorstelling

$$S_I(\alpha) = H'_1 \underline{E}^t = \sum_{i=0}^{14} E_i \alpha^i = \alpha^v + \alpha^j$$

$$S_3(\alpha) = H'_3 \underline{E}^t = \sum_{i=0}^{14} E_i \alpha^i = (\alpha^v)^3 + (\alpha^j)^3$$

Waarbij $\underline{S}_1$ de vektor voorstelling is van $S_I(\alpha)$.

Waarbij $\underline{S}_3$ de vektor voorstelling is van $S_3(\alpha)$.

De dekodeerschakeling kan het sydroom $\underline{S}$ berekenen.(zelfde type schakeling als bij hamming code) Hierdoor is de vektor notatie van de polynomen $S_1(\alpha)$ en $S_3(\alpha)$ bekend.
Er rest ons nog de volgende vergelijkingen op te lossen:

* $\underline{E} = (E_0, E_1, \ldots\ldots E_{14})$

$S_1(\alpha) = \alpha^{v} + \alpha^{l}$ (I)

$S_3(\alpha) = (\alpha^{v})^3 + (\alpha^{l})^3$ (2)

met α^{v} en α^{l} als onbekenden (α^{v} en α^{l} geven de plaatsen van de fouten aan).

In appendix 10.5 zijn de vergelijkingen (I) en (2) omgezet in een 2de graadsvergelijking

$$\sigma(z) = z^2 + \sigma_1 z + \sigma_2 = 0 \qquad \sigma_1 = S_1(\alpha)$$

$$\sigma_2 = S_1^3(\alpha) + \frac{S_3(\alpha)}{S_1(\alpha)}$$

α^{v} en α^{l} zijn de wortels van deze 2-de graadsvergelijking

Het dekodeer algorithme is nu :

1) Bereken het sydroom $\underline{S}^t = H\underline{R}^t$ Bij nul fouten is $S_1(\alpha) = S_3(\alpha) = 0$

bij één fout is $S_3(\alpha) = \{S_1(\alpha)\}^3$ $S_1(\alpha) \neq 0$

bij twee fouten is $S_1(\alpha) \neq 0$ $S_3(\alpha) \neq 0$

2) Bereken σ_2 Deze term levert bij het logische ontwerp de meeste moeilijkheden op. (zie sektie 9.2)

3) Los $\sigma(z)$ op. Het oplossen van $\sigma(z)$ wordt gerealiseerd door voor z alle elementen van $GF(2^4)$ te substitueren. Te beginnen bij $z = I$, daarna $z = \alpha$, $z = \alpha^2$ ....enz.
De waarden van z, waarvoor $\sigma(z)$ is nul, geven de plaats van de fouten aan.

In bovenstaande is de dekodering van een primitieve BCH code behandeld (Primitieve BCH code is een BCH code met woord lengte $2^k - 1$. k is een geheel getal).
Voor nonprimitieve BCH codes bestaat het dekoderingsalgorithme ook uit het oplossen van een 2-de graadsvergelijking. De problemen van het logische ontwerp zijn van dezelfde aard als bij de primitieve kode, nl. het delen van twee polynomen modulo $p(\alpha)$. $p(\alpha)$ is het primitieve polynoom van een galois lichaam.

9.2 Ontwerp parallel dekodeerschakeling m.b.v. cellulaire logica.

Een regelmatige verbindingsstruktuur en de toepassing van steeds dezelfde cellen vormen de karakteristieke eigenschappen van cellulaire logica. Er wordt getracht om het ontwerp van de dekodeerschakeling zoveel mogelijk aan deze twee eigenschappen te laten voldoen. De dekodeerschakeling wordt aan de hand van een voorbeeld (double error correcting BCH code met woord lengte 15) behandeld. Deze schakeling is gemakkelijk uit te breiden tot kodes met grotere woord lengte.

Het logisch ontwerp voor de berekening van het syndroom is gelijk aan die van de hamming codes. De toegepaste celfunktie is een exclusive-or of een doorverbinding.

9.2 A Berekening $\sigma_2(\alpha) = \sigma_2'(\alpha) + S_1^2(\alpha)$

$$\sigma_2(\alpha) = \frac{S_3(\alpha)}{S_1(\alpha)} + S_1^2(\alpha)$$

$\sigma_2(\alpha), S_2(\alpha), S_1(\alpha)$ zijn elementen van een $GF(2^4)$.

Daardoor kan $\sigma_2(\alpha)$ ook als volgt geschreven worden

$$\sigma_2'(\alpha) = \frac{S_3(\alpha) \bmod p(\alpha)}{S_1(\alpha) \bmod p(\alpha)}$$

$p(\alpha)$ is primitieve polynoom van $GF(2^4)$.

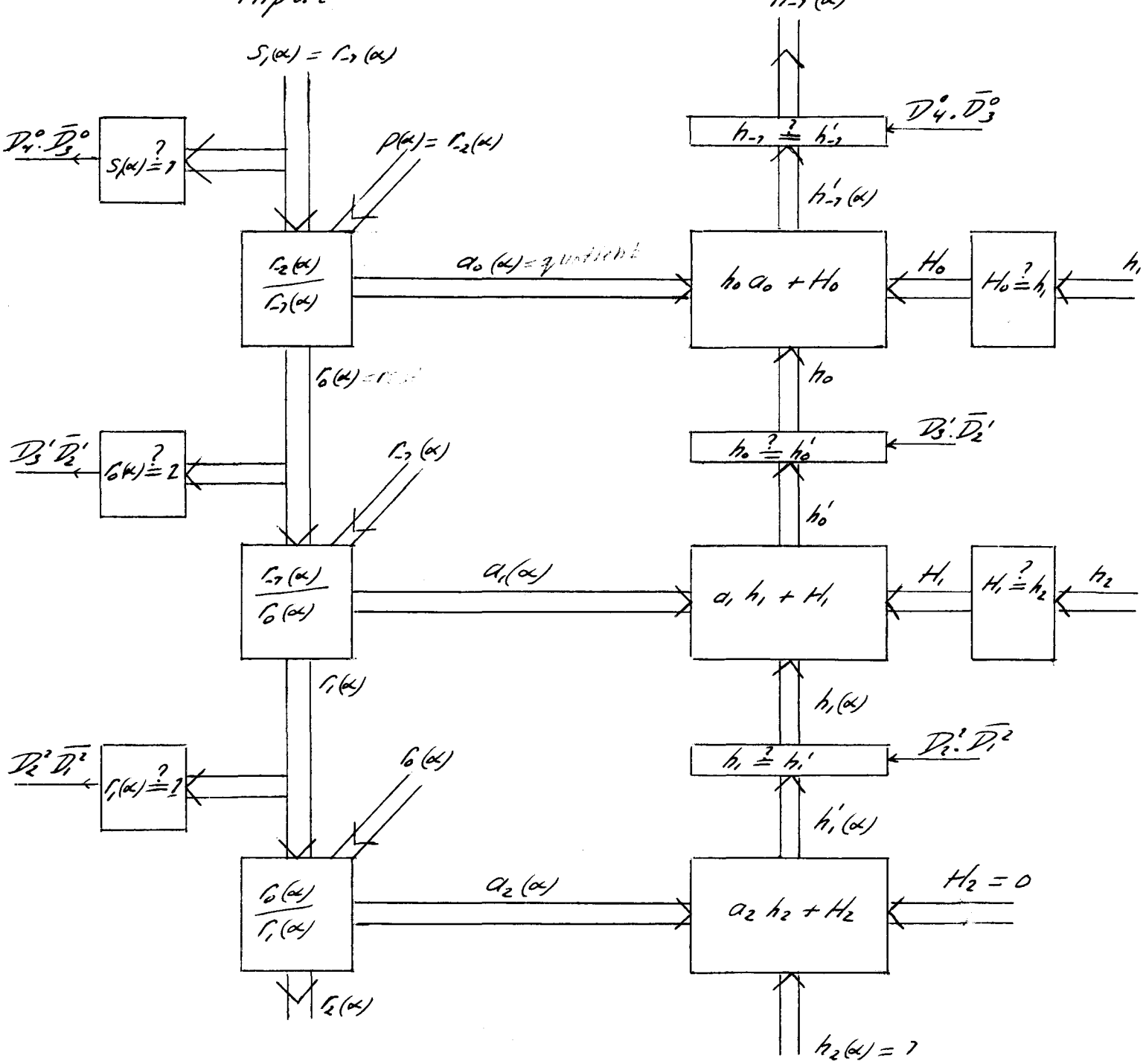
Stel $h_{-1}(\alpha)$ is de inverse van $S_1(\alpha)$

dan $\sigma_2'(\alpha) = h_{-1}(\alpha) S_3(\alpha) \bmod p(\alpha)$.

De deling is omgezet in een vermenigvuldiging. Deze vermenigvuldiging is gemakkelijk te realiseren.

Berekening inverse $h_{-1}(\alpha)$ van $S_1(\alpha)$.

Voor de berekening van de inverse van $S_1(\alpha)$ wordt gebruik gemaakt van de algorithme van Euclid. De procedure voor de berekening van deze inverse wordt behandeld in lit. 1 blz. 23 en 36. Het blokschema voor de berekening van de inverse van $S_1(\alpha)$ wordt in fig. 9.2 weergegeven.
Stel $p(\alpha) = r_{-2}(\alpha)$ en $S_1(\alpha) = r_{-1}(\alpha)$.
Het quotiënt van de eerste deling $\frac{r_{-2}(\alpha)}{r_{-1}(\alpha)}$ is gelijk aan $a_0(\alpha)$
de rest is $r_0(\alpha)$ (graad $r_0(\alpha) <$ graad $r_{-1}(\alpha) \leq 3$).



Blokschema van berekening inverse van $S_1(\alpha)$

Fig 9.2.

als $r_0(\alpha) = 1$ dan $h_0 = 1$ en $H_0 = 0$
De volgende delingen hebben dan geen invloed meer op de output $h_{-1}(\alpha)$.

als $r_0(\alpha) \neq 1$ dan wordt $h_0 = h_0'$ en $H_0 = h_1$
De volgende deling wordt uitgevoerd.

Het quotiënt van $\frac{r_{-1}(\alpha)}{r_0(\alpha)}$ wordt $a_1(\alpha)$ genoemd en de rest is $r_1(\alpha)$ graad $r_1(\alpha) <$ graad $r_0(\alpha) \leq 2$

als $r_1(\alpha) = 1$ dan wordt $h_1 = 1$ en $H_1 = 0$
De volgende delingen hebben geen invloed meer op de output $h_{-1}(\alpha)$.

als $r_1(\alpha) \neq 1$ dan wordt $h_1 = h_1'$ en $H_1 = h_2$

De laatste deling $\frac{r_0(\alpha)}{r_1(\alpha)}$ wordt op dezelfde wijze als de vorige delingen uitgevoerd.

De output $h_{-1}(\alpha)$ is uit de a_i's te berekenen met behulp van netwerken, die het produkt en de som van twee polynomen kunnen berekenen.

Voorbeeld:

stel $S_1(\alpha) = r_{-1}(\alpha) = \alpha^2 + 1$ en $p(\alpha) = r_{-2}(\alpha) = \alpha^4 + \alpha + 1$.

1-ste deling

$$\begin{array}{r|l} \alpha^2+1 & \alpha^4+\alpha+1 \\ \alpha^4+\alpha^2 & \alpha^2+1 \\ \hline \alpha^2+\alpha+1 & \\ \alpha^2+1 & \\ \hline \alpha & \end{array}$$

$a_0 = \alpha^2 + 1$

$r_0 = \alpha$

2-de deling

$$\begin{array}{r|l} \alpha & \alpha^2+1 \\ \alpha^2 & \\ \hline 1 & \end{array}$$

$a_1(\alpha) = \alpha$

$r_1(\alpha) = 1 \Rightarrow h_1 = 1 \quad H_1 = 0$

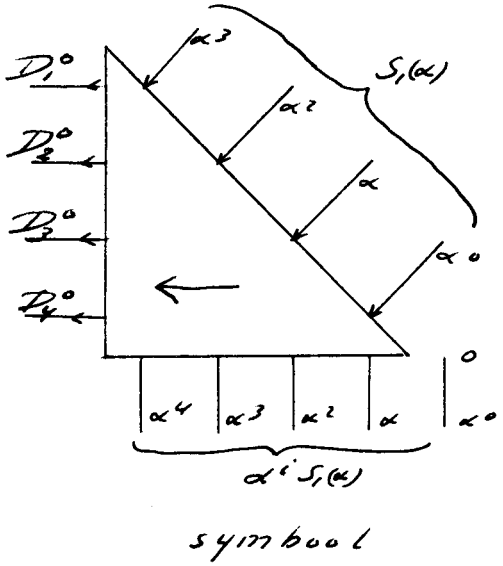
dan is $h_0 = a_1 h_1 + H_1 = a_1 = \alpha$

en $h_{-1} = h_0 a_0 + H_0 = \alpha(\alpha^2+1) + 1 = \alpha^3 + \alpha + 1.$

Het blokschema van fig. 9.2 is gemakkelijk uit te breiden voor polynomen gehorende tot $GF(2^k)$ $(k > 3)$. Er zijn dan $k-1$ netwerken nodig die een deling uitvoeren en $k-1$ netwerken die een vermenigvuldiging uitvoeren.

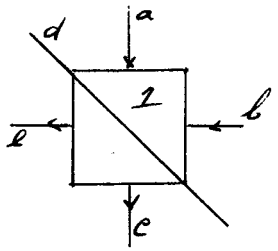
Logisch ontwerp deler.

Uit laatste voorbeeld blijkt dat delen bestaat uit herhaald schuiven en aftrekken.
De graad van $S_1(\alpha)$ is onbekend. Daarom wordt $S_1(\alpha)$ zodanig met α^i vermenigvuldigd, dat $\alpha^i . S_1(\alpha)$ een vierde graads-polynoom is.
$\alpha^i . S_1(\alpha)$ kan worden gerealiseerd door de coëfficiënten van de polynoom $S_1(\alpha)$ i plaatsen naar links te verschuiven. Fig. 9.3 geeft het symbool weer van dit verschuivingsnetwerk.
Appendix 10.6 behandelt de detail schakeling van dit netwerk.



symbool

toegepaste cel



$c = d(a+b)$
$e = \bar{d}(a+b)$

boolse schrijf-wijze.

cel 1

verschuivings netwerk

Fig. 9.3

De D_i^0 lijnen bevatten informatie over de graad van $S_1(\alpha)$.

$\alpha^i . S_1(\alpha)$ wordt tijdens de deling enige malen van $p(\alpha)$ afgetrokken, waarbij $\alpha^i . S_1(\alpha)$ steeds een plaats naar rechts verschoven wordt. Het symbool van het netwerk dat deze aftrekking uitvoert wordt in fig. 9.4 weergegeven. Appendix 10.7 behandelt de details van deze schakeling.

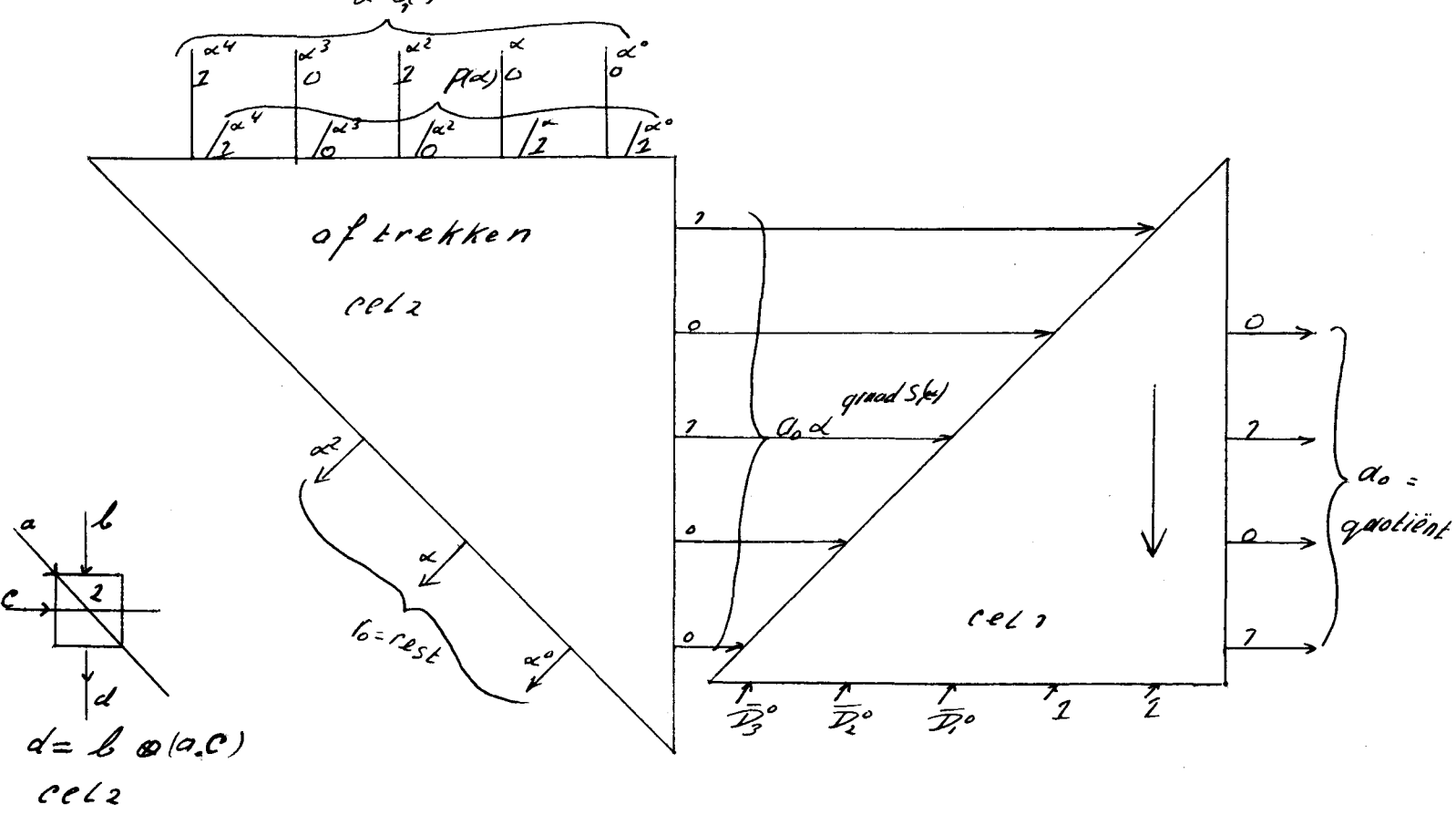


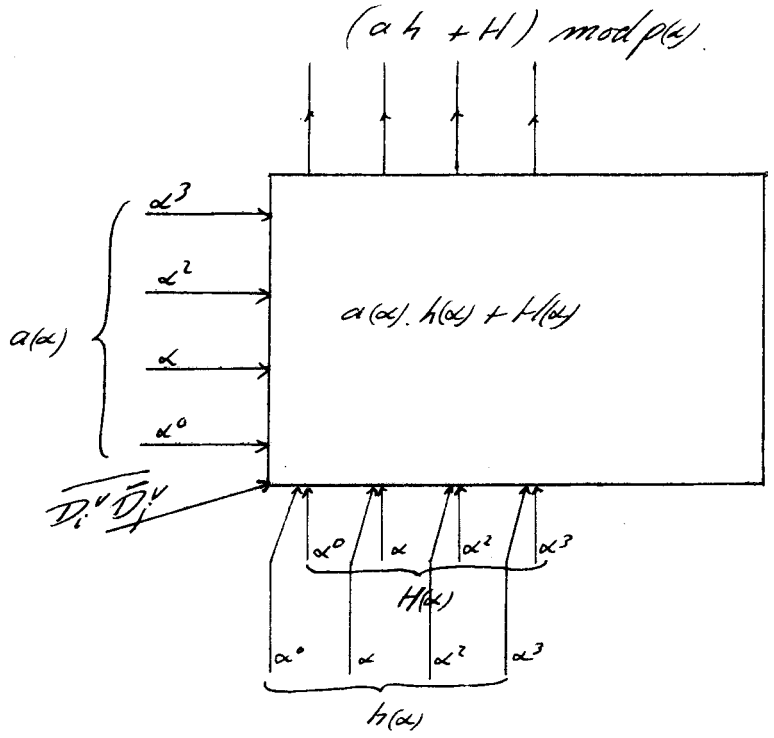
fig 9.4.

Het eigenlijke aftrekken en schuiven vindt plaats in het linker netwerk. Het rechter netwerk dient om het quotiënt in de juiste graad aan de outputs te laten verschijnen.

Ontwerp vermenigvuldigings-netwerk.

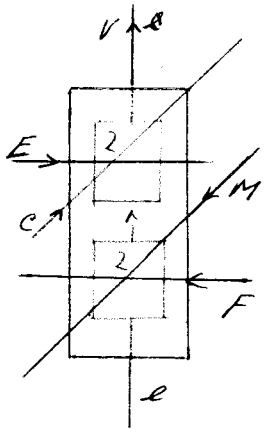
Het symbool van het netwerk, dat een vermenigvuldiging uitvoert, wordt in fig. 9.5 weergegeven.
Detail schakeling zie appendix 10.8. Het gehele netwerk bestaat uit cellen van het type cel 3 . Deze cel 3 bestaat uit twee deelcellen, die ieder identiek zijn aan cel 2 .
De input $D_i'' . \overline{D_j''}$ dient om de polynoom $H(\alpha)$ te beinvloeden. Als $D_i'' . \overline{D_j''} = 1$ dan wordt $H(\alpha)$ op nul gezet. Als $D_i'' . \overline{D_j''} = 0$ dan wordt $H(\alpha)$ onverandert aan het vermenigvuldigingsnetwerk toegevoerd.

Fig. 9.6 geeft de gehele logische schakeling weer, die de inverse van de polynoom $S_1(\alpha)$ berekent. $S_1(\alpha)$ is een element van $GF(2^4)$.



vermenigvuldigingsnetwerk.

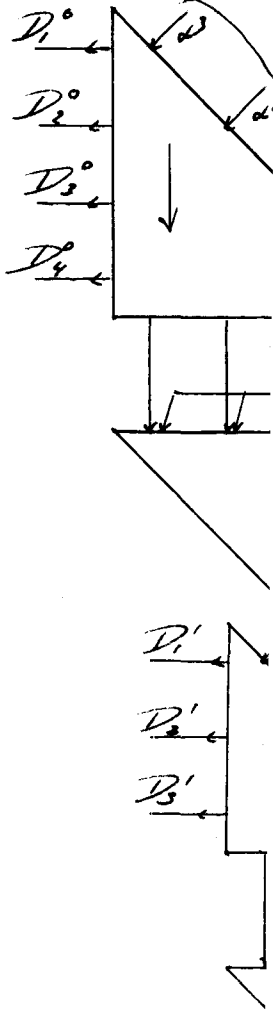
fig. 9.5.



$V = \ell \otimes (E.C) \otimes (M.F)$

$\ell \in L_3$

σ_2 is nu gemakkelijk te realiseren. (nl. $\sigma_2 = h_{-1}(\alpha).S_3(\alpha) \bmod p(\alpha) + S_1^3(\alpha) \bmod p(\alpha)$
σ_2 kan met de reeds behandelde vermenigvuldiger in fig. 9.5 gerealiseerd worden.


D_1°
D_2°
D_3°
D_4°

D_1'
D_2'
D_3'

9.2 B Het oplossen van $\sigma(z) = z^2 + \sigma_1 z + \sigma_2 = 0$

Fig. 9.7 geeft een blokschema voor het oplossen van $\sigma(z) = 0$

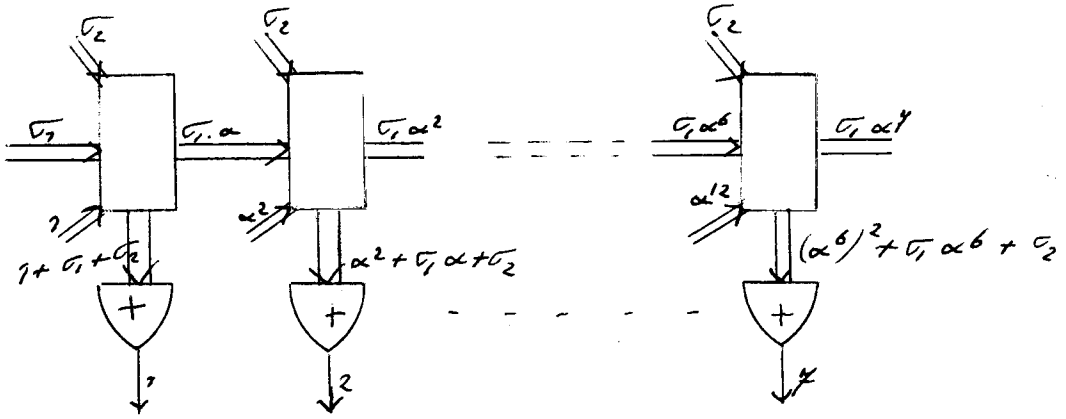


Fig. 9.7

Als de i-de output van het bovenstaande netwerk de waarde nul heeft, dan is er op de i-de plaats in het ontvangen kodewoord een fout opgetreden. Omdat het kodewoord 7 informatie bits bevat, is het voldoende alleen deze 7 bits te korrigeren. Deze informatie bits staan op de eerste 7 plaatsen van het ontvangen woord. Appendix 10.9 geeft een gedetailleerd schema van fig. 9.7.

Als de plaatsen van de fouten bekend zijn dan wordt de korrektie op dezelfde wijze uitgevoerd als bij de hamming code. Dit foutkorrektie netwerk bestaat uit cellen die een exclusive-or of een doorverbinding als celfunktie hebben.

Fig. 9.8 geeft de gehele dekodeer schakeling weer.

9.3 Berekening Pred.

De foutkans van de cel is evenredig met de oppervlakte. In de dekodeerschakeling worden 4 celtypes toegepast, die qua komplexiteit afwijken van de exclusive-or cel waarvan de foutkans $q = 1 - p$ is.
Om Pdek te berekenen, worden de foutkansen van de 4 celtypes afgeleid van de foutkans van deze exclusive-or cel.

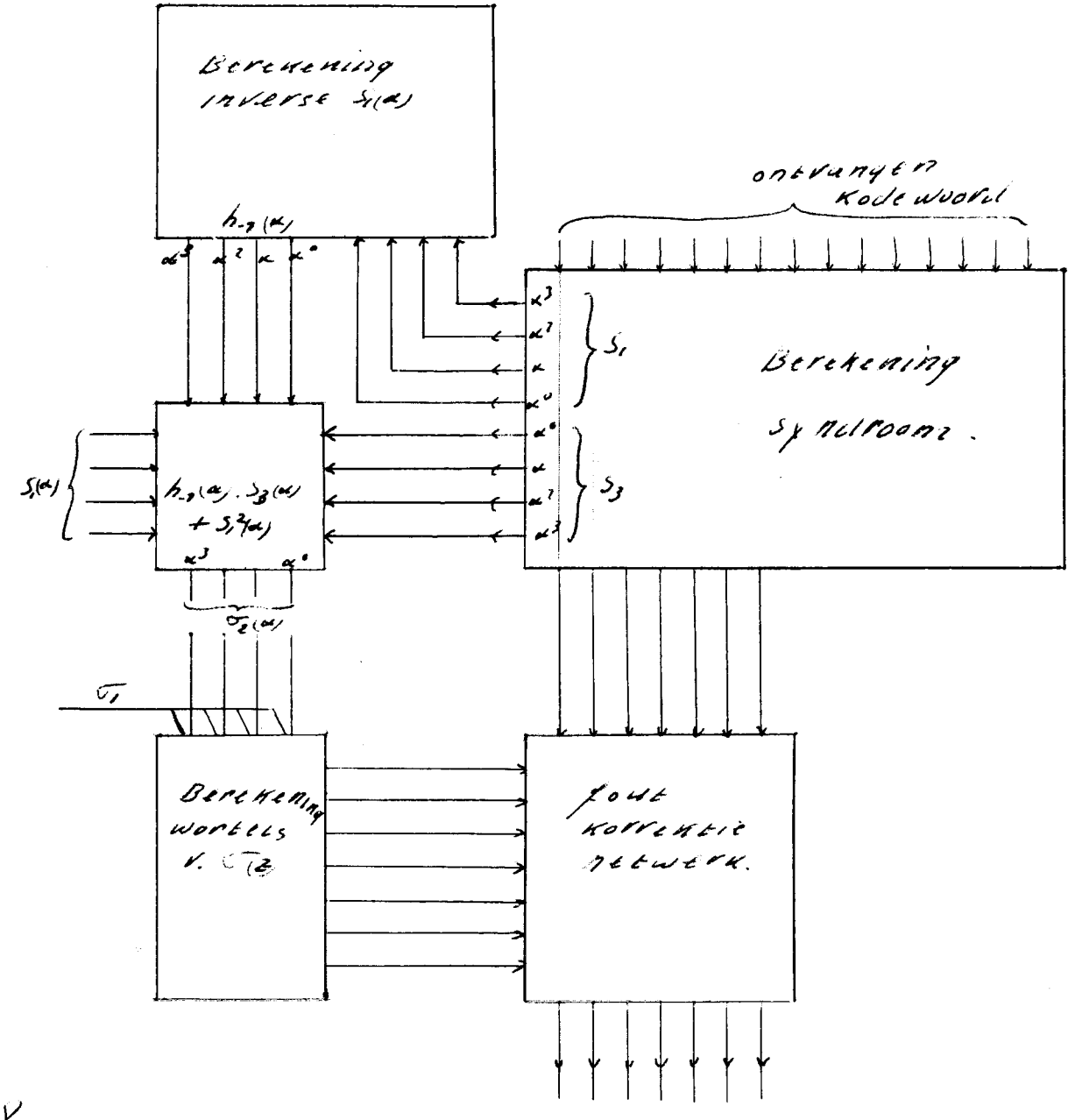


fig 9.5

komplete dekodeer schakeling

In onderstaande tabel worden de diverse foutkansen weergegeven.

cel type	aantal poorten	aantal oppervl. eenheden	foutkans
exclusive-or	2 and 1 or 2 invertors	5	q=I-p
cel type 1	I or 2 and 1 invertor	4	(4/5)q
cel type 2	1 and I exclusive-or en een extra diagonaal verbinding	6+I	(7/5)q
celtype 3	2 maal cel type 2	I4	(I4/5)q
cel type 4	2 exclusive-or I or en een extra diagonaal verbinding	1I+I	(I2/5)q

Het aantal cellen in de dekodeerschakeling is:

Onderdeel	aantal cellen bij woord lengte 2^k-I	
Syndroom	$s= 2k(2^k-2k-1)$	exclusive-or's
Inverse berekening $S_I(\alpha)$:		
a/ Delers;	$ia1= \sum_{i=2}^{4} (i^2+2i+1)$	cel type I
	$ib1= \sum_{i=2}^{4} (\frac{1}{2}i^2+(5/2)i+2)$	cel type 2
b/ Vermenig.:	$i2=\left(\sum_{a} a^2\right)+ (k-2-j)k^2$	cel type 3
	$a=2,2^2,2^3,.. 2^j$ waarbij $2^j \leq k < 2^{j+1}$	
Berekening van $b_1(\alpha) S_3(\alpha) + S_1(\alpha)^2$:	$b= 2k^2$	cel type 3
Oplossen van $z^2 + \sigma_1 z + \sigma_2 = 0$:	$0_1=(2^k-2k-1)k$	cel type 4
	$0_2=2^k-2k-1$	excl. or
	$f=(2^k-2k-1)^2$	excl. or
Fout korrektie netwerk:		

Nu wordt:

$$Pdek = p^{s} \times (I-(4/5)q)^{ia1} \times (I-(7/5)q)^{ibI} \times (I-(I4/5)q)^{i2} \times (I-(I4/5)q)^{b} \times (I-(I2/5)q)^{0I} \times p^{02} \times p^{f}.$$

Als $pi=p^{n}$ dan wordt:

$$Pred = (pi^{2k-I} + (2^{k}-I)(I-pi)pi^{2k-2} + \tfrac{1}{2}(2k-2)(2k-I)(I-pi)^{2}pi^{2k-3})Pdek.$$

Als $N=2^{k}-I-2k$ dan is: $Pc = p^{n \times N}$.

Voor $4 \leq k < 8$ en p is resp. p_1 en p_2 wordt door een komputer programma nagegaan of er een opbrengst verbetering mogelijk is. (Bijlage 8) Uit de reken resultaten blijkt dat voor $1 \leq n < 150$ geen opbrengst verbetering aanwezig is en dat de verhouding Pred/Pc zeer snel daalt bij toenemende k (Voor k=5 is dit 10^{-5}).

Konklusie.

De dekodeerschakeling bestaat uit 4 verschillende cel types, die veelal komplexer zijn dan de exclusive-or cellen van het q-funktie netwerk. De verbindingstruktuur van de dekodeerschakeling is ook komplexer, omdat er een extra derde verbindingslaag nodig is.
De rekenresultaten tonen aan dat de verhouding Pred/Pc zeer klein is voor toenemende waarden k. Voor k=4 is $Pred/Pc=10^{-1}$. Een opbrengst verbetering is met deze double error correcting codes in het geheel niet mogelijk.

Literatuur opgave.

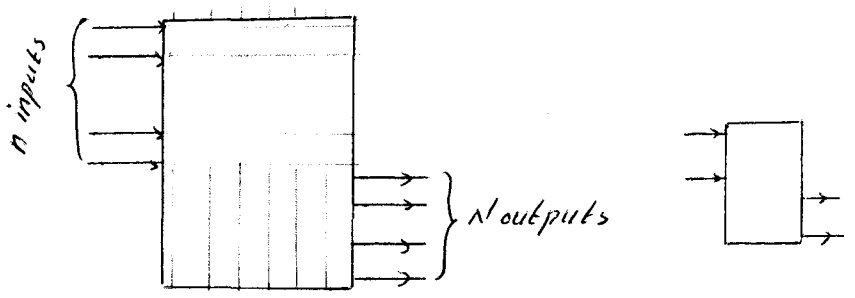
I Berlekamp E.R. Algebraic coding theory. Mc. Graw-Hill, Inc. New York. 1968.

2 Peterson W.W. Error correcting codes. Cambridge, Mass., M.I.T. Press. 1961.

hoofdstuk 10

APPENDICES.

10.1 Opbouw voternetwerk m.b.v. cellulaire logica.

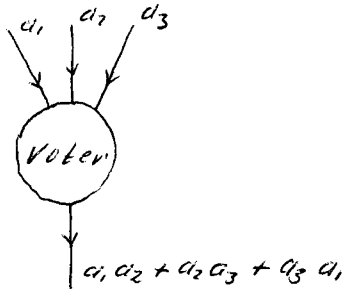


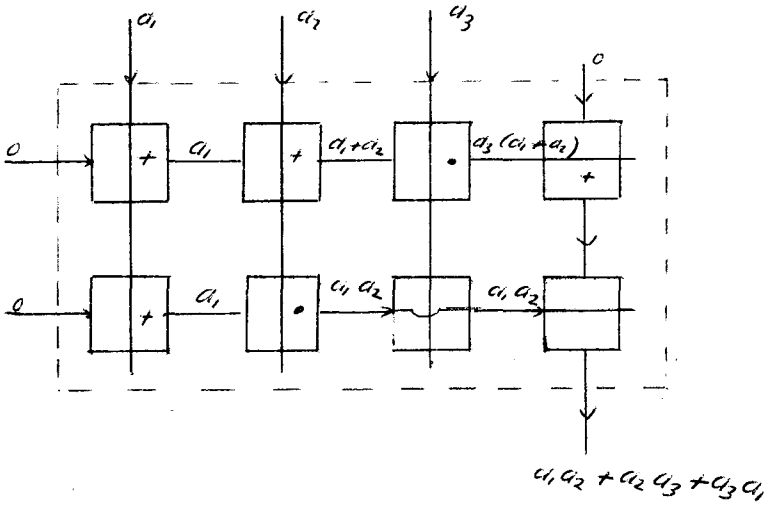
niet redundant cell. netwerk

Symbool

Fig. 10.1

Infig. 10.1 wordt het symbool van het randgevoed-bus type weergegeven.
De realisatie van een 3 inputs voternetwerk m.b.v. een cell. netwerk van het randgevoed-bus type wordt in fig. 10.2 weergegeven.





voternetwerk van het randgevoed bus type

Fig. 10.2

10.2 Worst-case berekening Pred van een redundant cell. netwerk van het randgevoed-bus type.

In deze sektie wordt Pred berekend van het netwerk in fig. 10.3. Eerst komen enige definities:

Pvoter	is de kans dat het voter netwerk goed werkt.
p	is de kans dat een cel goed werkt na het produktie proces.
Pc	is de kans dat het gehele niet redundante cell. netwerk goed werkt.
Pred	is de kans dat het redundante cell. netwerk goed werkt.
Pci	is de kans dat de i-de output van het niet redundante cell. netwerk goed werkt.
n	is het aantal inputs van het cell. netwerk.
m	is het aantal cascades in het hoofdnetwerk.
N	is het aantal outputs van het cell. netwerk.

Stel eerst dat de drie hoofdnetwerken goed zijn en dat er k kollektorcascades fout zijn. Het gehele netwerk werkt dan toch nog goed mits deze k kollektorcascades een verschillend rangnummer hebben en dat $k \leq N$.

De kans dat er k verschillende cascades fout zijn is:

$$3^k \binom{N}{k} (1-p^m)^k p^{(3N-k)m}$$

De kans dat het gehele netwerk nog goed werkt als de 3 hoofdnetwerken goed zijn is:

$$\sum_{k=0}^{N} 3^k \binom{N}{k} (1-p^m)^k p^{(3N-k)m} p^{3nm}$$

Daar komt nog bij de kans dat een hoofdnetwerk fout is, terwijl het erbij behorende kollektornetwerk er niet toe doet, maar de twee aandere kollektornetwerken goed zijn.
Het geheel moet nog met Pvoter vermenigvuldigd worden. $P_{voter} = p^{8N^2}$
Dan wordt:

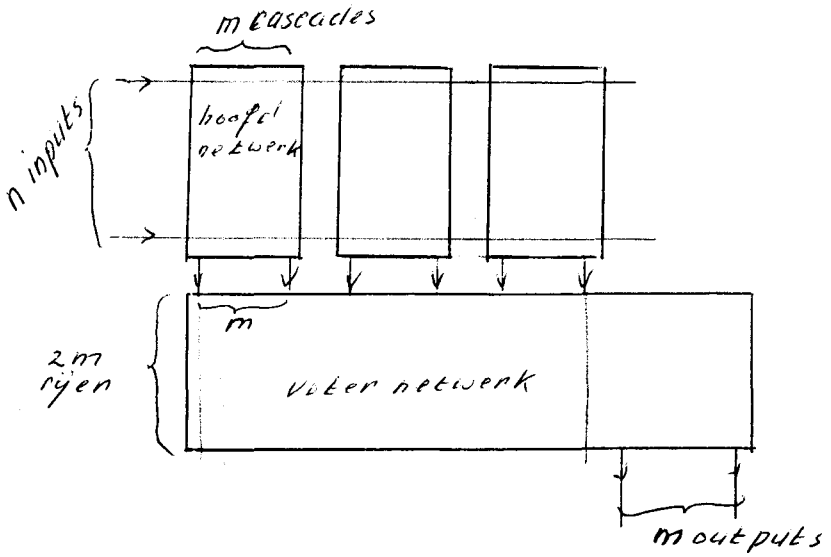
$$pred = \left\{ \sum_{k=0}^{N} 3^k \binom{N}{k} (1-p^m)^k p^{(3N-k)m} p^{3nm} + 3(1-p^{nm}) p^{2(nm+Nm)} \right\} P_{voter}$$

Het niet redundante netwerk bestaat uit (n + N)m cellen en daarom is:

$$Pc = p^{(n+N)m} .$$

Opmerking:
Bij Pci = 0.5 is punt A in de grafiek van fig. 7.2 bereikt. Hierdoor is bij Pci < 0.5 geen opbrengst verbetering mogelijk. Het komputer programma van bijlage 1 en 3 laat n oplopen totdat $Pci < p^{m+n} < 0.5$, daarna wordt n op nul gezet en een van de getallen m of N wordt met één verhoogd.

10.3 Berekening Pred van een redundant hoofd- of kollektor-netwerk.



Redundant hoofd- of kollektor-netwerk.

Fig. 10.4

Definitie: Pcas is de kans dan een cascade van het **niet redundante** hoofd- of kollektor- netwerk goed werkt.

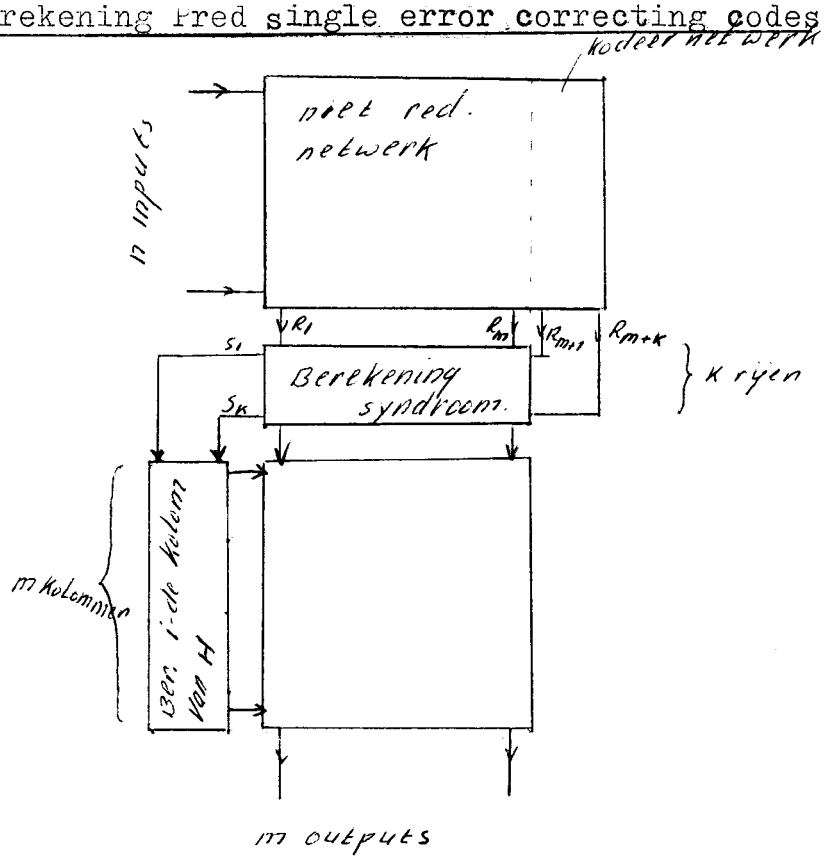
De cascade van het niet redundante hoofd- of kollektor-netwerk bevat n cellen dus

$$Pcas = p^n .$$

Het niet redundante hoofdnetwerk bevat nm cellen dus $Pc = p^{nm}$.

$$Pred = \left\{3(Pcas)^2 - 2(Pcas)^3\right\}^m . p^{8m^2} = p^{2nm} p^{8m^2} \left\{3 - 2p\right\}^m .$$

10.4 berekening Pred single error correcting codes



Redundant netwerk met n inputs en m outputs.

Fig. 10.5

Het aantal cellen in de dekodeerschakeling is m^2 +2km.

$P_{dek} = p^{m^2} + 2km$.

$P_i = p^n$ Pi is de kans dat de i-de output van het gekodeerde netwerk goed werkt.

Als één output van het gekodeerde netwerk defekt is, dan is de ouput van het redundante netwerk nog juist.

Dus $P_{red} = \left((m + k)(I - P_i)P_i^{m+k-I} + P_i^{m+k}\right) P_{dek}$.

↑ aantal mogelijke komb. van één fout.

10.5 2-de graadsvergelijking.

Gegeven $S_1(\alpha) = \alpha^v + \alpha^j$ (1)

$S_3(\alpha) = (\alpha^v)^3 + (\alpha^j)^3$ (2)

α^v en α^j zijn de onbekenden.

De algemene vorm van een 2-de graadsvergelijking is

$$\sigma(z) = z^2 + \sigma_1 z + \sigma_2 = 0$$

Als α^v en α^j wortels van deze vergelijking zijn dan moet

σ_1 gelijk zijn aan de som van beide wortels

σ_2 gelijk zijn aan het produkt van beide wortels

Dus

$$\sigma_1 = \alpha^v + \alpha^j = S_1(\alpha)$$

$$\sigma_2 = \alpha^v * \alpha^j$$

Uit (2) volgt

$$S_3(\alpha) = (\alpha^v + \alpha^j)\{(\alpha^v)^2 + \alpha^v\alpha^j + (\alpha^j)^2\} \quad *$$

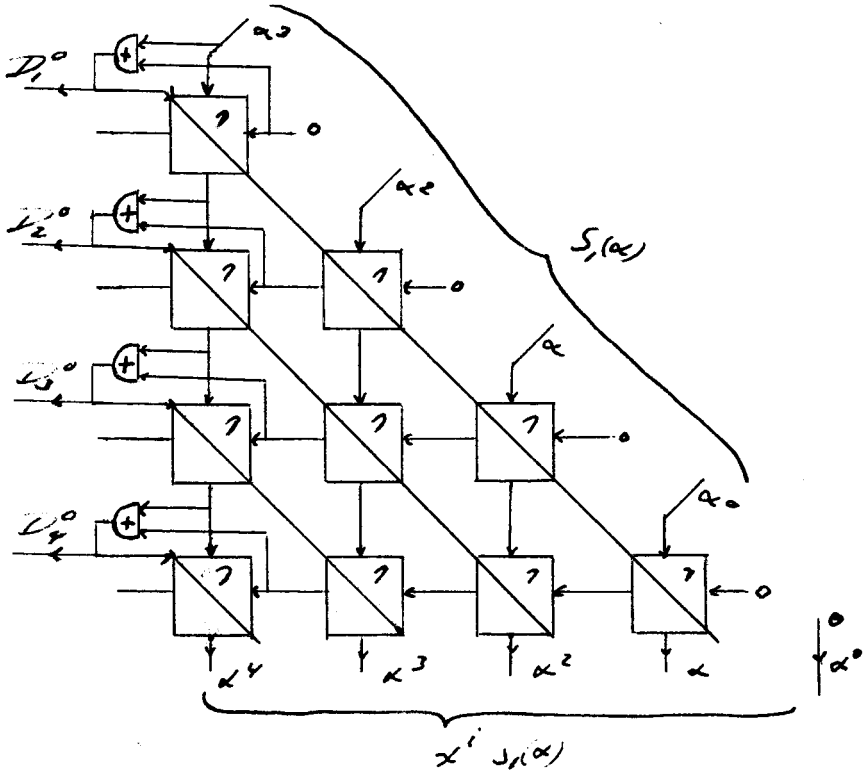
$$= S_1(\alpha)\{(\alpha^v)^2 + (\alpha^j)^2 + \alpha^v\alpha^j\} = S_1(\alpha)\{(S_1(\alpha))^2 + \alpha^v\alpha^j\}.$$

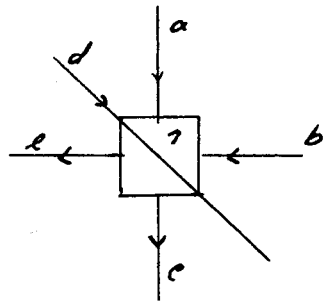
Dus $\alpha^v\alpha^j = (S_1(\alpha))^2 + \frac{S_3(\alpha)}{S_1(\alpha)} \Rightarrow \sigma_2 = (S_1(\alpha))^2 + \frac{S_3(\alpha)}{S_1(\alpha)}$

* modulo-2 optellen en aftrekken zijn dezelfde bewerkingen.

10.6 Verschuivingsnetwerk.

Gegeven een polynoom $S_1(\alpha)$ waarvan de graad kleiner of gelijk is aan 3. Dit polynoom moet zodanig met α^i vermenigvuldigd worden, dat $\alpha^i . S_1(\alpha)$ een 4-de graads polynoom is.
Dit kan worden gerealiseerd door het volgende netwerk.



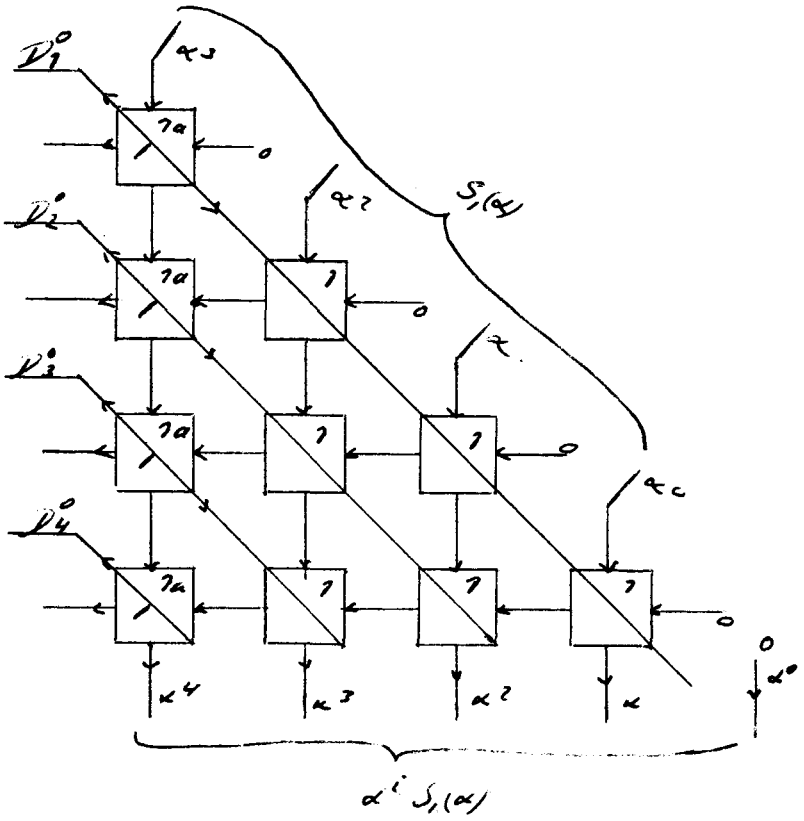


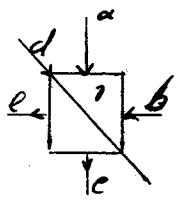
$c = (a+b).d$
$e = (a+b).\bar{d}$
} boolse schrijfwijze

cel 1

fig. 10.6

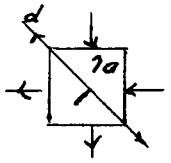
Als $D_i^0 = 0$ dan wordt $S_1(\alpha)$ één plaats naar links geschoven
Als $D_i^0 = 1$ dan wordt $S_1(\alpha)$ niet verschoven.
De or-poorten vormen de enige onregelmatigheid. Dit is op te heffen door de volgende schakeling.(fig. 10.7)





cel 1 $\quad c = (a+b).d$

$e = (a+b)\bar{d}$



cel 1a $\quad d = a+b$

$c = (a+b).d = a+b$

$e = (a+b).\bar{d} = 0$

⇑

boolse schrijf wijze

fig 10.7

Er is een tweede cel bijgekomen, die zeer weinig verschilt van cel 1. Deze cel bezit een extra verbinding meer dan cel 1 (nl. de verbinding tussen lijn d en de uitgang van de or-poort a+b).
De D_i° lijnen geven de graad van $S_i(x)$ aan.
Voorbeeld:

$D_1^\circ = D_2^\circ = D_3^\circ = D_4^\circ = 1 \Rightarrow$ graad $S_i(x)$ is 3

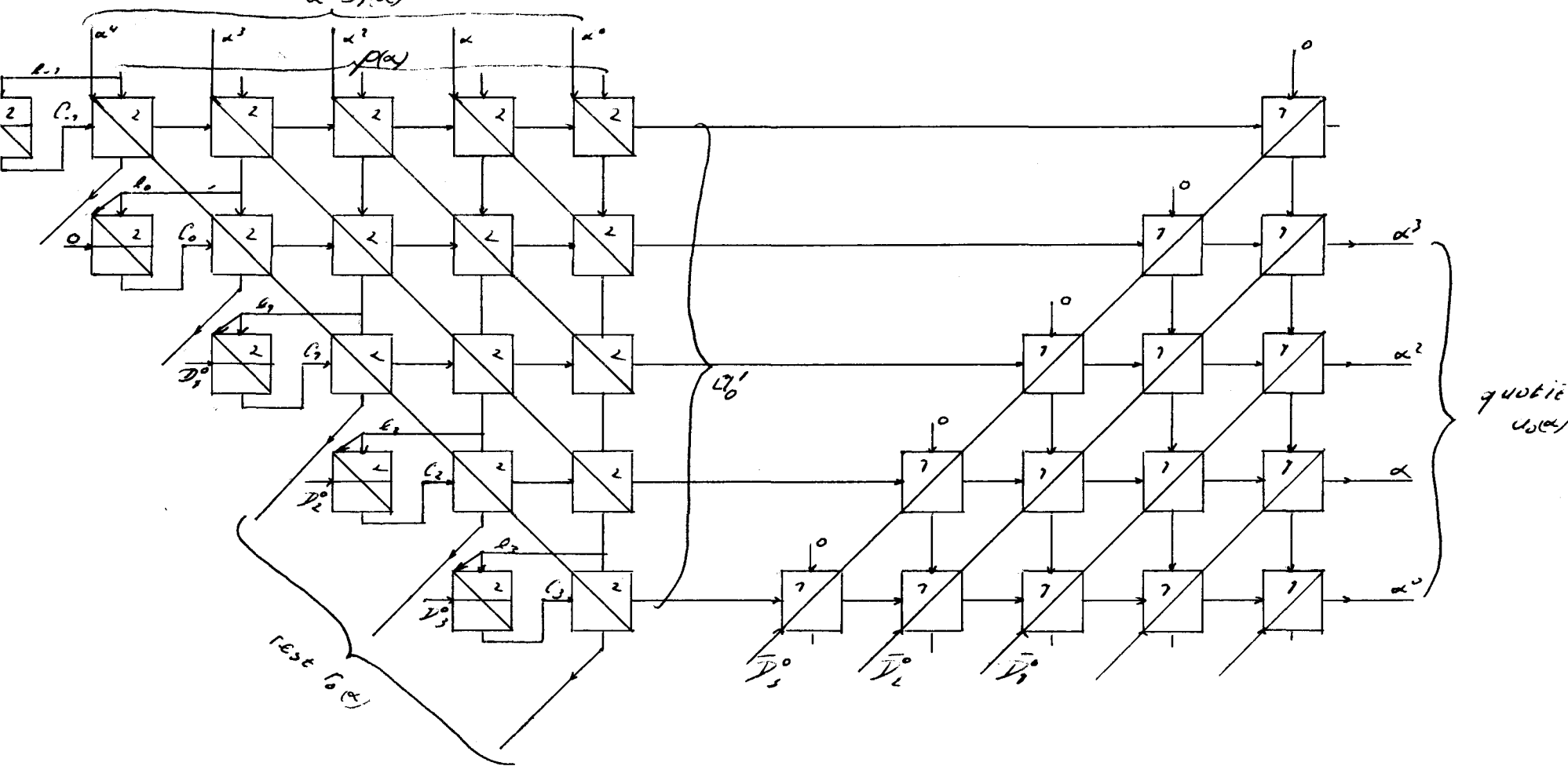
$D_1^\circ = D_2^\circ = D_3^\circ = 0$ en $D_4^\circ = 1 \Rightarrow$ graad $S_i(x)$ is 0

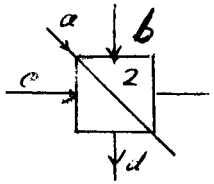
10.7 Aftrek en verschuivingsnetwerk.

$\alpha^i . S_i(\alpha)$ is een 4-de graads polynoom en moet herhaalde malen van het deeltal $p(\alpha) = \alpha^4 + \alpha + 1$ afgetrokken worden, waarbij $\alpha^i . S_i(\alpha)$ steeds een plaats naar rechts wordt geschoven.
Het aftrekken wordt beëindigd als de graad van de rest kleiner is dan de graad van $S_i(\alpha)$.
Fig. 10.9 geeft het gehele aftreknetwerk weer.

Als $c_i = \overline{D_i^0} . e_i = 1$ dan wordt afgetrokken.
Als $c_i = \overline{D_i^0} . e_i = 0$ dan wordt **niet** afgetrokken.

De output a_0' van het aftrek-netwerk is gelijk aan $a_0' = a_0 . \alpha^{\text{graad } S(\alpha)}$
a_0 is het quotiënt van de deling.
a_0' moet een aantal plaatsen verschoven worden om het quotiënt a_0 te verkrijgen.





$d = (a \oplus b) \cdot c$ boolse schrijfwijze cel

Fig. 10.4 Opzicht en verschuivingsnetwerk

10.8 Vermenigvuldigingsnetwerk

Een vermenigvuldigingsnetwerk, dat $a(\alpha).h(\alpha) \bmod p(\alpha)$ berekent, kan gerealiseerd worden door eerst $a(\alpha).h(\alpha)$ te berekenen en daarna dit produkt te delen door $p(\alpha)$. De rest is dan gelijk aan $a(\alpha).h(\alpha) \bmod p(\alpha)$.
Het vermenigvuldigingsnetwerk bestaat dus uit een netwerk, dat het produkt van twee polynomen berekent, en een netwerk dat een deling uitvoert.(Zie fig. 10.10) De cel waaruit beide netwerken bestaan is cel 2 .
De hoogte van het vermenigvuldigingsnetwerk moet even groot zijn als die van de deler, behandeld in appendix 10.7. Om daaraan te voldoen wordt het delingsnetwerk om de EE lijn gedraaid en op het produkt netwerk gelegd.
Het daaruit ontstane vermenigvuldigingsnetwerk wordt in fig. 10.11 weergegeven.
Iedere cel van dit netwerk (cel 3) bestaat uit twee deelcellen van het cel type 2.
De meest rechtse cellen van het delingsnetwerk moeten nu achterwege gelaten worden. (Anders terugkoppeling!)

De $\overline{D_i^v}\,\overline{D_j^v}$ lijn dient om $H(\alpha)$ de juiste waarde te geven.
Als $\overline{D_i^v}.\overline{D_j^v}=1$ dan $H(\alpha)=0$.

als $\overline{D_i^v}\,\overline{D_j^v}=0$ dan $H(\alpha)=H'(\alpha)$.

Veronderstel dat;

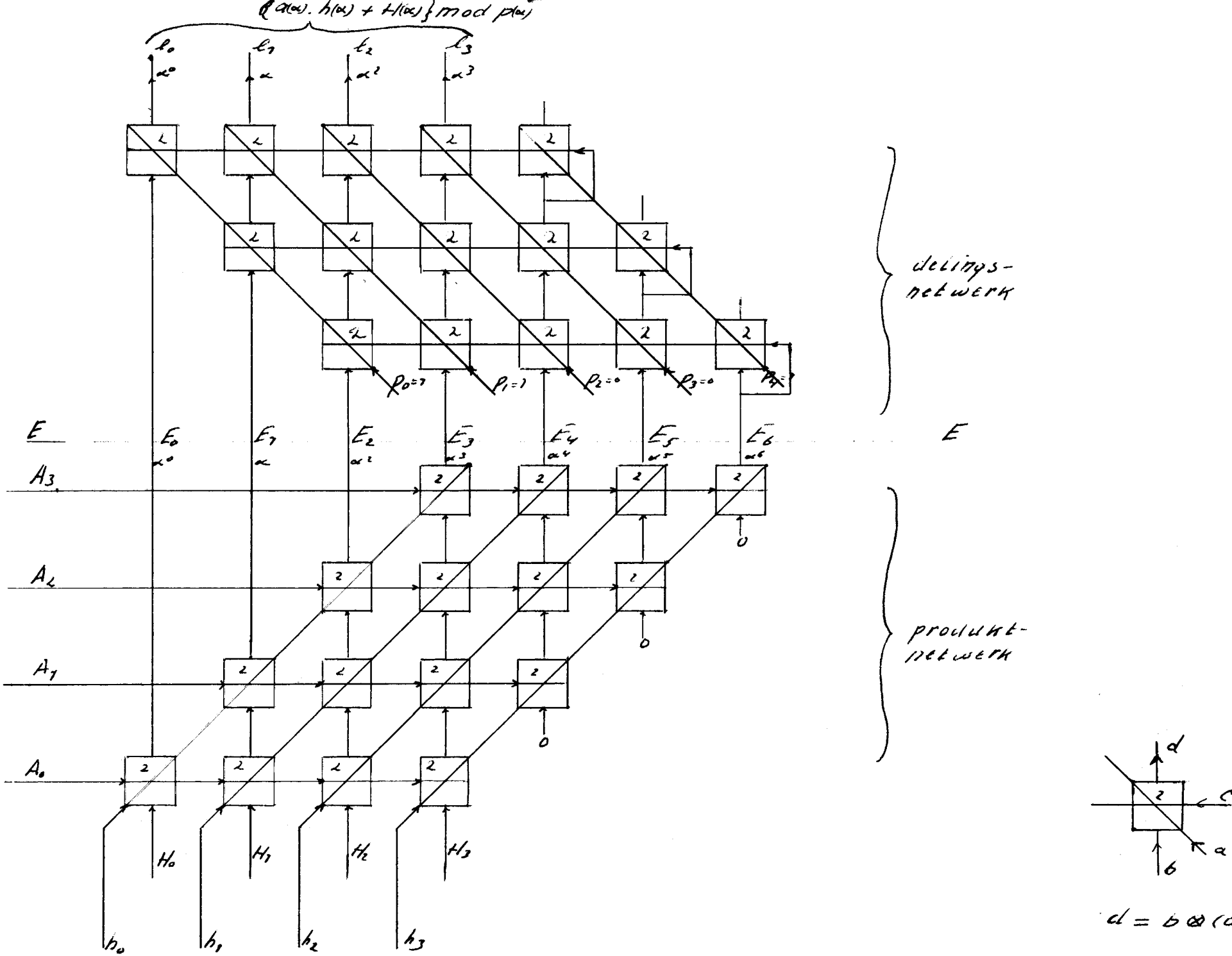
$$a(\alpha) = A_3\alpha^3 + A_2\alpha^2 + A_1\alpha + A_0 .$$

$$h(\alpha) = h_3\alpha^3 + h_2\alpha^2 + h_1\alpha + h_0 .$$

$$H(\alpha) = H_3\alpha^3 + H_2\alpha^2 + H_1\alpha + H_0 .$$

$$p(\alpha) = p_3\alpha^3 + p_2\alpha^2 + p_1\alpha + p_0 + p_4\alpha^4$$

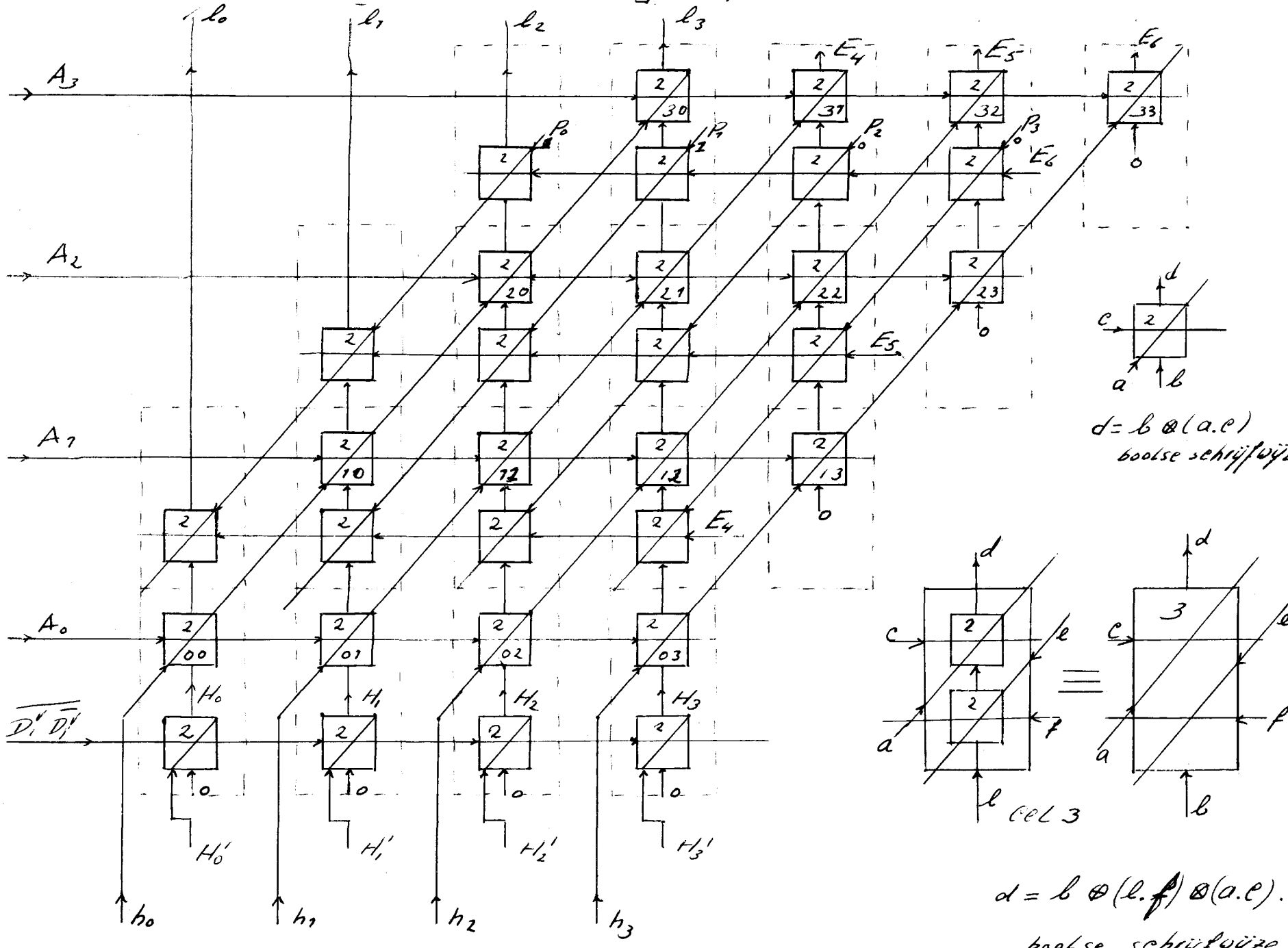
voor $GF(24)$ is $p_4 = p_1 = p_0 = 1$
$p_3 = p_2 = 0$



$d = b \otimes c$

cel 2.

FIG 12.1 vermenigvuldigingsnetwerk



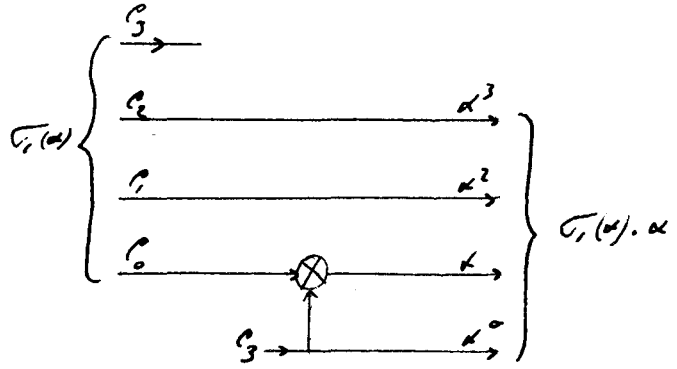
vermenigvuldigings netwerk fig 10.11

10.9 Hardware oplossing van $\sigma(z) = z^2 + \sigma_1 z + \sigma_2 = 0$

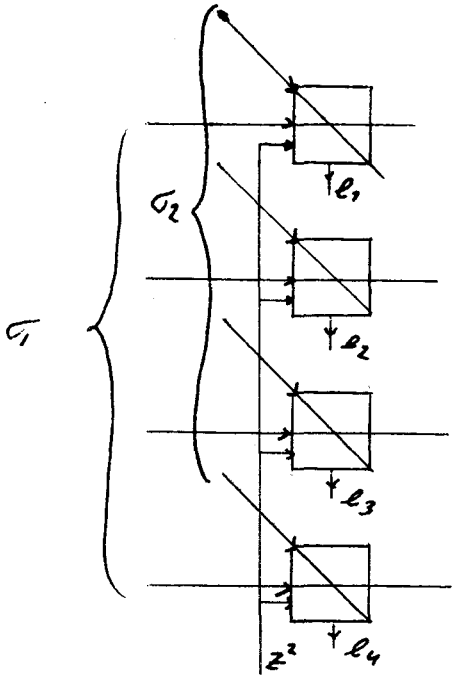
Stel $\sigma_1(\alpha) = c_3 \alpha^3 + c_2 \alpha^2 + c_1 \alpha + c_0$

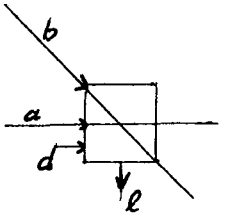
$\sigma_1(\alpha) \cdot \alpha = c_2 \alpha^3 + c_1 \alpha^2 + (c_0 + c_3)\alpha + c_3$

De realisatie van dit produkt ziet er dan als volgt uit:



De realisatie van $z^2 + \sigma_1 z + \sigma_2 = \sigma(z)$ is:

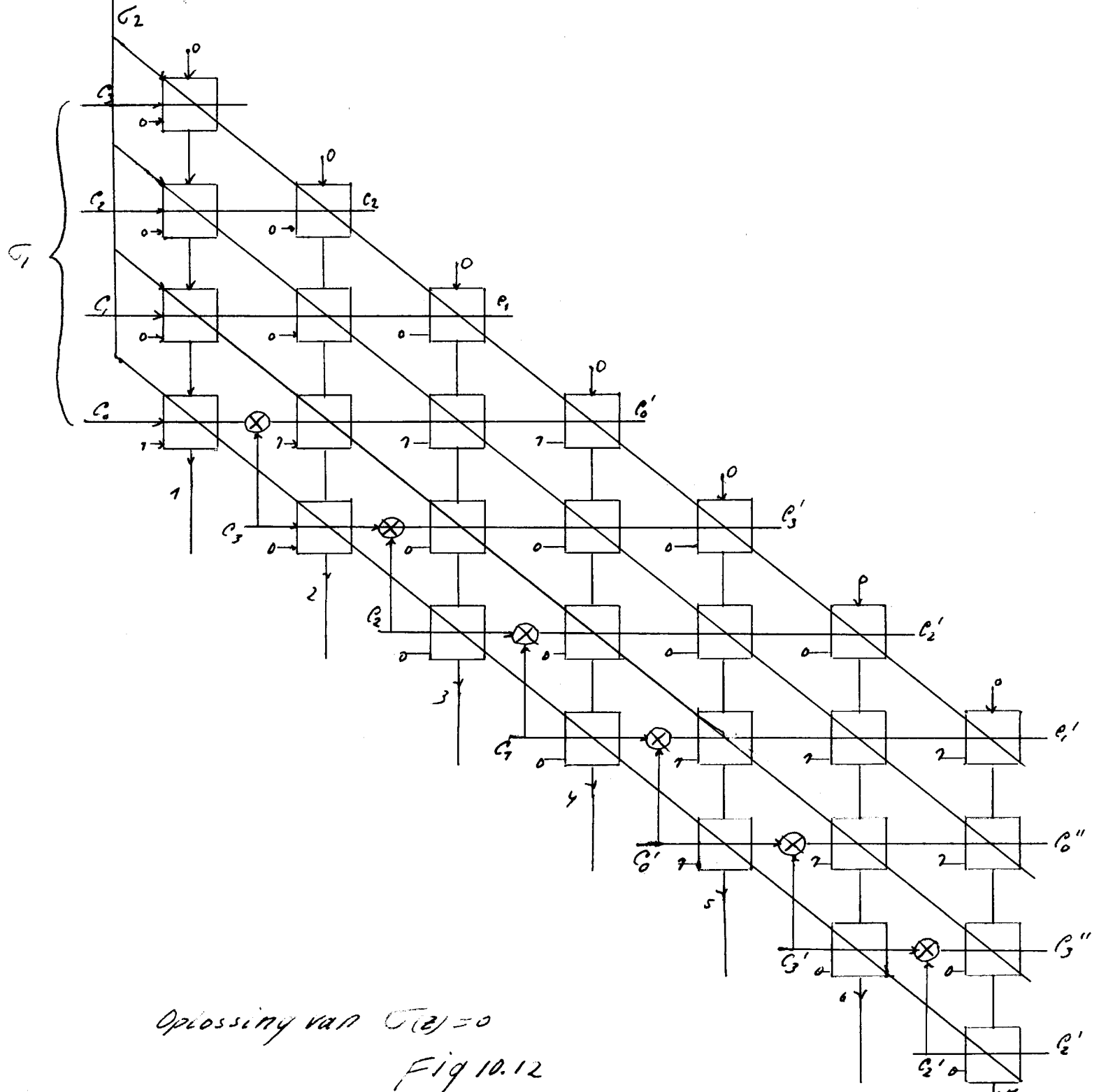


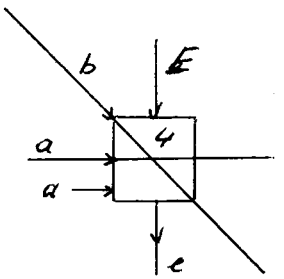


$e = a \oplus b \oplus d$ boolse schrijfwijze

$\sigma(z) = (e_1, e_2, e_3, e_4)$

Als $e_1 \vee e_2 \vee e_3 \vee e_4 = 0$ ($\vee$ is hier or-funktie)
dan $\sigma(z) = 0$ en de output van deze or-poort geeft dan de plaats van de fout aan.
Fig. 10.12 geeft de gehele schakeling weer.





$e = (a \oplus b \oplus d) + E$

boolse se

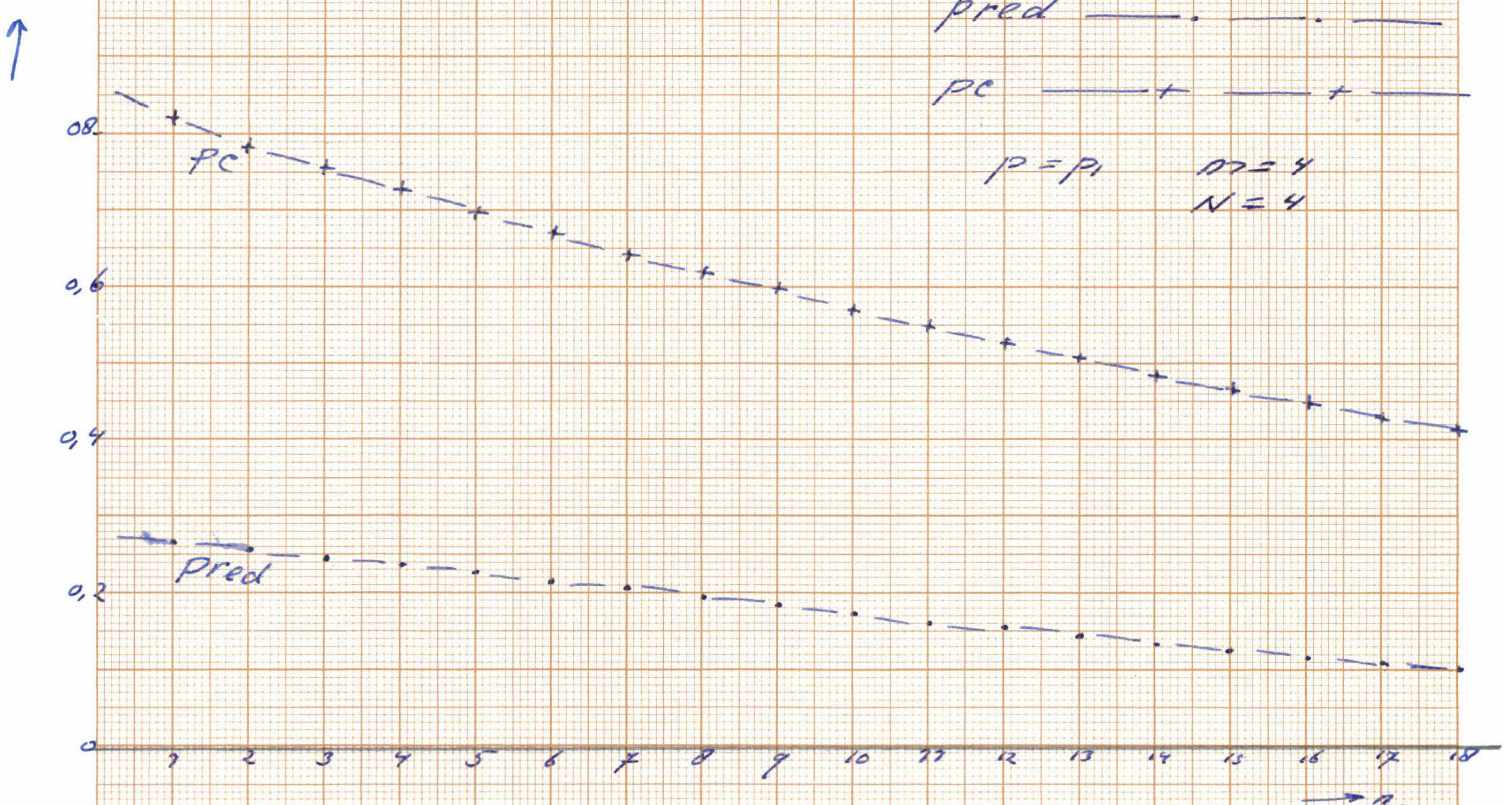
cel 4

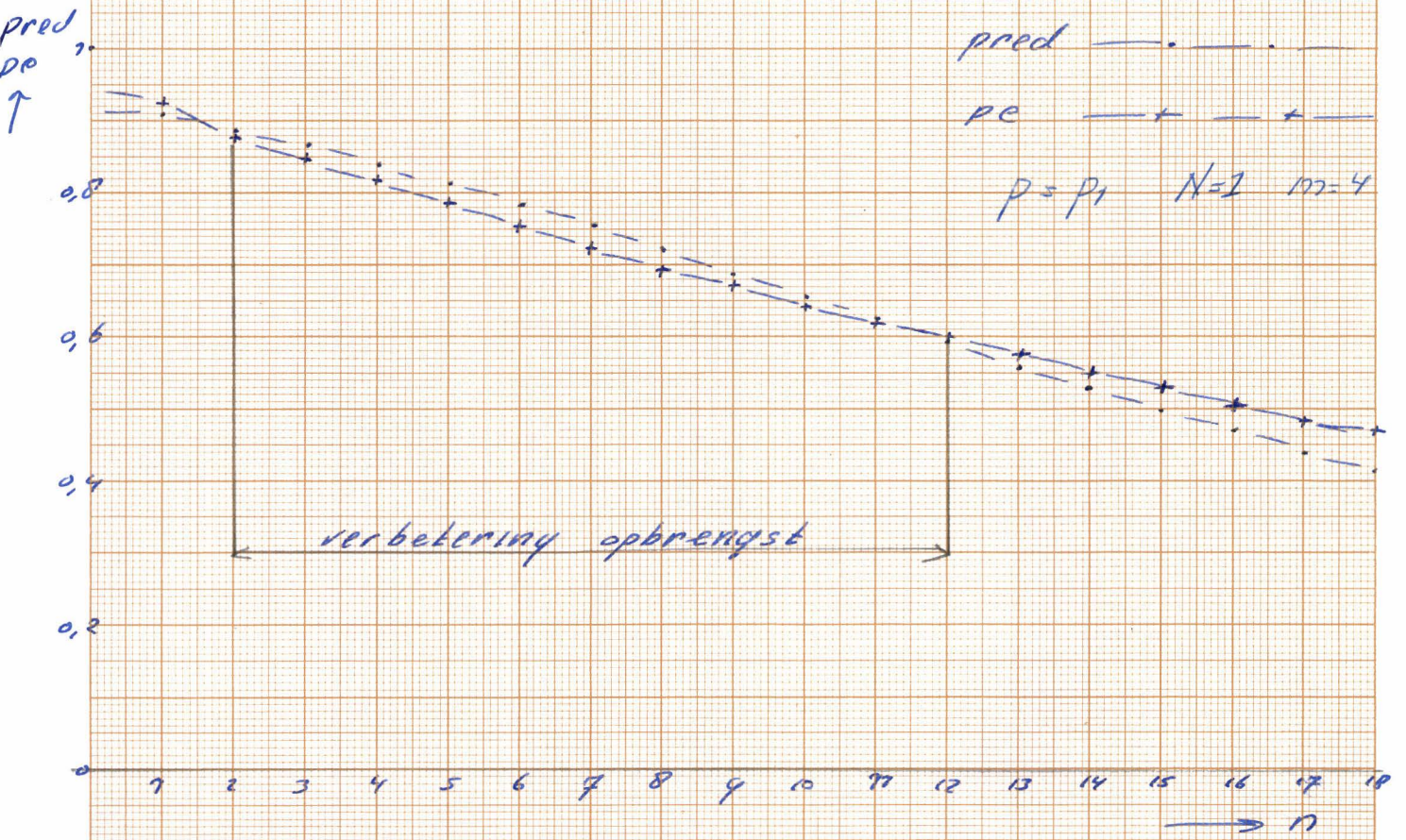
Oplossing van $C(2) = 0$

Fig 10.12

Lalgol 0618479 steltenpool

```
begin comment berekening opbrengst-verbetering van cellulaire netwerken, (worst-case berekening).

      n = aantal inputs cell. netwerk

      N = aantal outputs cell. netwerk

      m = aantal cascades hoofdnetwerk

      1 - p is fout kans cel;
      integer Nmax, mmax, nmax, t, N, m, n, j, v, i, k;
      real p, p8N2, pm, pNm, p2Nm, pmn, pred, pc, Nsubf, som, kfak, Novk, p2mn, p3mn, pn;
      j:= 0;
B:    Nmax:= read; mmax:= read; nmax:= read;
      t:= read;
    p:= exp(ln(0.05)/t); FLOT(3, 2, p); NLCR; NLCR;
      for N:= 1 step 1 until Nmax do
      begin p8N2:= p↑(8 × (N ↑ 2));
            for m:= 1 step 1 until mmax do
            begin pm:= p ↑ m;
                  pNm:= p ↑ (N × m); p2Nm:= pNm × pNm;
                  pmn:= 1; pn:= 1; som:= p2Nm × pNm;
                  for k:= 1 step 1 until N do
                  begin  Nsubf:= 1; kfak:= 1;  for v:= N-k+1 step 1 until N do Nsubf:= Nsubf × v;
                        for i:= 1 step 1 until k do kfak:= kfak × i; Novk:= (Nsubf/kfak);
                        som:= som + (3↑k)× Novk × (( 1 - pm )↑k) × ( pm↑( 3 × N - k ));
                  end;
                  for n:= 1 step 1 until nmax do
                  begin pmn:= pmn × pm; pn:= p × pn;
                        if (pn × pm) < 0.5 then goto A;
                        p2mn:= pmn × pmn; p3mn:= p2mn ×pmn;
                        pred:= ( som × p3mn + 3×( 1 - pmn ) × p2mn × p2Nm ) × p8N2;
                        pc:= pmn × pNm;
                        if pred < pc then goto C;
                        PRINTTEXT(‹verbetering›);
                        goto E;
C:                      if m ≠ 4 then goto D;
E:                    NLCR; SPACE(15); FLOT(3, 2, n); FLOT(3, 2, m);
                        FLOT(3, 2, N); FLOT(3, 2, pred); FLOT(3, 2, pc); FLOT(3, 2, som); FLOT(3, 2, Novk); NLCR;
D:                  end;
A:
            end;
      end; j:= j + 1; if j = 1 then goto B;
```

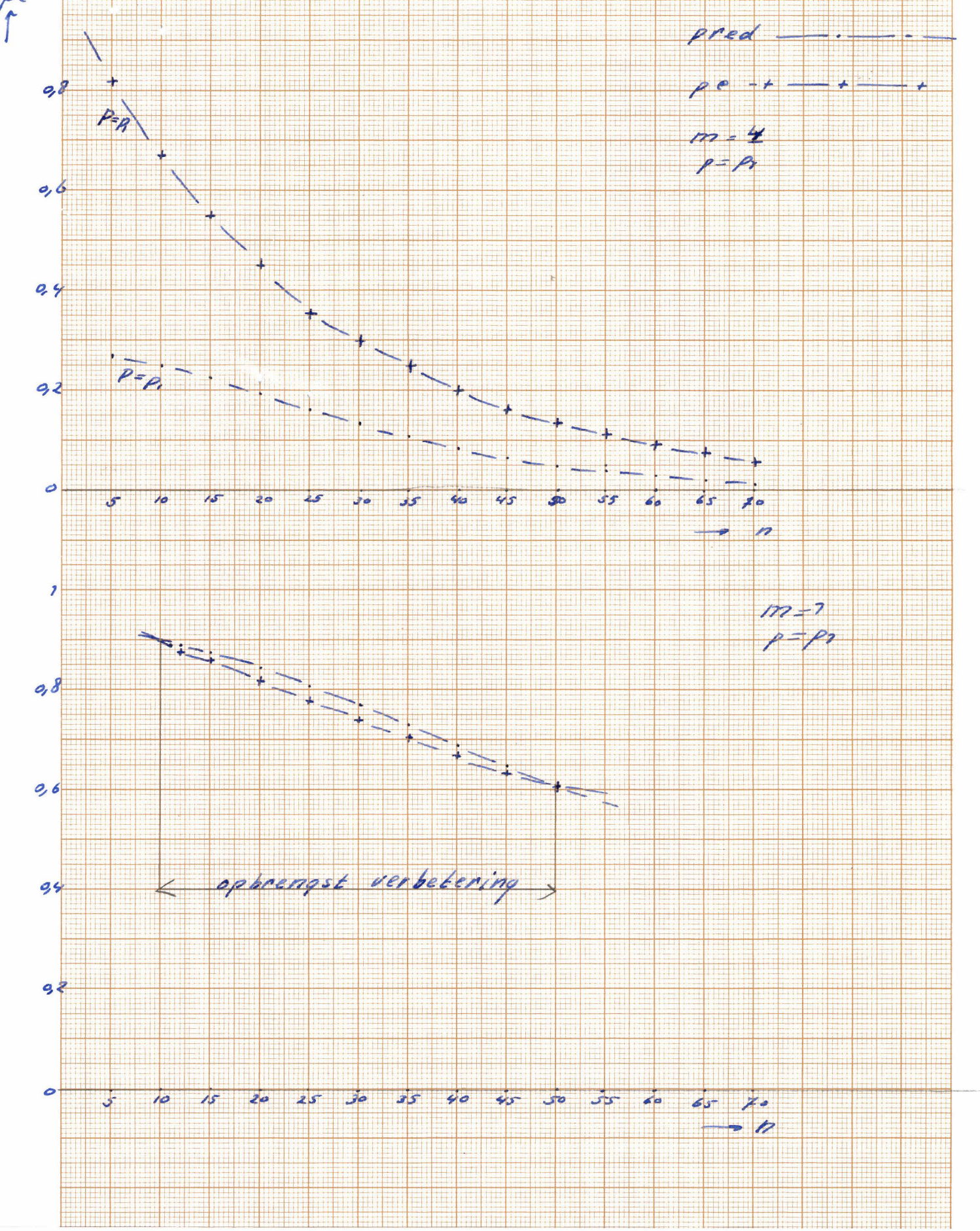




pred en pc als funktie van n bij konstante m en N

```
Ialgol 06185479 steltenpool
begin comment berekening pred en pc van kollektor netwerk door toepassing van voternetwerk.
      n is aantal inputs
      m is aantal outputs;
      integer m, t, n, j;
      real p, p↑m, p↑8m↑2, pm, p↑mn, pn, pc, pred; j := 0;
B:    t := read; p := exp(ln(0.05)/t); FLOT(3, 2, p); NLCR; NLCR;
      for m := 1 step 1 until 10 do
      begin p↑m := p ↑ (2 × m); p↑8m↑2 := p ↑ (8 × (m ↑ 2));
            pm := p ↑ m; p↑mn := 1; pn := 1; pc := 1;
            for n := 1 step 1 until 600 do
            begin pn := pn × p;
                  if pn < 0.5 then goto A;
                  p↑mn := p↑mn  × p↑m; pc := pc × pm;
                  pred := p↑mn × ((3 - 2 × pn)↑ m) × p↑8m↑2;
                  if pred < pc then goto C;
                  PRINTTEXT(‡verbetering‡); NLCR; goto E;
C:                if m ≠ 4 then goto D;
E:                SPACE(15); FLOT(3, 2, n); FLOT(3, 2, m);
                  FLOT(3, 2, pred); FLOT(3, 2, pc); NLCR;
D:
            end;
A:
      end; j := j + 1; if j = 1 then goto B;
end
progend

300, 600,
```

Bylage 3


pred
pe
m = 4
p = p1
p = p0
0,8
0,6
0,4
0,2
0
5 10 15 20 25 30 35 40 45 50 55 60 65 70
m
m = 7
p = p2
1
0,8
0,6
0,4
0,2
0
opbrengst verbetering
5 10 15 20 25 30 35 40 45 50 55 60 65 70
m

```
Lalgol 06185479 steltenpool
begin comment berekening opbrengst verbetering van kollektor netwerk door toepassing
      van hamming code;
      real p, pi, pdek, pred, pc;
      integer n, m, k, i, t;
      i := 0;
B:    t := read; p := exp(ln (0.05)/t); NEW PAGE;
      PRINTTEXT(‹p›); m := 0; k := 1;
A:    n := 1; m := m+1;
      if m > (2 ↑ k - k - 1) then k := k+1; pi := p;
      pdek := p ↑ (2 × m × k + (m ↑ 2));
C:    n := n+1; pi := p × pi; pc := pi ↑ m;
      pred := ((m+k) × (1-pi) × (pi ↑ (m + k - 1)) + pi ↑ (m+k)) × pdek;
      if (pred > pc ) ∨ (m = 4) ∨ (m = 3) then
      begin if pred > pc then
            begin PRINTTEXT(‹verbetering›);
                  NLCR; NLCR;
            end;
            SPACE(15); FLOT(3, 2, n); FLOT(3, 2, m);
            FLOT(3, 2, k); FLOT(3, 2, pred); FLOT(3, 2, pc);
                  NLCR;
      end;
      if pi > 0.5 then goto C;
      if m < 57 then goto A;
      i := i+1; if i = 1 then goto B;
end
progend
```

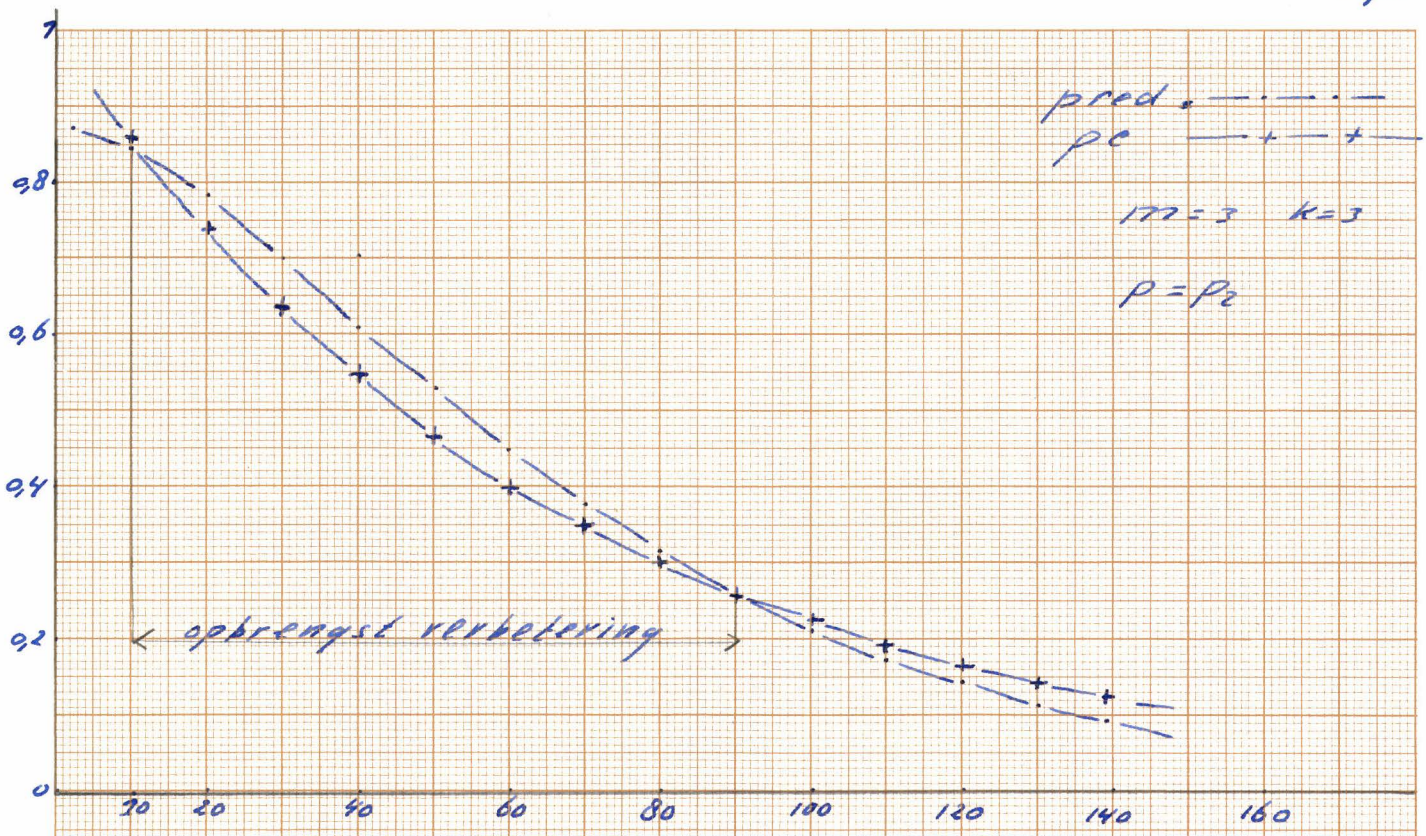
Bijlage 5

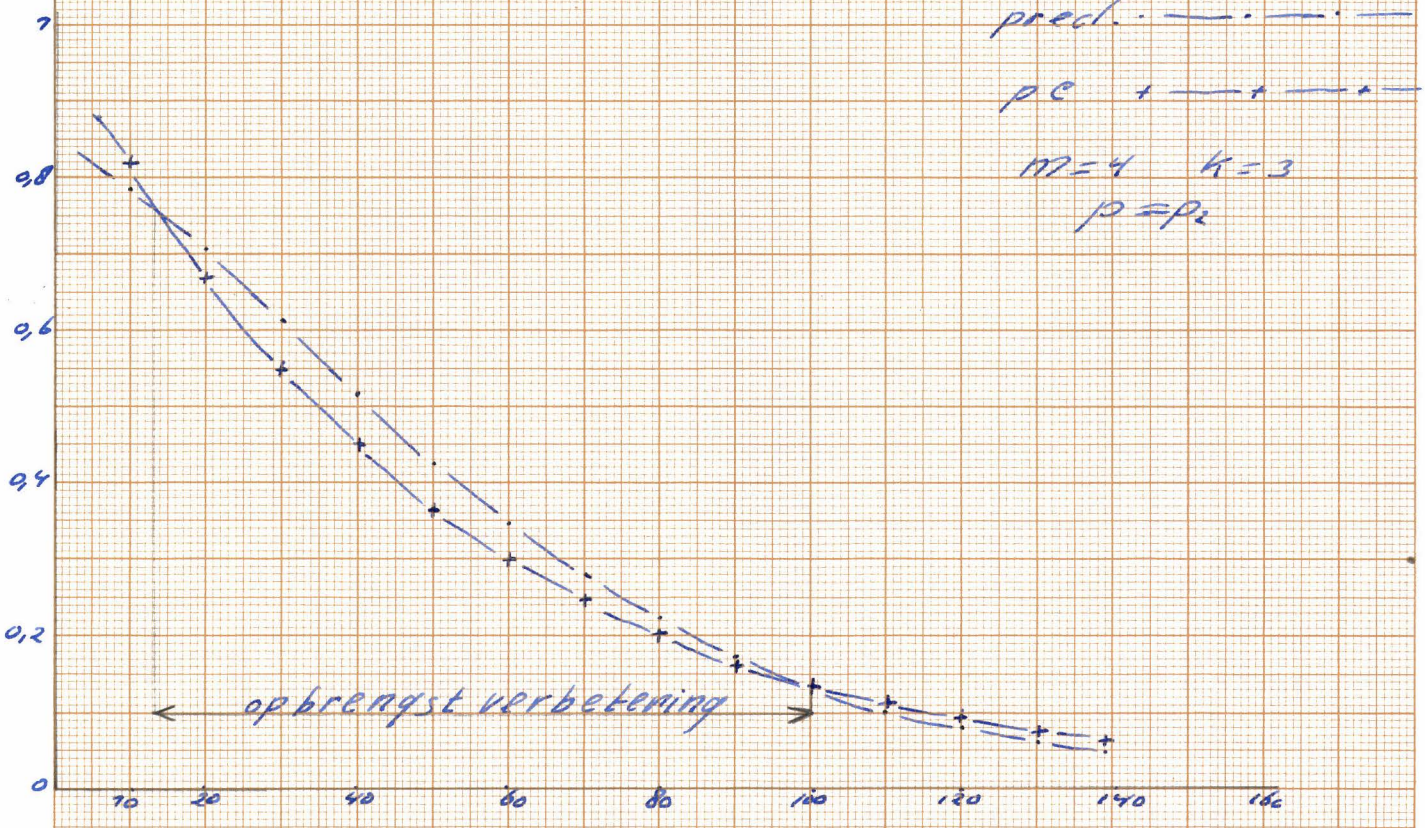
Table 1. Representation of $GF(2^4)$

$\alpha^0 = 1$		$= (1\ 0\ 0\ 0)$
$\alpha^1 = \alpha$		$= (0\ 1\ 0\ 0)$
$\alpha^2 = \alpha^2$		$= (0\ 0\ 1\ 0)$
$\alpha^3 = \alpha^3$		$= (0\ 0\ 0\ 1)$
$\alpha^4 = 1 + \alpha$		$= (1\ 1\ 0\ 0)$
$\alpha^5 = \alpha + \alpha^2$		$= (0\ 1\ 1\ 0)$
$\alpha^6 = \alpha^2 + \alpha^3$		$= (0\ 0\ 1\ 1)$
$\alpha^7 = 1 + \alpha + \alpha^3$		$= (1\ 1\ 0\ 1)$
$\alpha^8 = 1 + \alpha^2$		$= (1\ 0\ 1\ 0)$
$\alpha^9 = \alpha + \alpha^3$		$= (0\ 1\ 0\ 1)$
$\alpha^{10} = 1 + \alpha + \alpha^2$		$= (1\ 1\ 1\ 0)$
$\alpha^{11} = \alpha + \alpha^2 + \alpha^3$		$= (0\ 1\ 1\ 1)$
$\alpha^{12} = 1 + \alpha + \alpha^2 + \alpha^3$		$= (1\ 1\ 1\ 1)$
$\alpha^{13} = 1 + \alpha^2 + \alpha^3$		$= (1\ 0\ 1\ 1)$
$\alpha^{14} = 1 + \alpha^3$		$= (1\ 0\ 0\ 1)$
$\alpha^{15} = 1 = \alpha^0$		

primitieve polynoom $p(\alpha) = \alpha^4 + \alpha + 2$.

bijlage 6





pred en pe als funktie van n bij konstante m en p = p2.

```
Lalgol 06185479 steltenpool
begin comment berekening opbrengst verbetering van cell netwerken m.b.v. double error correcting BCH codes
      n = aantal inputs cell. netwerk
      m = (2 ↑ k) -1 is aantal outputs cell. netwerk;
      integer k, kmax, ia1, ib1,  i2, i, j, a, N, s, b, i0een, i0twee, f, i2k, v, nmax, n, t;
      real pdek, p, pc, pred, pi, q;
      v := 0; kmax := read;
B:    t := read; nmax := read; p := exp(ln(0.05)/t); q:= 1 - p;
      NEW PAGE; FLOT(5, 2, p); NLCR; NLCR; NLCR;
      for k := 4 step 1 until kmax do
      begin ia1 := 0; ib1 := 0; i2 := 0; for i := 2 step 1 until  k do
                              ia1 := (i ↑2) + 2 × i + 1 + ia1;  ib1 := 0.5 ×(i ↑2) + (5/2) × i + 2 + ib1;
            for j := 1 step 1 until k do begin a := (2 ↑ j);
            if a < k then i2 := (a ↑ 2) + i2 else goto A; end;
A:          i2 := i2 + (k - 2 - j) × (k ↑ 2); i2k := 2 ↑ k; N := i2k - 1 - 2 × k;
            s := 2 × k × N; b := 2 × (k ↑ 2); i0een := k × N; i0twee := N;
            f := N ↑ 2;
            pdek := p ↑ s × (1 - (4/5) × q) ↑ ia1 × (1 - (7/5) × q) ↑ ib1 × (1 - ( 14/5) × q ) ↑ i2 × (1 - (14/5) × q) ↑ b
            × (1 - (12/5) × q) ↑ i0een × p ↑ i0twee × p ↑ f;
            for n := 1 step 1 until nmax do
            begin pi := p ↑ n;
                  pred := (pi ↑ (i2k - 1) + (1 - pi) × (i2k - 1) × ( pi ↑ (i2k - 2))
                 + 0.5 × (i2k - 2) × (i2k - 1) × ((1 - pi) ↑ 2) × (pi ↑ (i2k - 3))) × pdek;
                  pc := p ↑ (n × N); if pred < pc then goto C;
                  PRINTTEXT(⟨verbetering⟩); NLCR; goto E;
C:                if k ≠ 5 ∧ k ≠ 4  then goto D;⊖
E:                SPACE(15); FLOT(3, 2, n); FLOT(3, 2, k); FLOT(5, 2, pred);
                  FLOT(3, 2, pc); FLOT(5, 2, pdek); FLOT(3, 2, ia1); FLOT(3, 2, i2);NLCR;
            D:          end;
      end;  v := v + 1; if v = 1 then goto B;
end
progend

7   300   70   600   150
```

Bijlage 8

$\rho_1 = +.990_{10}+0$

	n	m	pred	pc
VERBETERING	$+.120_{10}+2$	$+.100_{10}+1$	$+.891_{10}+0$	$+.887_{10}+0$
VERBETERING	$+.130_{10}+2$	$+.100_{10}+1$	$+.886_{10}+0$	$+.878_{10}+0$
VERBETERING	$+.140_{10}+2$	$+.100_{10}+1$	$+.880_{10}+0$	$+.870_{10}+0$
VERBETERING	$+.150_{10}+2$	$+.100_{10}+1$	$+.875_{10}+0$	$+.861_{10}+0$
VERBETERING	$+.160_{10}+2$	$+.100_{10}+1$	$+.869_{10}+0$	$+.852_{10}+0$
VERBETERING	$+.170_{10}+2$	$+.100_{10}+1$	$+.863_{10}+0$	$+.844_{10}+0$
VERBETERING	$+.180_{10}+2$	$+.100_{10}+1$	$+.856_{10}+0$	$+.835_{10}+0$
VERBETERING	$+.190_{10}+2$	$+.100_{10}+1$	$+.850_{10}+0$	$+.827_{10}+0$
VERBETERING	$+.200_{10}+2$	$+.100_{10}+1$	$+.843_{10}+0$	$+.819_{10}+0$
VERBETERING	$+.210_{10}+2$	$+.100_{10}+1$	$+.837_{10}+0$	$+.811_{10}+0$
VERBETERING	$+.220_{10}+2$	$+.100_{10}+1$	$+.830_{10}+0$	$+.803_{10}+0$
VERBETERING	$+.230_{10}+2$	$+.100_{10}+1$	$+.823_{10}+0$	$+.795_{10}+0$
VERBETERING	$+.240_{10}+2$	$+.100_{10}+1$	$+.815_{10}+0$	$+.787_{10}+0$
VERBETERING	$+.250_{10}+2$	$+.100_{10}+1$	$+.808_{10}+0$	$+.779_{10}+0$
VERBETERING	$+.260_{10}+2$	$+.100_{10}+1$	$+.800_{10}+0$	$+.771_{10}+0$
VERBETERING	$+.270_{10}+2$	$+.100_{10}+1$	$+.793_{10}+0$	$+.764_{10}+0$
VERBETERING	$+.280_{10}+2$	$+.100_{10}+1$	$+.785_{10}+0$	$+.756_{10}+0$
VERBETERING	$+.290_{10}+2$	$+.100_{10}+1$	$+.777_{10}+0$	$+.749_{10}+0$
VERBETERING	$+.300_{10}+2$	$+.100_{10}+1$	$+.770_{10}+0$	$+.741_{10}+0$
VERBETERING	$+.310_{10}+2$	$+.100_{10}+1$	$+.762_{10}+0$	$+.734_{10}+0$
VERBETERING	$+.320_{10}+2$	$+.100_{10}+1$	$+.754_{10}+0$	$+.726_{10}+0$
VERBETERING	$+.330_{10}+2$	$+.100_{10}+1$	$+.746_{10}+0$	$+.719_{10}+0$
VERBETERING	$+.340_{10}+2$	$+.100_{10}+1$	$+.738_{10}+0$	$+.712_{10}+0$
VERBETERING	$+.350_{10}+2$	$+.100_{10}+1$	$+.730_{10}+0$	$+.705_{10}+0$
VERBETERING	$+.360_{10}+2$	$+.100_{10}+1$	$+.722_{10}+0$	$+.698_{10}+0$
VERBETERING	$+.370_{10}+2$	$+.100_{10}+1$	$+.713_{10}+0$	$+.691_{10}+0$
VERBETERING	$+.380_{10}+2$	$+.100_{10}+1$	$+.705_{10}+0$	$+.684_{10}+0$
VERBETERING	$+.390_{10}+2$	$+.100_{10}+1$	$+.697_{10}+0$	$+.677_{10}+0$
VERBETERING	$+.400_{10}+2$	$+.100_{10}+1$	$+.689_{10}+0$	$+.671_{10}+0$

368 * 2.- 260472 - 06185479 S ELTENPOO

	n	m	pred	pc
VERBETERING				
	+.410₁₀+ 2	+.100₁₀+ 1	+.681₁₀+ 0	+.664₁₀+ 0
VERBETERING				
	+.420₁₀+ 2	+.100₁₀+ 1	+.672₁₀+ 0	+.657₁₀+ 0
VERBETERING				
	+.430₁₀+ 2	+.100₁₀+ 1	+.664₁₀+ 0	+.651₁₀+ 0
VERBETERING				
	+.440₁₀+ 2	+.100₁₀+ 1	+.656₁₀+ 0	+.644₁₀+ 0
VERBETERING				
	+.450₁₀+ 2	+.100₁₀+ 1	+.648₁₀+ 0	+.638₁₀+ 0
VERBETERING				
	+.460₁₀+ 2	+.100₁₀+ 1	+.640₁₀+ 0	+.632₁₀+ 0
VERBETERING				
	+.470₁₀+ 2	+.100₁₀+ 1	+.632₁₀+ 0	+.625₁₀+ 0
VERBETERING				
	+.480₁₀+ 2	+.100₁₀+ 1	+.624₁₀+ 0	+.619₁₀+ 0
VERBETERING				
	+.490₁₀+ 2	+.100₁₀+ 1	+.616₁₀+ 0	+.613₁₀+ 0
VERBETERING				
	+.500₁₀+ 2	+.100₁₀+ 1	+.607₁₀+ 0	+.607₁₀+ 0
	+.100₁₀+ 1	+.400₁₀+ 1	+.278₁₀+ 0	+.961₁₀+ 0
	+.200₁₀+ 1	+.400₁₀+ 1	+.277₁₀+ 0	+.923₁₀+ 0
	+.300₁₀+ 1	+.400₁₀+ 1	+.276₁₀+ 0	+.887₁₀+ 0
	+.400₁₀+ 1	+.400₁₀+ 1	+.274₁₀+ 0	+.852₁₀+ 0
	+.500₁₀+ 1	+.400₁₀+ 1	+.271₁₀+ 0	+.819₁₀+ 0
	+.600₁₀+ 1	+.400₁₀+ 1	+.268₁₀+ 0	+.787₁₀+ 0
	+.700₁₀+ 1	+.400₁₀+ 1	+.264₁₀+ 0	+.756₁₀+ 0
	+.800₁₀+ 1	+.400₁₀+ 1	+.260₁₀+ 0	+.726₁₀+ 0
	+.900₁₀+ 1	+.400₁₀+ 1	+.256₁₀+ 0	+.698₁₀+ 0
	+.100₁₀+ 2	+.400₁₀+ 1	+.251₁₀+ 0	+.671₁₀+ 0
	+.110₁₀+ 2	+.400₁₀+ 1	+.246₁₀+ 0	+.644₁₀+ 0
	+.120₁₀+ 2	+.400₁₀+ 1	+.241₁₀+ 0	+.619₁₀+ 0
	+.130₁₀+ 2	+.400₁₀+ 1	+.236₁₀+ 0	+.595₁₀+ 0
	+.140₁₀+ 2	+.400₁₀+ 1	+.230₁₀+ 0	+.572₁₀+ 0
	+.150₁₀+ 2	+.400₁₀+ 1	+.224₁₀+ 0	+.549₁₀+ 0
	+.160₁₀+ 2	+.400₁₀+ 1	+.218₁₀+ 0	+.528₁₀+ 0
	+.170₁₀+ 2	+.400₁₀+ 1	+.212₁₀+ 0	+.507₁₀+ 0
	+.180₁₀+ 2	+.400₁₀+ 1	+.206₁₀+ 0	+.487₁₀+ 0
	+.190₁₀+ 2	+.400₁₀+ 1	+.200₁₀+ 0	+.468₁₀+ 0
	+.200₁₀+ 2	+.400₁₀+ 1	+.194₁₀+ 0	+.450₁₀+ 0
	+.210₁₀+ 2	+.400₁₀+ 1	+.188₁₀+ 0	+.432₁₀+ 0
	+.220₁₀+ 2	+.400₁₀+ 1	+.182₁₀+ 0	+.415₁₀+ 0
	+.230₁₀+ 2	+.400₁₀+ 1	+.176₁₀+ 0	+.399₁₀+ 0
	+.240₁₀+ 2	+.400₁₀+ 1	+.169₁₀+ 0	+.383₁₀+ 0
	+.250₁₀+ 2	+.400₁₀+ 1	+.163₁₀+ 0	+.368₁₀+ 0
	+.260₁₀+ 2	+.400₁₀+ 1	+.157₁₀+ 0	+.354₁₀+ 0
	+.270₁₀+ 2	+.400₁₀+ 1	+.152₁₀+ 0	+.340₁₀+ 0
	+.280₁₀+ 2	+.400₁₀+ 1	+.146₁₀+ 0	+.327₁₀+ 0
	+.290₁₀+ 2	+.400₁₀+ 1	+.140₁₀+ 0	+.314₁₀+ 0
	+.300₁₀+ 2	+.400₁₀+ 1	+.135₁₀+ 0	+.302₁₀+ 0
	+.310₁₀+ 2	+.400₁₀+ 1	+.129₁₀+ 0	+.290₁₀+ 0
	+.320₁₀+ 2	+.400₁₀+ 1	+.124₁₀+ 0	+.279₁₀+ 0
	+.330₁₀+ 2	+.400₁₀+ 1	+.119₁₀+ 0	+.268₁₀+ 0
	+.340₁₀+ 2	+.400₁₀+ 1	+.114₁₀+ 0	+.257₁₀+ 0
	+.350₁₀+ 2	+.400₁₀+ 1	+.109₁₀+ 0	+.247₁₀+ 0
	+.360₁₀+ 2	+.400₁₀+ 1	+.104₁₀+ 0	+.237₁₀+ 0
	+.370₁₀+ 2	+.400₁₀+ 1	+.993₁₀- 1	+.228₁₀+ 0
	+.380₁₀+ 2	+.400₁₀+ 1	+.948₁₀- 1	+.219₁₀+ 0
	+.390₁₀+ 2	+.400₁₀+ 1	+.905₁₀- 1	+.211₁₀+ 0
	+.400₁₀+ 2	+.400₁₀+ 1	+.863₁₀- 1	+.202₁₀+ 0

```
   368 -   0 -   260472 - 06185479 STELTENPOO

 1 BEGIN COMMENT BEREKENING PRED EN PC VAN KOLLEKTOR NETWERK DOOR TOEPASSING VAN VOTERNETWERK,
 2       N IS AANTAL INPUTS
 3       M IS AANTAL OUTPUTS;
 4       INTEGER M, T, N, J,
 5       REAL P, P2M, P8M2, PM, P2MN, PN, PC, PRED; J := 0;
 6 B:    T := READ; P := EXP(LN(0.05)/T); FLOT(3, 2, P); NLCR; NLCR;
 7       FOR M := 1 STEP 1 UNTIL 10 DO
 8       BEGIN P2M := P ↑ (2 * M); P8M2 := P ↑ (8 * (M ↑ 2));
 9             PM := P ↑ M; P2MN := 1; PN := 1; PC := 1;
10             FOR N := 1 STEP 1 UNTIL 600 DO
11             BEGIN PN := PN * P;
12                   IF PN < 0,5 THEN GOTO A;
13                   P2MN := P2MN * P2M; PC := PC * PM;
14                   PRED := P2MN * ((3 - 2 * PN)↑ M) * P8M2;
15                   IF PRED < PC THEN GOTO C;
16                   PRINTTEXT(¢VERBETERING¢); NLCR; GOTO E;
17 C:                IF M ≠ 4 THEN GOTO D;
18 E:                SPACE(15); FLOT(3, 2, N); FLOT(3, 2, M);
19                   FLOT(3, 2, PRED); FLOT(3, 2, PC); NLCR;
20 D:
21             END;
22 A:
23       END; J := J + 1; IF J = 1 THEN GOTO B;
24 END
25 EROGEND
```

n	m	pred	pe
+.410₁₀+ 2	+.400₁₀+ 1	+.823₁₀- 1	+.194₁₀+ 0
+.420₁₀+ 2	+.400₁₀+ 1	+.784₁₀- 1	+.187₁₀+ 0
+.430₁₀+ 2	+.400₁₀+ 1	+.746₁₀- 1	+.180₁₀+ 0
+.440₁₀+ 2	+.400₁₀+ 1	+.710₁₀- 1	+.172₁₀+ 0
+.450₁₀+ 2	+.400₁₀+ 1	+.676₁₀- 1	+.166₁₀+ 0
+.460₁₀+ 2	+.400₁₀+ 1	+.642₁₀- 1	+.159₁₀+ 0
+.470₁₀+ 2	+.400₁₀+ 1	+.610₁₀- 1	+.153₁₀+ 0
+.480₁₀+ 2	+.400₁₀+ 1	+.580₁₀- 1	+.147₁₀+ 0
+.490₁₀+ 2	+.400₁₀+ 1	+.550₁₀- 1	+.141₁₀+ 0
+.500₁₀+ 2	+.400₁₀+ 1	+.522₁₀- 1	+.136₁₀+ 0
+.510₁₀+ 2	+.400₁₀+ 1	+.495₁₀- 1	+.130₁₀+ 0
+.520₁₀+ 2	+.400₁₀+ 1	+.469₁₀- 1	+.125₁₀+ 0
+.530₁₀+ 2	+.400₁₀+ 1	+.445₁₀- 1	+.120₁₀+ 0
+.540₁₀+ 2	+.400₁₀+ 1	+.421₁₀- 1	+.116₁₀+ 0
+.550₁₀+ 2	+.400₁₀+ 1	+.399₁₀- 1	+.111₁₀+ 0
+.560₁₀+ 2	+.400₁₀+ 1	+.378₁₀- 1	+.107₁₀+ 0
+.570₁₀+ 2	+.400₁₀+ 1	+.357₁₀- 1	+.103₁₀+ 0
+.580₁₀+ 2	+.400₁₀+ 1	+.338₁₀- 1	+.986₁₀- 1
+.590₁₀+ 2	+.400₁₀+ 1	+.319₁₀- 1	+.947₁₀- 1
+.600₁₀+ 2	+.400₁₀+ 1	+.302₁₀- 1	+.910₁₀- 1
+.610₁₀+ 2	+.400₁₀+ 1	+.285₁₀- 1	+.875₁₀- 1
+.620₁₀+ 2	+.400₁₀+ 1	+.269₁₀- 1	+.840₁₀- 1
+.630₁₀+ 2	+.400₁₀+ 1	+.254₁₀- 1	+.807₁₀- 1
+.640₁₀+ 2	+.400₁₀+ 1	+.240₁₀- 1	+.776₁₀- 1
+.650₁₀+ 2	+.400₁₀+ 1	+.226₁₀- 1	+.745₁₀- 1
+.660₁₀+ 2	+.400₁₀+ 1	+.213₁₀- 1	+.716₁₀- 1
+.670₁₀+ 2	+.400₁₀+ 1	+.201₁₀- 1	+.688₁₀- 1
+.680₁₀+ 2	+.400₁₀+ 1	+.189₁₀- 1	+.661₁₀- 1
+.690₁₀+ 2	+.400₁₀+ 1	+.178₁₀- 1	+.635₁₀- 1

pe = +.995₁₀+ 0

VERBETERING

VERBETERING

VERBETERING

VERBETERING

VERBETERING

VERBETERING

VERBETERING

VERBETERING

VERBETERING

VERBETERING

VERBETERING

VERBETERING

VERBETERING

VERBETERING

VERBETERING

n	m	pred	pe
+.100₁₀+ 2	+.100₁₀+ 1	+.954₁₀+ 0	+.951₁₀+ 0
+.110₁₀+ 2	+.100₁₀+ 1	+.953₁₀+ 0	+.947₁₀+ 0
+.120₁₀+ 2	+.100₁₀+ 1	+.951₁₀+ 0	+.942₁₀+ 0
+.130₁₀+ 2	+.100₁₀+ 1	+.950₁₀+ 0	+.937₁₀+ 0
+.140₁₀+ 2	+.100₁₀+ 1	+.948₁₀+ 0	+.932₁₀+ 0
+.150₁₀+ 2	+.100₁₀+ 1	+.947₁₀+ 0	+.928₁₀+ 0
+.160₁₀+ 2	+.100₁₀+ 1	+.945₁₀+ 0	+.923₁₀+ 0
+.170₁₀+ 2	+.100₁₀+ 1	+.943₁₀+ 0	+.919₁₀+ 0
+.180₁₀+ 2	+.100₁₀+ 1	+.941₁₀+ 0	+.914₁₀+ 0
+.190₁₀+ 2	+.100₁₀+ 1	+.939₁₀+ 0	+.909₁₀+ 0
+.200₁₀+ 2	+.100₁₀+ 1	+.936₁₀+ 0	+.905₁₀+ 0
+.210₁₀+ 2	+.100₁₀+ 1	+.934₁₀+ 0	+.900₁₀+ 0
+.220₁₀+ 2	+.100₁₀+ 1	+.932₁₀+ 0	+.896₁₀+ 0
+.230₁₀+ 2	+.100₁₀+ 1	+.929₁₀+ 0	+.892₁₀+ 0

	r	m	Prod	pc	
	+.240₁₀+ 2	+.100₁₀+ 1	+.927₁₀+ 0	+.887₁₀+ 0	
VERBETERING					
	+.250₁₀+ 2	+.100₁₀+ 1	+.924₁₀+ 0	+.883₁₀+ 0	
VERBETERING					
	+.260₁₀+ 2	+.100₁₀+ 1	+.922₁₀+ 0	+.878₁₀+ 0	
VERBETERING					
	+.270₁₀+ 2	+.100₁₀+ 1	+.919₁₀+ 0	+.874₁₀+ 0	
VERBETERING					
	+.280₁₀+ 2	+.100₁₀+ 1	+.916₁₀+ 0	+.870₁₀+ 0	
VERBETERING					
	+.290₁₀+ 2	+.100₁₀+ 1	+.913₁₀+ 0	+.865₁₀+ 0	
VERBETERING					
	+.300₁₀+ 2	+.100₁₀+ 1	+.910₁₀+ 0	+.861₁₀+ 0	
VERBETERING					
	+.310₁₀+ 2	+.100₁₀+ 1	+.907₁₀+ 0	+.857₁₀+ 0	
VERBETERING					
	+.320₁₀+ 2	+.100₁₀+ 1	+.904₁₀+ 0	+.852₁₀+ 0	
VERBETERING					
	+.330₁₀+ 2	+.100₁₀+ 1	+.901₁₀+ 0	+.848₁₀+ 0	
VERBETERING					
	+.340₁₀+ 2	+.100₁₀+ 1	+.898₁₀+ 0	+.844₁₀+ 0	
VERBETERING					
	+.350₁₀+ 2	+.100₁₀+ 1	+.895₁₀+ 0	+.840₁₀+ 0	
VERBETERING					
	+.360₁₀+ 2	+.100₁₀+ 1	+.891₁₀+ 0	+.835₁₀+ 0	
VERBETERING					
	+.370₁₀+ 2	+.100₁₀+ 1	+.888₁₀+ 0	+.831₁₀+ 0	
VERBETERING					
	+.380₁₀+ 2	+.100₁₀+ 1	+.885₁₀+ 0	+.827₁₀+ 0	
VERBETERING					
	+.390₁₀+ 2	+.100₁₀+ 1	+.881₁₀+ 0	+.823₁₀+ 0	
VERBETERING					
	+.400₁₀+ 2	+.100₁₀+ 1	+.878₁₀+ 0	+.819₁₀+ 0	
VERBETERING					
	+.410₁₀+ 2	+.100₁₀+ 1	+.874₁₀+ 0	+.815₁₀+ 0	
VERBETERING					
	+.420₁₀+ 2	+.100₁₀+ 1	+.871₁₀+ 0	+.811₁₀+ 0	
VERBETERING					
	+.430₁₀+ 2	+.100₁₀+ 1	+.867₁₀+ 0	+.807₁₀+ 0	
VERBETERING					
	+.440₁₀+ 2	+.100₁₀+ 1	+.863₁₀+ 0	+.803₁₀+ 0	
VERBETERING					
	+.450₁₀+ 2	+.100₁₀+ 1	+.860₁₀+ 0	+.799₁₀+ 0	
VERBETERING					
	+.460₁₀+ 2	+.100₁₀+ 1	+.856₁₀+ 0	+.795₁₀+ 0	
VERBETERING					
	+.470₁₀+ 2	+.100₁₀+ 1	+.852₁₀+ 0	+.791₁₀+ 0	
VERBETERING					
	+.480₁₀+ 2	+.100₁₀+ 1	+.849₁₀+ 0	+.787₁₀+ 0	
VERBETERING					
	+.490₁₀+ 2	+.100₁₀+ 1	+.845₁₀+ 0	+.783₁₀+ 0	
VERBETERING					
	+.500₁₀+ 2	+.100₁₀+ 1	+.841₁₀+ 0	+.779₁₀+ 0	
VERBETERING					
	+.510₁₀+ 2	+.100₁₀+ 1	+.837₁₀+ 0	+.775₁₀+ 0	
VERBETERING					
	+.520₁₀+ 2	+.100₁₀+ 1	+.833₁₀+ 0	+.771₁₀+ 0	
VERBETERING					
	+.530₁₀+ 2	+.100₁₀+ 1	+.829₁₀+ 0	+.767₁₀+ 0	
VERBETERING					

	n	m	prod	pe
	+.540₁₀+ 2	+.100₁₀+ 1	+.825₁₀+ 0	+.764₁₀+ 0
VERBETERING				
	+.550₁₀+ 2	+.100₁₀+ 1	+.821₁₀+ 0	+.760₁₀+ 0
VERBETERING				
	+.560₁₀+ 2	+.100₁₀+ 1	+.817₁₀+ 0	+.756₁₀+ 0
VERBETERING				
	+.570₁₀+ 2	+.100₁₀+ 1	+.813₁₀+ 0	+.752₁₀+ 0
VERBETERING				
	+.580₁₀+ 2	+.100₁₀+ 1	+.809₁₀+ 0	+.749₁₀+ 0
VERBETERING				
	+.590₁₀+ 2	+.100₁₀+ 1	+.805₁₀+ 0	+.745₁₀+ 0
VERBETERING				
	+.600₁₀+ 2	+.100₁₀+ 1	+.801₁₀+ 0	+.741₁₀+ 0
VERBETERING				
	+.610₁₀+ 2	+.100₁₀+ 1	+.797₁₀+ 0	+.737₁₀+ 0
VERBETERING				
	+.620₁₀+ 2	+.100₁₀+ 1	+.793₁₀+ 0	+.734₁₀+ 0
VERBETERING				
	+.630₁₀+ 2	+.100₁₀+ 1	+.789₁₀+ 0	+.730₁₀+ 0
VERBETERING				
	+.640₁₀+ 2	+.100₁₀+ 1	+.785₁₀+ 0	+.726₁₀+ 0
VERBETERING				
	+.650₁₀+ 2	+.100₁₀+ 1	+.780₁₀+ 0	+.723₁₀+ 0
VERBETERING				
	+.660₁₀+ 2	+.100₁₀+ 1	+.776₁₀+ 0	+.719₁₀+ 0
VERBETERING				
	+.670₁₀+ 2	+.100₁₀+ 1	+.772₁₀+ 0	+.716₁₀+ 0
VERBETERING				
	+.680₁₀+ 2	+.100₁₀+ 1	+.768₁₀+ 0	+.712₁₀+ 0
VERBETERING				
	+.690₁₀+ 2	+.100₁₀+ 1	+.764₁₀+ 0	+.709₁₀+ 0
VERBETERING				
	+.700₁₀+ 2	+.100₁₀+ 1	+.759₁₀+ 0	+.705₁₀+ 0
VERBETERING				
	+.710₁₀+ 2	+.100₁₀+ 1	+.755₁₀+ 0	+.702₁₀+ 0
VERBETERING				
	+.720₁₀+ 2	+.100₁₀+ 1	+.751₁₀+ 0	+.698₁₀+ 0
VERBETERING				
	+.730₁₀+ 2	+.100₁₀+ 1	+.747₁₀+ 0	+.695₁₀+ 0
VERBETERING				
	+.740₁₀+ 2	+.100₁₀+ 1	+.742₁₀+ 0	+.691₁₀+ 0
VERBETERING				
	+.750₁₀+ 2	+.100₁₀+ 1	+.738₁₀+ 0	+.688₁₀+ 0
VERBETERING				
	+.760₁₀+ 2	+.100₁₀+ 1	+.734₁₀+ 0	+.684₁₀+ 0
VERBETERING				
	+.770₁₀+ 2	+.100₁₀+ 1	+.730₁₀+ 0	+.681₁₀+ 0
VERBETERING				
	+.780₁₀+ 2	+.100₁₀+ 1	+.725₁₀+ 0	+.677₁₀+ 0
VERBETERING				
	+.790₁₀+ 2	+.100₁₀+ 1	+.721₁₀+ 0	+.674₁₀+ 0
VERBETERING				
	+.800₁₀+ 2	+.100₁₀+ 1	+.717₁₀+ 0	+.671₁₀+ 0
VERBETERING				
	+.810₁₀+ 2	+.100₁₀+ 1	+.713₁₀+ 0	+.667₁₀+ 0
VERBETERING				
	+.820₁₀+ 2	+.100₁₀+ 1	+.708₁₀+ 0	+.664₁₀+ 0
VERBETERING				
	+.830₁₀+ 2	+.100₁₀+ 1	+.704₁₀+ 0	+.661₁₀+ 0
VERBETERING				

	n		m		pred		pc	
	+.840$_{10}$+	2	+.100$_{10}$+	1	+.700$_{10}$+	0	+.657$_{10}$+	0
VERBETERING								
	+.850$_{10}$+	2	+.100$_{10}$+	1	+.696$_{10}$+	0	+.654$_{10}$+	0
VERBETERING								
	+.860$_{10}$+	2	+.100$_{10}$+	1	+.691$_{10}$+	0	+.651$_{10}$+	0
VERBETERING								
	+.870$_{10}$+	2	+.100$_{10}$+	1	+.687$_{10}$+	0	+.648$_{10}$+	0
VERBETERING								
	+.880$_{10}$+	2	+.100$_{10}$+	1	+.683$_{10}$+	0	+.644$_{10}$+	0
VERBETERING								
	+.890$_{10}$+	2	+.100$_{10}$+	1	+.679$_{10}$+	0	+.641$_{10}$+	0
VERBETERING								
	+.900$_{10}$+	2	+.100$_{10}$+	1	+.674$_{10}$+	0	+.638$_{10}$+	0
VERBETERING								
	+.910$_{10}$+	2	+.100$_{10}$+	1	+.670$_{10}$+	0	+.635$_{10}$+	0
VERBETERING								
	+.920$_{10}$+	2	+.100$_{10}$+	1	+.666$_{10}$+	0	+.632$_{10}$+	0
VERBETERING								
	+.930$_{10}$+	2	+.100$_{10}$+	1	+.662$_{10}$+	0	+.629$_{10}$+	0
VERBETERING								
	+.940$_{10}$+	2	+.100$_{10}$+	1	+.657$_{10}$+	0	+.625$_{10}$+	0
VERBETERING								
	+.950$_{10}$+	2	+.100$_{10}$+	1	+.653$_{10}$+	0	+.622$_{10}$+	0
VERBETERING								
	+.960$_{10}$+	2	+.100$_{10}$+	1	+.649$_{10}$+	0	+.619$_{10}$+	0
VERBETERING								
	+.970$_{10}$+	2	+.100$_{10}$+	1	+.645$_{10}$+	0	+.616$_{10}$+	0
VERBETERING								
	+.980$_{10}$+	2	+.100$_{10}$+	1	+.641$_{10}$+	0	+.613$_{10}$+	0
VERBETERING								
	+.990$_{10}$+	2	+.100$_{10}$+	1	+.636$_{10}$+	0	+.610$_{10}$+	0
VERBETERING								
	+.100$_{10}$+	3	+.100$_{10}$+	1	+.632$_{10}$+	0	+.607$_{10}$+	0
VERBETERING								
	+.101$_{10}$+	3	+.100$_{10}$+	1	+.628$_{10}$+	0	+.604$_{10}$+	0
VERBETERING								
	+.102$_{10}$+	3	+.100$_{10}$+	1	+.624$_{10}$+	0	+.601$_{10}$+	0
VERBETERING								
	+.103$_{10}$+	3	+.100$_{10}$+	1	+.620$_{10}$+	0	+.598$_{10}$+	0
VERBETERING								
	+.104$_{10}$+	3	+.100$_{10}$+	1	+.616$_{10}$+	0	+.595$_{10}$+	0
VERBETERING								
	+.105$_{10}$+	3	+.100$_{10}$+	1	+.612$_{10}$+	0	+.592$_{10}$+	0
VERBETERING								
	+.106$_{10}$+	3	+.100$_{10}$+	1	+.607$_{10}$+	0	+.589$_{10}$+	0
VERBETERING								
	+.107$_{10}$+	3	+.100$_{10}$+	1	+.603$_{10}$+	0	+.586$_{10}$+	0
VERBETERING								
	+.108$_{10}$+	3	+.100$_{10}$+	1	+.599$_{10}$+	0	+.583$_{10}$+	0
VERBETERING								
	+.109$_{10}$+	3	+.100$_{10}$+	1	+.595$_{10}$+	0	+.580$_{10}$+	0
VERBETERING								
	+.110$_{10}$+	3	+.100$_{10}$+	1	+.591$_{10}$+	0	+.577$_{10}$+	0
VERBETERING								
	+.111$_{10}$+	3	+.100$_{10}$+	1	+.587$_{10}$+	0	+.575$_{10}$+	0
VERBETERING								
	+.112$_{10}$+	3	+.100$_{10}$+	1	+.583$_{10}$+	0	+.572$_{10}$+	0
VERBETERING								
	+.113$_{10}$+	3	+.100$_{10}$+	1	+.579$_{10}$+	0	+.569$_{10}$+	0
VERBETERING								

	σ	m	pred	pc
	+.114₁₀+ 3	+.100₁₀+ 1	+.575₁₀+ 0	+.566₁₀+ 0
VERBETERING				
	+.115₁₀+ 3	+.100₁₀+ 1	+.571₁₀+ 0	+.563₁₀+ 0
VERBETERING				
	+.116₁₀+ 3	+.100₁₀+ 1	+.567₁₀+ 0	+.560₁₀+ 0
VERBETERING				
	+.117₁₀+ 3	+.100₁₀+ 1	+.563₁₀+ 0	+.558₁₀+ 0
VERBETERING				
	+.118₁₀+ 3	+.100₁₀+ 1	+.559₁₀+ 0	+.555₁₀+ 0
VERBETERING				
	+.119₁₀+ 3	+.100₁₀+ 1	+.555₁₀+ 0	+.552₁₀+ 0
VERBETERING				
	+.120₁₀+ 3	+.100₁₀+ 1	+.551₁₀+ 0	+.549₁₀+ 0
VERBETERING				
	+.121₁₀+ 3	+.100₁₀+ 1	+.547₁₀+ 0	+.547₁₀+ 0
VERBETERING				
	+.230₁₀+ 2	+.200₁₀+ 1	+.797₁₀+ 0	+.795₁₀+ 0
VERBETERING				
	+.240₁₀+ 2	+.200₁₀+ 1	+.793₁₀+ 0	+.787₁₀+ 0
VERBETERING				
	+.250₁₀+ 2	+.200₁₀+ 1	+.789₁₀+ 0	+.779₁₀+ 0
VERBETERING				
	+.260₁₀+ 2	+.200₁₀+ 1	+.784₁₀+ 0	+.771₁₀+ 0
VERBETERING				
	+.270₁₀+ 2	+.200₁₀+ 1	+.779₁₀+ 0	+.764₁₀+ 0
VERBETERING				
	+.280₁₀+ 2	+.200₁₀+ 1	+.775₁₀+ 0	+.756₁₀+ 0
VERBETERING				
	+.290₁₀+ 2	+.200₁₀+ 1	+.770₁₀+ 0	+.749₁₀+ 0
VERBETERING				
	+.300₁₀+ 2	+.200₁₀+ 1	+.765₁₀+ 0	+.741₁₀+ 0
VERBETERING				
	+.310₁₀+ 2	+.200₁₀+ 1	+.760₁₀+ 0	+.734₁₀+ 0
VERBETERING				
	+.320₁₀+ 2	+.200₁₀+ 1	+.755₁₀+ 0	+.726₁₀+ 0
VERBETERING				
	+.330₁₀+ 2	+.200₁₀+ 1	+.750₁₀+ 0	+.719₁₀+ 0
VERBETERING				
	+.340₁₀+ 2	+.200₁₀+ 1	+.744₁₀+ 0	+.712₁₀+ 0
VERBETERING				
	+.350₁₀+ 2	+.200₁₀+ 1	+.739₁₀+ 0	+.705₁₀+ 0
VERBETERING				
	+.360₁₀+ 2	+.200₁₀+ 1	+.734₁₀+ 0	+.698₁₀+ 0
VERBETERING				
	+.370₁₀+ 2	+.200₁₀+ 1	+.728₁₀+ 0	+.691₁₀+ 0
VERBETERING				
	+.380₁₀+ 2	+.200₁₀+ 1	+.723₁₀+ 0	+.684₁₀+ 0
VERBETERING				
	+.390₁₀+ 2	+.200₁₀+ 1	+.717₁₀+ 0	+.677₁₀+ 0
VERBETERING				
	+.400₁₀+ 2	+.200₁₀+ 1	+.711₁₀+ 0	+.671₁₀+ 0
VERBETERING				
	+.410₁₀+ 2	+.200₁₀+ 1	+.706₁₀+ 0	+.664₁₀+ 0
VERBETERING				
	+.420₁₀+ 2	+.200₁₀+ 1	+.700₁₀+ 0	+.657₁₀+ 0
VERBETERING				
	+.430₁₀+ 2	+.200₁₀+ 1	+.694₁₀+ 0	+.651₁₀+ 0
VERBETERING				
	+.440₁₀+ 2	+.200₁₀+ 1	+.688₁₀+ 0	+.644₁₀+ 0
VERBETERING				

	p	m	pred	pc
	+.450₁₀+ 2	+.200₁₀+ 1	+.682₁₀+ 0	+.638₁₀+ 0
VERBETERING				
	+.460₁₀+ 2	+.200₁₀+ 1	+.677₁₀+ 0	+.632₁₀+ 0
VERBETERING				
	+.470₁₀+ 2	+.200₁₀+ 1	+.671₁₀+ 0	+.625₁₀+ 0
VERBETERING				
	+.480₁₀+ 2	+.200₁₀+ 1	+.665₁₀+ 0	+.619₁₀+ 0
VERBETERING				
	+.490₁₀+ 2	+.200₁₀+ 1	+.659₁₀+ 0	+.613₁₀+ 0
VERBETERING				
	+.500₁₀+ 2	+.200₁₀+ 1	+.653₁₀+ 0	+.607₁₀+ 0
VERBETERING				
	+.510₁₀+ 2	+.200₁₀+ 1	+.647₁₀+ 0	+.601₁₀+ 0
VERBETERING				
	+.520₁₀+ 2	+.200₁₀+ 1	+.641₁₀+ 0	+.595₁₀+ 0
VERBETERING				
	+.530₁₀+ 2	+.200₁₀+ 1	+.635₁₀+ 0	+.589₁₀+ 0
VERBETERING				
	+.540₁₀+ 2	+.200₁₀+ 1	+.629₁₀+ 0	+.583₁₀+ 0
VERBETERING				
	+.550₁₀+ 2	+.200₁₀+ 1	+.623₁₀+ 0	+.577₁₀+ 0
VERBETERING				
	+.560₁₀+ 2	+.200₁₀+ 1	+.617₁₀+ 0	+.572₁₀+ 0
VERBETERING				
	+.570₁₀+ 2	+.200₁₀+ 1	+.611₁₀+ 0	+.566₁₀+ 0
VERBETERING				
	+.580₁₀+ 2	+.200₁₀+ 1	+.604₁₀+ 0	+.560₁₀+ 0
VERBETERING				
	+.590₁₀+ 2	+.200₁₀+ 1	+.598₁₀+ 0	+.555₁₀+ 0
VERBETERING				
	+.600₁₀+ 2	+.200₁₀+ 1	+.592₁₀+ 0	+.549₁₀+ 0
VERBETERING				
	+.610₁₀+ 2	+.200₁₀+ 1	+.586₁₀+ 0	+.544₁₀+ 0
VERBETERING				
	+.620₁₀+ 2	+.200₁₀+ 1	+.580₁₀+ 0	+.538₁₀+ 0
VERBETERING				
	+.630₁₀+ 2	+.200₁₀+ 1	+.574₁₀+ 0	+.533₁₀+ 0
VERBETERING				
	+.640₁₀+ 2	+.200₁₀+ 1	+.568₁₀+ 0	+.528₁₀+ 0
VERBETERING				
	+.650₁₀+ 2	+.200₁₀+ 1	+.562₁₀+ 0	+.523₁₀+ 0
VERBETERING				
	+.660₁₀+ 2	+.200₁₀+ 1	+.556₁₀+ 0	+.517₁₀+ 0
VERBETERING				
	+.670₁₀+ 2	+.200₁₀+ 1	+.550₁₀+ 0	+.512₁₀+ 0
VERBETERING				
	+.680₁₀+ 2	+.200₁₀+ 1	+.544₁₀+ 0	+.507₁₀+ 0
VERBETERING				
	+.690₁₀+ 2	+.200₁₀+ 1	+.538₁₀+ 0	+.502₁₀+ 0
VERBETERING				
	+.700₁₀+ 2	+.200₁₀+ 1	+.532₁₀+ 0	+.497₁₀+ 0
VERBETERING				
	+.710₁₀+ 2	+.200₁₀+ 1	+.526₁₀+ 0	+.492₁₀+ 0
VERBETERING				
	+.720₁₀+ 2	+.200₁₀+ 1	+.521₁₀+ 0	+.487₁₀+ 0
VERBETERING				
	+.730₁₀+ 2	+.200₁₀+ 1	+.515₁₀+ 0	+.482₁₀+ 0
VERBETERING				
	+.740₁₀+ 2	+.200₁₀+ 1	+.509₁₀+ 0	+.478₁₀+ 0
VERBETERING				

360 - 0 - 260472 - 0618547? S'ELTENROO

	n	m	Prod	Pc
	+.750$_{10}$+ 2	+.200$_{10}$+ 1	+.503$_{10}$+ 0	+.473$_{10}$+ 0
VERBETERING				
	+.760$_{10}$+ 2	+.200$_{10}$+ 1	+.497$_{10}$+ 0	+.468$_{10}$+ 0
VERBETERING				
	+.770$_{10}$+ 2	+.200$_{10}$+ 1	+.492$_{10}$+ 0	+.464$_{10}$+ 0
VERBETERING				
	+.780$_{10}$+ 2	+.200$_{10}$+ 1	+.486$_{10}$+ 0	+.459$_{10}$+ 0
VERBETERING				
	+.790$_{10}$+ 2	+.200$_{10}$+ 1	+.480$_{10}$+ 0	+.454$_{10}$+ 0
VERBETERING				
	+.800$_{10}$+ 2	+.200$_{10}$+ 1	+.474$_{10}$+ 0	+.450$_{10}$+ 0
VERBETERING				
	+.810$_{10}$+ 2	+.200$_{10}$+ 1	+.469$_{10}$+ 0	+.445$_{10}$+ 0
VERBETERING				
	+.820$_{10}$+ 2	+.200$_{10}$+ 1	+.463$_{10}$+ 0	+.441$_{10}$+ 0
VERBETERING				
	+.830$_{10}$+ 2	+.200$_{10}$+ 1	+.458$_{10}$+ 0	+.437$_{10}$+ 0
VERBETERING				
	+.840$_{10}$+ 2	+.200$_{10}$+ 1	+.452$_{10}$+ 0	+.432$_{10}$+ 0
VERBETERING				
	+.850$_{10}$+ 2	+.200$_{10}$+ 1	+.447$_{10}$+ 0	+.428$_{10}$+ 0
VERBETERING				
	+.860$_{10}$+ 2	+.200$_{10}$+ 1	+.441$_{10}$+ 0	+.424$_{10}$+ 0
VERBETERING				
	+.870$_{10}$+ 2	+.200$_{10}$+ 1	+.436$_{10}$+ 0	+.419$_{10}$+ 0
VERBETERING				
	+.880$_{10}$+ 2	+.200$_{10}$+ 1	+.430$_{10}$+ 0	+.415$_{10}$+ 0
VERBETERING				
	+.890$_{10}$+ 2	+.200$_{10}$+ 1	+.425$_{10}$+ 0	+.411$_{10}$+ 0
VERBETERING				
	+.900$_{10}$+ 2	+.200$_{10}$+ 1	+.420$_{10}$+ 0	+.407$_{10}$+ 0
VERBETERING				
	+.910$_{10}$+ 2	+.200$_{10}$+ 1	+.415$_{10}$+ 0	+.403$_{10}$+ 0
VERBETERING				
	+.920$_{10}$+ 2	+.200$_{10}$+ 1	+.409$_{10}$+ 0	+.399$_{10}$+ 0
VERBETERING				
	+.930$_{10}$+ 2	+.200$_{10}$+ 1	+.404$_{10}$+ 0	+.395$_{10}$+ 0
VERBETERING				
	+.940$_{10}$+ 2	+.200$_{10}$+ 1	+.399$_{10}$+ 0	+.391$_{10}$+ 0
VERBETERING				
	+.950$_{10}$+ 2	+.200$_{10}$+ 1	+.394$_{10}$+ 0	+.387$_{10}$+ 0
VERBETERING				
	+.960$_{10}$+ 2	+.200$_{10}$+ 1	+.389$_{10}$+ 0	+.383$_{10}$+ 0
VERBETERING				
	+.970$_{10}$+ 2	+.200$_{10}$+ 1	+.384$_{10}$+ 0	+.380$_{10}$+ 0
VERBETERING				
	+.980$_{10}$+ 2	+.200$_{10}$+ 1	+.379$_{10}$+ 0	+.376$_{10}$+ 0
VERBETERING				
	+.990$_{10}$+ 2	+.200$_{10}$+ 1	+.374$_{10}$+ 0	+.372$_{10}$+ 0
VERBETERING				
	+.100$_{10}$+ 3	+.200$_{10}$+ 1	+.369$_{10}$+ 0	+.368$_{10}$+ 0
	+.100$_{10}$+ 1	+.400$_{10}$+ 1	+.528$_{10}$+ 0	+.930$_{10}$+ 0
	+.200$_{10}$+ 1	+.400$_{10}$+ 1	+.527$_{10}$+ 0	+.961$_{10}$+ 0
	+.300$_{10}$+ 1	+.400$_{10}$+ 1	+.526$_{10}$+ 0	+.942$_{10}$+ 0
	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.525$_{10}$+ 0	+.923$_{10}$+ 0
	+.500$_{10}$+ 1	+.400$_{10}$+ 1	+.524$_{10}$+ 0	+.905$_{10}$+ 0
	+.600$_{10}$+ 1	+.400$_{10}$+ 1	+.522$_{10}$+ 0	+.887$_{10}$+ 0
	+.700$_{10}$+ 1	+.400$_{10}$+ 1	+.521$_{10}$+ 0	+.870$_{10}$+ 0
	+.800$_{10}$+ 1	+.400$_{10}$+ 1	+.518$_{10}$+ 0	+.852$_{10}$+ 0
	+.900$_{10}$+ 1	+.400$_{10}$+ 1	+.516$_{10}$+ 0	+.835$_{10}$+ 0

n	m	pred	pc
+.100$_{10}$+ 2	+.400$_{10}$+ 1	+.513$_{10}$+ 0	+.819$_{10}$+ 0
+.110$_{10}$+ 2	+.400$_{10}$+ 1	+.511$_{10}$+ 0	+.803$_{10}$+ 0
+.120$_{10}$+ 2	+.400$_{10}$+ 1	+.507$_{10}$+ 0	+.787$_{10}$+ 0
+.130$_{10}$+ 2	+.400$_{10}$+ 1	+.504$_{10}$+ 0	+.771$_{10}$+ 0
+.140$_{10}$+ 2	+.400$_{10}$+ 1	+.501$_{10}$+ 0	+.756$_{10}$+ 0
+.150$_{10}$+ 2	+.400$_{10}$+ 1	+.497$_{10}$+ 0	+.741$_{10}$+ 0
+.160$_{10}$+ 2	+.400$_{10}$+ 1	+.493$_{10}$+ 0	+.726$_{10}$+ 0
+.170$_{10}$+ 2	+.400$_{10}$+ 1	+.489$_{10}$+ 0	+.712$_{10}$+ 0
+.180$_{10}$+ 2	+.400$_{10}$+ 1	+.485$_{10}$+ 0	+.698$_{10}$+ 0
+.190$_{10}$+ 2	+.400$_{10}$+ 1	+.481$_{10}$+ 0	+.684$_{10}$+ 0
+.200$_{10}$+ 2	+.400$_{10}$+ 1	+.476$_{10}$+ 0	+.671$_{10}$+ 0
+.210$_{10}$+ 2	+.400$_{10}$+ 1	+.472$_{10}$+ 0	+.657$_{10}$+ 0
+.220$_{10}$+ 2	+.400$_{10}$+ 1	+.467$_{10}$+ 0	+.644$_{10}$+ 0
+.230$_{10}$+ 2	+.400$_{10}$+ 1	+.462$_{10}$+ 0	+.632$_{10}$+ 0
+.240$_{10}$+ 2	+.400$_{10}$+ 1	+.457$_{10}$+ 0	+.619$_{10}$+ 0
+.250$_{10}$+ 2	+.400$_{10}$+ 1	+.452$_{10}$+ 0	+.607$_{10}$+ 0
+.260$_{10}$+ 2	+.400$_{10}$+ 1	+.447$_{10}$+ 0	+.595$_{10}$+ 0
+.270$_{10}$+ 2	+.400$_{10}$+ 1	+.441$_{10}$+ 0	+.583$_{10}$+ 0
+.280$_{10}$+ 2	+.400$_{10}$+ 1	+.436$_{10}$+ 0	+.572$_{10}$+ 0
+.290$_{10}$+ 2	+.400$_{10}$+ 1	+.431$_{10}$+ 0	+.560$_{10}$+ 0
+.300$_{10}$+ 2	+.400$_{10}$+ 1	+.425$_{10}$+ 0	+.549$_{10}$+ 0
+.310$_{10}$+ 2	+.400$_{10}$+ 1	+.419$_{10}$+ 0	+.538$_{10}$+ 0
+.320$_{10}$+ 2	+.400$_{10}$+ 1	+.414$_{10}$+ 0	+.528$_{10}$+ 0
+.330$_{10}$+ 2	+.400$_{10}$+ 1	+.408$_{10}$+ 0	+.517$_{10}$+ 0
+.340$_{10}$+ 2	+.400$_{10}$+ 1	+.402$_{10}$+ 0	+.507$_{10}$+ 0
+.350$_{10}$+ 2	+.400$_{10}$+ 1	+.397$_{10}$+ 0	+.497$_{10}$+ 0
+.360$_{10}$+ 2	+.400$_{10}$+ 1	+.391$_{10}$+ 0	+.487$_{10}$+ 0
+.370$_{10}$+ 2	+.400$_{10}$+ 1	+.385$_{10}$+ 0	+.478$_{10}$+ 0
+.380$_{10}$+ 2	+.400$_{10}$+ 1	+.379$_{10}$+ 0	+.468$_{10}$+ 0
+.390$_{10}$+ 2	+.400$_{10}$+ 1	+.373$_{10}$+ 0	+.459$_{10}$+ 0
+.400$_{10}$+ 2	+.400$_{10}$+ 1	+.368$_{10}$+ 0	+.450$_{10}$+ 0
+.410$_{10}$+ 2	+.400$_{10}$+ 1	+.362$_{10}$+ 0	+.441$_{10}$+ 0
+.420$_{10}$+ 2	+.400$_{10}$+ 1	+.356$_{10}$+ 0	+.432$_{10}$+ 0
+.430$_{10}$+ 2	+.400$_{10}$+ 1	+.350$_{10}$+ 0	+.424$_{10}$+ 0
+.440$_{10}$+ 2	+.400$_{10}$+ 1	+.344$_{10}$+ 0	+.415$_{10}$+ 0
+.450$_{10}$+ 2	+.400$_{10}$+ 1	+.338$_{10}$+ 0	+.407$_{10}$+ 0
+.460$_{10}$+ 2	+.400$_{10}$+ 1	+.333$_{10}$+ 0	+.399$_{10}$+ 0
+.470$_{10}$+ 2	+.400$_{10}$+ 1	+.327$_{10}$+ 0	+.391$_{10}$+ 0
+.480$_{10}$+ 2	+.400$_{10}$+ 1	+.321$_{10}$+ 0	+.383$_{10}$+ 0
+.490$_{10}$+ 2	+.400$_{10}$+ 1	+.315$_{10}$+ 0	+.376$_{10}$+ 0
+.500$_{10}$+ 2	+.400$_{10}$+ 1	+.310$_{10}$+ 0	+.368$_{10}$+ 0
+.510$_{10}$+ 2	+.400$_{10}$+ 1	+.304$_{10}$+ 0	+.361$_{10}$+ 0
+.520$_{10}$+ 2	+.400$_{10}$+ 1	+.298$_{10}$+ 0	+.354$_{10}$+ 0
+.530$_{10}$+ 2	+.400$_{10}$+ 1	+.293$_{10}$+ 0	+.347$_{10}$+ 0
+.540$_{10}$+ 2	+.400$_{10}$+ 1	+.287$_{10}$+ 0	+.340$_{10}$+ 0
+.550$_{10}$+ 2	+.400$_{10}$+ 1	+.282$_{10}$+ 0	+.333$_{10}$+ 0
+.560$_{10}$+ 2	+.400$_{10}$+ 1	+.276$_{10}$+ 0	+.327$_{10}$+ 0
+.570$_{10}$+ 2	+.400$_{10}$+ 1	+.271$_{10}$+ 0	+.320$_{10}$+ 0
+.580$_{10}$+ 2	+.400$_{10}$+ 1	+.265$_{10}$+ 0	+.314$_{10}$+ 0
+.590$_{10}$+ 2	+.400$_{10}$+ 1	+.260$_{10}$+ 0	+.308$_{10}$+ 0
+.600$_{10}$+ 2	+.400$_{10}$+ 1	+.255$_{10}$+ 0	+.302$_{10}$+ 0
+.610$_{10}$+ 2	+.400$_{10}$+ 1	+.250$_{10}$+ 0	+.296$_{10}$+ 0
+.620$_{10}$+ 2	+.400$_{10}$+ 1	+.245$_{10}$+ 0	+.290$_{10}$+ 0
+.630$_{10}$+ 2	+.400$_{10}$+ 1	+.240$_{10}$+ 0	+.284$_{10}$+ 0
+.640$_{10}$+ 2	+.400$_{10}$+ 1	+.235$_{10}$+ 0	+.279$_{10}$+ 0
+.650$_{10}$+ 2	+.400$_{10}$+ 1	+.230$_{10}$+ 0	+.273$_{10}$+ 0
+.660$_{10}$+ 2	+.400$_{10}$+ 1	+.225$_{10}$+ 0	+.268$_{10}$+ 0
+.670$_{10}$+ 2	+.400$_{10}$+ 1	+.220$_{10}$+ 0	+.262$_{10}$+ 0
+.680$_{10}$+ 2	+.400$_{10}$+ 1	+.215$_{10}$+ 0	+.257$_{10}$+ 0
+.690$_{10}$+ 2	+.400$_{10}$+ 1	+.211$_{10}$+ 0	+.252$_{10}$+ 0

n	m	prod	Pc
+.700$_{10}$+ 2	+.400$_{10}$+ 1	+.206$_{10}$+ 0	+.247$_{10}$+ 0
+.710$_{10}$+ 2	+.400$_{10}$+ 1	+.201$_{10}$+ 0	+.242$_{10}$+ 0
+.720$_{10}$+ 2	+.400$_{10}$+ 1	+.197$_{10}$+ 0	+.237$_{10}$+ 0
+.730$_{10}$+ 2	+.400$_{10}$+ 1	+.192$_{10}$+ 0	+.233$_{10}$+ 0
+.740$_{10}$+ 2	+.400$_{10}$+ 1	+.188$_{10}$+ 0	+.228$_{10}$+ 0
+.750$_{10}$+ 2	+.400$_{10}$+ 1	+.184$_{10}$+ 0	+.224$_{10}$+ 0
+.760$_{10}$+ 2	+.400$_{10}$+ 1	+.180$_{10}$+ 0	+.219$_{10}$+ 0
+.770$_{10}$+ 2	+.400$_{10}$+ 1	+.176$_{10}$+ 0	+.215$_{10}$+ 0
+.780$_{10}$+ 2	+.400$_{10}$+ 1	+.171$_{10}$+ 0	+.211$_{10}$+ 0
+.790$_{10}$+ 2	+.400$_{10}$+ 1	+.167$_{10}$+ 0	+.206$_{10}$+ 0
+.800$_{10}$+ 2	+.400$_{10}$+ 1	+.164$_{10}$+ 0	+.202$_{10}$+ 0
+.810$_{10}$+ 2	+.400$_{10}$+ 1	+.160$_{10}$+ 0	+.198$_{10}$+ 0
+.820$_{10}$+ 2	+.400$_{10}$+ 1	+.156$_{10}$+ 0	+.194$_{10}$+ 0
+.830$_{10}$+ 2	+.400$_{10}$+ 1	+.152$_{10}$+ 0	+.191$_{10}$+ 0
+.840$_{10}$+ 2	+.400$_{10}$+ 1	+.149$_{10}$+ 0	+.187$_{10}$+ 0
+.850$_{10}$+ 2	+.400$_{10}$+ 1	+.145$_{10}$+ 0	+.183$_{10}$+ 0
+.860$_{10}$+ 2	+.400$_{10}$+ 1	+.141$_{10}$+ 0	+.180$_{10}$+ 0
+.870$_{10}$+ 2	+.400$_{10}$+ 1	+.138$_{10}$+ 0	+.176$_{10}$+ 0
+.880$_{10}$+ 2	+.400$_{10}$+ 1	+.135$_{10}$+ 0	+.172$_{10}$+ 0
+.890$_{10}$+ 2	+.400$_{10}$+ 1	+.131$_{10}$+ 0	+.169$_{10}$+ 0
+.900$_{10}$+ 2	+.400$_{10}$+ 1	+.128$_{10}$+ 0	+.166$_{10}$+ 0
+.910$_{10}$+ 2	+.400$_{10}$+ 1	+.125$_{10}$+ 0	+.162$_{10}$+ 0
+.920$_{10}$+ 2	+.400$_{10}$+ 1	+.122$_{10}$+ 0	+.159$_{10}$+ 0
+.930$_{10}$+ 2	+.400$_{10}$+ 1	+.119$_{10}$+ 0	+.156$_{10}$+ 0
+.940$_{10}$+ 2	+.400$_{10}$+ 1	+.116$_{10}$+ 0	+.153$_{10}$+ 0
+.950$_{10}$+ 2	+.400$_{10}$+ 1	+.113$_{10}$+ 0	+.150$_{10}$+ 0
+.960$_{10}$+ 2	+.400$_{10}$+ 1	+.110$_{10}$+ 0	+.147$_{10}$+ 0
+.970$_{10}$+ 2	+.400$_{10}$+ 1	+.107$_{10}$+ 0	+.144$_{10}$+ 0
+.980$_{10}$+ 2	+.400$_{10}$+ 1	+.104$_{10}$+ 0	+.141$_{10}$+ 0
+.990$_{10}$+ 2	+.400$_{10}$+ 1	+.102$_{10}$+ 0	+.138$_{10}$+ 0
+.100$_{10}$+ 3	+.400$_{10}$+ 1	+.989$_{10}$- 1	+.136$_{10}$+ 0
+.101$_{10}$+ 3	+.400$_{10}$+ 1	+.964$_{10}$- 1	+.133$_{10}$+ 0
+.102$_{10}$+ 3	+.400$_{10}$+ 1	+.938$_{10}$- 1	+.130$_{10}$+ 0
+.103$_{10}$+ 3	+.400$_{10}$+ 1	+.914$_{10}$- 1	+.128$_{10}$+ 0
+.104$_{10}$+ 3	+.400$_{10}$+ 1	+.889$_{10}$- 1	+.125$_{10}$+ 0
+.105$_{10}$+ 3	+.400$_{10}$+ 1	+.866$_{10}$- 1	+.123$_{10}$+ 0
+.106$_{10}$+ 3	+.400$_{10}$+ 1	+.843$_{10}$- 1	+.120$_{10}$+ 0
+.107$_{10}$+ 3	+.400$_{10}$+ 1	+.820$_{10}$- 1	+.118$_{10}$+ 0
+.108$_{10}$+ 3	+.400$_{10}$+ 1	+.798$_{10}$- 1	+.116$_{10}$+ 0
+.109$_{10}$+ 3	+.400$_{10}$+ 1	+.777$_{10}$- 1	+.113$_{10}$+ 0
+.110$_{10}$+ 3	+.400$_{10}$+ 1	+.756$_{10}$- 1	+.111$_{10}$+ 0
+.111$_{10}$+ 3	+.400$_{10}$+ 1	+.735$_{10}$- 1	+.109$_{10}$+ 0
+.112$_{10}$+ 3	+.400$_{10}$+ 1	+.715$_{10}$- 1	+.107$_{10}$+ 0
+.113$_{10}$+ 3	+.400$_{10}$+ 1	+.696$_{10}$- 1	+.105$_{10}$+ 0
+.114$_{10}$+ 3	+.400$_{10}$+ 1	+.677$_{10}$- 1	+.103$_{10}$+ 0
+.115$_{10}$+ 3	+.400$_{10}$+ 1	+.658$_{10}$- 1	+.101$_{10}$+ 0
+.116$_{10}$+ 3	+.400$_{10}$+ 1	+.640$_{10}$- 1	+.986$_{10}$- 1
+.117$_{10}$+ 3	+.400$_{10}$+ 1	+.622$_{10}$- 1	+.966$_{10}$- 1
+.118$_{10}$+ 3	+.400$_{10}$+ 1	+.605$_{10}$- 1	+.947$_{10}$- 1
+.119$_{10}$+ 3	+.400$_{10}$+ 1	+.588$_{10}$- 1	+.929$_{10}$- 1
+.120$_{10}$+ 3	+.400$_{10}$+ 1	+.572$_{10}$- 1	+.910$_{10}$- 1
+.121$_{10}$+ 3	+.400$_{10}$+ 1	+.556$_{10}$- 1	+.892$_{10}$- 1
+.122$_{10}$+ 3	+.400$_{10}$+ 1	+.540$_{10}$- 1	+.875$_{10}$- 1
+.123$_{10}$+ 3	+.400$_{10}$+ 1	+.525$_{10}$- 1	+.857$_{10}$- 1
+.124$_{10}$+ 3	+.400$_{10}$+ 1	+.510$_{10}$- 1	+.840$_{10}$- 1
+.125$_{10}$+ 3	+.400$_{10}$+ 1	+.495$_{10}$- 1	+.824$_{10}$- 1
+.126$_{10}$+ 3	+.400$_{10}$+ 1	+.481$_{10}$- 1	+.807$_{10}$- 1
+.127$_{10}$+ 3	+.400$_{10}$+ 1	+.468$_{10}$- 1	+.792$_{10}$- 1
+.128$_{10}$+ 3	+.400$_{10}$+ 1	+.454$_{10}$- 1	+.776$_{10}$- 1
+.129$_{10}$+ 3	+.400$_{10}$+ 1	+.441$_{10}$- 1	+.761$_{10}$- 1

n	m2	prod	pc
+.130₁₀+ 3	+.400₁₀+ 1	+.428₁₀- 1	+.745₁₀- 1
+.131₁₀+ 3	+.400₁₀+ 1	+.416₁₀- 1	+.731₁₀- 1
+.132₁₀+ 3	+.400₁₀+ 1	+.404₁₀- 1	+.716₁₀- 1
+.133₁₀+ 3	+.400₁₀+ 1	+.392₁₀- 1	+.702₁₀- 1
+.134₁₀+ 3	+.400₁₀+ 1	+.381₁₀- 1	+.688₁₀- 1
+.135₁₀+ 3	+.400₁₀+ 1	+.370₁₀- 1	+.675₁₀- 1
+.136₁₀+ 3	+.400₁₀+ 1	+.359₁₀- 1	+.661₁₀- 1
+.137₁₀+ 3	+.400₁₀+ 1	+.348₁₀- 1	+.648₁₀- 1
+.138₁₀+ 3	+.400₁₀+ 1	+.338₁₀- 1	+.635₁₀- 1

```
VERTAALTIJD          5
EXECUTIETIJD        12

AANTAL OVER STROOM 1 GELEZEN GETALLEN              2
LAATST OVER STROOM 1 GELEZEN GETAL              +600

AANTAL OVER STROOM 2 GELEZEN GETALLEN              0
```

XXXXXXXXXXXXXX

217 - 0 - 80672 - 06185479 STELTENPOO

```
 1 BEGIN COMMENT BEREKENING OPBRENGST VERBETERING VAN CELL NETWERKEN M.B.V. DOUBLE ERROR CORRECTING BCH CODES
 2       N = AANTAL INPUTS CELL. NETWERK
 3       M = (2 ↑ K) -1 IS AANTAL OUTPUTS CELL. NETWERK;
 4       INTEGER K, KMAX, IA1, IB1,  I2, I, J, A, N, S, B, IOEEN, IOTWEE, F, I2K, V, NMAX, N, T;
 5       REAL PDEK, P; PC, PRED, PI, Q;
 6       V := 0; KMAX := READ;
 7 B:    T := READ; NMAX := READ: P := EXP(LN(0.95)/T); Q:= 1 - P;
 8       NEW PAGE; ELOT(5, 2, P): NLCR; NLCR; NLCR;
 9       FOR K := 4 STEP 1 UNTIL KMAX DO
10       BEGIN IA1 := 0; IB1 := 0; I2 := 0; FOR I := 2 STEP 1 UNTIL  K DO
11                              IA1 := (I ↑2) + 2 * I + 1 + IA1;  IB1 := 0.5 *(I ↑2) * (5/2) * I + 2 + IB1;
12             FOR J := 1 STEP 1 UNTIL K DO BEGIN A := (2 ↑ J);
13             IF A < K THEN I2 := (A ↑ 2) + I2 ELSE GOTO A; END;
14 A:          I2 := I2 + (K - 2 - J) * (K ↑ 2); I2K := 2 ↑ K; N := I2K - 1 - 2 * K;
15             S := 2 * K * N; B := 2 * (K ↑ 2); IOEEN := K * N; IOTWEE := N;
16             F := N ↑ 2;
17             PDEK := P ↑ S * (1 - (4/5) * Q) ↑ IA1 * (1 - (7/5) * Q) ↑ IB1 * (1 - ( 14/5) * Q ) ↑ I2 * (1 - (14/5) * Q) ↑ B
18             * (1 - (12/5) * Q) ↑ IOEEN * P ↑ IOTWEE * P ↑ F;
19             FOR N := 1 STEP 1 UNTIL NMAX DO
20             BEGIN PI := P ↑ N;
21                   PRED := (PI ↑ (I2K - 1) + (1 - PI) * (I2K - 1) * ( PI ↑ (I2K - 2))
22                  + 0.5 * (I2K - 2) * (I2K - 1) * ((1 - PI) ↑ 2) * (PI ↑ (I2K - 3))) * PDEK;
23                   PC := P ↑ (N * N); IF PRED < PC THEN GOTO C;
24                   PRINTTEXT($VERBETERING$); NLCR; GOTO E;
25 C:                IF K ≠ 5 ∧ K ≠ 4  THEN GOTO D;
26 E:                SPACE(15); ELOT(3, 2, N); ELOT(3, 2, K); ELOT(5, 2, PRED);
27                   ELOT(3, 2, PC); ELOT(5, 2, PDEK); ELOT(3, 2, IA1); ELOT(3, 2, I2);NLCR;
28             D:          END;
29       END;  V := V + 1: IF V = 1 THEN GOTO B;
30 END
31 PROGEND
```

$= +.99006_{10}+ 0$

p	k	Pred	pc	Pdck	ia1	i2
$+.100_{10}+ 1$	$+.400_{10}+ 1$	$+.27664_{10}- 1$	$+.932_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.200_{10}+ 1$	$+.400_{10}+ 1$	$+.27594_{10}- 1$	$+.870_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.300_{10}+ 1$	$+.400_{10}+ 1$	$+.27428_{10}- 1$	$+.811_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.400_{10}+ 1$	$+.400_{10}+ 1$	$+.27145_{10}- 1$	$+.756_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.500_{10}+ 1$	$+.400_{10}+ 1$	$+.26739_{10}- 1$	$+.705_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.600_{10}+ 1$	$+.400_{10}+ 1$	$+.26211_{10}- 1$	$+.657_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.700_{10}+ 1$	$+.400_{10}+ 1$	$+.25571_{10}- 1$	$+.613_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.800_{10}+ 1$	$+.400_{10}+ 1$	$+.24830_{10}- 1$	$+.572_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.900_{10}+ 1$	$+.400_{10}+ 1$	$+.24002_{10}- 1$	$+.533_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.100_{10}+ 2$	$+.400_{10}+ 1$	$+.23102_{10}- 1$	$+.497_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.110_{10}+ 2$	$+.400_{10}+ 1$	$+.22146_{10}- 1$	$+.464_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.120_{10}+ 2$	$+.400_{10}+ 1$	$+.21149_{10}- 1$	$+.432_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.130_{10}+ 2$	$+.400_{10}+ 1$	$+.20125_{10}- 1$	$+.403_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.140_{10}+ 2$	$+.400_{10}+ 1$	$+.19085_{10}- 1$	$+.376_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.150_{10}+ 2$	$+.400_{10}+ 1$	$+.18042_{10}- 1$	$+.350_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.160_{10}+ 2$	$+.400_{10}+ 1$	$+.17005_{10}- 1$	$+.327_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.170_{10}+ 2$	$+.400_{10}+ 1$	$+.15983_{10}- 1$	$+.305_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.180_{10}+ 2$	$+.400_{10}+ 1$	$+.14983_{10}- 1$	$+.284_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.190_{10}+ 2$	$+.400_{10}+ 1$	$+.14011_{10}- 1$	$+.265_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.200_{10}+ 2$	$+.400_{10}+ 1$	$+.13072_{10}- 1$	$+.247_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.210_{10}+ 2$	$+.400_{10}+ 1$	$+.12169_{10}- 1$	$+.230_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.220_{10}+ 2$	$+.400_{10}+ 1$	$+.11305_{10}- 1$	$+.215_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.230_{10}+ 2$	$+.400_{10}+ 1$	$+.10482_{10}- 1$	$+.200_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.240_{10}+ 2$	$+.400_{10}+ 1$	$+.97009_{10}- 2$	$+.187_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.250_{10}+ 2$	$+.400_{10}+ 1$	$+.89624_{10}- 2$	$+.174_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.260_{10}+ 2$	$+.400_{10}+ 1$	$+.82665_{10}- 2$	$+.162_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.270_{10}+ 2$	$+.400_{10}+ 1$	$+.76126_{10}- 2$	$+.151_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.280_{10}+ 2$	$+.400_{10}+ 1$	$+.70000_{10}- 2$	$+.141_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.290_{10}+ 2$	$+.400_{10}+ 1$	$+.64277_{10}- 2$	$+.132_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.300_{10}+ 2$	$+.400_{10}+ 1$	$+.58942_{10}- 2$	$+.123_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.310_{10}+ 2$	$+.400_{10}+ 1$	$+.53981_{10}- 2$	$+.115_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.320_{10}+ 2$	$+.400_{10}+ 1$	$+.49377_{10}- 2$	$+.107_{10}+ 0$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.330_{10}+ 2$	$+.400_{10}+ 1$	$+.45113_{10}- 2$	$+.996_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.340_{10}+ 2$	$+.400_{10}+ 1$	$+.41172_{10}- 2$	$+.929_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.350_{10}+ 2$	$+.400_{10}+ 1$	$+.37535_{10}- 2$	$+.866_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.360_{10}+ 2$	$+.400_{10}+ 1$	$+.34184_{10}- 2$	$+.807_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.370_{10}+ 2$	$+.400_{10}+ 1$	$+.31103_{10}- 2$	$+.753_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.380_{10}+ 2$	$+.400_{10}+ 1$	$+.28273_{10}- 2$	$+.702_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.390_{10}+ 2$	$+.400_{10}+ 1$	$+.25677_{10}- 2$	$+.655_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.400_{10}+ 2$	$+.400_{10}+ 1$	$+.23300_{10}- 2$	$+.611_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.410_{10}+ 2$	$+.400_{10}+ 1$	$+.21125_{10}- 2$	$+.569_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.420_{10}+ 2$	$+.400_{10}+ 1$	$+.19138_{10}- 2$	$+.531_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.430_{10}+ 2$	$+.400_{10}+ 1$	$+.17325_{10}- 2$	$+.495_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.440_{10}+ 2$	$+.400_{10}+ 1$	$+.15673_{10}- 2$	$+.462_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.450_{10}+ 2$	$+.400_{10}+ 1$	$+.14168_{10}- 2$	$+.430_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.460_{10}+ 2$	$+.400_{10}+ 1$	$+.12798_{10}- 2$	$+.401_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.470_{10}+ 2$	$+.400_{10}+ 1$	$+.11554_{10}- 2$	$+.374_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.480_{10}+ 2$	$+.400_{10}+ 1$	$+.10424_{10}- 2$	$+.349_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.490_{10}+ 2$	$+.400_{10}+ 1$	$+.93985_{10}- 3$	$+.325_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.500_{10}+ 2$	$+.400_{10}+ 1$	$+.84692_{10}- 3$	$+.303_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.510_{10}+ 2$	$+.400_{10}+ 1$	$+.76273_{10}- 3$	$+.283_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.520_{10}+ 2$	$+.400_{10}+ 1$	$+.68654_{10}- 3$	$+.264_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.530_{10}+ 2$	$+.400_{10}+ 1$	$+.61764_{10}- 3$	$+.246_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.540_{10}+ 2$	$+.400_{10}+ 1$	$+.55536_{10}- 3$	$+.229_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.550_{10}+ 2$	$+.400_{10}+ 1$	$+.49912_{10}- 3$	$+.214_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.560_{10}+ 2$	$+.400_{10}+ 1$	$+.44835_{10}- 3$	$+.200_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$
$+.570_{10}+ 2$	$+.400_{10}+ 1$	$+.40256_{10}- 3$	$+.186_{10}- 1$	$+.27676_{10}- 1$	$+.500_{10}+ 2$	$+.400_{10}+ 1$

n	k	pred	pc	pdek		
+.580$_{10}$+ 2	+,400$_{10}$+ 1	+.38777$_{10}$- 1	+.132$_{10}$+ 0	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.590$_{10}$+ 2	+,400$_{10}$+ 1	+.37139$_{10}$- 1	+.127$_{10}$+ 0	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.600$_{10}$+ 2	+,400$_{10}$+ 1	+.35558$_{10}$- 1	+.123$_{10}$+ 0	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.610$_{10}$+ 2	+,400$_{10}$+ 1	+.34034$_{10}$- 1	+.119$_{10}$+ 0	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.620$_{10}$+ 2	+,400$_{10}$+ 1	+.32565$_{10}$- 1	+.115$_{10}$+ 0	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.630$_{10}$+ 2	+,400$_{10}$+ 1	+.31150$_{10}$- 1	+.111$_{10}$+ 0	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.640$_{10}$+ 2	+,400$_{10}$+ 1	+.29788$_{10}$- 1	+.107$_{10}$+ 0	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.650$_{10}$+ 2	+,400$_{10}$+ 1	+.28477$_{10}$- 1	+.103$_{10}$+ 0	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.660$_{10}$+ 2	+,400$_{10}$+ 1	+.27216$_{10}$- 1	+.996$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.670$_{10}$+ 2	+,400$_{10}$+ 1	+.26003$_{10}$- 1	+.962$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.680$_{10}$+ 2	+,400$_{10}$+ 1	+.24838$_{10}$- 1	+.929$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.690$_{10}$+ 2	+,400$_{10}$+ 1	+.23719$_{10}$- 1	+.897$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.700$_{10}$+ 2	+,400$_{10}$+ 1	+.22644$_{10}$- 1	+.866$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.710$_{10}$+ 2	+,400$_{10}$+ 1	+.21612$_{10}$- 1	+.836$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.720$_{10}$+ 2	+,400$_{10}$+ 1	+.20623$_{10}$- 1	+.807$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.730$_{10}$+ 2	+,400$_{10}$+ 1	+.19673$_{10}$- 1	+.780$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.740$_{10}$+ 2	+,400$_{10}$+ 1	+.18764$_{10}$- 1	+.753$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.750$_{10}$+ 2	+,400$_{10}$+ 1	+.17892$_{10}$- 1	+.727$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.760$_{10}$+ 2	+,400$_{10}$+ 1	+.17056$_{10}$- 1	+.702$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.770$_{10}$+ 2	+,400$_{10}$+ 1	+.16256$_{10}$- 1	+.678$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.780$_{10}$+ 2	+,400$_{10}$+ 1	+.15490$_{10}$- 1	+.655$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.790$_{10}$+ 2	+,400$_{10}$+ 1	+.14757$_{10}$- 1	+.632$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.800$_{10}$+ 2	+,400$_{10}$+ 1	+.14056$_{10}$- 1	+.611$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.810$_{10}$+ 2	+,400$_{10}$+ 1	+.13386$_{10}$- 1	+.590$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.820$_{10}$+ 2	+,400$_{10}$+ 1	+.12744$_{10}$- 1	+.569$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.830$_{10}$+ 2	+,400$_{10}$+ 1	+.12131$_{10}$- 1	+.550$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.840$_{10}$+ 2	+,400$_{10}$+ 1	+.11546$_{10}$- 1	+.531$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.850$_{10}$+ 2	+,400$_{10}$+ 1	+.10986$_{10}$- 1	+.513$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.860$_{10}$+ 2	+,400$_{10}$+ 1	+.10452$_{10}$- 1	+.495$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.870$_{10}$+ 2	+,400$_{10}$+ 1	+.99418$_{10}$- 2	+.478$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.880$_{10}$+ 2	+,400$_{10}$+ 1	+.94549$_{10}$- 2	+.462$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.890$_{10}$+ 2	+,400$_{10}$+ 1	+.89902$_{10}$- 2	+.446$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.900$_{10}$+ 2	+,400$_{10}$+ 1	+.85469$_{10}$- 2	+.430$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.910$_{10}$+ 2	+,400$_{10}$+ 1	+.81241$_{10}$- 2	+.416$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.920$_{10}$+ 2	+,400$_{10}$+ 1	+.77209$_{10}$- 2	+.401$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.930$_{10}$+ 2	+,400$_{10}$+ 1	+.73365$_{10}$- 2	+.388$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.940$_{10}$+ 2	+,400$_{10}$+ 1	+.69701$_{10}$- 2	+.374$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.950$_{10}$+ 2	+,400$_{10}$+ 1	+.66210$_{10}$- 2	+.361$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.960$_{10}$+ 2	+,400$_{10}$+ 1	+.62884$_{10}$- 2	+.349$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.970$_{10}$+ 2	+,400$_{10}$+ 1	+.59716$_{10}$- 2	+.337$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.980$_{10}$+ 2	+,400$_{10}$+ 1	+.56699$_{10}$- 2	+.325$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.990$_{10}$+ 2	+,400$_{10}$+ 1	+.53827$_{10}$- 2	+.314$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.100$_{10}$+ 3	+,400$_{10}$+ 1	+.51092$_{10}$- 2	+.303$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.101$_{10}$+ 3	+,400$_{10}$+ 1	+.48490$_{10}$- 2	+.293$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.102$_{10}$+ 3	+,400$_{10}$+ 1	+.46014$_{10}$- 2	+.283$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.103$_{10}$+ 3	+,400$_{10}$+ 1	+.43658$_{10}$- 2	+.273$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.104$_{10}$+ 3	+,400$_{10}$+ 1	+.41418$_{10}$- 2	+.264$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.105$_{10}$+ 3	+,400$_{10}$+ 1	+.39287$_{10}$- 2	+.255$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.106$_{10}$+ 3	+,400$_{10}$+ 1	+.37261$_{10}$- 2	+.246$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.107$_{10}$+ 3	+,400$_{10}$+ 1	+.35334$_{10}$- 2	+.238$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.108$_{10}$+ 3	+,400$_{10}$+ 1	+.33504$_{10}$- 2	+.229$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.109$_{10}$+ 3	+,400$_{10}$+ 1	+.31764$_{10}$- 2	+.222$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.110$_{10}$+ 3	+,400$_{10}$+ 1	+.30110$_{10}$- 2	+.214$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.111$_{10}$+ 3	+,400$_{10}$+ 1	+.28540$_{10}$- 2	+.207$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.112$_{10}$+ 3	+,400$_{10}$+ 1	+.27048$_{10}$- 2	+.200$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.113$_{10}$+ 3	+,400$_{10}$+ 1	+.25631$_{10}$- 2	+.193$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.114$_{10}$+ 3	+,400$_{10}$+ 1	+.24286$_{10}$- 2	+.186$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.115$_{10}$+ 3	+,400$_{10}$+ 1	+.23008$_{10}$- 2	+.180$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.116$_{10}$+ 3	+,400$_{10}$+ 1	+.21796$_{10}$- 2	+.173$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.117$_{10}$+ 3	+,400$_{10}$+ 1	+.20645$_{10}$- 2	+.168$_{10}$- 1	+,16696$_{10}$+ 0	+,500$_{10}$+ 2	+,400$_{10}$+ 1

n	k	pred	pe	pdek		
+.118+ 3	+.400+ 1	+.19552- 2	+.162- 1	+.16696+ 0	+.500+ 2	+.400+ 1
+.119+ 3	+.400+ 1	+.18516- 2	+.156- 1	+.16696+ 0	+.500+ 2	+.400+ 1
+.120+ 3	+.400+ 1	+.17532- 2	+.151- 1	+.16696+ 0	+.500+ 2	+.400+ 1
+.121+ 3	+.400+ 1	+.16599- 2	+.146- 1	+.16696+ 0	+.500+ 2	+.400+ 1
+.122+ 3	+.400+ 1	+.15715- 2	+.141- 1	+.16696+ 0	+.500+ 2	+.400+ 1
+.123+ 3	+.400+ 1	+.14876- 2	+.136- 1	+.16696+ 0	+.500+ 2	+.400+ 1
+.124+ 3	+.400+ 1	+.14080- 2	+.131- 1	+.16696+ 0	+.500+ 2	+.400+ 1
+.125+ 3	+.400+ 1	+.13325- 2	+.127- 1	+.16696+ 0	+.500+ 2	+.400+ 1
+.126+ 3	+.400+ 1	+.12610- 2	+.122- 1	+.16696+ 0	+.500+ 2	+.400+ 1
+.127+ 3	+.400+ 1	+.11932- 2	+.118- 1	+.16696+ 0	+.500+ 2	+.400+ 1
+.128+ 3	+.400+ 1	+.11290- 2	+.114- 1	+.16696+ 0	+.500+ 2	+.400+ 1
+.129+ 3	+.400+ 1	+.10681- 2	+.110- 1	+.16696+ 0	+.500+ 2	+.400+ 1
+.130+ 3	+.400+ 1	+.10104- 2	+.106- 1	+.16696+ 0	+.500+ 2	+.400+ 1
+.131+ 3	+.400+ 1	+.95569- 3	+.103- 1	+.16696+ 0	+.500+ 2	+.400+ 1
+.132+ 3	+.400+ 1	+.90390- 3	+.992- 2	+.16696+ 0	+.500+ 2	+.400+ 1
+.133+ 3	+.400+ 1	+.85484- 3	+.958- 2	+.16696+ 0	+.500+ 2	+.400+ 1
+.134+ 3	+.400+ 1	+.80837- 3	+.925- 2	+.16696+ 0	+.500+ 2	+.400+ 1
+.135+ 3	+.400+ 1	+.76437- 3	+.893- 2	+.16696+ 0	+.500+ 2	+.400+ 1
+.136+ 3	+.400+ 1	+.72270- 3	+.862- 2	+.16696+ 0	+.500+ 2	+.400+ 1
+.137+ 3	+.400+ 1	+.68325- 3	+.833- 2	+.16696+ 0	+.500+ 2	+.400+ 1
+.138+ 3	+.400+ 1	+.64590- 3	+.804- 2	+.16696+ 0	+.500+ 2	+.400+ 1
+.139+ 3	+.400+ 1	+.61054- 3	+.777- 2	+.16696+ 0	+.500+ 2	+.400+ 1
+.140+ 3	+.400+ 1	+.57707- 3	+.750- 2	+.16696+ 0	+.500+ 2	+.400+ 1
+.141+ 3	+.400+ 1	+.54539- 3	+.724- 2	+.16696+ 0	+.500+ 2	+.400+ 1
+.142+ 3	+.400+ 1	+.51542- 3	+.699- 2	+.16696+ 0	+.500+ 2	+.400+ 1
+.143+ 3	+.400+ 1	+.48705- 3	+.675- 2	+.16696+ 0	+.500+ 2	+.400+ 1
+.144+ 3	+.400+ 1	+.46021- 3	+.652- 2	+.16696+ 0	+.500+ 2	+.400+ 1
+.145+ 3	+.400+ 1	+.43482- 3	+.630- 2	+.16696+ 0	+.500+ 2	+.400+ 1
+.146+ 3	+.400+ 1	+.41080- 3	+.608- 2	+.16696+ 0	+.500+ 2	+.400+ 1
+.147+ 3	+.400+ 1	+.38807- 3	+.587- 2	+.16696+ 0	+.500+ 2	+.400+ 1
+.148+ 3	+.400+ 1	+.36658- 3	+.567- 2	+.16696+ 0	+.500+ 2	+.400+ 1
+.149+ 3	+.400+ 1	+.34626- 3	+.547- 2	+.16696+ 0	+.500+ 2	+.400+ 1
+.150+ 3	+.400+ 1	+.32704- 3	+.529- 2	+.16696+ 0	+.500+ 2	+.400+ 1
+.100+ 1	+.500+ 1	+.20506- 2	+.900+ 0	+.20516- 2	+.860+ 2	+.200+ 2
+.200+ 1	+.500+ 1	+.20443- 2	+.811+ 0	+.20516- 2	+.860+ 2	+.200+ 2
+.300+ 1	+.500+ 1	+.20294- 2	+.730+ 0	+.20516- 2	+.860+ 2	+.200+ 2
+.400+ 1	+.500+ 1	+.20044- 2	+.657+ 0	+.20516- 2	+.860+ 2	+.200+ 2
+.500+ 1	+.500+ 1	+.19689- 2	+.592+ 0	+.20516- 2	+.860+ 2	+.200+ 2
+.600+ 1	+.500+ 1	+.19232- 2	+.533+ 0	+.20516- 2	+.860+ 2	+.200+ 2
+.700+ 1	+.500+ 1	+.18683- 2	+.480+ 0	+.20516- 2	+.860+ 2	+.200+ 2
+.800+ 1	+.500+ 1	+.18054- 2	+.432+ 0	+.20516- 2	+.860+ 2	+.200+ 2
+.900+ 1	+.500+ 1	+.17358- 2	+.389+ 0	+.20516- 2	+.860+ 2	+.200+ 2
+.100+ 2	+.500+ 1	+.16609- 2	+.350+ 0	+.20516- 2	+.860+ 2	+.200+ 2
+.110+ 2	+.500+ 1	+.15822- 2	+.316+ 0	+.20516- 2	+.860+ 2	+.200+ 2
+.120+ 2	+.500+ 1	+.15009- 2	+.284+ 0	+.20516- 2	+.860+ 2	+.200+ 2
+.130+ 2	+.500+ 1	+.14182- 2	+.256+ 0	+.20516- 2	+.860+ 2	+.200+ 2
+.140+ 2	+.500+ 1	+.13351- 2	+.230+ 0	+.20516- 2	+.860+ 2	+.200+ 2
+.150+ 2	+.500+ 1	+.12526- 2	+.207+ 0	+.20516- 2	+.860+ 2	+.200+ 2
+.160+ 2	+.500+ 1	+.11714- 2	+.187+ 0	+.20516- 2	+.860+ 2	+.200+ 2
+.170+ 2	+.500+ 1	+.10922- 2	+.168+ 0	+.20516- 2	+.860+ 2	+.200+ 2
+.180+ 2	+.500+ 1	+.10155- 2	+.151+ 0	+.20516- 2	+.860+ 2	+.200+ 2
+.190+ 2	+.500+ 1	+.94169- 3	+.136+ 0	+.20516- 2	+.860+ 2	+.200+ 2
+.200+ 2	+.500+ 1	+.87108- 3	+.123+ 0	+.20516- 2	+.860+ 2	+.200+ 2
+.210+ 2	+.500+ 1	+.80389- 3	+.111+ 0	+.20516- 2	+.860+ 2	+.200+ 2
+.220+ 2	+.500+ 1	+.74026- 3	+.996- 1	+.20516- 2	+.860+ 2	+.200+ 2
+.230+ 2	+.500+ 1	+.68026- 3	+.897- 1	+.20516- 2	+.860+ 2	+.200+ 2
+.240+ 2	+.500+ 1	+.62390- 3	+.807- 1	+.20516- 2	+.860+ 2	+.200+ 2
+.250+ 2	+.500+ 1	+.57116- 3	+.727- 1	+.20516- 2	+.860+ 2	+.200+ 2
+.260+ 2	+.500+ 1	+.52197- 3	+.655- 1	+.20516- 2	+.860+ 2	+.200+ 2
+.270+ 2	+.500+ 1	+.47623- 3	+.590- 1	+.20516- 2	+.860+ 2	+.200+ 2

n	k	Pred	Pe	Pdek		
+.280$_{10}$+ 2	+.500$_{10}$+ 1	+.43382$_{10}$- 3	+.531$_{10}$- 1	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.290$_{10}$+ 2	+.500$_{10}$+ 1	+.39460$_{10}$- 3	+.478$_{10}$- 1	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.300$_{10}$+ 2	+.500$_{10}$+ 1	+.35842$_{10}$- 3	+.430$_{10}$- 1	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.310$_{10}$+ 2	+.500$_{10}$+ 1	+.32511$_{10}$- 3	+.388$_{10}$- 1	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.320$_{10}$+ 2	+.500$_{10}$+ 1	+.29453$_{10}$- 3	+.349$_{10}$- 1	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.330$_{10}$+ 2	+.500$_{10}$+ 1	+.26650$_{10}$- 3	+.314$_{10}$- 1	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.340$_{10}$+ 2	+.500$_{10}$+ 1	+.24085$_{10}$- 3	+.283$_{10}$- 1	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.350$_{10}$+ 2	+.500$_{10}$+ 1	+.21743$_{10}$- 3	+.255$_{10}$- 1	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.360$_{10}$+ 2	+.500$_{10}$+ 1	+.19608$_{10}$- 3	+.229$_{10}$- 1	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.370$_{10}$+ 2	+.500$_{10}$+ 1	+.17665$_{10}$- 3	+.207$_{10}$- 1	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.380$_{10}$+ 2	+.500$_{10}$+ 1	+.15899$_{10}$- 3	+.186$_{10}$- 1	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.390$_{10}$+ 2	+.500$_{10}$+ 1	+.14296$_{10}$- 3	+.168$_{10}$- 1	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.400$_{10}$+ 2	+.500$_{10}$+ 1	+.12843$_{10}$- 3	+.151$_{10}$- 1	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.410$_{10}$+ 2	+.500$_{10}$+ 1	+.11528$_{10}$- 3	+.136$_{10}$- 1	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.420$_{10}$+ 2	+.500$_{10}$+ 1	+.10339$_{10}$- 3	+.122$_{10}$- 1	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.430$_{10}$+ 2	+.500$_{10}$+ 1	+.92653$_{10}$- 4	+.110$_{10}$- 1	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.440$_{10}$+ 2	+.500$_{10}$+ 1	+.82967$_{10}$- 4	+.992$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.450$_{10}$+ 2	+.500$_{10}$+ 1	+.74239$_{10}$- 4	+.893$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.460$_{10}$+ 2	+.500$_{10}$+ 1	+.66383$_{10}$- 4	+.804$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.470$_{10}$+ 2	+.500$_{10}$+ 1	+.59317$_{10}$- 4	+.724$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.480$_{10}$+ 2	+.500$_{10}$+ 1	+.52969$_{10}$- 4	+.652$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.490$_{10}$+ 2	+.500$_{10}$+ 1	+.47270$_{10}$- 4	+.587$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.500$_{10}$+ 2	+.500$_{10}$+ 1	+.42158$_{10}$- 4	+.529$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.510$_{10}$+ 2	+.500$_{10}$+ 1	+.37576$_{10}$- 4	+.476$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.520$_{10}$+ 2	+.500$_{10}$+ 1	+.33474$_{10}$- 4	+.429$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.530$_{10}$+ 2	+.500$_{10}$+ 1	+.29803$_{10}$- 4	+.386$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.540$_{10}$+ 2	+.500$_{10}$+ 1	+.26520$_{10}$- 4	+.348$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.550$_{10}$+ 2	+.500$_{10}$+ 1	+.23586$_{10}$- 4	+.313$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.560$_{10}$+ 2	+.500$_{10}$+ 1	+.20967$_{10}$- 4	+.282$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.570$_{10}$+ 2	+.500$_{10}$+ 1	+.18629$_{10}$- 4	+.254$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.580$_{10}$+ 2	+.500$_{10}$+ 1	+.16544$_{10}$- 4	+.229$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.590$_{10}$+ 2	+.500$_{10}$+ 1	+.14686$_{10}$- 4	+.206$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.600$_{10}$+ 2	+.500$_{10}$+ 1	+.13030$_{10}$- 4	+.185$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.610$_{10}$+ 2	+.500$_{10}$+ 1	+.11557$_{10}$- 4	+.167$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.620$_{10}$+ 2	+.500$_{10}$+ 1	+.10245$_{10}$- 4	+.150$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.630$_{10}$+ 2	+.500$_{10}$+ 1	+.90789$_{10}$- 5	+.135$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.640$_{10}$+ 2	+.500$_{10}$+ 1	+.80422$_{10}$- 5	+.122$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.650$_{10}$+ 2	+.500$_{10}$+ 1	+.71211$_{10}$- 5	+.110$_{10}$- 2	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.660$_{10}$+ 2	+.500$_{10}$+ 1	+.63031$_{10}$- 5	+.988$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.670$_{10}$+ 2	+.500$_{10}$+ 1	+.55771$_{10}$- 5	+.889$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.680$_{10}$+ 2	+.500$_{10}$+ 1	+.49329$_{10}$- 5	+.801$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.690$_{10}$+ 2	+.500$_{10}$+ 1	+.43617$_{10}$- 5	+.721$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.700$_{10}$+ 2	+.500$_{10}$+ 1	+.38553$_{10}$- 5	+.649$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.710$_{10}$+ 2	+.500$_{10}$+ 1	+.34065$_{10}$- 5	+.585$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.720$_{10}$+ 2	+.500$_{10}$+ 1	+.30091$_{10}$- 5	+.527$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.730$_{10}$+ 2	+.500$_{10}$+ 1	+.26571$_{10}$- 5	+.474$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.740$_{10}$+ 2	+.500$_{10}$+ 1	+.23457$_{10}$- 5	+.427$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.750$_{10}$+ 2	+.500$_{10}$+ 1	+.20701$_{10}$- 5	+.384$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.760$_{10}$+ 2	+.500$_{10}$+ 1	+.18264$_{10}$- 5	+.346$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.770$_{10}$+ 2	+.500$_{10}$+ 1	+.16109$_{10}$- 5	+.312$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.780$_{10}$+ 2	+.500$_{10}$+ 1	+.14204$_{10}$- 5	+.281$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.790$_{10}$+ 2	+.500$_{10}$+ 1	+.12521$_{10}$- 5	+.253$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.800$_{10}$+ 2	+.500$_{10}$+ 1	+.11035$_{10}$- 5	+.228$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.810$_{10}$+ 2	+.500$_{10}$+ 1	+.97226$_{10}$- 6	+.205$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.820$_{10}$+ 2	+.500$_{10}$+ 1	+.85642$_{10}$- 6	+.185$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.830$_{10}$+ 2	+.500$_{10}$+ 1	+.75419$_{10}$- 6	+.166$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.840$_{10}$+ 2	+.500$_{10}$+ 1	+.66400$_{10}$- 6	+.150$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.850$_{10}$+ 2	+.500$_{10}$+ 1	+.58446$_{10}$- 6	+.135$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.860$_{10}$+ 2	+.500$_{10}$+ 1	+.51433$_{10}$- 6	+.121$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.870$_{10}$+ 2	+.500$_{10}$+ 1	+.45252$_{10}$- 6	+.109$_{10}$- 3	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2

n	k	Pred	Pc	Pdek		
+.580$_{10}$+ 2	+,400$_{10}$+ 1	+.36129$_{10}$- 3	+.173$_{10}$- 1	+,27676$_{10}$- 1	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.590$_{10}$+ 2	+,400$_{10}$+ 1	+.32410$_{10}$- 3	+.162$_{10}$- 1	+.27676$_{10}$- 1	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.600$_{10}$+ 2	+,400$_{10}$+ 1	+.29062$_{10}$- 3	+.151$_{10}$- 1	+,27676$_{10}$- 1	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.610$_{10}$+ 2	+,400$_{10}$+ 1	+.26049$_{10}$- 3	+.141$_{10}$- 1	+,27676$_{10}$- 1	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.620$_{10}$+ 2	+,400$_{10}$+ 1	+.23339$_{10}$- 3	+.131$_{10}$- 1	+,27676$_{10}$- 1	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.630$_{10}$+ 2	+,400$_{10}$+ 1	+.20903$_{10}$- 3	+.122$_{10}$- 1	+,27676$_{10}$- 1	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.640$_{10}$+ 2	+,400$_{10}$+ 1	+.18714$_{10}$- 3	+.114$_{10}$- 1	+,27676$_{10}$- 1	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.650$_{10}$+ 2	+,400$_{10}$+ 1	+.16748$_{10}$- 3	+.106$_{10}$- 1	+,27676$_{10}$- 1	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.660$_{10}$+ 2	+,400$_{10}$+ 1	+.14983$_{10}$- 3	+.992$_{10}$- 2	+,27676$_{10}$- 1	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.670$_{10}$+ 2	+,400$_{10}$+ 1	+.13400$_{10}$- 3	+.925$_{10}$- 2	+,27676$_{10}$- 1	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.680$_{10}$+ 2	+,400$_{10}$+ 1	+.11980$_{10}$- 3	+.862$_{10}$- 2	+,27676$_{10}$- 1	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.690$_{10}$+ 2	+,400$_{10}$+ 1	+.10706$_{10}$- 3	+.804$_{10}$- 2	+,27676$_{10}$- 1	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.700$_{10}$+ 2	+,400$_{10}$+ 1	+.95656$_{10}$- 4	+.750$_{10}$- 2	+,27676$_{10}$- 1	+,500$_{10}$+ 2	+,400$_{10}$+ 1
+.100$_{10}$+ 1	+,500$_{10}$+ 1	+.41188$_{10}$- 5	+.811$_{10}$+ 0	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.200$_{10}$+ 1	+,500$_{10}$+ 1	+.40385$_{10}$- 5	+.657$_{10}$+ 0	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.300$_{10}$+ 1	+,500$_{10}$+ 1	+.38749$_{10}$- 5	+.533$_{10}$+ 0	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.400$_{10}$+ 1	+,500$_{10}$+ 1	+.36375$_{10}$- 5	+.432$_{10}$+ 0	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.500$_{10}$+ 1	+,500$_{10}$+ 1	+.33464$_{10}$- 5	+.350$_{10}$+ 0	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.600$_{10}$+ 1	+,500$_{10}$+ 1	+.30240$_{10}$- 5	+.284$_{10}$+ 0	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.700$_{10}$+ 1	+,500$_{10}$+ 1	+.26900$_{10}$- 5	+.230$_{10}$+ 0	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.800$_{10}$+ 1	+,500$_{10}$+ 1	+.23602$_{10}$- 5	+.187$_{10}$+ 0	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.900$_{10}$+ 1	+,500$_{10}$+ 1	+.20460$_{10}$- 5	+.151$_{10}$+ 0	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.100$_{10}$+ 2	+,500$_{10}$+ 1	+.17550$_{10}$- 5	+.123$_{10}$+ 0	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.110$_{10}$+ 2	+,500$_{10}$+ 1	+.14915$_{10}$- 5	+.996$_{10}$- 1	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.120$_{10}$+ 2	+,500$_{10}$+ 1	+.12570$_{10}$- 5	+.807$_{10}$- 1	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.130$_{10}$+ 2	+,500$_{10}$+ 1	+.10517$_{10}$- 5	+.655$_{10}$- 1	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.140$_{10}$+ 2	+,500$_{10}$+ 1	+.87405$_{10}$- 6	+.531$_{10}$- 1	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.150$_{10}$+ 2	+,500$_{10}$+ 1	+.72213$_{10}$- 6	+.430$_{10}$- 1	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.160$_{10}$+ 2	+,500$_{10}$+ 1	+.59342$_{10}$- 6	+.349$_{10}$- 1	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.170$_{10}$+ 2	+,500$_{10}$+ 1	+.48527$_{10}$- 6	+.283$_{10}$- 1	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.180$_{10}$+ 2	+,500$_{10}$+ 1	+.39507$_{10}$- 6	+.229$_{10}$- 1	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.190$_{10}$+ 2	+,500$_{10}$+ 1	+.32033$_{10}$- 6	+.186$_{10}$- 1	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.200$_{10}$+ 2	+,500$_{10}$+ 1	+.25876$_{10}$- 6	+.151$_{10}$- 1	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.210$_{10}$+ 2	+,500$_{10}$+ 1	+.20831$_{10}$- 6	+.122$_{10}$- 1	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.220$_{10}$+ 2	+,500$_{10}$+ 1	+.16716$_{10}$- 6	+.992$_{10}$- 2	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.230$_{10}$+ 2	+,500$_{10}$+ 1	+.13375$_{10}$- 6	+.804$_{10}$- 2	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.240$_{10}$+ 2	+,500$_{10}$+ 1	+.10672$_{10}$- 6	+.652$_{10}$- 2	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.250$_{10}$+ 2	+,500$_{10}$+ 1	+.84039$_{10}$- 7	+.529$_{10}$- 2	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.260$_{10}$+ 2	+,500$_{10}$+ 1	+.67443$_{10}$- 7	+.429$_{10}$- 2	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.270$_{10}$+ 2	+,500$_{10}$+ 1	+.53432$_{10}$- 7	+.348$_{10}$- 2	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.280$_{10}$+ 2	+,500$_{10}$+ 1	+.42243$_{10}$- 7	+.282$_{10}$- 2	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.290$_{10}$+ 2	+,500$_{10}$+ 1	+.33333$_{10}$- 7	+.229$_{10}$- 2	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.300$_{10}$+ 2	+,500$_{10}$+ 1	+.26253$_{10}$- 7	+.185$_{10}$- 2	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.310$_{10}$+ 2	+,500$_{10}$+ 1	+.20642$_{10}$- 7	+.150$_{10}$- 2	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.320$_{10}$+ 2	+,500$_{10}$+ 1	+.16203$_{10}$- 7	+.122$_{10}$- 2	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.330$_{10}$+ 2	+,500$_{10}$+ 1	+.12700$_{10}$- 7	+.988$_{10}$- 3	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.340$_{10}$+ 2	+,500$_{10}$+ 1	+.99388$_{10}$- 8	+.801$_{10}$- 3	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.350$_{10}$+ 2	+,500$_{10}$+ 1	+.77676$_{10}$- 8	+.649$_{10}$- 3	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.360$_{10}$+ 2	+,500$_{10}$+ 1	+.60626$_{10}$- 8	+.527$_{10}$- 3	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.370$_{10}$+ 2	+,500$_{10}$+ 1	+.47260$_{10}$- 8	+.427$_{10}$- 3	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.380$_{10}$+ 2	+,500$_{10}$+ 1	+.36797$_{10}$- 8	+.346$_{10}$- 3	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.390$_{10}$+ 2	+,500$_{10}$+ 1	+.28618$_{10}$- 8	+.281$_{10}$- 3	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.400$_{10}$+ 2	+,500$_{10}$+ 1	+.22233$_{10}$- 8	+.228$_{10}$- 3	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.410$_{10}$+ 2	+,500$_{10}$+ 1	+.17255$_{10}$- 8	+.185$_{10}$- 3	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.420$_{10}$+ 2	+,500$_{10}$+ 1	+.13378$_{10}$- 8	+.150$_{10}$- 3	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.430$_{10}$+ 2	+,500$_{10}$+ 1	+.10363$_{10}$- 8	+.121$_{10}$- 3	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.440$_{10}$+ 2	+,500$_{10}$+ 1	+.80197$_{10}$- 9	+.984$_{10}$- 4	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.450$_{10}$+ 2	+,500$_{10}$+ 1	+.62011$_{10}$- 9	+.798$_{10}$- 4	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.460$_{10}$+ 2	+,500$_{10}$+ 1	+.47909$_{10}$- 9	+.647$_{10}$- 4	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.470$_{10}$+ 2	+,500$_{10}$+ 1	+.36985$_{10}$- 9	+.524$_{10}$- 4	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2

n	k	pred	pc	pdek		
+.480$_{10}$+ 2	+,500$_{10}$+ 1	+.28530$_{10}$- 9	+.425$_{10}$- 4	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.490$_{10}$+ 2	+,500$_{10}$+ 1	+.21992$_{10}$- 9	+.345$_{10}$- 4	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.500$_{10}$+ 2	+,500$_{10}$+ 1	+.16940$_{10}$- 9	+.280$_{10}$- 4	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.510$_{10}$+ 2	+,500$_{10}$+ 1	+.13040$_{10}$- 9	+.227$_{10}$- 4	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.520$_{10}$+ 2	+,500$_{10}$+ 1	+.10031$_{10}$- 9	+.184$_{10}$- 4	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.530$_{10}$+ 2	+,500$_{10}$+ 1	+.77114$_{10}$-10	+.149$_{10}$- 4	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.540$_{10}$+ 2	+,500$_{10}$+ 1	+.59247$_{10}$-10	+.121$_{10}$- 4	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.550$_{10}$+ 2	+,500$_{10}$+ 1	+.45493$_{10}$-10	+.980$_{10}$- 5	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.560$_{10}$+ 2	+,500$_{10}$+ 1	+.34913$_{10}$-10	+.794$_{10}$- 5	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.570$_{10}$+ 2	+,500$_{10}$+ 1	+.26778$_{10}$-10	+.644$_{10}$- 5	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.580$_{10}$+ 2	+,500$_{10}$+ 1	+.20528$_{10}$-10	+.522$_{10}$- 5	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.590$_{10}$+ 2	+,500$_{10}$+ 1	+.15729$_{10}$-10	+.423$_{10}$- 5	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.600$_{10}$+ 2	+,500$_{10}$+ 1	+.12045$_{10}$-10	+.343$_{10}$- 5	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.610$_{10}$+ 2	+,500$_{10}$+ 1	+.92203$_{10}$-11	+.278$_{10}$- 5	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.620$_{10}$+ 2	+,500$_{10}$+ 1	+.70546$_{10}$-11	+.226$_{10}$- 5	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.630$_{10}$+ 2	+,500$_{10}$+ 1	+.53951$_{10}$-11	+.183$_{10}$- 5	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.640$_{10}$+ 2	+,500$_{10}$+ 1	+.41242$_{10}$-11	+.148$_{10}$- 5	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.650$_{10}$+ 2	+,500$_{10}$+ 1	+.31513$_{10}$-11	+.120$_{10}$- 5	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.660$_{10}$+ 2	+,500$_{10}$+ 1	+.24070$_{10}$-11	+.976$_{10}$- 6	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.670$_{10}$+ 2	+,500$_{10}$+ 1	+.18377$_{10}$-11	+.791$_{10}$- 6	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.680$_{10}$+ 2	+,500$_{10}$+ 1	+.14025$_{10}$-11	+.641$_{10}$- 6	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.690$_{10}$+ 2	+,500$_{10}$+ 1	+.10700$_{10}$-11	+.520$_{10}$- 6	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.700$_{10}$+ 2	+,500$_{10}$+ 1	+.81604$_{10}$-12	+.422$_{10}$- 6	+,41336$_{10}$- 5	+,860$_{10}$+ 2	+,200$_{10}$+ 2

217 - 4 - 80572 - 061854 9 STELTENPOO

$p - p_2$

+.99502$_{10}$+ 0

n	k	P_{red}	p_e	p_{dek}
+.100$_{10}$+ 1	+.400$_{10}$+ 1	+.16695$_{10}$+ 0	+.966$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.200$_{10}$+ 1	+.400$_{10}$+ 1	+.16689$_{10}$+ 0	+.932$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.300$_{10}$+ 1	+.400$_{10}$+ 1	+.16674$_{10}$+ 0	+.900$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.16647$_{10}$+ 0	+.870$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.500$_{10}$+ 1	+.400$_{10}$+ 1	+.16605$_{10}$+ 0	+.840$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.600$_{10}$+ 1	+.400$_{10}$+ 1	+.16546$_{10}$+ 0	+.811$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.700$_{10}$+ 1	+.400$_{10}$+ 1	+.16470$_{10}$+ 0	+.783$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.800$_{10}$+ 1	+.400$_{10}$+ 1	+.16376$_{10}$+ 0	+.756$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.900$_{10}$+ 1	+.400$_{10}$+ 1	+.16263$_{10}$+ 0	+.730$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.100$_{10}$+ 2	+.400$_{10}$+ 1	+.16131$_{10}$+ 0	+.705$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.110$_{10}$+ 2	+.400$_{10}$+ 1	+.15981$_{10}$+ 0	+.681$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.120$_{10}$+ 2	+.400$_{10}$+ 1	+.15813$_{10}$+ 0	+.657$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.130$_{10}$+ 2	+.400$_{10}$+ 1	+.15628$_{10}$+ 0	+.635$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.140$_{10}$+ 2	+.400$_{10}$+ 1	+.15426$_{10}$+ 0	+.613$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.150$_{10}$+ 2	+.400$_{10}$+ 1	+.15210$_{10}$+ 0	+.592$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.160$_{10}$+ 2	+.400$_{10}$+ 1	+.14979$_{10}$+ 0	+.572$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.170$_{10}$+ 2	+.400$_{10}$+ 1	+.14735$_{10}$+ 0	+.552$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.180$_{10}$+ 2	+.400$_{10}$+ 1	+.14480$_{10}$+ 0	+.533$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.190$_{10}$+ 2	+.400$_{10}$+ 1	+.14213$_{10}$+ 0	+.515$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.200$_{10}$+ 2	+.400$_{10}$+ 1	+.13937$_{10}$+ 0	+.497$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.210$_{10}$+ 2	+.400$_{10}$+ 1	+.13652$_{10}$+ 0	+.480$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.220$_{10}$+ 2	+.400$_{10}$+ 1	+.13360$_{10}$+ 0	+.464$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.230$_{10}$+ 2	+.400$_{10}$+ 1	+.13062$_{10}$+ 0	+.448$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.240$_{10}$+ 2	+.400$_{10}$+ 1	+.12759$_{10}$+ 0	+.432$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.250$_{10}$+ 2	+.400$_{10}$+ 1	+.12451$_{10}$+ 0	+.417$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.260$_{10}$+ 2	+.400$_{10}$+ 1	+.12141$_{10}$+ 0	+.403$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.270$_{10}$+ 2	+.400$_{10}$+ 1	+.11828$_{10}$+ 0	+.389$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.280$_{10}$+ 2	+.400$_{10}$+ 1	+.11514$_{10}$+ 0	+.376$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.290$_{10}$+ 2	+.400$_{10}$+ 1	+.11199$_{10}$+ 0	+.363$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.300$_{10}$+ 2	+.400$_{10}$+ 1	+.10884$_{10}$+ 0	+.350$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.310$_{10}$+ 2	+.400$_{10}$+ 1	+.10571$_{10}$+ 0	+.338$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.320$_{10}$+ 2	+.400$_{10}$+ 1	+.10259$_{10}$+ 0	+.327$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.330$_{10}$+ 2	+.400$_{10}$+ 1	+.99491$_{10}$- 1	+.316$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.340$_{10}$+ 2	+.400$_{10}$+ 1	+.96423$_{10}$- 1	+.305$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.350$_{10}$+ 2	+.400$_{10}$+ 1	+.93388$_{10}$- 1	+.294$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.360$_{10}$+ 2	+.400$_{10}$+ 1	+.90391$_{10}$- 1	+.284$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.370$_{10}$+ 2	+.400$_{10}$+ 1	+.87436$_{10}$- 1	+.274$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.380$_{10}$+ 2	+.400$_{10}$+ 1	+.84527$_{10}$- 1	+.265$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.390$_{10}$+ 2	+.400$_{10}$+ 1	+.81668$_{10}$- 1	+.256$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.400$_{10}$+ 2	+.400$_{10}$+ 1	+.78861$_{10}$- 1	+.247$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.410$_{10}$+ 2	+.400$_{10}$+ 1	+.76109$_{10}$- 1	+.239$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.420$_{10}$+ 2	+.400$_{10}$+ 1	+.73413$_{10}$- 1	+.230$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.430$_{10}$+ 2	+.400$_{10}$+ 1	+.70777$_{10}$- 1	+.222$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.440$_{10}$+ 2	+.400$_{10}$+ 1	+.68201$_{10}$- 1	+.215$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.450$_{10}$+ 2	+.400$_{10}$+ 1	+.65687$_{10}$- 1	+.207$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.460$_{10}$+ 2	+.400$_{10}$+ 1	+.63235$_{10}$- 1	+.200$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.470$_{10}$+ 2	+.400$_{10}$+ 1	+.60847$_{10}$- 1	+.193$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.480$_{10}$+ 2	+.400$_{10}$+ 1	+.58523$_{10}$- 1	+.187$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.490$_{10}$+ 2	+.400$_{10}$+ 1	+.56264$_{10}$- 1	+.180$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.500$_{10}$+ 2	+.400$_{10}$+ 1	+.54068$_{10}$- 1	+.174$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.510$_{10}$+ 2	+.400$_{10}$+ 1	+.51937$_{10}$- 1	+.168$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.520$_{10}$+ 2	+.400$_{10}$+ 1	+.49869$_{10}$- 1	+.162$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.530$_{10}$+ 2	+.400$_{10}$+ 1	+.47866$_{10}$- 1	+.157$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.540$_{10}$+ 2	+.400$_{10}$+ 1	+.45925$_{10}$- 1	+.151$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.550$_{10}$+ 2	+.400$_{10}$+ 1	+.44046$_{10}$- 1	+.146$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.560$_{10}$+ 2	+.400$_{10}$+ 1	+.42230$_{10}$- 1	+.141$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1
+.570$_{10}$+ 2	+.400$_{10}$+ 1	+.40473$_{10}$- 1	+.136$_{10}$+ 0	+.16696$_{10}$+ 0	+.500$_{10}$+ 2	+.400$_{10}$+ 1

```
      217 -  10 -     80672 - 06185479 STELTENPOO


VERTAALTIJD          10
EXECUTIETIJD         20

AANTAL OVER STROOM 1 GELEZEN GETALLEN              5
LAATST OVER STROOM 1 GELEZEN GETAL              +150

AANTAL OVER STROOM 2 GELEZEN GETALLEN              0
```

217 - 9 - 80672 - D6185479 STELTENPOO

n	k	Pred	Pe	Pdek		
+.148$_{10}$+ 3	+,500$_{10}$+ 1	+.13654$_{10}$- 9	+.182$_{10}$- 6	+.20516$_{10}$- 2	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.149$_{10}$+ 3	+,500$_{10}$+ 1	+.11914$_{10}$- 9	+.164$_{10}$- 6	+.20516$_{10}$- 2	+,860$_{10}$+ 2	+,200$_{10}$+ 2
+.150$_{10}$+ 3	+,500$_{10}$+ 1	+.10396$_{10}$- 9	+.148$_{10}$- 6	+.20516$_{10}$- 2	+,860$_{10}$+ 2	+,200$_{10}$+ 2

$p = p_2$

σ	k	p_{red}	p_c	p_{dek}		
+.880$_{10}$+ 2	+.500$_{10}$+ 1	+.39804$_{10}$- 6	+.984$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.890$_{10}$+ 2	+.500$_{10}$+ 1	+.35005$_{10}$- 6	+.886$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.900$_{10}$+ 2	+.500$_{10}$+ 1	+.30778$_{10}$- 6	+.798$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.910$_{10}$+ 2	+.500$_{10}$+ 1	+.27056$_{10}$- 6	+.718$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.920$_{10}$+ 2	+.500$_{10}$+ 1	+.23779$_{10}$- 6	+.647$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.930$_{10}$+ 2	+.500$_{10}$+ 1	+.20895$_{10}$- 6	+.582$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.940$_{10}$+ 2	+.500$_{10}$+ 1	+.18357$_{10}$- 6	+.524$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.950$_{10}$+ 2	+.500$_{10}$+ 1	+.16124$_{10}$- 6	+.472$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.960$_{10}$+ 2	+.500$_{10}$+ 1	+.14160$_{10}$- 6	+.425$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.970$_{10}$+ 2	+.500$_{10}$+ 1	+.12433$_{10}$- 6	+.383$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.980$_{10}$+ 2	+.500$_{10}$+ 1	+.10915$_{10}$- 6	+.345$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.990$_{10}$+ 2	+.500$_{10}$+ 1	+.95806$_{10}$- 7	+.310$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.100$_{10}$+ 3	+.500$_{10}$+ 1	+.84077$_{10}$- 7	+.280$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.101$_{10}$+ 3	+.500$_{10}$+ 1	+.73772$_{10}$- 7	+.252$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.102$_{10}$+ 3	+.500$_{10}$+ 1	+.64719$_{10}$- 7	+.227$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.103$_{10}$+ 3	+.500$_{10}$+ 1	+.56768$_{10}$- 7	+.204$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.104$_{10}$+ 3	+.500$_{10}$+ 1	+.49786$_{10}$- 7	+.184$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.105$_{10}$+ 3	+.500$_{10}$+ 1	+.43655$_{10}$- 7	+.165$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.106$_{10}$+ 3	+.500$_{10}$+ 1	+.38274$_{10}$- 7	+.149$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.107$_{10}$+ 3	+.500$_{10}$+ 1	+.33551$_{10}$- 7	+.134$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.108$_{10}$+ 3	+.500$_{10}$+ 1	+.29406$_{10}$- 7	+.121$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.109$_{10}$+ 3	+.500$_{10}$+ 1	+.25770$_{10}$- 7	+.109$_{10}$- 4	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.110$_{10}$+ 3	+.500$_{10}$+ 1	+.22580$_{10}$- 7	+.980$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.111$_{10}$+ 3	+.500$_{10}$+ 1	+.19782$_{10}$- 7	+.882$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.112$_{10}$+ 3	+.500$_{10}$+ 1	+.17328$_{10}$- 7	+.794$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.113$_{10}$+ 3	+.500$_{10}$+ 1	+.15177$_{10}$- 7	+.715$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.114$_{10}$+ 3	+.500$_{10}$+ 1	+.13291$_{10}$- 7	+.644$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.115$_{10}$+ 3	+.500$_{10}$+ 1	+.11638$_{10}$- 7	+.580$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.116$_{10}$+ 3	+.500$_{10}$+ 1	+.10189$_{10}$- 7	+.522$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.117$_{10}$+ 3	+.500$_{10}$+ 1	+.89190$_{10}$- 8	+.470$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.118$_{10}$+ 3	+.500$_{10}$+ 1	+.78066$_{10}$- 8	+.423$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.119$_{10}$+ 3	+.500$_{10}$+ 1	+.68321$_{10}$- 8	+.381$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.120$_{10}$+ 3	+.500$_{10}$+ 1	+.59785$_{10}$- 8	+.343$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.121$_{10}$+ 3	+.500$_{10}$+ 1	+.52310$_{10}$- 8	+.309$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.122$_{10}$+ 3	+.500$_{10}$+ 1	+.45763$_{10}$- 8	+.278$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.123$_{10}$+ 3	+.500$_{10}$+ 1	+.40032$_{10}$- 8	+.251$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.124$_{10}$+ 3	+.500$_{10}$+ 1	+.35014$_{10}$- 8	+.226$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.125$_{10}$+ 3	+.500$_{10}$+ 1	+.30622$_{10}$- 8	+.203$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.126$_{10}$+ 3	+.500$_{10}$+ 1	+.26777$_{10}$- 8	+.183$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.127$_{10}$+ 3	+.500$_{10}$+ 1	+.23413$_{10}$- 8	+.165$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.128$_{10}$+ 3	+.500$_{10}$+ 1	+.20469$_{10}$- 8	+.148$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.129$_{10}$+ 3	+.500$_{10}$+ 1	+.17894$_{10}$- 8	+.134$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.130$_{10}$+ 3	+.500$_{10}$+ 1	+.15641$_{10}$- 8	+.120$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.131$_{10}$+ 3	+.500$_{10}$+ 1	+.13670$_{10}$- 8	+.108$_{10}$- 5	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.132$_{10}$+ 3	+.500$_{10}$+ 1	+.11947$_{10}$- 8	+.976$_{10}$- 6	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.133$_{10}$+ 3	+.500$_{10}$+ 1	+.10439$_{10}$- 8	+.878$_{10}$- 6	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.134$_{10}$+ 3	+.500$_{10}$+ 1	+.91211$_{10}$- 9	+.791$_{10}$- 6	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.135$_{10}$+ 3	+.500$_{10}$+ 1	+.79687$_{10}$- 9	+.712$_{10}$- 6	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.136$_{10}$+ 3	+.500$_{10}$+ 1	+.69612$_{10}$- 9	+.641$_{10}$- 6	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.137$_{10}$+ 3	+.500$_{10}$+ 1	+.60806$_{10}$- 9	+.578$_{10}$- 6	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.138$_{10}$+ 3	+.500$_{10}$+ 1	+.53108$_{10}$- 9	+.520$_{10}$- 6	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.139$_{10}$+ 3	+.500$_{10}$+ 1	+.46381$_{10}$- 9	+.468$_{10}$- 6	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.140$_{10}$+ 3	+.500$_{10}$+ 1	+.40502$_{10}$- 9	+.422$_{10}$- 6	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.141$_{10}$+ 3	+.500$_{10}$+ 1	+.35366$_{10}$- 9	+.380$_{10}$- 6	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.142$_{10}$+ 3	+.500$_{10}$+ 1	+.30878$_{10}$- 9	+.342$_{10}$- 6	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.143$_{10}$+ 3	+.500$_{10}$+ 1	+.26957$_{10}$- 9	+.308$_{10}$- 6	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.144$_{10}$+ 3	+.500$_{10}$+ 1	+.23532$_{10}$- 9	+.277$_{10}$- 6	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.145$_{10}$+ 3	+.500$_{10}$+ 1	+.20541$_{10}$- 9	+.250$_{10}$- 6	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2
+.146$_{10}$+ 3	+.500$_{10}$+ 1	+.17928$_{10}$- 9	+.225$_{10}$- 6	+.20516$_{10}$- 2	+.860$_{10}$+ 2	+.200$_{10}$+ 2

	n	m	N	pred	pc							
VERBETERING	+.140$_{10}$+	2	+.200$_{10}$+	1	+.100$_{10}$+	1	+.770$_{10}$+	0	+.741$_{10}$+	0	+.999$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.150$_{10}$+	2	+.200$_{10}$+	1	+.100$_{10}$+	1	+.754$_{10}$+	0	+.726$_{10}$+	0	+.999$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.160$_{10}$+	2	+.200$_{10}$+	1	+.100$_{10}$+	1	+.738$_{10}$+	0	+.712$_{10}$+	0	+.999$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.170$_{10}$+	2	+.200$_{10}$+	1	+.100$_{10}$+	1	+.722$_{10}$+	0	+.698$_{10}$+	0	+.999$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.180$_{10}$+	2	+.200$_{10}$+	1	+.100$_{10}$+	1	+.705$_{10}$+	0	+.684$_{10}$+	0	+.999$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.190$_{10}$+	2	+.200$_{10}$+	1	+.100$_{10}$+	1	+.689$_{10}$+	0	+.671$_{10}$+	0	+.999$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.200$_{10}$+	2	+.200$_{10}$+	1	+.100$_{10}$+	1	+.672$_{10}$+	0	+.657$_{10}$+	0	+.999$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.210$_{10}$+	2	+.200$_{10}$+	1	+.100$_{10}$+	1	+.656$_{10}$+	0	+.644$_{10}$+	0	+.999$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.220$_{10}$+	2	+.200$_{10}$+	1	+.100$_{10}$+	1	+.640$_{10}$+	0	+.632$_{10}$+	0	+.999$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.230$_{10}$+	2	+.200$_{10}$+	1	+.100$_{10}$+	1	+.624$_{10}$+	0	+.619$_{10}$+	0	+.999$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.240$_{10}$+	2	+.200$_{10}$+	1	+.100$_{10}$+	1	+.607$_{10}$+	0	+.607$_{10}$+	0	+.999$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.300$_{10}$+	1	+.300$_{10}$+	1	+.100$_{10}$+	1	+.891$_{10}$+	0	+.887$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.400$_{10}$+	1	+.300$_{10}$+	1	+.100$_{10}$+	1	+.875$_{10}$+	0	+.861$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.500$_{10}$+	1	+.300$_{10}$+	1	+.100$_{10}$+	1	+.856$_{10}$+	0	+.835$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.600$_{10}$+	1	+.300$_{10}$+	1	+.100$_{10}$+	1	+.837$_{10}$+	0	+.811$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.700$_{10}$+	1	+.300$_{10}$+	1	+.100$_{10}$+	1	+.815$_{10}$+	0	+.787$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.800$_{10}$+	1	+.300$_{10}$+	1	+.100$_{10}$+	1	+.793$_{10}$+	0	+.764$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.900$_{10}$+	1	+.300$_{10}$+	1	+.100$_{10}$+	1	+.770$_{10}$+	0	+.741$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.100$_{10}$+	2	+.300$_{10}$+	1	+.100$_{10}$+	1	+.746$_{10}$+	0	+.719$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.110$_{10}$+	2	+.300$_{10}$+	1	+.100$_{10}$+	1	+.722$_{10}$+	0	+.698$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.120$_{10}$+	2	+.300$_{10}$+	1	+.100$_{10}$+	1	+.697$_{10}$+	0	+.677$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.130$_{10}$+	2	+.300$_{10}$+	1	+.100$_{10}$+	1	+.672$_{10}$+	0	+.657$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.140$_{10}$+	2	+.300$_{10}$+	1	+.100$_{10}$+	1	+.648$_{10}$+	0	+.638$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.150$_{10}$+	2	+.300$_{10}$+	1	+.100$_{10}$+	1	+.624$_{10}$+	0	+.619$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+	1
	+.100$_{10}$+	1	+.400$_{10}$+	1	+.100$_{10}$+	1	+.908$_{10}$+	0	+.923$_{10}$+	0	+.996$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.200$_{10}$+	1	+.400$_{10}$+	1	+.100$_{10}$+	1	+.891$_{10}$+	0	+.887$_{10}$+	0	+.996$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.300$_{10}$+	1	+.400$_{10}$+	1	+.100$_{10}$+	1	+.869$_{10}$+	0	+.852$_{10}$+	0	+.996$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.400$_{10}$+	1	+.400$_{10}$+	1	+.100$_{10}$+	1	+.843$_{10}$+	0	+.819$_{10}$+	0	+.996$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.500$_{10}$+	1	+.400$_{10}$+	1	+.100$_{10}$+	1	+.815$_{10}$+	0	+.787$_{10}$+	0	+.996$_{10}$+	0	+.100$_{10}$+	1
VERBETERING	+.600$_{10}$+	1	+.400$_{10}$+	1	+.100$_{10}$+	1	+.785$_{10}$+	0	+.756$_{10}$+	0	+.996$_{10}$+	0	+.100$_{10}$+	1

```
 29 -    0 -     80672 - 0618547? STELTENPOO

 1 BEGIN COMMENT BEREKENING OPBRENGST-VERBETERING VAN CELLULAIRE NETWERKEN, (WORST-CASE BEREKENING),
 2       N = AANTAL INPUTS CELL. NETWERK
 3       W = AANTAL OUTPUTS CELL. NETWERK
 4       M = AANTAL CASCADES HOOFDNETWERK
 5       1 - P IS FOUT KANS CEL;
 6       INTEGER WMAX, MMAX, NMAX, T, N, M, N, J, V, I, K;
 7       REAL P, P8N2, PM, PNM, P2NM, PMN, PRED, PC, NSUBF, SOM, KFAK, NOVK, P2MN, P3MN, PN;
 8       J:= 0;
 9 B:    NMAX:= READ; MMAX:= READ; NMAX:= READ;
10       T:= READ;
11     P:= EXP(LN(0.95)/T); FLOT(3, 2, P); NLCR; NLCR;
12       FOR N:= 1 STEP 1 UNTIL NMAX DO
13       BEGIN P8N2:= P↑(8 * (N ↑ 2));
14             FOR M:= 1 STEP 1 UNTIL MMAX DO
15             BEGIN PM:= P ↑ M;
16                   PNM:= P ↑ (N * M); P2NM:= PNM * PNM;
17                   PMN:= 1; PN:= 1; SOM:= P2NM * PNM;
18                    FOR K:= 1 STEP 1 UNTIL N DO
19                   BEGIN  NSUBF:= 1; KFAK:= 1;  FOR V:= N-K+1 STEP 1 UNTIL N DO NSUBF:= NSUBF * V;
20                          FOR I:= 1 STEP 1 UNTIL K DO KFAK:= KFAK * I; NOVK:= (NSUBF/KFAK);
21                          SOM:= SOM + (3↑K)* NOVK * (( 1 - PM )↑K) * ( PM↑( 3 * N - K ));
22                   END;
23                   FOR N:= 1 STEP 1 UNTIL NMAX DO
24                   BEGIN PMN:= PMN * PM; PN:= P * PN;
25                         IF (PN * PM) < 0.5 THEN GOTO A;
26                         P2MN:= PMN * PMN; P3MN:= P2MN *PMN;
27                         PRED:= ( SOM * P3MN + 3*( 1 - PMN ) * P2MN * P2NM ) * P8N2;
28                         PC:= PMN * PNM;
29                         IF PRED < PC THEN GOTO C;
30                         PRINTTEXT('(VERBETERING)');
31                         GOTO E;
32 C:                      IF M ≠ 4 THEN GOTO D;
33 E:                     NLCR; SPACE(15); FLOT(3, 2, N); FLOT(3, 2, M);
34                         FLOT(3, 2, N); FLOT(3, 2, PRED); FLOT(3, 2, PC); FLOT(3, 2, SOM); FLOT(3, 2, NOVK); NLCR;
35 D:                 END;
36 A:
37           END;
38       END; J:= J + 1; IF J = 1 THEN GOTO B;
39 END
40 EOQGEND
```

29 - 1 - 80672 - 06185479 S ELTENPOO

$p=p_1$

+.990$_{10}$+ 0

	n	m	N	pred	pe	som	
VERBETERING	+.110$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.891$_{10}$+ 0	+.887$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.120$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.886$_{10}$+ 0	+.878$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.130$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.880$_{10}$+ 0	+.870$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.140$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.875$_{10}$+ 0	+.861$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.150$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.869$_{10}$+ 0	+.852$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.160$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.863$_{10}$+ 0	+.844$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.170$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.856$_{10}$+ 0	+.835$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.180$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.850$_{10}$+ 0	+.827$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.190$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.843$_{10}$+ 0	+.819$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.200$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.837$_{10}$+ 0	+.811$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.210$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.830$_{10}$+ 0	+.803$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.220$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.823$_{10}$+ 0	+.795$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.230$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.815$_{10}$+ 0	+.787$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.240$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.808$_{10}$+ 0	+.779$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.250$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.800$_{10}$+ 0	+.771$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.260$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.793$_{10}$+ 0	+.764$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.270$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.785$_{10}$+ 0	+.756$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.280$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.777$_{10}$+ 0	+.749$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.290$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.770$_{10}$+ 0	+.741$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.300$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.762$_{10}$+ 0	+.734$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.500$_{10}$+ 1	+.200$_{10}$+ 1	+.100$_{10}$+ 1	+.891$_{10}$+ 0	+.887$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.600$_{10}$+ 1	+.200$_{10}$+ 1	+.100$_{10}$+ 1	+.880$_{10}$+ 0	+.870$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.700$_{10}$+ 1	+.200$_{10}$+ 1	+.100$_{10}$+ 1	+.869$_{10}$+ 0	+.852$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.800$_{10}$+ 1	+.200$_{10}$+ 1	+.100$_{10}$+ 1	+.856$_{10}$+ 0	+.835$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.900$_{10}$+ 1	+.200$_{10}$+ 1	+.100$_{10}$+ 1	+.843$_{10}$+ 0	+.819$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.100$_{10}$+ 2	+.200$_{10}$+ 1	+.100$_{10}$+ 1	+.830$_{10}$+ 0	+.803$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.110$_{10}$+ 2	+.200$_{10}$+ 1	+.100$_{10}$+ 1	+.815$_{10}$+ 0	+.787$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.120$_{10}$+ 2	+.200$_{10}$+ 1	+.100$_{10}$+ 1	+.800$_{10}$+ 0	+.771$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.130$_{10}$+ 2	+.200$_{10}$+ 1	+.100$_{10}$+ 1	+.785$_{10}$+ 0	+.756$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1

p	m	N	pred	pe									
+.300$_{10}$+	1	+.400$_{10}$+	1	+.200$_{10}$+	1	+.668$_{10}$+	0	+.819$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.400$_{10}$+	1	+.400$_{10}$+	1	+.200$_{10}$+	1	+.645$_{10}$+	0	+.787$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.500$_{10}$+	1	+.400$_{10}$+	1	+.200$_{10}$+	1	+.621$_{10}$+	0	+.756$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.600$_{10}$+	1	+.400$_{10}$+	1	+.200$_{10}$+	1	+.596$_{10}$+	0	+.726$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.700$_{10}$+	1	+.400$_{10}$+	1	+.200$_{10}$+	1	+.570$_{10}$+	0	+.698$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.800$_{10}$+	1	+.400$_{10}$+	1	+.200$_{10}$+	1	+.544$_{10}$+	0	+.671$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.900$_{10}$+	1	+.400$_{10}$+	1	+.200$_{10}$+	1	+.518$_{10}$+	0	+.644$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.100$_{10}$+	2	+.400$_{10}$+	1	+.200$_{10}$+	1	+.492$_{10}$+	0	+.619$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.110$_{10}$+	2	+.400$_{10}$+	1	+.200$_{10}$+	1	+.467$_{10}$+	0	+.595$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.120$_{10}$+	2	+.400$_{10}$+	1	+.200$_{10}$+	1	+.442$_{10}$+	0	+.572$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.130$_{10}$+	2	+.400$_{10}$+	1	+.200$_{10}$+	1	+.418$_{10}$+	0	+.549$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.140$_{10}$+	2	+.400$_{10}$+	1	+.200$_{10}$+	1	+.395$_{10}$+	0	+.528$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.150$_{10}$+	2	+.400$_{10}$+	1	+.200$_{10}$+	1	+.372$_{10}$+	0	+.507$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.160$_{10}$+	2	+.400$_{10}$+	1	+.200$_{10}$+	1	+.350$_{10}$+	0	+.487$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.170$_{10}$+	2	+.400$_{10}$+	1	+.200$_{10}$+	1	+.329$_{10}$+	0	+.468$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.180$_{10}$+	2	+.400$_{10}$+	1	+.200$_{10}$+	1	+.309$_{10}$+	0	+.450$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.190$_{10}$+	2	+.400$_{10}$+	1	+.200$_{10}$+	1	+.290$_{10}$+	0	+.432$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.200$_{10}$+	2	+.400$_{10}$+	1	+.200$_{10}$+	1	+.272$_{10}$+	0	+.415$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.210$_{10}$+	2	+.400$_{10}$+	1	+.200$_{10}$+	1	+.255$_{10}$+	0	+.399$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.220$_{10}$+	2	+.400$_{10}$+	1	+.200$_{10}$+	1	+.239$_{10}$+	0	+.383$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.230$_{10}$+	2	+.400$_{10}$+	1	+.200$_{10}$+	1	+.224$_{10}$+	0	+.368$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.240$_{10}$+	2	+.400$_{10}$+	1	+.200$_{10}$+	1	+.209$_{10}$+	0	+.354$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.250$_{10}$+	2	+.400$_{10}$+	1	+.200$_{10}$+	1	+.195$_{10}$+	0	+.340$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.260$_{10}$+	2	+.400$_{10}$+	1	+.200$_{10}$+	1	+.182$_{10}$+	0	+.327$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.270$_{10}$+	2	+.400$_{10}$+	1	+.200$_{10}$+	1	+.170$_{10}$+	0	+.314$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.280$_{10}$+	2	+.400$_{10}$+	1	+.200$_{10}$+	1	+.159$_{10}$+	0	+.302$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.290$_{10}$+	2	+.400$_{10}$+	1	+.200$_{10}$+	1	+.148$_{10}$+	0	+.290$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.300$_{10}$+	2	+.400$_{10}$+	1	+.200$_{10}$+	1	+.138$_{10}$+	0	+.279$_{10}$+	0	+.991$_{10}$+	0	+.100$_{10}$+	1
+.100$_{10}$+	1	+.400$_{10}$+	1	+.300$_{10}$+	1	+.468$_{10}$+	0	+.852$_{10}$+	0	+.987$_{10}$+	0	+.100$_{10}$+	1
+.200$_{10}$+	1	+.400$_{10}$+	1	+.300$_{10}$+	1	+.454$_{10}$+	0	+.819$_{10}$+	0	+.987$_{10}$+	0	+.100$_{10}$+	1

	n	m	N	Pred	Pe
VERBETERING	+.800$_{10}$+ 1	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.648$_{10}$+ 0	+.638$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.900$_{10}$+ 1	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.607$_{10}$+ 0	+.607$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.100$_{10}$+ 1	+.600$_{10}$+ 1	+.100$_{10}$+ 1	+.891$_{10}$+ 0	+.887$_{10}$+ 0	+.990$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.200$_{10}$+ 1	+.600$_{10}$+ 1	+.100$_{10}$+ 1	+.856$_{10}$+ 0	+.835$_{10}$+ 0	+.990$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.300$_{10}$+ 1	+.600$_{10}$+ 1	+.100$_{10}$+ 1	+.815$_{10}$+ 0	+.787$_{10}$+ 0	+.990$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.100$_{10}$+ 1	+.770$_{10}$+ 0	+.741$_{10}$+ 0	+.990$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.500$_{10}$+ 1	+.600$_{10}$+ 1	+.100$_{10}$+ 1	+.722$_{10}$+ 0	+.698$_{10}$+ 0	+.990$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.600$_{10}$+ 1	+.600$_{10}$+ 1	+.100$_{10}$+ 1	+.672$_{10}$+ 0	+.657$_{10}$+ 0	+.990$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.700$_{10}$+ 1	+.600$_{10}$+ 1	+.100$_{10}$+ 1	+.624$_{10}$+ 0	+.619$_{10}$+ 0	+.990$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.100$_{10}$+ 1	+.700$_{10}$+ 1	+.100$_{10}$+ 1	+.880$_{10}$+ 0	+.870$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.200$_{10}$+ 1	+.700$_{10}$+ 1	+.100$_{10}$+ 1	+.837$_{10}$+ 0	+.811$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.300$_{10}$+ 1	+.700$_{10}$+ 1	+.100$_{10}$+ 1	+.785$_{10}$+ 0	+.756$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.400$_{10}$+ 1	+.700$_{10}$+ 1	+.100$_{10}$+ 1	+.730$_{10}$+ 0	+.705$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.500$_{10}$+ 1	+.700$_{10}$+ 1	+.100$_{10}$+ 1	+.672$_{10}$+ 0	+.657$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.600$_{10}$+ 1	+.700$_{10}$+ 1	+.100$_{10}$+ 1	+.616$_{10}$+ 0	+.613$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.100$_{10}$+ 1	+.800$_{10}$+ 1	+.100$_{10}$+ 1	+.869$_{10}$+ 0	+.852$_{10}$+ 0	+.983$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.200$_{10}$+ 1	+.800$_{10}$+ 1	+.100$_{10}$+ 1	+.815$_{10}$+ 0	+.787$_{10}$+ 0	+.983$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.300$_{10}$+ 1	+.800$_{10}$+ 1	+.100$_{10}$+ 1	+.754$_{10}$+ 0	+.726$_{10}$+ 0	+.983$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.400$_{10}$+ 1	+.800$_{10}$+ 1	+.100$_{10}$+ 1	+.689$_{10}$+ 0	+.671$_{10}$+ 0	+.983$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.500$_{10}$+ 1	+.800$_{10}$+ 1	+.100$_{10}$+ 1	+.624$_{10}$+ 0	+.619$_{10}$+ 0	+.983$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.100$_{10}$+ 1	+.900$_{10}$+ 1	+.100$_{10}$+ 1	+.856$_{10}$+ 0	+.835$_{10}$+ 0	+.979$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.200$_{10}$+ 1	+.900$_{10}$+ 1	+.100$_{10}$+ 1	+.793$_{10}$+ 0	+.764$_{10}$+ 0	+.979$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.300$_{10}$+ 1	+.900$_{10}$+ 1	+.100$_{10}$+ 1	+.722$_{10}$+ 0	+.698$_{10}$+ 0	+.979$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.400$_{10}$+ 1	+.900$_{10}$+ 1	+.100$_{10}$+ 1	+.648$_{10}$+ 0	+.638$_{10}$+ 0	+.979$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.100$_{10}$+ 1	+.100$_{10}$+ 2	+.100$_{10}$+ 1	+.843$_{10}$+ 0	+.819$_{10}$+ 0	+.975$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.200$_{10}$+ 1	+.100$_{10}$+ 2	+.100$_{10}$+ 1	+.770$_{10}$+ 0	+.741$_{10}$+ 0	+.975$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.300$_{10}$+ 1	+.100$_{10}$+ 2	+.100$_{10}$+ 1	+.689$_{10}$+ 0	+.671$_{10}$+ 0	+.975$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.400$_{10}$+ 1	+.100$_{10}$+ 2	+.100$_{10}$+ 1	+.607$_{10}$+ 0	+.607$_{10}$+ 0	+.975$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.100$_{10}$+ 1	+.400$_{10}$+ 1	+.200$_{10}$+ 1	+.706$_{10}$+ 0	+.887$_{10}$+ 0	+.991$_{10}$+ 0	+.100$_{10}$+ 1
	+.200$_{10}$+ 1	+.400$_{10}$+ 1	+.200$_{10}$+ 1	+.688$_{10}$+ 0	+.852$_{10}$+ 0	+.991$_{10}$+ 0	+.100$_{10}$+ 1

	n	m	N	pred	pc
VERBETERING	+.700$_{10}$+ 1	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.754$_{10}$+ 0	+.726$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.800$_{10}$+ 1	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.722$_{10}$+ 0	+.698$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.900$_{10}$+ 1	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.689$_{10}$+ 0	+.671$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.100$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.656$_{10}$+ 0	+.644$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.110$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.624$_{10}$+ 0	+.619$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
	+.120$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.592$_{10}$+ 0	+.595$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
	+.130$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.560$_{10}$+ 0	+.572$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
	+.140$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.530$_{10}$+ 0	+.549$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
	+.150$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.500$_{10}$+ 0	+.528$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
	+.160$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.471$_{10}$+ 0	+.507$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
	+.170$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.444$_{10}$+ 0	+.487$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
	+.180$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.418$_{10}$+ 0	+.468$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
	+.190$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.392$_{10}$+ 0	+.450$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
	+.200$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.368$_{10}$+ 0	+.432$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
	+.210$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.345$_{10}$+ 0	+.415$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
	+.220$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.324$_{10}$+ 0	+.399$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
	+.230$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.303$_{10}$+ 0	+.383$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
	+.240$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.284$_{10}$+ 0	+.368$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
	+.250$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.265$_{10}$+ 0	+.354$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
	+.260$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.248$_{10}$+ 0	+.340$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
	+.270$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.231$_{10}$+ 0	+.327$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
	+.280$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.216$_{10}$+ 0	+.314$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
	+.290$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.201$_{10}$+ 0	+.302$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
	+.300$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.188$_{10}$+ 0	+.290$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.200$_{10}$+ 1	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.875$_{10}$+ 0	+.861$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.300$_{10}$+ 1	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.843$_{10}$+ 0	+.819$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.808$_{10}$+ 0	+.779$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.500$_{10}$+ 1	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.770$_{10}$+ 0	+.741$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.600$_{10}$+ 1	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.730$_{10}$+ 0	+.705$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.700$_{10}$+ 1	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.689$_{10}$+ 0	+.671$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1

n	m	N	pred	Pc
+.300$_{10}$+ 1	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.117$_{10}$+ 0	+.726$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.111$_{10}$+ 0	+.698$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.500$_{10}$+ 1	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.106$_{10}$+ 0	+.671$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.600$_{10}$+ 1	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.101$_{10}$+ 0	+.644$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.700$_{10}$+ 1	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.954$_{10}$- 1	+.619$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.800$_{10}$+ 1	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.903$_{10}$- 1	+.595$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.900$_{10}$+ 1	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.853$_{10}$- 1	+.572$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.100$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.805$_{10}$- 1	+.549$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.110$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.758$_{10}$- 1	+.528$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.120$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.714$_{10}$- 1	+.507$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.130$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.671$_{10}$- 1	+.487$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.140$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.630$_{10}$- 1	+.468$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.150$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.591$_{10}$- 1	+.450$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.160$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.554$_{10}$- 1	+.432$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.170$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.519$_{10}$- 1	+.415$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.180$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.486$_{10}$- 1	+.399$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.190$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.455$_{10}$- 1	+.383$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.200$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.425$_{10}$- 1	+.368$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.210$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.397$_{10}$- 1	+.354$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.220$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.370$_{10}$- 1	+.340$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.230$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.346$_{10}$- 1	+.327$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.240$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.322$_{10}$- 1	+.314$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.250$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.300$_{10}$- 1	+.302$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.260$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.280$_{10}$- 1	+.290$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.270$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.261$_{10}$- 1	+.279$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.280$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.243$_{10}$- 1	+.268$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.290$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.226$_{10}$- 1	+.257$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.300$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.210$_{10}$- 1	+.247$_{10}$+ 0	+.978$_{10}$+ 0	+.100$_{10}$+ 1
+.100$_{10}$+ 1	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.525$_{10}$- 1	+.756$_{10}$+ 0	+.973$_{10}$+ 0	+.100$_{10}$+ 1
+.200$_{10}$+ 1	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.500$_{10}$- 1	+.726$_{10}$+ 0	+.973$_{10}$+ 0	+.100$_{10}$+ 1

$p = p_3$

29 - 7 - 80672 - 06185479 STELTENPOO

n	m	N	pred	pe		
$+.300_{10}+1$	$+.400_{10}+1$	$+.400_{10}+1$	$+.245_{10}+0$	$+.756_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.400_{10}+1$	$+.400_{10}+1$	$+.400_{10}+1$	$+.235_{10}+0$	$+.726_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.500_{10}+1$	$+.400_{10}+1$	$+.400_{10}+1$	$+.224_{10}+0$	$+.698_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.600_{10}+1$	$+.400_{10}+1$	$+.400_{10}+1$	$+.213_{10}+0$	$+.671_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.700_{10}+1$	$+.400_{10}+1$	$+.400_{10}+1$	$+.203_{10}+0$	$+.644_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.800_{10}+1$	$+.400_{10}+1$	$+.400_{10}+1$	$+.193_{10}+0$	$+.619_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.900_{10}+1$	$+.400_{10}+1$	$+.400_{10}+1$	$+.182_{10}+0$	$+.595_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.100_{10}+2$	$+.400_{10}+1$	$+.400_{10}+1$	$+.172_{10}+0$	$+.572_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.110_{10}+2$	$+.400_{10}+1$	$+.400_{10}+1$	$+.163_{10}+0$	$+.549_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.120_{10}+2$	$+.400_{10}+1$	$+.400_{10}+1$	$+.154_{10}+0$	$+.528_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.130_{10}+2$	$+.400_{10}+1$	$+.400_{10}+1$	$+.145_{10}+0$	$+.507_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.140_{10}+2$	$+.400_{10}+1$	$+.400_{10}+1$	$+.136_{10}+0$	$+.487_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.150_{10}+2$	$+.400_{10}+1$	$+.400_{10}+1$	$+.128_{10}+0$	$+.468_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.160_{10}+2$	$+.400_{10}+1$	$+.400_{10}+1$	$+.120_{10}+0$	$+.450_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.170_{10}+2$	$+.400_{10}+1$	$+.400_{10}+1$	$+.113_{10}+0$	$+.432_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.180_{10}+2$	$+.400_{10}+1$	$+.400_{10}+1$	$+.106_{10}+0$	$+.415_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.190_{10}+2$	$+.400_{10}+1$	$+.400_{10}+1$	$+.988_{10}-1$	$+.399_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.200_{10}+2$	$+.400_{10}+1$	$+.400_{10}+1$	$+.925_{10}-1$	$+.383_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.210_{10}+2$	$+.400_{10}+1$	$+.400_{10}+1$	$+.865_{10}-1$	$+.368_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.220_{10}+2$	$+.400_{10}+1$	$+.400_{10}+1$	$+.808_{10}-1$	$+.354_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.230_{10}+2$	$+.400_{10}+1$	$+.400_{10}+1$	$+.755_{10}-1$	$+.340_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.240_{10}+2$	$+.400_{10}+1$	$+.400_{10}+1$	$+.704_{10}-1$	$+.327_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.250_{10}+2$	$+.400_{10}+1$	$+.400_{10}+1$	$+.657_{10}-1$	$+.314_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.260_{10}+2$	$+.400_{10}+1$	$+.400_{10}+1$	$+.613_{10}-1$	$+.302_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.270_{10}+2$	$+.400_{10}+1$	$+.400_{10}+1$	$+.571_{10}-1$	$+.290_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.280_{10}+2$	$+.400_{10}+1$	$+.400_{10}+1$	$+.532_{10}-1$	$+.279_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.290_{10}+2$	$+.400_{10}+1$	$+.400_{10}+1$	$+.495_{10}-1$	$+.268_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.300_{10}+2$	$+.400_{10}+1$	$+.400_{10}+1$	$+.461_{10}-1$	$+.257_{10}+0$	$+.982_{10}+0$	$+.100_{10}+1$
$+.100_{10}+1$	$+.400_{10}+1$	$+.500_{10}+1$	$+.128_{10}+0$	$+.787_{10}+0$	$+.978_{10}+0$	$+.100_{10}+1$
$+.200_{10}+1$	$+.400_{10}+1$	$+.500_{10}+1$	$+.122_{10}+0$	$+.756_{10}+0$	$+.978_{10}+0$	$+.100_{10}+1$

$p=p_1$

n	m	N	pred	pc		
+.300$_{10}$+ 1	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.438$_{10}$+ 0	+.787$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.421$_{10}$+ 0	+.756$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.500$_{10}$+ 1	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.404$_{10}$+ 0	+.726$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.600$_{10}$+ 1	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.386$_{10}$+ 0	+.698$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.700$_{10}$+ 1	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.368$_{10}$+ 0	+.671$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.800$_{10}$+ 1	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.350$_{10}$+ 0	+.644$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.900$_{10}$+ 1	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.333$_{10}$+ 0	+.619$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.100$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.315$_{10}$+ 0	+.595$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.110$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.299$_{10}$+ 0	+.572$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.120$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.282$_{10}$+ 0	+.549$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.130$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.266$_{10}$+ 0	+.528$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.140$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.251$_{10}$+ 0	+.507$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.150$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.236$_{10}$+ 0	+.487$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.160$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.222$_{10}$+ 0	+.468$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.170$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.208$_{10}$+ 0	+.450$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.180$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.196$_{10}$+ 0	+.432$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.190$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.183$_{10}$+ 0	+.415$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.200$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.172$_{10}$+ 0	+.399$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.210$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.161$_{10}$+ 0	+.383$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.220$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.150$_{10}$+ 0	+.368$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.230$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.141$_{10}$+ 0	+.354$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.240$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.131$_{10}$+ 0	+.340$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.250$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.123$_{10}$+ 0	+.327$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.260$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.114$_{10}$+ 0	+.314$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.270$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.107$_{10}$+ 0	+.302$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.280$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.995$_{10}$- 1	+.290$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.290$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.927$_{10}$- 1	+.279$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.300$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.863$_{10}$- 1	+.268$_{10}$+ 0	+.987$_{10}$+ 0	+.100$_{10}$+ 1
+.100$_{10}$+ 1	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.265$_{10}$+ 0	+.819$_{10}$+ 0	+.982$_{10}$+ 0	+.100$_{10}$+ 1
+.200$_{10}$+ 1	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.255$_{10}$+ 0	+.787$_{10}$+ 0	+.982$_{10}$+ 0	+.100$_{10}$+ 1

29 = 12 - 80672 - 0618547 STELTENPOO

VERBETERING +.700$_{10}$+ 2 +.100$_{10}$+ 1 +.100$_{10}$+ 1 +.755$_{10}$+ 0 +.702$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.954$_{10}$+ 0 +.951$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.500$_{10}$+ 1 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.951$_{10}$+ 0 +.942$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.600$_{10}$+ 1 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.948$_{10}$+ 0 +.932$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.700$_{10}$+ 1 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.945$_{10}$+ 0 +.923$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.800$_{10}$+ 1 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.941$_{10}$+ 0 +.914$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.900$_{10}$+ 1 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.936$_{10}$+ 0 +.905$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.100$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.932$_{10}$+ 0 +.896$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.110$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.927$_{10}$+ 0 +.887$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.120$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.922$_{10}$+ 0 +.878$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.130$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.916$_{10}$+ 0 +.870$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.140$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.910$_{10}$+ 0 +.861$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.150$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.904$_{10}$+ 0 +.852$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.160$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.898$_{10}$+ 0 +.844$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.170$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.891$_{10}$+ 0 +.835$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.180$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.885$_{10}$+ 0 +.827$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.190$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.878$_{10}$+ 0 +.819$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.200$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.871$_{10}$+ 0 +.811$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.210$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.863$_{10}$+ 0 +.803$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.220$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.856$_{10}$+ 0 +.795$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.230$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.849$_{10}$+ 0 +.787$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.240$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.841$_{10}$+ 0 +.779$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.250$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.833$_{10}$+ 0 +.771$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.260$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.825$_{10}$+ 0 +.764$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.270$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.817$_{10}$+ 0 +.756$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.280$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.809$_{10}$+ 0 +.749$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.290$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.801$_{10}$+ 0 +.741$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.300$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.793$_{10}$+ 0 +.734$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.310$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.785$_{10}$+ 0 +.726$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

VERBETERING +.320$_{10}$+ 2 +.200$_{10}$+ 1 +.100$_{10}$+ 1 +.776$_{10}$+ 0 +.719$_{10}$+ 0 +.100$_{10}$+ 1 +.100$_{10}$+ 1

	n	m	N	Pacd	pe					
VERBETERING	+.400$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.874$_{10}$+ 0	+.815$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.410$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.871$_{10}$+ 0	+.811$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.420$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.867$_{10}$+ 0	+.807$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.430$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.863$_{10}$+ 0	+.803$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.440$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.860$_{10}$+ 0	+.799$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.450$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.856$_{10}$+ 0	+.795$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.460$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.852$_{10}$+ 0	+.791$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.470$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.849$_{10}$+ 0	+.787$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.480$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.845$_{10}$+ 0	+.783$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.490$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.841$_{10}$+ 0	+.779$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.500$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.837$_{10}$+ 0	+.775$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.510$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.833$_{10}$+ 0	+.771$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.520$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.829$_{10}$+ 0	+.767$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.530$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.825$_{10}$+ 0	+.764$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.540$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.821$_{10}$+ 0	+.760$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.550$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.817$_{10}$+ 0	+.756$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.560$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.813$_{10}$+ 0	+.752$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.570$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.809$_{10}$+ 0	+.749$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.580$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.805$_{10}$+ 0	+.745$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.590$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.801$_{10}$+ 0	+.741$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.600$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.797$_{10}$+ 0	+.737$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.610$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.793$_{10}$+ 0	+.734$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.620$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.789$_{10}$+ 0	+.730$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.630$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.785$_{10}$+ 0	+.726$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.640$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.780$_{10}$+ 0	+.723$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.650$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.776$_{10}$+ 0	+.719$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.660$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.772$_{10}$+ 0	+.716$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.670$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.768$_{10}$+ 0	+.712$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.680$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.764$_{10}$+ 0	+.709$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1
VERBETERING	+.690$_{10}$+ 2	+.100$_{10}$+ 1	+.100$_{10}$+ 1	+.759$_{10}$+ 0	+.705$_{10}$+ 0	+.100$_{10}$+ 1	+.100$_{10}$+ 1

$p = p_2$

29 - 10 - 80672 - 06185479 STELTENPOO

	n	m	N	pred	pc		
VERBETERING	$+.100_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.953_{10}+0$	$+.947_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.110_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.951_{10}+0$	$+.942_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.120_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.950_{10}+0$	$+.937_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.130_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.948_{10}+0$	$+.932_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.140_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.947_{10}+0$	$+.928_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.150_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.945_{10}+0$	$+.923_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.160_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.943_{10}+0$	$+.919_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.170_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.941_{10}+0$	$+.914_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.180_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.939_{10}+0$	$+.909_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.190_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.936_{10}+0$	$+.905_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.200_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.934_{10}+0$	$+.900_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.210_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.932_{10}+0$	$+.896_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.220_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.929_{10}+0$	$+.892_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.230_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.927_{10}+0$	$+.887_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.240_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.924_{10}+0$	$+.883_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.250_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.922_{10}+0$	$+.878_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.260_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.919_{10}+0$	$+.874_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.270_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.916_{10}+0$	$+.870_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.280_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.913_{10}+0$	$+.865_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.290_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.910_{10}+0$	$+.861_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.300_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.907_{10}+0$	$+.857_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.310_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.904_{10}+0$	$+.852_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.320_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.901_{10}+0$	$+.848_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.330_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.898_{10}+0$	$+.844_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.340_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.895_{10}+0$	$+.840_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.350_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.891_{10}+0$	$+.835_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.360_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.888_{10}+0$	$+.831_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.370_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.895_{10}+0$	$+.827_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.380_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.881_{10}+0$	$+.823_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.390_{10}+2$	$+.100_{10}+1$	$+.100_{10}+1$	$+.878_{10}+0$	$+.819_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$

$p = p_1$

29 - 9 - 80672 - 06185479 STELTENPOO

n	m	N	pred	pc		
$+.300_{10}+1$	$+.400_{10}+1$	$+.600_{10}+1$	$+.476_{10}-1$	$+.698_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.400_{10}+1$	$+.400_{10}+1$	$+.600_{10}+1$	$+.452_{10}-1$	$+.671_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.500_{10}+1$	$+.400_{10}+1$	$+.600_{10}+1$	$+.429_{10}-1$	$+.644_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.600_{10}+1$	$+.400_{10}+1$	$+.600_{10}+1$	$+.406_{10}-1$	$+.619_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.700_{10}+1$	$+.400_{10}+1$	$+.600_{10}+1$	$+.383_{10}-1$	$+.595_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.800_{10}+1$	$+.400_{10}+1$	$+.600_{10}+1$	$+.362_{10}-1$	$+.572_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.900_{10}+1$	$+.400_{10}+1$	$+.600_{10}+1$	$+.341_{10}-1$	$+.549_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.100_{10}+2$	$+.400_{10}+1$	$+.600_{10}+1$	$+.321_{10}-1$	$+.528_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.110_{10}+2$	$+.400_{10}+1$	$+.600_{10}+1$	$+.301_{10}-1$	$+.507_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.120_{10}+2$	$+.400_{10}+1$	$+.600_{10}+1$	$+.283_{10}-1$	$+.487_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.130_{10}+2$	$+.400_{10}+1$	$+.600_{10}+1$	$+.266_{10}-1$	$+.468_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.140_{10}+2$	$+.400_{10}+1$	$+.600_{10}+1$	$+.249_{10}-1$	$+.450_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.150_{10}+2$	$+.400_{10}+1$	$+.600_{10}+1$	$+.233_{10}-1$	$+.432_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.160_{10}+2$	$+.400_{10}+1$	$+.600_{10}+1$	$+.218_{10}-1$	$+.415_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.170_{10}+2$	$+.400_{10}+1$	$+.600_{10}+1$	$+.204_{10}-1$	$+.399_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.180_{10}+2$	$+.400_{10}+1$	$+.600_{10}+1$	$+.191_{10}-1$	$+.383_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.190_{10}+2$	$+.400_{10}+1$	$+.600_{10}+1$	$+.178_{10}-1$	$+.368_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.200_{10}+2$	$+.400_{10}+1$	$+.600_{10}+1$	$+.167_{10}-1$	$+.354_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.210_{10}+2$	$+.400_{10}+1$	$+.600_{10}+1$	$+.155_{10}-1$	$+.340_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.220_{10}+2$	$+.400_{10}+1$	$+.600_{10}+1$	$+.145_{10}-1$	$+.327_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.230_{10}+2$	$+.400_{10}+1$	$+.600_{10}+1$	$+.135_{10}-1$	$+.314_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.240_{10}+2$	$+.400_{10}+1$	$+.600_{10}+1$	$+.126_{10}-1$	$+.302_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.250_{10}+2$	$+.400_{10}+1$	$+.600_{10}+1$	$+.117_{10}-1$	$+.290_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.260_{10}+2$	$+.400_{10}+1$	$+.600_{10}+1$	$+.109_{10}-1$	$+.279_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.270_{10}+2$	$+.400_{10}+1$	$+.600_{10}+1$	$+.102_{10}-1$	$+.268_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.280_{10}+2$	$+.400_{10}+1$	$+.600_{10}+1$	$+.944_{10}-2$	$+.257_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.290_{10}+2$	$+.400_{10}+1$	$+.600_{10}+1$	$+.878_{10}-2$	$+.247_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$
$+.300_{10}+2$	$+.400_{10}+1$	$+.600_{10}+1$	$+.816_{10}-2$	$+.237_{10}+0$	$+.973_{10}+0$	$+.100_{10}+1$

$p = p_2 = +.995_{10}+0$

VERBETERING

n	m	N	pred	pc		
$+.900_{10}+1$	$+.100_{10}+1$	$+.100_{10}+1$	$+.954_{10}+0$	$+.951_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$

VERBETERING

VERBETERING

VERBETERING

$+.270_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.593_{10}+0\ +.572_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.280_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.567_{10}+0\ +.560_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.290_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.551_{10}+0\ +.549_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.300_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.536_{10}+0\ +.538_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.310_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.520_{10}+0\ +.528_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.320_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.505_{10}+0\ +.517_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.330_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.491_{10}+0\ +.507_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.340_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.476_{10}+0\ +.497_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.350_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.462_{10}+0\ +.487_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.360_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.448_{10}+0\ +.478_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.370_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.435_{10}+0\ +.468_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.380_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.421_{10}+0\ +.459_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.390_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.408_{10}+0\ +.450_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.400_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.396_{10}+0\ +.441_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.410_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.383_{10}+0\ +.432_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.420_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.371_{10}+0\ +.424_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.430_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.360_{10}+0\ +.415_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.440_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.348_{10}+0\ +.407_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.450_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.337_{10}+0\ +.399_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.460_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.326_{10}+0\ +.391_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.470_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.315_{10}+0\ +.383_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.480_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.305_{10}+0\ +.376_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.490_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.295_{10}+0\ +.368_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.500_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.285_{10}+0\ +.361_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.510_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.276_{10}+0\ +.354_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.520_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.267_{10}+0\ +.347_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.530_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.258_{10}+0\ +.340_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.540_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.249_{10}+0\ +.333_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.550_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.241_{10}+0\ +.327_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$+.560_{10}+2\ +.400_{10}+1\ +.100_{10}+1\ +.233_{10}+0\ +.320_{10}+0\ +.999_{10}+0\ +.100_{10}+1$

$p = p_2$

	n	m	N	$Prod$	Pc		
VERBETERING	$+.360_{10}+2$	$+.300_{10}+1$	$+.100_{10}+1$	$+.587_{10}+0$	$+.575_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.370_{10}+2$	$+.300_{10}+1$	$+.100_{10}+1$	$+.575_{10}+0$	$+.566_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.380_{10}+2$	$+.300_{10}+1$	$+.100_{10}+1$	$+.563_{10}+0$	$+.558_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.390_{10}+2$	$+.300_{10}+1$	$+.100_{10}+1$	$+.551_{10}+0$	$+.549_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
	$+.100_{10}+1$	$+.400_{10}+1$	$+.100_{10}+1$	$+.957_{10}+0$	$+.961_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.200_{10}+1$	$+.400_{10}+1$	$+.100_{10}+1$	$+.951_{10}+0$	$+.942_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.300_{10}+1$	$+.400_{10}+1$	$+.100_{10}+1$	$+.945_{10}+0$	$+.923_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.400_{10}+1$	$+.400_{10}+1$	$+.100_{10}+1$	$+.936_{10}+0$	$+.905_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.500_{10}+1$	$+.400_{10}+1$	$+.100_{10}+1$	$+.927_{10}+0$	$+.887_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.600_{10}+1$	$+.400_{10}+1$	$+.100_{10}+1$	$+.916_{10}+0$	$+.870_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.700_{10}+1$	$+.400_{10}+1$	$+.100_{10}+1$	$+.904_{10}+0$	$+.852_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.800_{10}+1$	$+.400_{10}+1$	$+.100_{10}+1$	$+.891_{10}+0$	$+.835_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.900_{10}+1$	$+.400_{10}+1$	$+.100_{10}+1$	$+.878_{10}+0$	$+.819_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.100_{10}+2$	$+.400_{10}+1$	$+.100_{10}+1$	$+.863_{10}+0$	$+.803_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.110_{10}+2$	$+.400_{10}+1$	$+.100_{10}+1$	$+.849_{10}+0$	$+.787_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.120_{10}+2$	$+.400_{10}+1$	$+.100_{10}+1$	$+.833_{10}+0$	$+.771_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.130_{10}+2$	$+.400_{10}+1$	$+.100_{10}+1$	$+.817_{10}+0$	$+.756_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.140_{10}+2$	$+.400_{10}+1$	$+.100_{10}+1$	$+.801_{10}+0$	$+.741_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.150_{10}+2$	$+.400_{10}+1$	$+.100_{10}+1$	$+.785_{10}+0$	$+.726_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.160_{10}+2$	$+.400_{10}+1$	$+.100_{10}+1$	$+.768_{10}+0$	$+.712_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.170_{10}+2$	$+.400_{10}+1$	$+.100_{10}+1$	$+.751_{10}+0$	$+.698_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.180_{10}+2$	$+.400_{10}+1$	$+.100_{10}+1$	$+.734_{10}+0$	$+.684_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.190_{10}+2$	$+.400_{10}+1$	$+.100_{10}+1$	$+.717_{10}+0$	$+.671_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.200_{10}+2$	$+.400_{10}+1$	$+.100_{10}+1$	$+.700_{10}+0$	$+.657_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.210_{10}+2$	$+.400_{10}+1$	$+.100_{10}+1$	$+.683_{10}+0$	$+.644_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.220_{10}+2$	$+.400_{10}+1$	$+.100_{10}+1$	$+.666_{10}+0$	$+.632_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.230_{10}+2$	$+.400_{10}+1$	$+.100_{10}+1$	$+.649_{10}+0$	$+.619_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.240_{10}+2$	$+.400_{10}+1$	$+.100_{10}+1$	$+.632_{10}+0$	$+.607_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.250_{10}+2$	$+.400_{10}+1$	$+.100_{10}+1$	$+.616_{10}+0$	$+.595_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.260_{10}+2$	$+.400_{10}+1$	$+.100_{10}+1$	$+.599_{10}+0$	$+.583_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$

VERBETERING
+.600$_{10}$+ 1 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.934$_{10}$+ 0 +.900$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.700$_{10}$+ 1 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.927$_{10}$+ 0 +.887$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.800$_{10}$+ 1 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.919$_{10}$+ 0 +.874$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.900$_{10}$+ 1 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.910$_{10}$+ 0 +.861$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.100$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.901$_{10}$+ 0 +.848$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.110$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.891$_{10}$+ 0 +.835$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.120$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.881$_{10}$+ 0 +.823$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.130$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.871$_{10}$+ 0 +.811$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.140$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.860$_{10}$+ 0 +.799$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.150$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.849$_{10}$+ 0 +.787$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.160$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.837$_{10}$+ 0 +.775$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.170$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.825$_{10}$+ 0 +.764$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.180$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.813$_{10}$+ 0 +.752$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.190$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.801$_{10}$+ 0 +.741$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.200$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.789$_{10}$+ 0 +.730$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.210$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.776$_{10}$+ 0 +.719$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.220$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.764$_{10}$+ 0 +.709$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.230$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.751$_{10}$+ 0 +.698$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.240$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.738$_{10}$+ 0 +.688$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.250$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.725$_{10}$+ 0 +.677$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.260$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.713$_{10}$+ 0 +.667$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.270$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.700$_{10}$+ 0 +.657$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.280$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.687$_{10}$+ 0 +.648$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.290$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.674$_{10}$+ 0 +.638$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.300$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.662$_{10}$+ 0 +.629$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.310$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.649$_{10}$+ 0 +.619$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.320$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.636$_{10}$+ 0 +.610$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.330$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.624$_{10}$+ 0 +.601$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.340$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.612$_{10}$+ 0 +.592$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1
VERBETERING
+.350$_{10}$+ 2 +.300$_{10}$+ 1 +.100$_{10}$+ 1 +.599$_{10}$+ 0 +.583$_{10}$+ 0 +.999$_{10}$+ 0 +.100$_{10}$+ 1

$\rho = \rho_2$

	ρ	m	N	ρ_{med}	ρ_c		
VERBETERING	$+.330_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.768_{10}+0$	$+.712_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.340_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.759_{10}+0$	$+.705_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.350_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.751_{10}+0$	$+.698_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.360_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.742_{10}+0$	$+.691_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.370_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.734_{10}+0$	$+.684_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.380_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.725_{10}+0$	$+.677_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.390_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.717_{10}+0$	$+.671_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.400_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.708_{10}+0$	$+.664_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.410_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.700_{10}+0$	$+.657_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.420_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.691_{10}+0$	$+.651_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.430_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.683_{10}+0$	$+.644_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.440_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.674_{10}+0$	$+.638_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.450_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.666_{10}+0$	$+.632_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.460_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.657_{10}+0$	$+.625_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.470_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.649_{10}+0$	$+.619_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.480_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.641_{10}+0$	$+.613_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.490_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.632_{10}+0$	$+.607_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.500_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.624_{10}+0$	$+.601_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.510_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.616_{10}+0$	$+.595_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.520_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.607_{10}+0$	$+.589_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.530_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.599_{10}+0$	$+.583_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.540_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.591_{10}+0$	$+.577_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.550_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.583_{10}+0$	$+.572_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.560_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.575_{10}+0$	$+.566_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.570_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.567_{10}+0$	$+.560_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.580_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.559_{10}+0$	$+.555_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.590_{10}+2$	$+.200_{10}+1$	$+.100_{10}+1$	$+.551_{10}+0$	$+.549_{10}+0$	$+.100_{10}+1$	$+.100_{10}+1$
VERBETERING	$+.300_{10}+1$	$+.300_{10}+1$	$+.100_{10}+1$	$+.951_{10}+0$	$+.942_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.400_{10}+1$	$+.300_{10}+1$	$+.100_{10}+1$	$+.947_{10}+0$	$+.928_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$
VERBETERING	$+.500_{10}+1$	$+.300_{10}+1$	$+.100_{10}+1$	$+.941_{10}+0$	$+.914_{10}+0$	$+.999_{10}+0$	$+.100_{10}+1$

(XXXXXXXXXXXXX

29 - 20 - 80672 - 06185479 STELTENPOO

VERBETERING +.500$_{10}$+ 1 +.900$_{10}$+ 1 +.100$_{10}$+ 1 +.825$_{10}$+ 0 +.764$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

VERBETERING +.600$_{10}$+ 1 +.900$_{10}$+ 1 +.100$_{10}$+ 1 +.789$_{10}$+ 0 +.730$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

VERBETERING +.700$_{10}$+ 1 +.900$_{10}$+ 1 +.100$_{10}$+ 1 +.751$_{10}$+ 0 +.698$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

VERBETERING +.800$_{10}$+ 1 +.900$_{10}$+ 1 +.100$_{10}$+ 1 +.713$_{10}$+ 0 +.667$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

VERBETERING +.900$_{10}$+ 1 +.900$_{10}$+ 1 +.100$_{10}$+ 1 +.674$_{10}$+ 0 +.638$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

VERBETERING +.100$_{10}$+ 2 +.900$_{10}$+ 1 +.100$_{10}$+ 1 +.636$_{10}$+ 0 +.610$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

VERBETERING +.110$_{10}$+ 2 +.900$_{10}$+ 1 +.100$_{10}$+ 1 +.599$_{10}$+ 0 +.583$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

VERBETERING +.120$_{10}$+ 2 +.900$_{10}$+ 1 +.100$_{10}$+ 1 +.563$_{10}$+ 0 +.558$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

VERBETERING +.100$_{10}$+ 1 +.100$_{10}$+ 2 +.100$_{10}$+ 1 +.936$_{10}$+ 0 +.905$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

VERBETERING +.200$_{10}$+ 1 +.100$_{10}$+ 2 +.100$_{10}$+ 1 +.910$_{10}$+ 0 +.861$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

VERBETERING +.300$_{10}$+ 1 +.100$_{10}$+ 2 +.100$_{10}$+ 1 +.878$_{10}$+ 0 +.819$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

VERBETERING +.400$_{10}$+ 1 +.100$_{10}$+ 2 +.100$_{10}$+ 1 +.841$_{10}$+ 0 +.779$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

VERBETERING +.500$_{10}$+ 1 +.100$_{10}$+ 2 +.100$_{10}$+ 1 +.801$_{10}$+ 0 +.741$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

VERBETERING +.600$_{10}$+ 1 +.100$_{10}$+ 2 +.100$_{10}$+ 1 +.759$_{10}$+ 0 +.705$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

VERBETERING +.700$_{10}$+ 1 +.100$_{10}$+ 2 +.100$_{10}$+ 1 +.717$_{10}$+ 0 +.671$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

VERBETERING +.800$_{10}$+ 1 +.100$_{10}$+ 2 +.100$_{10}$+ 1 +.674$_{10}$+ 0 +.638$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

VERBETERING +.900$_{10}$+ 1 +.100$_{10}$+ 2 +.100$_{10}$+ 1 +.632$_{10}$+ 0 +.607$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

VERBETERING +.100$_{10}$+ 2 +.100$_{10}$+ 2 +.100$_{10}$+ 1 +.591$_{10}$+ 0 +.577$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

VERBETERING +.110$_{10}$+ 2 +.100$_{10}$+ 2 +.100$_{10}$+ 1 +.551$_{10}$+ 0 +.549$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.100$_{10}$+ 1 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.846$_{10}$+ 0 +.942$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.200$_{10}$+ 1 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.840$_{10}$+ 0 +.923$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.300$_{10}$+ 1 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.832$_{10}$+ 0 +.905$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.400$_{10}$+ 1 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.824$_{10}$+ 0 +.887$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.500$_{10}$+ 1 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.814$_{10}$+ 0 +.870$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.600$_{10}$+ 1 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.803$_{10}$+ 0 +.852$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.700$_{10}$+ 1 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.792$_{10}$+ 0 +.835$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.800$_{10}$+ 1 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.780$_{10}$+ 0 +.819$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.900$_{10}$+ 1 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.767$_{10}$+ 0 +.803$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.100$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.754$_{10}$+ 0 +.787$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.110$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.740$_{10}$+ 0 +.771$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

	P	M	N	Pred	Pc		
VERBETERING	+.500$_{10}$+ 1	+.700$_{10}$+ 1	+.100$_{10}$+ 1	+.871$_{10}$+ 0	+.811$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.600$_{10}$+ 1	+.700$_{10}$+ 1	+.100$_{10}$+ 1	+.845$_{10}$+ 0	+.783$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.700$_{10}$+ 1	+.700$_{10}$+ 1	+.100$_{10}$+ 1	+.817$_{10}$+ 0	+.756$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.800$_{10}$+ 1	+.700$_{10}$+ 1	+.100$_{10}$+ 1	+.789$_{10}$+ 0	+.730$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.900$_{10}$+ 1	+.700$_{10}$+ 1	+.100$_{10}$+ 1	+.759$_{10}$+ 0	+.705$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.100$_{10}$+ 2	+.700$_{10}$+ 1	+.100$_{10}$+ 1	+.730$_{10}$+ 0	+.681$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.110$_{10}$+ 2	+.700$_{10}$+ 1	+.100$_{10}$+ 1	+.700$_{10}$+ 0	+.657$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.120$_{10}$+ 2	+.700$_{10}$+ 1	+.100$_{10}$+ 1	+.670$_{10}$+ 0	+.635$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.130$_{10}$+ 2	+.700$_{10}$+ 1	+.100$_{10}$+ 1	+.641$_{10}$+ 0	+.613$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.140$_{10}$+ 2	+.700$_{10}$+ 1	+.100$_{10}$+ 1	+.612$_{10}$+ 0	+.592$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.150$_{10}$+ 2	+.700$_{10}$+ 1	+.100$_{10}$+ 1	+.583$_{10}$+ 0	+.572$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.160$_{10}$+ 2	+.700$_{10}$+ 1	+.100$_{10}$+ 1	+.555$_{10}$+ 0	+.552$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.100$_{10}$+ 1	+.800$_{10}$+ 1	+.100$_{10}$+ 1	+.945$_{10}$+ 0	+.923$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.200$_{10}$+ 1	+.800$_{10}$+ 1	+.100$_{10}$+ 1	+.927$_{10}$+ 0	+.887$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.300$_{10}$+ 1	+.800$_{10}$+ 1	+.100$_{10}$+ 1	+.904$_{10}$+ 0	+.852$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.400$_{10}$+ 1	+.800$_{10}$+ 1	+.100$_{10}$+ 1	+.878$_{10}$+ 0	+.819$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.500$_{10}$+ 1	+.800$_{10}$+ 1	+.100$_{10}$+ 1	+.849$_{10}$+ 0	+.787$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.600$_{10}$+ 1	+.800$_{10}$+ 1	+.100$_{10}$+ 1	+.817$_{10}$+ 0	+.756$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.700$_{10}$+ 1	+.800$_{10}$+ 1	+.100$_{10}$+ 1	+.785$_{10}$+ 0	+.726$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.800$_{10}$+ 1	+.800$_{10}$+ 1	+.100$_{10}$+ 1	+.751$_{10}$+ 0	+.698$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.900$_{10}$+ 1	+.800$_{10}$+ 1	+.100$_{10}$+ 1	+.717$_{10}$+ 0	+.671$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.100$_{10}$+ 2	+.800$_{10}$+ 1	+.100$_{10}$+ 1	+.683$_{10}$+ 0	+.644$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.110$_{10}$+ 2	+.800$_{10}$+ 1	+.100$_{10}$+ 1	+.649$_{10}$+ 0	+.619$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.120$_{10}$+ 2	+.800$_{10}$+ 1	+.100$_{10}$+ 1	+.616$_{10}$+ 0	+.595$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.130$_{10}$+ 2	+.800$_{10}$+ 1	+.100$_{10}$+ 1	+.583$_{10}$+ 0	+.572$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.140$_{10}$+ 2	+.800$_{10}$+ 1	+.100$_{10}$+ 1	+.551$_{10}$+ 0	+.549$_{10}$+ 0	+.996$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.100$_{10}$+ 1	+.900$_{10}$+ 1	+.100$_{10}$+ 1	+.941$_{10}$+ 0	+.914$_{10}$+ 0	+.994$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.200$_{10}$+ 1	+.900$_{10}$+ 1	+.100$_{10}$+ 1	+.919$_{10}$+ 0	+.874$_{10}$+ 0	+.994$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.300$_{10}$+ 1	+.900$_{10}$+ 1	+.100$_{10}$+ 1	+.891$_{10}$+ 0	+.835$_{10}$+ 0	+.994$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.400$_{10}$+ 1	+.900$_{10}$+ 1	+.100$_{10}$+ 1	+.860$_{10}$+ 0	+.799$_{10}$+ 0	+.994$_{10}$+ 0	+.100$_{10}$+ 1

	n	m	N	Pred	Pe								
VERBETERING	+.170$_{10}$+	2	+,500$_{10}$+	1	+.100$_{10}$+	1	+.674$_{10}$+	0	+.638$_{10}$+	0	+.998$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.180$_{10}$+	2	+,500$_{10}$+	1	+.100$_{10}$+	1	+.653$_{10}$+	0	+.622$_{10}$+	0	+.998$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.190$_{10}$+	2	+,500$_{10}$+	1	+.100$_{10}$+	1	+.632$_{10}$+	0	+.607$_{10}$+	0	+.998$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.200$_{10}$+	2	+,500$_{10}$+	1	+.100$_{10}$+	1	+.612$_{10}$+	0	+.592$_{10}$+	0	+.998$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.210$_{10}$+	2	+,500$_{10}$+	1	+.100$_{10}$+	1	+.591$_{10}$+	0	+.577$_{10}$+	0	+.998$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.220$_{10}$+	2	+,500$_{10}$+	1	+.100$_{10}$+	1	+.571$_{10}$+	0	+.563$_{10}$+	0	+.998$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.230$_{10}$+	2	+,500$_{10}$+	1	+.100$_{10}$+	1	+.551$_{10}$+	0	+.549$_{10}$+	0	+.998$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.100$_{10}$+	1	+,600$_{10}$+	1	+.100$_{10}$+	1	+.951$_{10}$+	0	+.942$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.200$_{10}$+	1	+,600$_{10}$+	1	+.100$_{10}$+	1	+.941$_{10}$+	0	+.914$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.300$_{10}$+	1	+,600$_{10}$+	1	+.100$_{10}$+	1	+.927$_{10}$+	0	+.887$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.400$_{10}$+	1	+,600$_{10}$+	1	+.100$_{10}$+	1	+.910$_{10}$+	0	+.861$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.500$_{10}$+	1	+,600$_{10}$+	1	+.100$_{10}$+	1	+.891$_{10}$+	0	+.835$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.600$_{10}$+	1	+,600$_{10}$+	1	+.100$_{10}$+	1	+.871$_{10}$+	0	+.811$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.700$_{10}$+	1	+,600$_{10}$+	1	+.100$_{10}$+	1	+.849$_{10}$+	0	+.787$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.800$_{10}$+	1	+,600$_{10}$+	1	+.100$_{10}$+	1	+.825$_{10}$+	0	+.764$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.900$_{10}$+	1	+,600$_{10}$+	1	+.100$_{10}$+	1	+.801$_{10}$+	0	+.741$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.100$_{10}$+	2	+,600$_{10}$+	1	+.100$_{10}$+	1	+.776$_{10}$+	0	+.719$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.110$_{10}$+	2	+,600$_{10}$+	1	+.100$_{10}$+	1	+.751$_{10}$+	0	+.698$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.120$_{10}$+	2	+,600$_{10}$+	1	+.100$_{10}$+	1	+.725$_{10}$+	0	+.677$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.130$_{10}$+	2	+,600$_{10}$+	1	+.100$_{10}$+	1	+.700$_{10}$+	0	+.657$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.140$_{10}$+	2	+,600$_{10}$+	1	+.100$_{10}$+	1	+.674$_{10}$+	0	+.638$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.150$_{10}$+	2	+,600$_{10}$+	1	+.100$_{10}$+	1	+.649$_{10}$+	0	+.619$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.160$_{10}$+	2	+,600$_{10}$+	1	+.100$_{10}$+	1	+.624$_{10}$+	0	+.601$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.170$_{10}$+	2	+,600$_{10}$+	1	+.100$_{10}$+	1	+.599$_{10}$+	0	+.583$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.180$_{10}$+	2	+,600$_{10}$+	1	+.100$_{10}$+	1	+.575$_{10}$+	0	+.566$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.190$_{10}$+	2	+,600$_{10}$+	1	+.100$_{10}$+	1	+.551$_{10}$+	0	+.549$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.100$_{10}$+	1	+,700$_{10}$+	1	+.100$_{10}$+	1	+.948$_{10}$+	0	+.932$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.200$_{10}$+	1	+,700$_{10}$+	1	+.100$_{10}$+	1	+.934$_{10}$+	0	+.900$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.300$_{10}$+	1	+,700$_{10}$+	1	+.100$_{10}$+	1	+.916$_{10}$+	0	+.870$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1
VERBETERING	+.400$_{10}$+	1	+,700$_{10}$+	1	+.100$_{10}$+	1	+.895$_{10}$+	0	+.840$_{10}$+	0	+.997$_{10}$+	0	+.100$_{10}$+ 1

$P=P_2$

	n	m	N	Pred	Pc		
	+.570$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.225$_{10}$+ 0	+.314$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
	+.580$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.217$_{10}$+ 0	+.308$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
	+.590$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.210$_{10}$+ 0	+.302$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
	+.600$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.202$_{10}$+ 0	+.296$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
	+.610$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.195$_{10}$+ 0	+.290$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
	+.620$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.189$_{10}$+ 0	+.284$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
	+.630$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.182$_{10}$+ 0	+.279$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
	+.640$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.176$_{10}$+ 0	+.273$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
	+.650$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.170$_{10}$+ 0	+.268$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
	+.660$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.164$_{10}$+ 0	+.262$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
	+.670$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.158$_{10}$+ 0	+.257$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
	+.680$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.152$_{10}$+ 0	+.252$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
	+.690$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.147$_{10}$+ 0	+.247$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
	+.700$_{10}$+ 2	+.400$_{10}$+ 1	+.100$_{10}$+ 1	+.142$_{10}$+ 0	+.242$_{10}$+ 0	+.999$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.100$_{10}$+ 1	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.954$_{10}$+ 0	+.951$_{10}$+ 0	+.998$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.200$_{10}$+ 1	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.947$_{10}$+ 0	+.928$_{10}$+ 0	+.998$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.300$_{10}$+ 1	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.936$_{10}$+ 0	+.905$_{10}$+ 0	+.998$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.924$_{10}$+ 0	+.883$_{10}$+ 0	+.998$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.500$_{10}$+ 1	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.910$_{10}$+ 0	+.861$_{10}$+ 0	+.998$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.600$_{10}$+ 1	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.895$_{10}$+ 0	+.840$_{10}$+ 0	+.998$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.700$_{10}$+ 1	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.878$_{10}$+ 0	+.819$_{10}$+ 0	+.998$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.800$_{10}$+ 1	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.860$_{10}$+ 0	+.799$_{10}$+ 0	+.998$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.900$_{10}$+ 1	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.841$_{10}$+ 0	+.779$_{10}$+ 0	+.998$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.100$_{10}$+ 2	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.821$_{10}$+ 0	+.760$_{10}$+ 0	+.998$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.110$_{10}$+ 2	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.801$_{10}$+ 0	+.741$_{10}$+ 0	+.998$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.120$_{10}$+ 2	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.780$_{10}$+ 0	+.723$_{10}$+ 0	+.998$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.130$_{10}$+ 2	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.759$_{10}$+ 0	+.705$_{10}$+ 0	+.998$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.140$_{10}$+ 2	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.738$_{10}$+ 0	+.688$_{10}$+ 0	+.998$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.150$_{10}$+ 2	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.717$_{10}$+ 0	+.671$_{10}$+ 0	+.998$_{10}$+ 0	+.100$_{10}$+ 1
VERBETERING	+.160$_{10}$+ 2	+.500$_{10}$+ 1	+.100$_{10}$+ 1	+.696$_{10}$+ 0	+.654$_{10}$+ 0	+.998$_{10}$+ 0	+.100$_{10}$+ 1

+.320₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.347₁₀+ 0 +.497₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.330₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.336₁₀+ 0 +.487₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.340₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.326₁₀+ 0 +.478₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.350₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.316₁₀+ 0 +.468₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.360₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.307₁₀+ 0 +.459₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.370₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.297₁₀+ 0 +.450₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.380₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.288₁₀+ 0 +.441₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.390₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.279₁₀+ 0 +.432₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.400₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.270₁₀+ 0 +.424₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.410₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.262₁₀+ 0 +.415₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.420₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.253₁₀+ 0 +.407₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.430₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.245₁₀+ 0 +.399₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.440₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.237₁₀+ 0 +.391₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.450₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.229₁₀+ 0 +.383₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.460₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.222₁₀+ 0 +.376₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.470₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.215₁₀+ 0 +.368₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.480₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.208₁₀+ 0 +.361₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.490₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.201₁₀+ 0 +.354₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.500₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.194₁₀+ 0 +.347₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.510₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.188₁₀+ 0 +.340₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.520₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.181₁₀+ 0 +.333₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.530₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.175₁₀+ 0 +.327₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.540₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.169₁₀+ 0 +.320₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.550₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.163₁₀+ 0 +.314₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.560₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.158₁₀+ 0 +.308₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.570₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.152₁₀+ 0 +.302₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.580₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.147₁₀+ 0 +.296₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.590₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.142₁₀+ 0 +.290₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.600₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.137₁₀+ 0 +.284₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

+.610₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.132₁₀+ 0 +.279₁₀+ 0 +.997₁₀+ 0 +.100₁₀+ 1

$p = p_2$

29 - 23 - 80672 - 06185479 STELTENPOO

p	m	N	pred	pc		
+.200$_{10}$+ 1	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.684$_{10}$+ 0	+.905$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.300$_{10}$+ 1	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.677$_{10}$+ 0	+.887$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.400$_{10}$+ 1	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.669$_{10}$+ 0	+.870$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.500$_{10}$+ 1	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.660$_{10}$+ 0	+.852$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.600$_{10}$+ 1	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.651$_{10}$+ 0	+.835$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.700$_{10}$+ 1	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.641$_{10}$+ 0	+.819$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.800$_{10}$+ 1	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.630$_{10}$+ 0	+.803$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.900$_{10}$+ 1	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.619$_{10}$+ 0	+.787$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.100$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.608$_{10}$+ 0	+.771$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.110$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.596$_{10}$+ 0	+.756$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.120$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.584$_{10}$+ 0	+.741$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.130$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.572$_{10}$+ 0	+.726$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.140$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.560$_{10}$+ 0	+.712$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.150$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.547$_{10}$+ 0	+.698$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.160$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.535$_{10}$+ 0	+.684$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.170$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.522$_{10}$+ 0	+.671$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.180$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.510$_{10}$+ 0	+.657$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.190$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.497$_{10}$+ 0	+.644$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.200$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.485$_{10}$+ 0	+.632$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.210$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.473$_{10}$+ 0	+.619$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.220$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.460$_{10}$+ 0	+.607$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.230$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.448$_{10}$+ 0	+.595$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.240$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.436$_{10}$+ 0	+.583$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.250$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.425$_{10}$+ 0	+.572$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.260$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.413$_{10}$+ 0	+.560$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.270$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.401$_{10}$+ 0	+.549$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.280$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.390$_{10}$+ 0	+.538$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.290$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.379$_{10}$+ 0	+.528$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.300$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.368$_{10}$+ 0	+.517$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.310$_{10}$+ 2	+,400$_{10}$+ 1	+.300$_{10}$+ 1	+.357$_{10}$+ 0	+.507$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1

+.420$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.319$_{10}$+ 0 +.415$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.430$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.309$_{10}$+ 0 +.407$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.440$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.299$_{10}$+ 0 +.399$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.450$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.289$_{10}$+ 0 +.391$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.460$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.280$_{10}$+ 0 +.383$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.470$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.271$_{10}$+ 0 +.376$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.480$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.262$_{10}$+ 0 +.368$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.490$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.253$_{10}$+ 0 +.361$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.500$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.245$_{10}$+ 0 +.354$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.510$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.237$_{10}$+ 0 +.347$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.520$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.229$_{10}$+ 0 +.340$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.530$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.221$_{10}$+ 0 +.333$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.540$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.214$_{10}$+ 0 +.327$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.550$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.206$_{10}$+ 0 +.320$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.560$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.199$_{10}$+ 0 +.314$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.570$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.193$_{10}$+ 0 +.308$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.580$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.186$_{10}$+ 0 +.302$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.590$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.180$_{10}$+ 0 +.296$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.600$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.173$_{10}$+ 0 +.290$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.610$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.167$_{10}$+ 0 +.284$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.620$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.162$_{10}$+ 0 +.279$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.630$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.156$_{10}$+ 0 +.273$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.640$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.151$_{10}$+ 0 +.268$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.650$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.145$_{10}$+ 0 +.262$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.660$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.140$_{10}$+ 0 +.257$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.670$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.135$_{10}$+ 0 +.252$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.680$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.130$_{10}$+ 0 +.247$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.690$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.126$_{10}$+ 0 +.242$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.700$_{10}$+ 2 +.400$_{10}$+ 1 +.200$_{10}$+ 1 +.121$_{10}$+ 0 +.237$_{10}$+ 0 +.998$_{10}$+ 0 +.100$_{10}$+ 1

+.100$_{10}$+ 1 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.690$_{10}$+ 0 +.923$_{10}$+ 0 +.997$_{10}$+ 0 +.100$_{10}$+ 1

$p = p_2$

n	m	N	pnev	pe		
+.120$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,726$_{10}$+ 0	+.756$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.130$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,711$_{10}$+ 0	+,741$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.140$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,697$_{10}$+ 0	+,726$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.150$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,682$_{10}$+ 0	+,712$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.160$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,667$_{10}$+ 0	+,698$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.170$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,652$_{10}$+ 0	+,684$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.180$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,637$_{10}$+ 0	+,671$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.190$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,621$_{10}$+ 0	+,657$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.200$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,606$_{10}$+ 0	+,644$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.210$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,591$_{10}$+ 0	+,632$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.220$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,576$_{10}$+ 0	+,619$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.230$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,561$_{10}$+ 0	+,607$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.240$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,547$_{10}$+ 0	+,595$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.250$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,532$_{10}$+ 0	+,583$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.260$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,518$_{10}$+ 0	+,572$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.270$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,503$_{10}$+ 0	+,560$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.280$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,489$_{10}$+ 0	+,549$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.290$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,476$_{10}$+ 0	+,538$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.300$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,462$_{10}$+ 0	+,528$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.310$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,449$_{10}$+ 0	+,517$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.320$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,436$_{10}$+ 0	+,507$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.330$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,423$_{10}$+ 0	+,497$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.340$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,410$_{10}$+ 0	+,487$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.350$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,398$_{10}$+ 0	+,478$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.360$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,386$_{10}$+ 0	+,468$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.370$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,374$_{10}$+ 0	+,459$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.380$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,362$_{10}$+ 0	+,450$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.390$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,351$_{10}$+ 0	+,441$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.400$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,340$_{10}$+ 0	+,432$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1
+.410$_{10}$+ 2	+,400$_{10}$+ 1	+.200$_{10}$+ 1	+,330$_{10}$+ 0	+,424$_{10}$+ 0	+,998$_{10}$+ 0	+,100$_{10}$+ 1

+.120$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.298$_{10}$+ 0 +.712$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.130$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.291$_{10}$+ 0 +.698$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.140$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.285$_{10}$+ 0 +.684$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.150$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.278$_{10}$+ 0 +.671$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.160$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.271$_{10}$+ 0 +.657$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.170$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.264$_{10}$+ 0 +.644$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.180$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.258$_{10}$+ 0 +.632$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.190$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.251$_{10}$+ 0 +.619$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.200$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.245$_{10}$+ 0 +.607$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.210$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.238$_{10}$+ 0 +.595$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.220$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.232$_{10}$+ 0 +.583$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.230$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.225$_{10}$+ 0 +.572$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.240$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.219$_{10}$+ 0 +.560$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.250$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.213$_{10}$+ 0 +.549$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.260$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.207$_{10}$+ 0 +.538$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.270$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.201$_{10}$+ 0 +.528$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.280$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.195$_{10}$+ 0 +.517$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.290$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.189$_{10}$+ 0 +.507$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.300$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.184$_{10}$+ 0 +.497$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.310$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.178$_{10}$+ 0 +.487$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.320$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.173$_{10}$+ 0 +.478$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.330$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.168$_{10}$+ 0 +.468$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.340$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.162$_{10}$+ 0 +.459$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.350$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.157$_{10}$+ 0 +.450$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.360$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.153$_{10}$+ 0 +.441$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.370$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.148$_{10}$+ 0 +.432$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.380$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.143$_{10}$+ 0 +.424$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.390$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.139$_{10}$+ 0 +.415$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.400$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.134$_{10}$+ 0 +.407$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

+.410$_{10}$+ 2 +.400$_{10}$+ 1 +.500$_{10}$+ 1 +.130$_{10}$+ 0 +.399$_{10}$+ 0 +.994$_{10}$+ 0 +.100$_{10}$+ 1

$P - P_2$

n	m	N	Pred	PC		
+.520$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.133$_{10}$+ 0	+.327$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.530$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.128$_{10}$+ 0	+.320$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.540$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.124$_{10}$+ 0	+.314$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.550$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.119$_{10}$+ 0	+.308$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.560$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.115$_{10}$+ 0	+.302$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.570$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.111$_{10}$+ 0	+.296$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.580$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.108$_{10}$+ 0	+.290$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.590$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.104$_{10}$+ 0	+.284$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.600$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.100$_{10}$+ 0	+.279$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.610$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.967$_{10}$- 1	+.273$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.620$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.933$_{10}$- 1	+.268$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.630$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.901$_{10}$- 1	+.262$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.640$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.869$_{10}$- 1	+.257$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.650$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.838$_{10}$- 1	+.252$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.660$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.809$_{10}$- 1	+.247$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.670$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.780$_{10}$- 1	+.242$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.680$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.752$_{10}$- 1	+.237$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.690$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.725$_{10}$- 1	+.233$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.700$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.700$_{10}$- 1	+.228$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.100$_{10}$+ 1	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.362$_{10}$+ 0	+.887$_{10}$+ 0	+.994$_{10}$+ 0	+.100$_{10}$+ 1
+.200$_{10}$+ 1	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.358$_{10}$+ 0	+.870$_{10}$+ 0	+.994$_{10}$+ 0	+.100$_{10}$+ 1
+.300$_{10}$+ 1	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.353$_{10}$+ 0	+.852$_{10}$+ 0	+.994$_{10}$+ 0	+.100$_{10}$+ 1
+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.347$_{10}$+ 0	+.835$_{10}$+ 0	+.994$_{10}$+ 0	+.100$_{10}$+ 1
+.500$_{10}$+ 1	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.342$_{10}$+ 0	+.819$_{10}$+ 0	+.994$_{10}$+ 0	+.100$_{10}$+ 1
+.600$_{10}$+ 1	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.336$_{10}$+ 0	+.803$_{10}$+ 0	+.994$_{10}$+ 0	+.100$_{10}$+ 1
+.700$_{10}$+ 1	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.330$_{10}$+ 0	+.787$_{10}$+ 0	+.994$_{10}$+ 0	+.100$_{10}$+ 1
+.800$_{10}$+ 1	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.324$_{10}$+ 0	+.771$_{10}$+ 0	+.994$_{10}$+ 0	+.100$_{10}$+ 1
+.900$_{10}$+ 1	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.318$_{10}$+ 0	+.756$_{10}$+ 0	+.994$_{10}$+ 0	+.100$_{10}$+ 1
+.100$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.311$_{10}$+ 0	+.741$_{10}$+ 0	+.994$_{10}$+ 0	+.100$_{10}$+ 1
+.110$_{10}$+ 2	+.400$_{10}$+ 1	+.500$_{10}$+ 1	+.305$_{10}$+ 0	+.726$_{10}$+ 0	+.994$_{10}$+ 0	+.100$_{10}$+ 1

+.220$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.340$_{10}$+ 0 +.595$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.230$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.331$_{10}$+ 0 +.583$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.240$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.322$_{10}$+ 0 +.572$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.250$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.313$_{10}$+ 0 +.560$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.260$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.304$_{10}$+ 0 +.549$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.270$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.296$_{10}$+ 0 +.538$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.280$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.287$_{10}$+ 0 +.528$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.290$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.279$_{10}$+ 0 +.517$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.300$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.271$_{10}$+ 0 +.507$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.310$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.263$_{10}$+ 0 +.497$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.320$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.255$_{10}$+ 0 +.487$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.330$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.247$_{10}$+ 0 +.478$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.340$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.240$_{10}$+ 0 +.468$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.350$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.232$_{10}$+ 0 +.459$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.360$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.225$_{10}$+ 0 +.450$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.370$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.218$_{10}$+ 0 +.441$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.380$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.211$_{10}$+ 0 +.432$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.390$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.205$_{10}$+ 0 +.424$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.400$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.198$_{10}$+ 0 +.415$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.410$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.192$_{10}$+ 0 +.407$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.420$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.186$_{10}$+ 0 +.399$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.430$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.180$_{10}$+ 0 +.391$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.440$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.174$_{10}$+ 0 +.383$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.450$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.168$_{10}$+ 0 +.376$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.460$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.163$_{10}$+ 0 +.368$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.470$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.157$_{10}$+ 0 +.361$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.480$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.152$_{10}$+ 0 +.354$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.490$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.147$_{10}$+ 0 +.347$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.500$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.142$_{10}$+ 0 +.340$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

+.510$_{10}$+ 2 +.400$_{10}$+ 1 +.400$_{10}$+ 1 +.137$_{10}$+ 0 +.333$_{10}$+ 0 +.995$_{10}$+ 0 +.100$_{10}$+ 1

n	m	N	pmed	pe
+.620$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.123$_{10}$+ 0	+.273$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.630$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.123$_{10}$+ 0	+.268$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.640$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.119$_{10}$+ 0	+.262$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.650$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.115$_{10}$+ 0	+.257$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.660$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.111$_{10}$+ 0	+.252$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.670$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.107$_{10}$+ 0	+.247$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.680$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.103$_{10}$+ 0	+.242$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.690$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.994$_{10}$- 1	+.237$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.700$_{10}$+ 2	+.400$_{10}$+ 1	+.300$_{10}$+ 1	+.959$_{10}$- 1	+.233$_{10}$+ 0	+.997$_{10}$+ 0	+.100$_{10}$+ 1
+.100$_{10}$+ 1	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.520$_{10}$+ 0	+.905$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.200$_{10}$+ 1	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.515$_{10}$+ 0	+.887$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.300$_{10}$+ 1	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.509$_{10}$+ 0	+.870$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.502$_{10}$+ 0	+.852$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.500$_{10}$+ 1	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.494$_{10}$+ 0	+.835$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.600$_{10}$+ 1	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.487$_{10}$+ 0	+.819$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.700$_{10}$+ 1	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.478$_{10}$+ 0	+.803$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.800$_{10}$+ 1	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.470$_{10}$+ 0	+.787$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.900$_{10}$+ 1	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.461$_{10}$+ 0	+.771$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.100$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.452$_{10}$+ 0	+.756$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.110$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.443$_{10}$+ 0	+.741$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.120$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.434$_{10}$+ 0	+.726$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.130$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.425$_{10}$+ 0	+.712$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.140$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.415$_{10}$+ 0	+.698$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.150$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.406$_{10}$+ 0	+.684$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.160$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.396$_{10}$+ 0	+.671$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.170$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.387$_{10}$+ 0	+.657$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.180$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.377$_{10}$+ 0	+.644$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.190$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.368$_{10}$+ 0	+.632$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.200$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.358$_{10}$+ 0	+.619$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1
+.210$_{10}$+ 2	+.400$_{10}$+ 1	+.400$_{10}$+ 1	+.349$_{10}$+ 0	+.607$_{10}$+ 0	+.995$_{10}$+ 0	+.100$_{10}$+ 1

$+.620_{10}+2$ $+.400_{10}+1$ $+.600_{10}+1$ $+.302_{10}-1$ $+.257_{10}+0$ $+.993_{10}+0$ $+.100_{10}+1$

$+.630_{10}+2$ $+.400_{10}+1$ $+.600_{10}+1$ $+.378_{10}-1$ $+.252_{10}+0$ $+.993_{10}+0$ $+.100_{10}+1$

$+.640_{10}+2$ $+.400_{10}+1$ $+.600_{10}+1$ $+.365_{10}-1$ $+.247_{10}+0$ $+.993_{10}+0$ $+.100_{10}+1$

$+.650_{10}+2$ $+.400_{10}+1$ $+.600_{10}+1$ $+.352_{10}-1$ $+.242_{10}+0$ $+.993_{10}+0$ $+.100_{10}+1$

$+.660_{10}+2$ $+.400_{10}+1$ $+.600_{10}+1$ $+.339_{10}-1$ $+.237_{10}+0$ $+.993_{10}+0$ $+.100_{10}+1$

$+.670_{10}+2$ $+.400_{10}+1$ $+.600_{10}+1$ $+.327_{10}-1$ $+.233_{10}+0$ $+.993_{10}+0$ $+.100_{10}+1$

$+.680_{10}+2$ $+.400_{10}+1$ $+.600_{10}+1$ $+.315_{10}-1$ $+.228_{10}+0$ $+.993_{10}+0$ $+.100_{10}+1$

$+.690_{10}+2$ $+.400_{10}+1$ $+.600_{10}+1$ $+.304_{10}-1$ $+.224_{10}+0$ $+.993_{10}+0$ $+.100_{10}+1$

$+.700_{10}+2$ $+.400_{10}+1$ $+.600_{10}+1$ $+.293_{10}-1$ $+.219_{10}+0$ $+.993_{10}+0$ $+.100_{10}+1$

p = p2

p	m	N	pmed	pc						
+.320$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.108$_{10}$+ 0	+.468$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.330$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.105$_{10}$+ 0	+.459$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.340$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.102$_{10}$+ 0	+.450$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.350$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.986$_{10}$- 1	+.441$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.360$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.955$_{10}$- 1	+.432$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.370$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.925$_{10}$- 1	+.424$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.380$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.895$_{10}$- 1	+.415$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.390$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.867$_{10}$- 1	+.407$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.400$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.839$_{10}$- 1	+.399$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.410$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.811$_{10}$- 1	+.391$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.420$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.785$_{10}$- 1	+.383$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.430$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.759$_{10}$- 1	+.376$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.440$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.734$_{10}$- 1	+.368$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.450$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.710$_{10}$- 1	+.361$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.460$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.686$_{10}$- 1	+.354$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.470$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.663$_{10}$- 1	+.347$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.480$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.641$_{10}$- 1	+.340$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.490$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.619$_{10}$- 1	+.333$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.500$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.598$_{10}$- 1	+.327$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.510$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.578$_{10}$- 1	+.320$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.520$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.558$_{10}$- 1	+.314$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.530$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.539$_{10}$- 1	+.308$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.540$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.521$_{10}$- 1	+.302$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.550$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.503$_{10}$- 1	+.296$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.560$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.485$_{10}$- 1	+.290$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.570$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.468$_{10}$- 1	+.284$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.580$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.452$_{10}$- 1	+.279$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.590$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.436$_{10}$- 1	+.273$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1
+.600$_{10}$+ 2	+.400$_{10}$+ 1	+.600$_{10}$+ 1	+.421$_{10}$- 1	+.268$_{10}$+ 0	+.993$_{10}$+ 0	+.100$_{10}$+ 1

+.200$_{10}$+ 1 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.229$_{10}$+ 0 +.852$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.300$_{10}$+ 1 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.226$_{10}$+ 0 +.835$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.400$_{10}$+ 1 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.222$_{10}$+ 0 +.819$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.500$_{10}$+ 1 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.218$_{10}$+ 0 +.803$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.600$_{10}$+ 1 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.214$_{10}$+ 0 +.787$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.700$_{10}$+ 1 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.210$_{10}$+ 0 +.771$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.800$_{10}$+ 1 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.206$_{10}$+ 0 +.756$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.900$_{10}$+ 1 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.202$_{10}$+ 0 +.741$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.100$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.198$_{10}$+ 0 +.726$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.110$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.193$_{10}$+ 0 +.712$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.120$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.189$_{10}$+ 0 +.698$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.130$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.184$_{10}$+ 0 +.684$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.140$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.180$_{10}$+ 0 +.671$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.150$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.176$_{10}$+ 0 +.657$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.160$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.171$_{10}$+ 0 +.644$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.170$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.167$_{10}$+ 0 +.632$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.180$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.163$_{10}$+ 0 +.619$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.190$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.158$_{10}$+ 0 +.607$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.200$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.154$_{10}$+ 0 +.595$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.210$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.150$_{10}$+ 0 +.583$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.220$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.146$_{10}$+ 0 +.572$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.230$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.142$_{10}$+ 0 +.560$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.240$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.138$_{10}$+ 0 +.549$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.250$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.134$_{10}$+ 0 +.538$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.260$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.130$_{10}$+ 0 +.528$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.270$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.126$_{10}$+ 0 +.517$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.280$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.122$_{10}$+ 0 +.507$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.290$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.119$_{10}$+ 0 +.497$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.300$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.115$_{10}$+ 0 +.487$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

+.310$_{10}$+ 2 +.400$_{10}$+ 1 +.600$_{10}$+ 1 +.112$_{10}$+ 0 +.478$_{10}$+ 0 +.993$_{10}$+ 0 +.100$_{10}$+ 1

$p = p_2$

n	m2	N	pred	pe			
+.420+ 2	+.400+ 1	+.500+ 1	+.126+ 0	+.391+ 0	+.994+ 0	+.100+ 1
+.430+ 2	+.400+ 1	+.500+ 1	+.121+ 0	+.383+ 0	+.994+ 0	+.100+ 1
+.440+ 2	+.400+ 1	+.500+ 1	+.118+ 0	+.376+ 0	+.994+ 0	+.100+ 1
+.450+ 2	+.400+ 1	+.500+ 1	+.114+ 0	+.368+ 0	+.994+ 0	+.100+ 1
+.460+ 2	+.400+ 1	+.500+ 1	+.110+ 0	+.361+ 0	+.994+ 0	+.100+ 1
+.470+ 2	+.400+ 1	+.500+ 1	+.106+ 0	+.354+ 0	+.994+ 0	+.100+ 1
+.480+ 2	+.400+ 1	+.500+ 1	+.103+ 0	+.347+ 0	+.994+ 0	+.100+ 1
+.490+ 2	+.400+ 1	+.500+ 1	+.992- 1	+.340+ 0	+.994+ 0	+.100+ 1
+.500+ 2	+.400+ 1	+.500+ 1	+.959- 1	+.333+ 0	+.994+ 0	+.100+ 1
+.510+ 2	+.400+ 1	+.500+ 1	+.927- 1	+.327+ 0	+.994+ 0	+.100+ 1
+.520+ 2	+.400+ 1	+.500+ 1	+.895- 1	+.320+ 0	+.994+ 0	+.100+ 1
+.530+ 2	+.400+ 1	+.500+ 1	+.865- 1	+.314+ 0	+.994+ 0	+.100+ 1
+.540+ 2	+.400+ 1	+.500+ 1	+.835- 1	+.308+ 0	+.994+ 0	+.100+ 1
+.550+ 2	+.400+ 1	+.500+ 1	+.806- 1	+.302+ 0	+.994+ 0	+.100+ 1
+.560+ 2	+.400+ 1	+.500+ 1	+.779- 1	+.296+ 0	+.994+ 0	+.100+ 1
+.570+ 2	+.400+ 1	+.500+ 1	+.752- 1	+.290+ 0	+.994+ 0	+.100+ 1
+.580+ 2	+.400+ 1	+.500+ 1	+.726- 1	+.284+ 0	+.994+ 0	+.100+ 1
+.590+ 2	+.400+ 1	+.500+ 1	+.700- 1	+.279+ 0	+.994+ 0	+.100+ 1
+.600+ 2	+.400+ 1	+.500+ 1	+.676- 1	+.273+ 0	+.994+ 0	+.100+ 1
+.610+ 2	+.400+ 1	+.500+ 1	+.652- 1	+.268+ 0	+.994+ 0	+.100+ 1
+.620+ 2	+.400+ 1	+.500+ 1	+.629- 1	+.262+ 0	+.994+ 0	+.100+ 1
+.630+ 2	+.400+ 1	+.500+ 1	+.607- 1	+.257+ 0	+.994+ 0	+.100+ 1
+.640+ 2	+.400+ 1	+.500+ 1	+.586- 1	+.252+ 0	+.994+ 0	+.100+ 1
+.650+ 2	+.400+ 1	+.500+ 1	+.565- 1	+.247+ 0	+.994+ 0	+.100+ 1
+.660+ 2	+.400+ 1	+.500+ 1	+.545- 1	+.242+ 0	+.994+ 0	+.100+ 1
+.670+ 2	+.400+ 1	+.500+ 1	+.526- 1	+.237+ 0	+.994+ 0	+.100+ 1
+.680+ 2	+.400+ 1	+.500+ 1	+.507- 1	+.233+ 0	+.994+ 0	+.100+ 1
+.690+ 2	+.400+ 1	+.500+ 1	+.489- 1	+.228+ 0	+.994+ 0	+.100+ 1
+.700+ 2	+.400+ 1	+.500+ 1	+.471- 1	+.224+ 0	+.994+ 0	+.100+ 1
+.100+ 1	+.400+ 1	+.600+ 1	+.233+ 0	+.870+ 0	+.993+ 0	+.100+ 1

$P = P_1$

PVERBETERING	n	m	K	p_{red}	p_c
	$+.600_{10}+1$	$+.100_{10}+1$	$+.200_{10}+1$	$+.942_{10}-0$	$+.942_{10}-0$
VERBETERING	$+.700_{10}+1$	$+.100_{10}+1$	$+.200_{10}+1$	$+.939_{10}-0$	$+.932_{10}-0$
VERBETERING	$+.800_{10}+1$	$+.100_{10}+1$	$+.200_{10}+1$	$+.935_{10}-0$	$+.923_{10}-0$
VERBETERING	$+.900_{10}+1$	$+.100_{10}+1$	$+.200_{10}+1$	$+.931_{10}-0$	$+.914_{10}-0$
VERBETERING	$+.100_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.927_{10}-0$	$+.905_{10}-0$
VERBETERING	$+.110_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.923_{10}-0$	$+.896_{10}-0$
VERBETERING	$+.120_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.918_{10}-0$	$+.887_{10}-0$
VERBETERING	$+.130_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.912_{10}-0$	$+.878_{10}-0$
VERBETERING	$+.140_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.907_{10}-0$	$+.870_{10}-0$
VERBETERING	$+.150_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.901_{10}-0$	$+.861_{10}-0$
VERBETERING	$+.160_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.895_{10}-0$	$+.852_{10}-0$
VERBETERING	$+.170_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.889_{10}-0$	$+.844_{10}-0$
VERBETERING	$+.180_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.883_{10}-0$	$+.835_{10}-0$
VERBETERING	$+.190_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.876_{10}-0$	$+.827_{10}-0$
VERBETERING	$+.200_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.869_{10}-0$	$+.819_{10}-0$
VERBETERING	$+.210_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.862_{10}-0$	$+.811_{10}-0$
VERBETERING	$+.220_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.855_{10}-0$	$+.803_{10}-0$
VERBETERING	$+.230_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.848_{10}-0$	$+.795_{10}-0$
VERBETERING	$+.240_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.840_{10}-0$	$+.787_{10}-0$
	$+.250_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.833_{10}-0$	$+.779_{10}-0$

```
1   BEGIN COMMENT BEREKENING OPBRENGST VERBETERING VAN KOLLEKTOR NETWERK DOOR TOEPASSING
2         VAN HAMMING CODE;
3         REAL P, PI, PDEK, PRED, PC;
4         INTEGER N, M, K, I, T;
5         I := 0;
6   B:    T := READ; P := EXP(LN (0.05)/T); NEW PAGE;
7         PRINTTEXT(*P*); M := 3; K := 1;
8   A:    N := 1; M := M+1;
9         IF M > (2 ↑ K - K - 1) THEN K := K+1; PI := P;
10        PDEK := P ↑ (2 * M * K + (M ↑ 2));
11  C:    N := N+1; PI := P * PI; PC := PI ↑ M;
12        PRED := ((M+K) * (1-PI) * (PI ↑ (M + K - 1)) + PI ↑ (M+K)) * PDEK;
13        IF (PRED > PC ) ∨ (M = 4) ∨ (M = 3) THEN
14        BEGIN IF PRED > PC THEN
15              BEGIN PRINTTEXT(*VERBETERING*);
16                    NLCR; NLCR;
17              END;
18              SPACE(15); FLOT(3, 2, N); FLOT(3, 2, M);
19              FLOT(3, 2, K); FLOT(3, 2, PRED); FLOT(3, 2, PC);
20              NLCR;
21        END;
22        IF PI > 0.5 THEN GOTO C;
23        IF M < 57 THEN GOTO A;
24        I := I+1; IF I = 1 THEN GOTO B;
25  END
```

VERTAALTIJD 9
EXECUTIETIJD 39

AANTAL OVER STROOM 1 GELEZEN GETALLEN 8
LAATST OVER STROOM 1 GELEZEN GETAL +600

AANTAL OVER STROOM 2 GELEZEN GETALLEN 0

+.140$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.517$_{10}$	0	+.572$_{10}$ 0
+.150$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.501$_{10}$	0	+.549$_{10}$ 0
+.160$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.485$_{10}$	0	+.528$_{10}$ 0
+.170$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.469$_{10}$	0	+.507$_{10}$ 0
+.180$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.453$_{10}$	0	+.487$_{10}$ 0
+.190$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.438$_{10}$	0	+.468$_{10}$ 0
+.200$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.422$_{10}$	0	+.450$_{10}$ 0
+.210$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.407$_{10}$	0	+.432$_{10}$ 0
+.220$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.392$_{10}$	0	+.415$_{10}$ 0
+.230$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.377$_{10}$	0	+.399$_{10}$ 0
+.240$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.363$_{10}$	0	+.383$_{10}$ 0
+.250$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.349$_{10}$	0	+.368$_{10}$ 0
+.260$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.335$_{10}$	0	+.354$_{10}$ 0
+.270$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.322$_{10}$	0	+.340$_{10}$ 0
+.280$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.309$_{10}$	0	+.327$_{10}$ 0
+.290$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.296$_{10}$	0	+.314$_{10}$ 0
+.300$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.284$_{10}$	0	+.302$_{10}$ 0
+.310$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.272$_{10}$	0	+.290$_{10}$ 0
+.320$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.260$_{10}$	0	+.279$_{10}$ 0
+.330$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.249$_{10}$	0	+.268$_{10}$ 0
+.340$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.239$_{10}$	0	+.257$_{10}$ 0
+.350$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.228$_{10}$	0	+.247$_{10}$ 0
+.360$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.218$_{10}$	0	+.237$_{10}$ 0
+.370$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.209$_{10}$	0	+.228$_{10}$ 0
+.380$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.199$_{10}$	0	+.219$_{10}$ 0
+.390$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.190$_{10}$	0	+.211$_{10}$ 0
+.400$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.182$_{10}$	0	+.202$_{10}$ 0
+.410$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.173$_{10}$	0	+.194$_{10}$ 0
+.420$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.165$_{10}$	0	+.187$_{10}$ 0
+.430$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.158$_{10}$	0	+.180$_{10}$ 0
+.440$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.151$_{10}$	0	+.172$_{10}$ 0
+.450$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.144$_{10}$	0	+.166$_{10}$ 0
+.460$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.137$_{10}$	0	+.159$_{10}$ 0
+.470$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.130$_{10}$	0	+.153$_{10}$ 0
+.480$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.124$_{10}$	0	+.147$_{10}$ 0
+.490$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.118$_{10}$	0	+.141$_{10}$ 0
+.500$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.113$_{10}$	0	+.136$_{10}$ 0
+.510$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.107$_{10}$	0	+.130$_{10}$ 0
+.520$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.102$_{10}$	0	+.125$_{10}$ 0
+.530$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.971$_{10}$	1	+.120$_{10}$ 0
+.540$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.924$_{10}$	1	+.116$_{10}$ 0
+.550$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.879$_{10}$	1	+.111$_{10}$ 0
+.560$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.836$_{10}$	1	+.107$_{10}$ 0
+.570$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.795$_{10}$	1	+.103$_{10}$ 0
+.580$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.755$_{10}$	1	+.986$_{10}$ 1
+.590$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.718$_{10}$	1	+.947$_{10}$ 1
+.600$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.682$_{10}$	1	+.910$_{10}$ 1
+.610$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.648$_{10}$	1	+.875$_{10}$ 1
+.620$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.616$_{10}$	1	+.840$_{10}$ 1
+.630$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.585$_{10}$	1	+.807$_{10}$ 1
+.640$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.556$_{10}$	1	+.776$_{10}$ 1
+.650$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.528$_{10}$	1	+.745$_{10}$ 1
+.660$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.501$_{10}$	1	+.716$_{10}$ 1
+.670$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.476$_{10}$	1	+.688$_{10}$ 1
+.680$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.451$_{10}$	1	+.661$_{10}$ 1
+.690$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.428$_{10}$	1	+.635$_{10}$ 1
+.700$_{10}$	2	+.400$_{10}$	1	+.300$_{10}$	1	+.406$_{10}$	1	+.611$_{10}$ 1

p=p4

n	m	k	prod	pc
+.230₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.491₁₀- 0	+.502₁₀- 0
+.240₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.476₁₀- 0	+.487₁₀- 0
+.250₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.461₁₀- 0	+.473₁₀- 0
+.260₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.447₁₀- 0	+.459₁₀- 0
+.270₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.433₁₀- 0	+.445₁₀- 0
+.280₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.419₁₀- 0	+.432₁₀- 0
+.290₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.405₁₀- 0	+.419₁₀- 0
+.300₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.392₁₀- 0	+.407₁₀- 0
+.310₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.379₁₀- 0	+.395₁₀- 0
+.320₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.366₁₀- 0	+.383₁₀- 0
+.330₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.353₁₀- 0	+.372₁₀- 0
+.340₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.341₁₀- 0	+.361₁₀- 0
+.350₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.329₁₀- 0	+.350₁₀- 0
+.360₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.318₁₀- 0	+.340₁₀- 0
+.370₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.306₁₀- 0	+.330₁₀- 0
+.380₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.295₁₀- 0	+.320₁₀- 0
+.390₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.285₁₀- 0	+.311₁₀- 0
+.400₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.274₁₀- 0	+.302₁₀- 0
+.410₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.264₁₀- 0	+.293₁₀- 0
+.420₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.254₁₀- 0	+.284₁₀- 0
+.430₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.245₁₀- 0	+.276₁₀- 0
+.440₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.236₁₀- 0	+.268₁₀- 0
+.450₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.227₁₀- 0	+.260₁₀- 0
+.460₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.218₁₀- 0	+.252₁₀- 0
+.470₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.210₁₀- 0	+.245₁₀- 0
+.480₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.202₁₀- 0	+.237₁₀- 0
+.490₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.194₁₀- 0	+.230₁₀- 0
+.500₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.187₁₀- 0	+.224₁₀- 0
+.510₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.179₁₀- 0	+.217₁₀- 0
+.520₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.172₁₀- 0	+.211₁₀- 0
+.530₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.165₁₀- 0	+.204₁₀- 0
+.540₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.159₁₀- 0	+.198₁₀- 0
+.550₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.153₁₀- 0	+.193₁₀- 0
+.560₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.146₁₀- 0	+.187₁₀- 0
+.570₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.141₁₀- 0	+.181₁₀- 0
+.580₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.135₁₀- 0	+.176₁₀- 0
+.590₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.129₁₀- 0	+.171₁₀- 0
+.600₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.124₁₀- 0	+.166₁₀- 0
+.610₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.119₁₀- 0	+.161₁₀- 0
+.620₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.114₁₀- 0	+.156₁₀- 0
+.630₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.110₁₀- 0	+.151₁₀- 0
+.640₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.105₁₀- 0	+.147₁₀- 0
+.650₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.101₁₀- 0	+.143₁₀- 0
+.660₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.966₁₀- 1	+.138₁₀- 0
+.670₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.926₁₀- 1	+.134₁₀- 0
+.680₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.887₁₀- 1	+.130₁₀- 0
+.690₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.850₁₀- 1	+.127₁₀- 0
+.700₁₀+ 2	+.300₁₀+ 1	+.300₁₀+ 1	+.815₁₀- 1	+.123₁₀- 0
+.200₁₀+ 1	+.400₁₀+ 1	+.300₁₀+ 1	+.666₁₀- 0	+.923₁₀- 0
+.300₁₀+ 1	+.400₁₀+ 1	+.300₁₀+ 1	+.660₁₀- 0	+.887₁₀- 0
+.400₁₀+ 1	+.400₁₀+ 1	+.300₁₀+ 1	+.652₁₀- 0	+.852₁₀- 0
+.500₁₀+ 1	+.400₁₀+ 1	+.300₁₀+ 1	+.642₁₀- 0	+.819₁₀- 0
+.600₁₀+ 1	+.400₁₀+ 1	+.300₁₀+ 1	+.632₁₀- 0	+.787₁₀- 0
+.700₁₀+ 1	+.400₁₀+ 1	+.300₁₀+ 1	+.620₁₀- 0	+.756₁₀- 0
+.800₁₀+ 1	+.400₁₀+ 1	+.300₁₀+ 1	+.607₁₀- 0	+.726₁₀- 0
+.900₁₀+ 1	+.400₁₀+ 1	+.300₁₀+ 1	+.593₁₀- 0	+.698₁₀- 0
+.100₁₀+ 2	+.400₁₀+ 1	+.300₁₀+ 1	+.578₁₀- 0	+.671₁₀- 0
+.110₁₀+ 2	+.400₁₀+ 1	+.300₁₀+ 1	+.564₁₀- 0	+.644₁₀- 0
+.120₁₀+ 2	+.400₁₀+ 1	+.300₁₀+ 1	+.548₁₀- 0	+.619₁₀- 0
+.130₁₀+ 2	+.400₁₀+ 1	+.300₁₀+ 1	+.533₁₀- 0	+.595₁₀- 0

VERBETERING

+.460$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.659$_{10}$- 0 +.632$_{10}$- 0

VERBETERING

+.470$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.651$_{10}$- 0 +.625$_{10}$- 0

VERBETERING

+.480$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.643$_{10}$- 0 +.619$_{10}$- 0

VERBETERING

+.490$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.634$_{10}$- 0 +.613$_{10}$- 0

VERBETERING

+.500$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.626$_{10}$- 0 +.607$_{10}$- 0

VERBETERING

+.510$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.618$_{10}$- 0 +.601$_{10}$- 0

VERBETERING

+.520$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.610$_{10}$- 0 +.595$_{10}$- 0

VERBETERING

+.530$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.601$_{10}$- 0 +.589$_{10}$- 0

VERBETERING

+.540$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.593$_{10}$- 0 +.583$_{10}$- 0

VERBETERING

+.550$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.585$_{10}$- 0 +.577$_{10}$- 0

VERBETERING

+.560$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.577$_{10}$- 0 +.572$_{10}$- 0

VERBETERING

+.570$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.569$_{10}$- 0 +.566$_{10}$- 0

VERBETERING

+.580$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.561$_{10}$- 0 +.560$_{10}$- 0
+.200$_{10}$+ 1 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.759$_{10}$- 0 +.942$_{10}$- 0
+.300$_{10}$+ 1 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.754$_{10}$- 0 +.914$_{10}$- 0
+.400$_{10}$+ 1 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.748$_{10}$- 0 +.887$_{10}$- 0
+.500$_{10}$+ 1 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.740$_{10}$- 0 +.861$_{10}$- 0
+.600$_{10}$+ 1 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.731$_{10}$- 0 +.835$_{10}$- 0
+.700$_{10}$+ 1 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.720$_{10}$- 0 +.811$_{10}$- 0
+.800$_{10}$+ 1 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.709$_{10}$- 0 +.787$_{10}$- 0
+.900$_{10}$+ 1 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.697$_{10}$- 0 +.764$_{10}$- 0
+.100$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.684$_{10}$- 0 +.741$_{10}$- 0
+.110$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.670$_{10}$- 0 +.719$_{10}$- 0
+.120$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.656$_{10}$- 0 +.698$_{10}$- 0
+.130$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.642$_{10}$- 0 +.677$_{10}$- 0
+.140$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.627$_{10}$- 0 +.657$_{10}$- 0
+.150$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.612$_{10}$- 0 +.638$_{10}$- 0
+.160$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.597$_{10}$- 0 +.619$_{10}$- 0
+.170$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.582$_{10}$- 0 +.601$_{10}$- 0
+.180$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.567$_{10}$- 0 +.583$_{10}$- 0
+.190$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.551$_{10}$- 0 +.566$_{10}$- 0
+.200$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.536$_{10}$- 0 +.549$_{10}$- 0
+.210$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.521$_{10}$- 0 +.533$_{10}$- 0
+.220$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.506$_{10}$- 0 +.517$_{10}$- 0

$p = p_1$

	p	m	k	p_{rel}	p_c
VERBETERING					
	$+.260_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.825_{10}-0$	$+.771_{10}-0$
VERBETERING					
	$+.270_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.817_{10}-0$	$+.764_{10}-0$
VERBETERING					
	$+.280_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.809_{10}-0$	$+.756_{10}-0$
VERBETERING					
	$+.290_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.801_{10}-0$	$+.749_{10}-0$
VERBETERING					
	$+.300_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.793_{10}-0$	$+.741_{10}-0$
VERBETERING					
	$+.310_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.785_{10}-0$	$+.734_{10}-0$
VERBETERING					
	$+.320_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.777_{10}-0$	$+.726_{10}-0$
VERBETERING					
	$+.330_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.768_{10}-0$	$+.719_{10}-0$
VERBETERING					
	$+.340_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.760_{10}-0$	$+.712_{10}-0$
VERBETERING					
	$+.350_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.752_{10}-0$	$+.705_{10}-0$
VERBETERING					
	$+.360_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.743_{10}-0$	$+.698_{10}-0$
VERBETERING					
	$+.370_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.735_{10}-0$	$+.691_{10}-0$
VERBETERING					
	$+.380_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.727_{10}-0$	$+.684_{10}-0$
VERBETERING					
	$+.390_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.718_{10}-0$	$+.677_{10}-0$
VERBETERING					
	$+.400_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.710_{10}-0$	$+.671_{10}-0$
VERBETERING					
	$+.410_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.701_{10}-0$	$+.664_{10}-0$
VERBETERING					
	$+.420_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.693_{10}-0$	$+.657_{10}-0$
VERBETERING					
	$+.430_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.684_{10}-0$	$+.651_{10}-0$
VERBETERING					
	$+.440_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.676_{10}-0$	$+.644_{10}-0$
VERBETERING					
	$+.450_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.668_{10}-0$	$+.638_{10}-0$

VERBETERING

$+.660_{10}+2$ $+.100_{10}+1$ $+.200_{10}+1$ $+.788_{10}-0$ $+.719_{10}-0$

VERBETERING

$+.670_{10}+2$ $+.100_{10}+1$ $+.200_{10}+1$ $+.784_{10}-0$ $+.716_{10}-0$

VERBETERING

$+.680_{10}+2$ $+.100_{10}+1$ $+.200_{10}+1$ $+.779_{10}-0$ $+.712_{10}-0$

VERBETERING

$+.690_{10}+2$ $+.100_{10}+1$ $+.200_{10}+1$ $+.775_{10}-0$ $+.709_{10}-0$

VERBETERING

$+.700_{10}+2$ $+.100_{10}+1$ $+.200_{10}+1$ $+.771_{10}-0$ $+.705_{10}-0$

VERBETERING

$+.710_{10}+2$ $+.100_{10}+1$ $+.200_{10}+1$ $+.767_{10}-0$ $+.702_{10}-0$

VERBETERING

$+.720_{10}+2$ $+.100_{10}+1$ $+.200_{10}+1$ $+.762_{10}-0$ $+.698_{10}-0$

VERBETERING

$+.730_{10}+2$ $+.100_{10}+1$ $+.200_{10}+1$ $+.758_{10}-0$ $+.695_{10}-0$

VERBETERING

$+.740_{10}+2$ $+.100_{10}+1$ $+.200_{10}+1$ $+.754_{10}-0$ $+.691_{10}-0$

VERBETERING

$+.750_{10}+2$ $+.100_{10}+1$ $+.200_{10}+1$ $+.749_{10}-0$ $+.688_{10}-0$

VERBETERING

$+.760_{10}+2$ $+.100_{10}+1$ $+.200_{10}+1$ $+.745_{10}-0$ $+.684_{10}-0$

VERBETERING

$+.770_{10}+2$ $+.100_{10}+1$ $+.200_{10}+1$ $+.741_{10}-0$ $+.681_{10}-0$

VERBETERING

$+.780_{10}+2$ $+.100_{10}+1$ $+.200_{10}+1$ $+.736_{10}-0$ $+.677_{10}-0$

VERBETERING

$+.790_{10}+2$ $+.100_{10}+1$ $+.200_{10}+1$ $+.732_{10}-0$ $+.674_{10}-0$

VERBETERING

$+.800_{10}+2$ $+.100_{10}+1$ $+.200_{10}+1$ $+.728_{10}-0$ $+.671_{10}-0$

VERBETERING

$+.810_{10}+2$ $+.100_{10}+1$ $+.200_{10}+1$ $+.723_{10}-0$ $+.667_{10}-0$

VERBETERING

$+.820_{10}+2$ $+.100_{10}+1$ $+.200_{10}+1$ $+.719_{10}-0$ $+.664_{10}-0$

VERBETERING

$+.830_{10}+2$ $+.100_{10}+1$ $+.200_{10}+1$ $+.715_{10}-0$ $+.661_{10}-0$

VERBETERING

$+.840_{10}+2$ $+.100_{10}+1$ $+.200_{10}+1$ $+.710_{10}-0$ $+.657_{10}-0$

VERBETERING

$+.850_{10}+2$ $+.100_{10}+1$ $+.200_{10}+1$ $+.706_{10}-0$ $+.654_{10}-0$

	n	m	k	$pred$	pc
VERBETERING					
VERBETERING	$+.460_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.869_{10}-0$	$+.795_{10}-0$
VERBETERING	$+.470_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.865_{10}-0$	$+.791_{10}-0$
VERBETERING	$+.480_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.861_{10}-0$	$+.787_{10}-0$
VERBETERING	$+.490_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.857_{10}-0$	$+.783_{10}-0$
VERBETERING	$+.500_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.854_{10}-0$	$+.779_{10}-0$
VERBETERING	$+.510_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.850_{10}-0$	$+.775_{10}-0$
VERBETERING	$+.520_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.846_{10}-0$	$+.771_{10}-0$
VERBETERING	$+.530_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.842_{10}-0$	$+.767_{10}-0$
VERBETERING	$+.540_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.838_{10}-0$	$+.764_{10}-0$
VERBETERING	$+.550_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.834_{10}-0$	$+.760_{10}-0$
VERBETERING	$+.560_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.830_{10}-0$	$+.756_{10}-0$
VERBETERING	$+.570_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.825_{10}-0$	$+.752_{10}-0$
VERBETERING	$+.580_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.821_{10}-0$	$+.749_{10}-0$
VERBETERING	$+.590_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.817_{10}-0$	$+.745_{10}-0$
VERBETERING	$+.600_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.813_{10}-0$	$+.741_{10}-0$
VERBETERING	$+.610_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.809_{10}-0$	$+.737_{10}-0$
VERBETERING	$+.620_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.805_{10}-0$	$+.734_{10}-0$
VERBETERING	$+.630_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.801_{10}-0$	$+.730_{10}-0$
VERBETERING	$+.640_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.796_{10}-0$	$+.726_{10}-0$
	$+.650_{10}+2$	$+.100_{10}+1$	$+.200_{10}+1$	$+.792_{10}-0$	$+.723_{10}-0$

VERBETERING

VERBETERING +.260$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.935$_{10}$- 0 +.878$_{10}$- 0

VERBETERING +.270$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.933$_{10}$- 0 +.874$_{10}$- 0

VERBETERING +.280$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.930$_{10}$- 0 +.870$_{10}$- 0

VERBETERING +.290$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.927$_{10}$- 0 +.865$_{10}$- 0

VERBETERING +.300$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.924$_{10}$- 0 +.861$_{10}$- 0

VERBETERING +.310$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.921$_{10}$- 0 +.857$_{10}$- 0

VERBETERING +.320$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.918$_{10}$- 0 +.852$_{10}$- 0

VERBETERING +.330$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.915$_{10}$- 0 +.848$_{10}$- 0

VERBETERING +.340$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.911$_{10}$- 0 +.844$_{10}$- 0

VERBETERING +.350$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.908$_{10}$- 0 +.840$_{10}$- 0

VERBETERING +.360$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.905$_{10}$- 0 +.835$_{10}$- 0

VERBETERING +.370$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.901$_{10}$- 0 +.831$_{10}$- 0

VERBETERING +.380$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.898$_{10}$- 0 +.827$_{10}$- 0

VERBETERING +.390$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.895$_{10}$- 0 +.823$_{10}$- 0

VERBETERING +.400$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.891$_{10}$- 0 +.819$_{10}$- 0

VERBETERING +.410$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.887$_{10}$- 0 +.815$_{10}$- 0

VERBETERING +.420$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.884$_{10}$- 0 +.811$_{10}$- 0

VERBETERING +.430$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.880$_{10}$- 0 +.807$_{10}$- 0

VERBETERING +.440$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.876$_{10}$- 0 +.803$_{10}$- 0

+.450$_{10}$+ 2 +.100$_{10}$+ 1 +.200$_{10}$+ 1 +.873$_{10}$- 0 +.799$_{10}$- 0

$P = P_2$

	n	m	k	$prob$	pc
PVERBETERING					
VERBETERING	$+.600_{10}+$ 1	$+.100_{10}+$ 1	$+.200_{10}+$ 1	$+.973_{10}-$ 0	$+.970_{10}-$ 0
VERBETERING	$+.700_{10}+$ 1	$+.100_{10}+$ 1	$+.200_{10}+$ 1	$+.972_{10}-$ 0	$+.966_{10}-$ 0
VERBETERING	$+.800_{10}+$ 1	$+.100_{10}+$ 1	$+.200_{10}+$ 1	$+.971_{10}-$ 0	$+.961_{10}-$ 0
VERBETERING	$+.900_{10}+$ 1	$+.100_{10}+$ 1	$+.200_{10}+$ 1	$+.970_{10}-$ 0	$+.956_{10}-$ 0
VERBETERING	$+.100_{10}+$ 2	$+.100_{10}+$ 1	$+.200_{10}+$ 1	$+.969_{10}-$ 0	$+.951_{10}-$ 0
VERBETERING	$+.110_{10}+$ 2	$+.100_{10}+$ 1	$+.200_{10}+$ 1	$+.967_{10}-$ 0	$+.947_{10}-$ 0
VERBETERING	$+.120_{10}+$ 2	$+.100_{10}+$ 1	$+.200_{10}+$ 1	$+.966_{10}-$ 0	$+.942_{10}-$ 0
VERBETERING	$+.130_{10}+$ 2	$+.100_{10}+$ 1	$+.200_{10}+$ 1	$+.964_{10}-$ 0	$+.937_{10}-$ 0
VERBETERING	$+.140_{10}+$ 2	$+.100_{10}+$ 1	$+.200_{10}+$ 1	$+.963_{10}-$ 0	$+.932_{10}-$ 0
VERBETERING	$+.150_{10}+$ 2	$+.100_{10}+$ 1	$+.200_{10}+$ 1	$+.961_{10}-$ 0	$+.928_{10}-$ 0
VERBETERING	$+.160_{10}+$ 2	$+.100_{10}+$ 1	$+.200_{10}+$ 1	$+.959_{10}-$ 0	$+.923_{10}-$ 0
VERBETERING	$+.170_{10}+$ 2	$+.100_{10}+$ 1	$+.200_{10}+$ 1	$+.957_{10}-$ 0	$+.919_{10}-$ 0
VERBETERING	$+.180_{10}+$ 2	$+.100_{10}+$ 1	$+.200_{10}+$ 1	$+.955_{10}-$ 0	$+.914_{10}-$ 0
VERBETERING	$+.190_{10}+$ 2	$+.100_{10}+$ 1	$+.200_{10}+$ 1	$+.953_{10}-$ 0	$+.909_{10}-$ 0
VERBETERING	$+.200_{10}+$ 2	$+.100_{10}+$ 1	$+.200_{10}+$ 1	$+.951_{10}-$ 0	$+.905_{10}-$ 0
VERBETERING	$+.210_{10}+$ 2	$+.100_{10}+$ 1	$+.200_{10}+$ 1	$+.948_{10}-$ 0	$+.900_{10}-$ 0
VERBETERING	$+.220_{10}+$ 2	$+.100_{10}+$ 1	$+.200_{10}+$ 1	$+.946_{10}-$ 0	$+.896_{10}-$ 0
VERBETERING	$+.230_{10}+$ 2	$+.100_{10}+$ 1	$+.200_{10}+$ 1	$+.943_{10}-$ 0	$+.892_{10}-$ 0
VERBETERING	$+.240_{10}+$ 2	$+.100_{10}+$ 1	$+.200_{10}+$ 1	$+.941_{10}-$ 0	$+.887_{10}-$ 0
	$+.250_{10}+$ 2	$+.100_{10}+$ 1	$+.200_{10}+$ 1	$+.938_{10}-$ 0	$+.883_{10}-$ 0

P=P2

	n	m	k	pred	pc
VERBETERING	+.860$_{10}$+ 2	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.702$_{10}$- 0	+.651$_{10}$- 0
VERBETERING	+.870$_{10}$+ 2	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.697$_{10}$- 0	+.648$_{10}$- 0
VERBETERING	+.880$_{10}$+ 2	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.693$_{10}$- 0	+.644$_{10}$- 0
VERBETERING	+.890$_{10}$+ 2	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.689$_{10}$- 0	+.641$_{10}$- 0
VERBETERING	+.900$_{10}$+ 2	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.684$_{10}$- 0	+.638$_{10}$- 0
VERBETERING	+.910$_{10}$+ 2	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.680$_{10}$- 0	+.635$_{10}$- 0
VERBETERING	+.920$_{10}$+ 2	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.676$_{10}$- 0	+.632$_{10}$- 0
VERBETERING	+.930$_{10}$+ 2	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.672$_{10}$- 0	+.629$_{10}$- 0
VERBETERING	+.940$_{10}$+ 2	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.667$_{10}$- 0	+.625$_{10}$- 0
VERBETERING	+.950$_{10}$+ 2	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.663$_{10}$- 0	+.622$_{10}$- 0
VERBETERING	+.960$_{10}$+ 2	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.659$_{10}$- 0	+.619$_{10}$- 0
VERBETERING	+.970$_{10}$+ 2	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.655$_{10}$- 0	+.616$_{10}$- 0
VERBETERING	+.980$_{10}$+ 2	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.650$_{10}$- 0	+.613$_{10}$- 0
VERBETERING	+.990$_{10}$+ 2	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.646$_{10}$- 0	+.610$_{10}$- 0
VERBETERING	+.100$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.642$_{10}$- 0	+.607$_{10}$- 0
VERBETERING	+.101$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.638$_{10}$- 0	+.604$_{10}$- 0
VERBETERING	+.102$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.633$_{10}$- 0	+.601$_{10}$- 0
VERBETERING	+.103$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.629$_{10}$- 0	+.598$_{10}$- 0
VERBETERING	+.104$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.625$_{10}$- 0	+.595$_{10}$- 0
	+.105$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.621$_{10}$- 0	+.592$_{10}$- 0

	n	m	k	prod	pc
VERBETERING	+.106$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.617$_{10}$- 0	+.589$_{10}$- 0
VERBETERING	+.107$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.612$_{10}$- 0	+.586$_{10}$- 0
VERBETERING	+.108$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.608$_{10}$- 0	+.583$_{10}$- 0
VERBETERING	+.109$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.604$_{10}$- 0	+.580$_{10}$- 0
VERBETERING	+.110$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.600$_{10}$- 0	+.577$_{10}$- 0
VERBETERING	+.111$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.596$_{10}$- 0	+.575$_{10}$- 0
VERBETERING	+.112$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.592$_{10}$- 0	+.572$_{10}$- 0
VERBETERING	+.113$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.588$_{10}$- 0	+.569$_{10}$- 0
VERBETERING	+.114$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.584$_{10}$- 0	+.566$_{10}$- 0
VERBETERING	+.115$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.580$_{10}$- 0	+.563$_{10}$- 0
VERBETERING	+.116$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.576$_{10}$- 0	+.560$_{10}$- 0
VERBETERING	+.117$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.572$_{10}$- 0	+.558$_{10}$- 0
VERBETERING	+.118$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.568$_{10}$- 0	+.555$_{10}$- 0
VERBETERING	+.119$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.564$_{10}$- 0	+.552$_{10}$- 0
VERBETERING	+.120$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.560$_{10}$- 0	+.549$_{10}$- 0
VERBETERING	+.121$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.556$_{10}$- 0	+.547$_{10}$- 0
VERBETERING	+.122$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.552$_{10}$- 0	+.544$_{10}$- 0
VERBETERING	+.123$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.548$_{10}$- 0	+.541$_{10}$- 0
VERBETERING	+.124$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.544$_{10}$- 0	+.538$_{10}$- 0
	+.125$_{10}$+ 3	+.100$_{10}$+ 1	+.200$_{10}$+ 1	+.540$_{10}$- 0	+.536$_{10}$- 0

VERBETERING

VERBETERING +.126₁₀+ 3 +.100₁₀+ 1 +.200₁₀+ 1 +.536₁₀- 0 +.533₁₀- 0

VERBETERING +.127₁₀+ 3 +.100₁₀+ 1 +.200₁₀+ 1 +.532₁₀- 0 +.530₁₀- 0

VERBETERING +.128₁₀+ 3 +.100₁₀+ 1 +.200₁₀+ 1 +.528₁₀- 0 +.528₁₀- 0

VERBETERING +.110₁₀+ 2 +.200₁₀+ 1 +.300₁₀+ 1 +.900₁₀- 0 +.896₁₀- 0

VERBETERING +.120₁₀+ 2 +.200₁₀+ 1 +.300₁₀+ 1 +.895₁₀- 0 +.887₁₀- 0

VERBETERING +.130₁₀+ 2 +.200₁₀+ 1 +.300₁₀+ 1 +.891₁₀- 0 +.878₁₀- 0

VERBETERING +.140₁₀+ 2 +.200₁₀+ 1 +.300₁₀+ 1 +.887₁₀- 0 +.870₁₀- 0

VERBETERING +.150₁₀+ 2 +.200₁₀+ 1 +.300₁₀+ 1 +.882₁₀- 0 +.861₁₀- 0

VERBETERING +.160₁₀+ 2 +.200₁₀+ 1 +.300₁₀+ 1 +.877₁₀- 0 +.852₁₀- 0

VERBETERING +.170₁₀+ 2 +.200₁₀+ 1 +.300₁₀+ 1 +.871₁₀- 0 +.844₁₀- 0

VERBETERING +.180₁₀+ 2 +.200₁₀+ 1 +.300₁₀+ 1 +.866₁₀- 0 +.835₁₀- 0

VERBETERING +.190₁₀+ 2 +.200₁₀+ 1 +.300₁₀+ 1 +.860₁₀- 0 +.827₁₀- 0

VERBETERING +.200₁₀+ 2 +.200₁₀+ 1 +.300₁₀+ 1 +.855₁₀- 0 +.819₁₀- 0

VERBETERING +.210₁₀+ 2 +.200₁₀+ 1 +.300₁₀+ 1 +.849₁₀- 0 +.811₁₀- 0

VERBETERING +.220₁₀+ 2 +.200₁₀+ 1 +.300₁₀+ 1 +.843₁₀- 0 +.803₁₀- 0

VERBETERING +.230₁₀+ 2 +.200₁₀+ 1 +.300₁₀+ 1 +.836₁₀- 0 +.795₁₀- 0

VERBETERING +.240₁₀+ 2 +.200₁₀+ 1 +.300₁₀+ 1 +.830₁₀- 0 +.787₁₀- 0

VERBETERING +.250₁₀+ 2 +.200₁₀+ 1 +.300₁₀+ 1 +.823₁₀- 0 +.779₁₀- 0

VERBETERING +.260₁₀+ 2 +.200₁₀+ 1 +.300₁₀+ 1 +.817₁₀- 0 +.771₁₀- 0

VERBETERING +.270₁₀+ 2 +.200₁₀+ 1 +.300₁₀+ 1 +.810₁₀- 0 +.764₁₀- 0

VERBETERING

VERBETERING +.280$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.803$_{10}$- 0 +.756$_{10}$- 0

VERBETERING +.290$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.796$_{10}$- 0 +.749$_{10}$- 0

VERBETERING +.300$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.789$_{10}$- 0 +.741$_{10}$- 0

VERBETERING +.310$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.782$_{10}$- 0 +.734$_{10}$- 0

VERBETERING +.320$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.775$_{10}$- 0 +.726$_{10}$- 0

VERBETERING +.330$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.768$_{10}$- 0 +.719$_{10}$- 0

VERBETERING +.340$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.761$_{10}$- 0 +.712$_{10}$- 0

VERBETERING +.350$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.753$_{10}$- 0 +.705$_{10}$- 0

VERBETERING +.360$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.746$_{10}$- 0 +.698$_{10}$- 0

VERBETERING +.370$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.738$_{10}$- 0 +.691$_{10}$- 0

VERBETERING +.380$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.731$_{10}$- 0 +.684$_{10}$- 0

VERBETERING +.390$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.724$_{10}$- 0 +.677$_{10}$- 0

VERBETERING +.400$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.716$_{10}$- 0 +.671$_{10}$- 0

VERBETERING +.410$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.709$_{10}$- 0 +.664$_{10}$- 0

VERBETERING +.420$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.701$_{10}$- 0 +.657$_{10}$- 0

VERBETERING +.430$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.693$_{10}$- 0 +.651$_{10}$- 0

VERBETERING +.440$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.686$_{10}$- 0 +.644$_{10}$- 0

VERBETERING +.450$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.678$_{10}$- 0 +.638$_{10}$- 0

VERBETERING +.460$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.671$_{10}$- 0 +.632$_{10}$- 0

+.470$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.663$_{10}$- 0 +.625$_{10}$- 0

$p = p_2$

	n	m	k	p_{max}	p_c
VERBETERING	$+.480_{10}+2$	$+.200_{10}+1$	$+.300_{10}+1$	$+.656_{10}-0$	$+.619_{10}-0$
VERBETERING	$+.490_{10}+2$	$+.200_{10}+1$	$+.300_{10}+1$	$+.648_{10}-0$	$+.613_{10}-0$
VERBETERING	$+.500_{10}+2$	$+.200_{10}+1$	$+.300_{10}+1$	$+.641_{10}-0$	$+.607_{10}-0$
VERBETERING	$+.510_{10}+2$	$+.200_{10}+1$	$+.300_{10}+1$	$+.633_{10}-0$	$+.601_{10}-0$
VERBETERING	$+.520_{10}+2$	$+.200_{10}+1$	$+.300_{10}+1$	$+.626_{10}-0$	$+.595_{10}-0$
VERBETERING	$+.530_{10}+2$	$+.200_{10}+1$	$+.300_{10}+1$	$+.618_{10}-0$	$+.589_{10}-0$
VERBETERING	$+.540_{10}+2$	$+.200_{10}+1$	$+.300_{10}+1$	$+.611_{10}-0$	$+.583_{10}-0$
VERBETERING	$+.550_{10}+2$	$+.200_{10}+1$	$+.300_{10}+1$	$+.603_{10}-0$	$+.577_{10}-0$
VERBETERING	$+.560_{10}+2$	$+.200_{10}+1$	$+.300_{10}+1$	$+.596_{10}-0$	$+.572_{10}-0$
VERBETERING	$+.570_{10}+2$	$+.200_{10}+1$	$+.300_{10}+1$	$+.589_{10}-0$	$+.566_{10}-0$
VERBETERING	$+.580_{10}+2$	$+.200_{10}+1$	$+.300_{10}+1$	$+.581_{10}-0$	$+.560_{10}-0$
VERBETERING	$+.590_{10}+2$	$+.200_{10}+1$	$+.300_{10}+1$	$+.574_{10}-0$	$+.555_{10}-0$
VERBETERING	$+.600_{10}+2$	$+.200_{10}+1$	$+.300_{10}+1$	$+.567_{10}-0$	$+.549_{10}-0$
VERBETERING	$+.610_{10}+2$	$+.200_{10}+1$	$+.300_{10}+1$	$+.560_{10}-0$	$+.544_{10}-0$
VERBETERING	$+.620_{10}+2$	$+.200_{10}+1$	$+.300_{10}+1$	$+.553_{10}-0$	$+.538_{10}-0$
VERBETERING	$+.630_{10}+2$	$+.200_{10}+1$	$+.300_{10}+1$	$+.546_{10}-0$	$+.533_{10}-0$
VERBETERING	$+.640_{10}+2$	$+.200_{10}+1$	$+.300_{10}+1$	$+.539_{10}-0$	$+.528_{10}-0$
VERBETERING	$+.650_{10}+2$	$+.200_{10}+1$	$+.300_{10}+1$	$+.532_{10}-0$	$+.523_{10}-0$
VERBETERING	$+.660_{10}+2$	$+.200_{10}+1$	$+.300_{10}+1$	$+.525_{10}-0$	$+.517_{10}-0$
	$+.670_{10}+2$	$+.200_{10}+1$	$+.300_{10}+1$	$+.518_{10}-0$	$+.512_{10}-0$

VERBETERING

+.680$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.511$_{10}$- 0 +.507$_{10}$- 0

VERBETERING

+.690$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.504$_{10}$- 0 +.502$_{10}$- 0

VERBETERING

+.700$_{10}$+ 2 +.200$_{10}$+ 1 +.300$_{10}$+ 1 +.497$_{10}$- 0 +.497$_{10}$- 0
+.200$_{10}$+ 1 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.873$_{10}$- 0 +.970$_{10}$- 0
+.300$_{10}$+ 1 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.871$_{10}$- 0 +.956$_{10}$- 0
+.400$_{10}$+ 1 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.869$_{10}$- 0 +.942$_{10}$- 0
+.500$_{10}$+ 1 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.866$_{10}$- 0 +.928$_{10}$- 0
+.600$_{10}$+ 1 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.863$_{10}$- 0 +.914$_{10}$- 0
+.700$_{10}$+ 1 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.860$_{10}$- 0 +.900$_{10}$- 0
+.800$_{10}$+ 1 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.856$_{10}$- 0 +.887$_{10}$- 0
+.900$_{10}$+ 1 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.851$_{10}$- 0 +.874$_{10}$- 0
+.100$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.847$_{10}$- 0 +.861$_{10}$- 0
+.110$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.841$_{10}$- 0 +.848$_{10}$- 0

VERBETERING

+.120$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.836$_{10}$- 0 +.835$_{10}$- 0

VERBETERING

+.130$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.830$_{10}$- 0 +.823$_{10}$- 0

VERBETERING

+.140$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.824$_{10}$- 0 +.811$_{10}$- 0

VERBETERING

+.150$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.818$_{10}$- 0 +.799$_{10}$- 0

VERBETERING

+.160$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.811$_{10}$- 0 +.787$_{10}$- 0

VERBETERING

+.170$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.804$_{10}$- 0 +.775$_{10}$- 0

VERBETERING

+.180$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.797$_{10}$- 0 +.764$_{10}$- 0

VERBETERING

+.190$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.790$_{10}$- 0 +.752$_{10}$- 0

VERBETERING

+.200$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.782$_{10}$- 0 +.741$_{10}$- 0

VERBETERING

+.210$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.775$_{10}$- 0 +.730$_{10}$- 0

VERBETERING

+.220$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.767$_{10}$- 0 +.719$_{10}$- 0

VERBETERING

+.230$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.759$_{10}$- 0 +.709$_{10}$- 0

VERBETERING

+.240$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.751$_{10}$- 0 +.698$_{10}$- 0

VERBETERING

$P_2 = P_2$

	n	m	K	pred	pc
VERBETERING	+.250$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.743$_{10}$- 0	+.688$_{10}$- 0
VERBETERING	+.260$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.735$_{10}$- 0	+.677$_{10}$- 0
VERBETERING	+.270$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.726$_{10}$- 0	+.667$_{10}$- 0
VERBETERING	+.280$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.718$_{10}$- 0	+.657$_{10}$- 0
VERBETERING	+.290$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.709$_{10}$- 0	+.648$_{10}$- 0
VERBETERING	+.300$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.701$_{10}$- 0	+.638$_{10}$- 0
VERBETERING	+.310$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.692$_{10}$- 0	+.629$_{10}$- 0
VERBETERING	+.320$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.683$_{10}$- 0	+.619$_{10}$- 0
VERBETERING	+.330$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.675$_{10}$- 0	+.610$_{10}$- 0
VERBETERING	+.340$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.666$_{10}$- 0	+.601$_{10}$- 0
VERBETERING	+.350$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.657$_{10}$- 0	+.592$_{10}$- 0
VERBETERING	+.360$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.648$_{10}$- 0	+.583$_{10}$- 0
VERBETERING	+.370$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.640$_{10}$- 0	+.575$_{10}$- 0
VERBETERING	+.380$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.631$_{10}$- 0	+.566$_{10}$- 0
VERBETERING	+.390$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.622$_{10}$- 0	+.558$_{10}$- 0
VERBETERING	+.400$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.613$_{10}$- 0	+.549$_{10}$- 0
VERBETERING	+.410$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.605$_{10}$- 0	+.541$_{10}$- 0
VERBETERING	+.420$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.596$_{10}$- 0	+.533$_{10}$- 0
VERBETERING	+.430$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.587$_{10}$- 0	+.525$_{10}$- 0
VERBETERING	+.440$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.579$_{10}$- 0	+.517$_{10}$- 0

+.450$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.570$_{10}$- 0 +.510$_{10}$- 0
VERBETERING

+.460$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.562$_{10}$- 0 +.502$_{10}$- 0
VERBETERING

+.470$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.553$_{10}$- 0 +.495$_{10}$- 0
VERBETERING

+.480$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.545$_{10}$- 0 +.487$_{10}$- 0
VERBETERING

+.490$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.536$_{10}$- 0 +.480$_{10}$- 0
VERBETERING

+.500$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.528$_{10}$- 0 +.473$_{10}$- 0
VERBETERING

+.510$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.520$_{10}$- 0 +.466$_{10}$- 0
VERBETERING

+.520$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.511$_{10}$- 0 +.459$_{10}$- 0
VERBETERING

+.530$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.503$_{10}$- 0 +.452$_{10}$- 0
VERBETERING

+.540$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.495$_{10}$- 0 +.445$_{10}$- 0
VERBETERING

+.550$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.487$_{10}$- 0 +.439$_{10}$- 0
VERBETERING

+.560$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.479$_{10}$- 0 +.432$_{10}$- 0
VERBETERING

+.570$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.471$_{10}$- 0 +.426$_{10}$- 0
VERBETERING

+.580$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.464$_{10}$- 0 +.419$_{10}$- 0
VERBETERING

+.590$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.456$_{10}$- 0 +.413$_{10}$- 0
VERBETERING

+.600$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.448$_{10}$- 0 +.407$_{10}$- 0
VERBETERING

+.610$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.441$_{10}$- 0 +.401$_{10}$- 0
VERBETERING

+.620$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.433$_{10}$- 0 +.395$_{10}$- 0
VERBETERING

+.630$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.426$_{10}$- 0 +.389$_{10}$- 0
VERBETERING

+.640$_{10}$+ 2 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.419$_{10}$- 0 +.383$_{10}$- 0
VERBETERING

$p = p_2$

	p	m	k	pred	pc
VERBETERING	+.650$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.411$_{10}$- 0	+.378$_{10}$- 0
VERBETERING	+.660$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.404$_{10}$- 0	+.372$_{10}$- 0
VERBETERING	+.670$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.397$_{10}$- 0	+.367$_{10}$- 0
VERBETERING	+.680$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.390$_{10}$- 0	+.361$_{10}$- 0
VERBETERING	+.690$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.384$_{10}$- 0	+.356$_{10}$- 0
VERBETERING	+.700$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.377$_{10}$- 0	+.350$_{10}$- 0
VERBETERING	+.710$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.370$_{10}$- 0	+.345$_{10}$- 0
VERBETERING	+.720$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.363$_{10}$- 0	+.340$_{10}$- 0
VERBETERING	+.730$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.357$_{10}$- 0	+.335$_{10}$- 0
VERBETERING	+.740$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.351$_{10}$- 0	+.330$_{10}$- 0
VERBETERING	+.750$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.344$_{10}$- 0	+.325$_{10}$- 0
VERBETERING	+.760$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.338$_{10}$- 0	+.320$_{10}$- 0
VERBETERING	+.770$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.332$_{10}$- 0	+.316$_{10}$- 0
VERBETERING	+.780$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.326$_{10}$- 0	+.311$_{10}$- 0
VERBETERING	+.790$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.320$_{10}$- 0	+.306$_{10}$- 0
VERBETERING	+.800$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.314$_{10}$- 0	+.302$_{10}$- 0
VERBETERING	+.810$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.308$_{10}$- 0	+.297$_{10}$- 0
VERBETERING	+.820$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.302$_{10}$- 0	+.293$_{10}$- 0
VERBETERING	+.830$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.297$_{10}$- 0	+.288$_{10}$- 0
VERBETERING	+.840$_{10}$+ 2	+.300$_{10}$+ 1	+.300$_{10}$+ 1	+.291$_{10}$- 0	+.284$_{10}$- 0

$+.850_{10}+2$ $+.300_{10}+1$ $+.300_{10}+1$ $+.286_{10}-0$ $+.280_{10}-0$

VERBETERING

$+.860_{10}+2$ $+.300_{10}+1$ $+.300_{10}+1$ $+.280_{10}-0$ $+.276_{10}-0$

VERBETERING

$+.870_{10}+2$ $+.300_{10}+1$ $+.300_{10}+1$ $+.275_{10}-0$ $+.272_{10}-0$

VERBETERING

$+.880_{10}+2$ $+.300_{10}+1$ $+.300_{10}+1$ $+.270_{10}-0$ $+.268_{10}-0$

VERBETERING

$+.890_{10}+2$ $+.300_{10}+1$ $+.300_{10}+1$ $+.265_{10}-0$ $+.264_{10}-0$
$+.900_{10}+2$ $+.300_{10}+1$ $+.300_{10}+1$ $+.260_{10}-0$ $+.260_{10}-0$
$+.910_{10}+2$ $+.300_{10}+1$ $+.300_{10}+1$ $+.255_{10}-0$ $+.256_{10}-0$
$+.920_{10}+2$ $+.300_{10}+1$ $+.300_{10}+1$ $+.250_{10}-0$ $+.252_{10}-0$
$+.930_{10}+2$ $+.300_{10}+1$ $+.300_{10}+1$ $+.245_{10}-0$ $+.248_{10}-0$
$+.940_{10}+2$ $+.300_{10}+1$ $+.300_{10}+1$ $+.240_{10}-0$ $+.245_{10}-0$
$+.950_{10}+2$ $+.300_{10}+1$ $+.300_{10}+1$ $+.236_{10}-0$ $+.241_{10}-0$
$+.960_{10}+2$ $+.300_{10}+1$ $+.300_{10}+1$ $+.231_{10}-0$ $+.237_{10}-0$
$+.970_{10}+2$ $+.300_{10}+1$ $+.300_{10}+1$ $+.227_{10}-0$ $+.234_{10}-0$
$+.980_{10}+2$ $+.300_{10}+1$ $+.300_{10}+1$ $+.222_{10}-0$ $+.230_{10}-0$
$+.990_{10}+2$ $+.300_{10}+1$ $+.300_{10}+1$ $+.218_{10}-0$ $+.227_{10}-0$
$+.100_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.213_{10}-0$ $+.224_{10}-0$
$+.101_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.209_{10}-0$ $+.220_{10}-0$
$+.102_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.205_{10}-0$ $+.217_{10}-0$
$+.103_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.201_{10}-0$ $+.214_{10}-0$
$+.104_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.197_{10}-0$ $+.211_{10}-0$
$+.105_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.193_{10}-0$ $+.207_{10}-0$
$+.106_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.189_{10}-0$ $+.204_{10}-0$
$+.107_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.186_{10}-0$ $+.201_{10}-0$
$+.108_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.182_{10}-0$ $+.198_{10}-0$
$+.109_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.178_{10}-0$ $+.195_{10}-0$
$+.110_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.175_{10}-0$ $+.193_{10}-0$
$+.111_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.171_{10}-0$ $+.190_{10}-0$
$+.112_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.168_{10}-0$ $+.187_{10}-0$
$+.113_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.164_{10}-0$ $+.184_{10}-0$
$+.114_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.161_{10}-0$ $+.181_{10}-0$
$+.115_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.158_{10}-0$ $+.179_{10}-0$
$+.116_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.154_{10}-0$ $+.176_{10}-0$
$+.117_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.151_{10}-0$ $+.173_{10}-0$
$+.118_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.148_{10}-0$ $+.171_{10}-0$
$+.119_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.145_{10}-0$ $+.168_{10}-0$
$+.120_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.142_{10}-0$ $+.166_{10}-0$
$+.121_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.139_{10}-0$ $+.163_{10}-0$
$+.122_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.136_{10}-0$ $+.161_{10}-0$
$+.123_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.134_{10}-0$ $+.158_{10}-0$
$+.124_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.131_{10}-0$ $+.156_{10}-0$
$+.125_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.128_{10}-0$ $+.154_{10}-0$
$+.126_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.125_{10}-0$ $+.151_{10}-0$
$+.127_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.123_{10}-0$ $+.149_{10}-0$
$+.128_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.120_{10}-0$ $+.147_{10}-0$
$+.129_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.118_{10}-0$ $+.145_{10}-0$
$+.130_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.115_{10}-0$ $+.143_{10}-0$
$+.131_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.113_{10}-0$ $+.141_{10}-0$
$+.132_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.111_{10}-0$ $+.138_{10}-0$
$+.133_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.108_{10}-0$ $+.136_{10}-0$
$+.134_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.106_{10}-0$ $+.134_{10}-0$
$+.135_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.104_{10}-0$ $+.132_{10}-0$
$+.136_{10}+3$ $+.300_{10}+1$ $+.300_{10}+1$ $+.102_{10}-0$ $+.130_{10}-0$

+.137$_{10}$+ 3 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.994$_{10}$- 1 +.128$_{10}$- 0
+.138$_{10}$+ 3 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.973$_{10}$- 1 +.127$_{10}$- 0
+.139$_{10}$+ 3 +.300$_{10}$+ 1 +.300$_{10}$+ 1 +.952$_{10}$- 1 +.125$_{10}$- 0
+.200$_{10}$+ 1 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.817$_{10}$- 0 +.961$_{10}$- 0
+.300$_{10}$+ 1 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.815$_{10}$- 0 +.942$_{10}$- 0
+.400$_{10}$+ 1 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.813$_{10}$- 0 +.923$_{10}$- 0
+.500$_{10}$+ 1 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.809$_{10}$- 0 +.905$_{10}$- 0
+.600$_{10}$+ 1 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.805$_{10}$- 0 +.887$_{10}$- 0
+.700$_{10}$+ 1 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.801$_{10}$- 0 +.870$_{10}$- 0
+.800$_{10}$+ 1 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.796$_{10}$- 0 +.852$_{10}$- 0
+.900$_{10}$+ 1 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.790$_{10}$- 0 +.835$_{10}$- 0
+.100$_{10}$+ 2 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.784$_{10}$- 0 +.819$_{10}$- 0
+.110$_{10}$+ 2 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.778$_{10}$- 0 +.803$_{10}$- 0
+.120$_{10}$+ 2 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.771$_{10}$- 0 +.787$_{10}$- 0
+.130$_{10}$+ 2 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.764$_{10}$- 0 +.771$_{10}$- 0
VERBETERING

+.140$_{10}$+ 2 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.757$_{10}$- 0 +.756$_{10}$- 0
VERBETERING

+.150$_{10}$+ 2 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.749$_{10}$- 0 +.741$_{10}$- 0
VERBETERING

+.160$_{10}$+ 2 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.741$_{10}$- 0 +.726$_{10}$- 0
VERBETERING

+.170$_{10}$+ 2 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.732$_{10}$- 0 +.712$_{10}$- 0
VERBETERING

+.180$_{10}$+ 2 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.724$_{10}$- 0 +.698$_{10}$- 0
VERBETERING

+.190$_{10}$+ 2 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.715$_{10}$- 0 +.684$_{10}$- 0
VERBETERING

+.200$_{10}$+ 2 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.706$_{10}$- 0 +.671$_{10}$- 0
VERBETERING

+.210$_{10}$+ 2 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.697$_{10}$- 0 +.657$_{10}$- 0
VERBETERING

+.220$_{10}$+ 2 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.688$_{10}$- 0 +.644$_{10}$- 0
VERBETERING

+.230$_{10}$+ 2 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.679$_{10}$- 0 +.632$_{10}$- 0
VERBETERING

+.240$_{10}$+ 2 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.669$_{10}$- 0 +.619$_{10}$- 0
VERBETERING

+.250$_{10}$+ 2 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.660$_{10}$- 0 +.607$_{10}$- 0
VERBETERING

+.260$_{10}$+ 2 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.650$_{10}$- 0 +.595$_{10}$- 0
VERBETERING

+.270$_{10}$+ 2 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.641$_{10}$- 0 +.583$_{10}$- 0
VERBETERING

+.280$_{10}$+ 2 +.400$_{10}$+ 1 +.300$_{10}$+ 1 +.631$_{10}$- 0 +.572$_{10}$- 0

VERBETERING

VERBETERING +.290₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.621₁₀- 0 +.560₁₀- 0

VERBETERING +.300₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.612₁₀- 0 +.549₁₀- 0

VERBETERING +.310₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.602₁₀- 0 +.538₁₀- 0

VERBETERING +.320₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.592₁₀- 0 +.528₁₀- 0

VERBETERING +.330₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.582₁₀- 0 +.517₁₀- 0

VERBETERING +.340₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.573₁₀- 0 +.507₁₀- 0

VERBETERING +.350₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.563₁₀- 0 +.497₁₀- 0

VERBETERING +.360₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.553₁₀- 0 +.487₁₀- 0

VERBETERING +.370₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.544₁₀- 0 +.478₁₀- 0

VERBETERING +.380₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.534₁₀- 0 +.468₁₀- 0

VERBETERING +.390₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.525₁₀- 0 +.459₁₀- 0

VERBETERING +.400₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.515₁₀- 0 +.450₁₀- 0

VERBETERING +.410₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.506₁₀- 0 +.441₁₀- 0

VERBETERING +.420₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.497₁₀- 0 +.432₁₀- 0

VERBETERING +.430₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.488₁₀- 0 +.424₁₀- 0

VERBETERING +.440₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.479₁₀- 0 +.415₁₀- 0

VERBETERING +.450₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.470₁₀- 0 +.407₁₀- 0

VERBETERING +.460₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.461₁₀- 0 +.399₁₀- 0

VERBETERING +.470₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.452₁₀- 0 +.391₁₀- 0

VERBETERING +.480₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.443₁₀- 0 +.383₁₀- 0

VERBETERING p m k pred pc

VERBETERING +.490₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.434₁₀- 0 +.376₁₀- 0

VERBETERING +.500₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.426₁₀- 0 +.368₁₀- 0

VERBETERING +.510₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.417₁₀- 0 +.361₁₀- 0

VERBETERING +.520₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.409₁₀- 0 +.354₁₀- 0

VERBETERING +.530₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.401₁₀- 0 +.347₁₀- 0

VERBETERING +.540₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.393₁₀- 0 +.340₁₀- 0

VERBETERING +.550₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.385₁₀- 0 +.333₁₀- 0

VERBETERING +.560₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.377₁₀- 0 +.327₁₀- 0

VERBETERING +.570₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.369₁₀- 0 +.320₁₀- 0

VERBETERING +.580₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.361₁₀- 0 +.314₁₀- 0

VERBETERING +.590₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.354₁₀- 0 +.308₁₀- 0

VERBETERING +.600₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.347₁₀- 0 +.302₁₀- 0

VERBETERING +.610₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.339₁₀- 0 +.296₁₀- 0

VERBETERING +.620₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.332₁₀- 0 +.290₁₀- 0

VERBETERING +.630₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.325₁₀- 0 +.284₁₀- 0

VERBETERING +.640₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.318₁₀- 0 +.279₁₀- 0

VERBETERING +.650₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.311₁₀- 0 +.273₁₀- 0

VERBETERING +.660₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.304₁₀- 0 +.268₁₀- 0

VERBETERING +.670₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.298₁₀- 0 +.262₁₀- 0

VERBETERING +.680₁₀+ 2 +.400₁₀+ 1 +.300₁₀+ 1 +.291₁₀- 0 +.257₁₀- 0

$+.121_{10}+3$ $+.400_{10}+1$ $+.300_{10}+1$ $+.812_{10}-1$ $+.892_{10}-1$
$+.122_{10}+3$ $+.400_{10}+1$ $+.300_{10}+1$ $+.792_{10}-1$ $+.875_{10}-1$
$+.123_{10}+3$ $+.400_{10}+1$ $+.300_{10}+1$ $+.772_{10}-1$ $+.857_{10}-1$
$+.124_{10}+3$ $+.400_{10}+1$ $+.300_{10}+1$ $+.752_{10}-1$ $+.840_{10}-1$
$+.125_{10}+3$ $+.400_{10}+1$ $+.300_{10}+1$ $+.733_{10}-1$ $+.824_{10}-1$
$+.126_{10}+3$ $+.400_{10}+1$ $+.300_{10}+1$ $+.714_{10}-1$ $+.807_{10}-1$
$+.127_{10}+3$ $+.400_{10}+1$ $+.300_{10}+1$ $+.696_{10}-1$ $+.792_{10}-1$
$+.128_{10}+3$ $+.400_{10}+1$ $+.300_{10}+1$ $+.678_{10}-1$ $+.776_{10}-1$
$+.129_{10}+3$ $+.400_{10}+1$ $+.300_{10}+1$ $+.661_{10}-1$ $+.761_{10}-1$
$+.130_{10}+3$ $+.400_{10}+1$ $+.300_{10}+1$ $+.644_{10}-1$ $+.745_{10}-1$
$+.131_{10}+3$ $+.400_{10}+1$ $+.300_{10}+1$ $+.628_{10}-1$ $+.731_{10}-1$
$+.132_{10}+3$ $+.400_{10}+1$ $+.300_{10}+1$ $+.612_{10}-1$ $+.716_{10}-1$
$+.133_{10}+3$ $+.400_{10}+1$ $+.300_{10}+1$ $+.596_{10}-1$ $+.702_{10}-1$
$+.134_{10}+3$ $+.400_{10}+1$ $+.300_{10}+1$ $+.581_{10}-1$ $+.688_{10}-1$
$+.135_{10}+3$ $+.400_{10}+1$ $+.300_{10}+1$ $+.566_{10}-1$ $+.675_{10}-1$
$+.136_{10}+3$ $+.400_{10}+1$ $+.300_{10}+1$ $+.551_{10}-1$ $+.661_{10}-1$
$+.137_{10}+3$ $+.400_{10}+1$ $+.300_{10}+1$ $+.537_{10}-1$ $+.648_{10}-1$
$+.138_{10}+3$ $+.400_{10}+1$ $+.300_{10}+1$ $+.523_{10}-1$ $+.635_{10}-1$
$+.139_{10}+3$ $+.400_{10}+1$ $+.300_{10}+1$ $+.510_{10}-1$ $+.623_{10}-1$