(19) DANMARK

(12) **PATENTSKRIFT** (11) **167828 B1**

Patentdirektoratet
TAASTRUP

(21) Patentansøgning nr.: **5820/87** (51) Int.Cl.5 **G 06 F 15/336**

(22) Indleveringsdag: **05 nov 1987**

(24) Løbedag: **06 mar 1987**

(41) Alm. tilgængelig: **05 nov 1987**

(45) Patentets meddelelse bkg. den: **20 dec 1993**

(86) International ansøgning nr.: **PCT/US87/00480**

(86) International indleveringsdag: **06 mar 1987**

(85) Videreførelsesdag: **05 nov 1987**

(30) Prioritet: **07 mar 1986 US 837260**

(73) Patenthaver: ***Adler Research Associates; 2333 Morris Avenue; Union; NJ 07083, US**

(72) Opfinder: **George *Caravannis; GR, Christos *Halkias; GR, Dimitris *Manolakis; US, Elias *Koukoutsis; GR**

(74) Fuldmægtig: **Budde, Schou & Co. A/S**

(54) **Optimal parametrisk signalprocessor og behandlingsfremgangsmåde i en signalprocessor**

(56) Fremdragne publikationer

(57) Sammendrag:

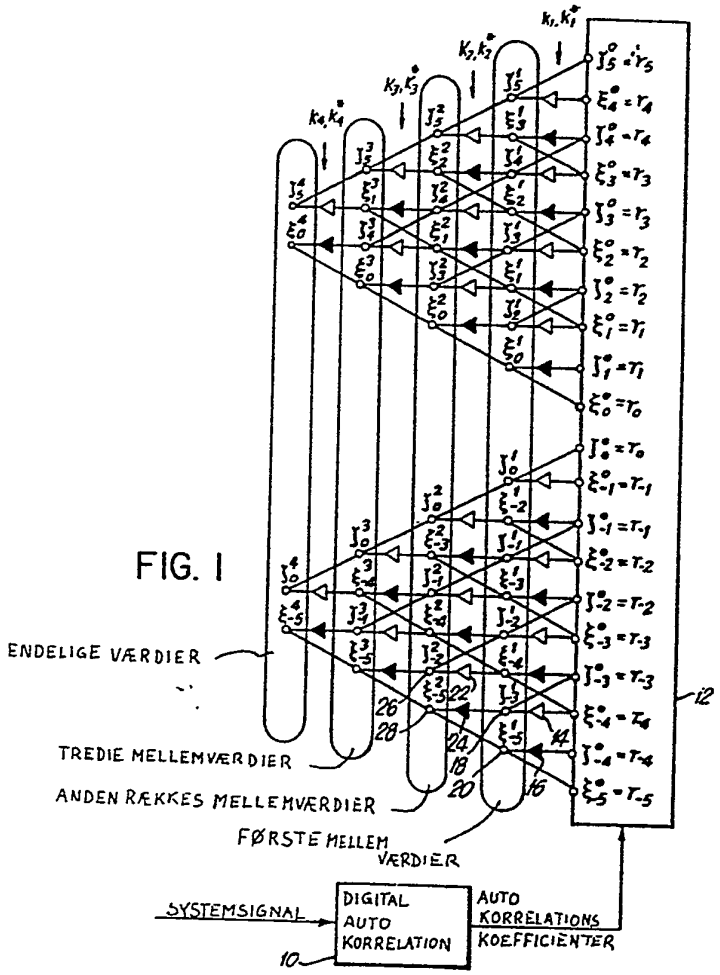
5820-87

En optimal parametrisk signalprocessor, som modtager autokorrelationskoefficienter, tilvejebringer gitterkoefficienter på optimal vis og muliggør anvendelsen af et hvilket som helst antal tilgængelige, parallelle behandlingsenheder. Signalprocessoren kan indsættes på fuld parallelvis og på fuld seriel vis, eller på en "parallel-opdelt" vis, hvorved frembringes fordelene ved parallel behandling, overskuelig materielkompleksitet og optimal signal behandling for et udpeget antal tilgængelige processorer.

DK 167828 B1

fortsættes

FIG. I


ENDELIGE VÆRDIER
TREDIE MELLEMVÆRDIER
ANDEN RÆKKES MELLEMVÆRDIER
FØRSTE MELLEM VÆRDIER
SYSTEMSIGNAL
DIGITAL AUTO KORRELATION
AUTO KORRELATIONS KOEFFICIENTER
10
12
14
16
18
20
22
24
26
28

Opfindelsen angår parametrisk signalbehandling. Parametrisk signalbehandling anvendes inden for mange områder, såsom tale- og billedanalyse, syntese og genkendelse, neurofysik, geofysik, behandling af datagrupper, datamatbehandlet tomografi, kommunikation og astronomi, for blot at nævne nogle områder.

Et eksempel på signalbehandling af særlig vigtighed er den lineære forudsigelsesteknik, som anvendes ved taleanalyse, syntese og genkendelse, og til behandling af seismiske signaler til fremme af rekonstruktion af geofysiske substrata. Den lineære forudsigelsesteknik anvender en særlig autokorrelationsfunktion.

En anden form for signalbehandling, som virker ved mange anvendelser, er fastlæggelsen af et optimalt (ud fra de mindste kvadraters synspunkt) endeligt impulssvarfilter. En signalprocessor, som anvender en sådan teknik, arbejder med autokorrelation af filterindgangssignalet, og krydskorrelation mellem indgangssignal og det udpegede svarsignal, hvilket kan virke i tilknytning til mange af de foran nævnte anvendelsesområder.

En anden form for signalbehandling af særlig betydning er inden for fagområdet kendt som "L-trin foran" forudsigelse og filtrering, til løsning af det "optimale forsinkelses"-problem. Denne teknik er især anvendelig ved beregning af spids- og formfiltre. Signalprocessorer som udøver denne funktion virker med en særlig autokorrelationsfunktion, som også tager hensyn til en tidsforsinkelse, som er knyttet til anlægget.

I almindelighed vil, ved stigende orden af det system, som undersøges, kompleksiteten af den nødvendige signalbehandling til frembringelse af brugbar information også forøges. F.eks. kan et system af p'te orden, ved anvendelse af den almindelige gaussiske eliminationsprocedure, blive behandlet i "$O(p^3)$" trin, hvilket angiver, at antallet af trin er "i størrelsesordenen" p^3, dvs. en

funktion af p^3. Det vil således kunne indses, at et system af en orden på p=100 udkræver behandlingstrin i størrelsesordenen en million til behandling af signalet, hvilket er en umiddelbar væsentlig begrænsning, især tilknytning til tidstro behandling.

Der er blevet udviklet signalbehandlingsteknikker, hvorved antallet af nødvendige operationer til behandling af et signal er blevet reduceret. En sådan måde er baseret på en teknik, som er udviklet af N. Levinson, hvortil udkræves $O(p^2)$ efter hinanden følgende operationer til behandling af signalet. Især kræver "Levinson's teknik" $O(2 \cdot p^2)$ efter hinanden følgende operationer til behandling af signalet. En forbedret udgave af denne teknik, som kendes som "Levinson-Durbin"--teknikken kræver $O(1 \cdot p^2)$ efter hinanden følgende operationer til behandling af signalet. Ingen af disse fremgangsmåder er egnet til parallelbehandling. En almindelig behandling af Levinsons og Levinson-Durbin-teknikkerne er omtalt i J. Math. Phys., bind 25, januar 1947, N. Levinson, "The Wiener RMS (Root-Mean-Square) Error Criterion in Filter Design and Prediction", side 261-278, og i Rev. Int. Statist. Inst., bind 28, 1960, J. Durbin, "The Filtering of Time Series Models", side 233-244.

Skønt de repræsenterer en forbedring af størrelsesordenen i forhold til den gaussiske eliminationsteknik, er Levinson og Levinson-Durbin-teknikkerne for langsomme for mange sammensatte systemer, hvor der kræves tidstro behandling.

En anden måde at anvende hovedgentagelsen af Levinson-Durbin-teknikken til beregning af, hvad der i almindelighed benævnes "gittercoefficienter", blev udviklet af Schur i 1917 for tilvejebringelse af et kriterium for systemstabilitet. J. Reine Angewandte Mathematik, bind 147, 1917, I. Schur, "Über Potenzreihen Die Im innern Des Einheitskreises Beschrankt Sind", side 205-232. Lev-Ari og Kailath ved Stanford universitet har udviklet en anden

løsningsmetode, som er baseret på teknikkerne fra Schur og Levinson, hvorved tilvejebringes en trekantet "stige"--struktur til signalbehandling. Ved Lev-Ari's og Kailath's teknik anvendes selve signalet som indgangssignal til processoren, i stedet for autokorrelationskoefficienter, og det virker i signaludformningssammenhæng. Se IEEE International Conference on Acoustics, Speech and Signal Processing, 1981, H. Lev-Ar og T. Kailath, "Schur and Levinson Algorithms for Non-Stationary Processes", side 860-864.

I en anden udformning af Schur's teknik, genafleder Le Roux og C. Gueguen Schuralgoritmen, idet der lægges vægt på implementeringen af den endelige ordlængde, under anvendelse af fastpunktaritmetik. Se IEEE Transactions on Acoustics, Speech, and Signal Processing, juni 1977, Le Roux og Geuguen, "A Fixed Point Computation of Partial Correlation, Coefficients", side 257-259.

Kung og Hu har udviklet en parallel plan, som er baseret på Schurteknikken, og hvortil anvendes et antal af parallelprocessorer, til behandling af et signal af p'te orden i O(p) operationer, hvilket er en væsentlig forbedring sammenlignet med Levison-Durbinteknikken. Se IEEE Transactions on Acoustics, Speech and Signal Processing, bind ASSP-31, nr. 1, februar 1983, Kung og Hu, "A Highly Concurrent Algorithm and Pipelined Architecture for Solving Toeplitz Systems", side 66-76. Imidlertid er anvendelsen af Kung eller Hu's teknik alvorligt begrænset, idet der udkræves et antal processorer lig med den systemorden, som skal løses. Kung og Hu's teknik kan således ikke virke ved behandling af et signal, som er tilvejebragt i et system med en orden større end antallet af parallelle processorer. Systemkompleksiteten er derfor en væsentlig begrænset faktor ved anvendelse af Kung og Hu's teknik, idet mange komplekse systemer er af en orden, som er meget større, end det antal parallelle processorer, som for øjeblikket er tilgængelige i moderne VLSI eller anden teknologi.

C.G. Carayannis et al. har beskrevet et signalbehandlingssystem, som indbefatter de træk, som er indeholdt i indledningen til kravene 1-4, og hvori anvendes en ren parallel behandling, hvor antallet af processorer er lig med systemets orden, og en ren rekursiv teknik, hvor der kun anvendes en enkelt processor. Se C.G. Carayannis et al., IEEE International Conference on Acoustics, Speech and Signal Processing, bind 4, marts 26-28, 1985, C.G. Carayannis et al., "A New Look on the Parallel Implementation of the Shur Algorithm for the Solution of Toeplitz Equations", side 1858-1861, især fig. 3 og 4.

Det er formålet med den foreliggende opfindelse at tilvejebringe en signalprocessor og en behandlingsfremgangsmåde, som kan virke under anvendelse af et mindre antal behandlingsenheder til behandling af et signal på "opdelt parallel" vis.

Formålet opnås med en signalprocessor ifølge den kendetegnende del af krav 1 og 3 og en fremgangsmåde ifølge den kendetegnende del af krav 2 og 4.

Krav 1 og 2 angår den symmetriske anvendelse og krav 3 og 4 angår den ikke-symmetriske anvendelse.

Ved opfindelsen er tilvejebragt en lineær forudsigelsessignalprocessor, som også danner basis for "L-trin forud"-processorer og endeligt impulssvarprocessorer baserede på de mindste kvadraters metode (LS-FIR processorer).

Opfindelsen forklares i det følgende nærmere under henvisning til tegningen, på hvilken:

Fig. 1 anskueliggør "overgitter"-behandlingsopbygningen i det ikke-symmetriske tilfælde, i overensstemmelse med den foreliggende opfindelse,

fig. 2 viser signalbehandlingsopbygningen i overensstemmelse med den foreliggende opfindelse, hvorved den direkte forudsigelseskoefficient a_i kan afledes fra gitterkoefficienten k_i, som er tilvejebragt ved den i fig. 1 viste opbygning,

fig. 3 viser "overgitter"-behandlingsopbygningen i det symmetriske tilfælde i overensstemmelse med den foreliggende opfindelsen,

fig. 4 viser en ved den foreliggende opfindelse tilvejebragt "grundcelle",

fig. 5 anskueliggør den gentagne brug af grundcellen i fig. 4 for tilvejebringelse af behandlingsopbygningen i fig. 1 og 3,

fig. 6 anskueliggør indsættelsen af overgitteropbygningen under anvendelse af tre grundceller i et system af 8. orden,

fig. 7 anskueliggør signalstrømmen gennem virkelige og virtuelle processorer, hvori anvendes opbygningen i fig. 6,

fig. 8 anskueliggør den materielmæssige indsættelse af den i fig. 6 og 7 viste opbygning i overensstemmelse med den foreliggende opfindelse, idet der er vist tre grundceller med dertil knyttet materiel.

I fig. 1 er vist en "overgitter"-opbygning af multiplikationskredse, som er angivet ved trekanter, og additionskredse, som er angivet ved cirkler, og som er opstillet på en sådan vis, at et signal kan behandles for tilvejebringelse af en lineær forudsigelsesstørrelse (prediktor) for et ikke-symmetrisk system. Overgitteropbygningen udgør også basis for L-trin forud og LS-FIR processorer.

Som vist i fig. 1 overføres et signal fra et system af P'ende orden, såsom et seismisk signal, til et kendt digitalt autokorrelationsled 10, som tilvejebringer autokorrelationskoefficienterne r_{-5} til r_5. Autokorrelationskoefficienterne overføres til en indgangsindretning 12, såsom et digitalt register, eller et bufferlager, for overføring til overgitteropbygningen. Med undtagelse af autokorrelationskoefficienterne r_{-5}, r_0 og r_5, overføres hver autokorrelationskoefficient til et par multiplikationskredse, som virker ved at multiplicere hver koefficient med et par "gitterkoefficienter" k_n og k_n^*, idet k_n er en almindelig gitterkoefficient, og k_n^* er den "tilgrænsende" gitterkoefficient. De multiplikationskredse, som er angivet med hvide trekanter, virker ved at multiplicere autokorrelationskoefficienterne med den almindelige gitterkoefficient k_n, medens de multiplikationskredse, som er angivet med de sorte trekanter, virker ved at multiplicere autokorrela-

tionskoefficienterne med de tilgrænsende gitterkoefficienter k_n^*. Herefter forklares frembringelsen af den almindelige og den tilgrænsende gitterkoefficient.

De to produkter, som er tilvejebragt ved multiplikation af hver autokorrelationskoefficient med den almindelige og den tilgrænsende gitterkoefficient adderes i de adderingskredse, som på fig. 1 er vist med cirkler, med et tilgrænsende par autokorrelationskoefficienter for tilvejebringelse af et sæt første mellemværdier J_n^1, hvor n = -3, -2, -1, 0, 2, 3, 4, 5, og ξ_m^1, hvor m = -5, -4, -3, -2, 0, 1, 2, 3. F.eks. multipliceres autokorrelationskoefficienten r_{-4} med k_1 og k_1^* i multiplikationskredsene henholdsvis 14 og 16, og produkterne adderes til autokorrelationskoeffientparret r_{-3} og r_{-5}, som grænser op til koefficienten r_{-4}, i additionskredsene henholdsvis 18 og 20. På tilsvarende vis adderes autokorrelationskoefficienten r_{-3} efter multiplikation med gitterkoefficienterne k_1 og k_1^* til hver af autokorrelationskoefficienterne r_{-4} og r_{-2}. Den samme proces udøves for autokorrelationskoefficienterne r_{-2} til r_4, for tilvejebringelse af det viste første sæt mellemværdier.

Af hensyn til sammenhængen benævnes også korrelationskoefficienterne r_{-5} til r_5 og for J_n^0 og ξ_m^0, hvor n = -4 til 5 og m = -5 til 4.

Gitterkoefficienterne beregnes som følger:

$$k_{m+1} = - J_{m+1}^m / J_0^m$$

$$k_{m+1}^* = - \xi_{m+1}^m / \xi_0^m$$

Gitterkoefficienterne k_1 og k_1^* frembringes direkte fra autokorrelationskoefficienterne, medens det andet sæt af gitterkoefficienter, k_2 og k_2^* beregnes ud fra de første mellemværdier.

På tilsvarende vis, som ved frembringelsen af de første mellemværdier, multipliceres udpegede par grænsende op til de første mellemværdier, f.eks. ξ_{-4}^1 og J_{-3}^1 med den

almindelige og den tilgrænsende gitterkoefficient henholdsvis k_2 og k_2^*, i multiplikationskredsene 22 og 24. To første mellemværdier J_{-2}^1 og ξ_{-5}^1 som grænser op til og er beliggende på hver sin side af det udpegede par adderes til de produkter, som frembringes ved multiplikationskredsene 22 og 24, for tilvejebringelse af to andenrække mellemværdier J_{-2}^2 og ξ_{-5}^2. De øvrige andenrækkes mellemværdier frembringes på lignende vis, nemlig ved multiplikation af et udpeget par tilgrænsende de første mellemværdier med den normale og den tilgrænsende gitterkoefficient k_2 og k_2^* og ved påfølgende addering af produkterne med de første mellemværdier tilgrænsende og på hver sin side af det udpegede par.

Det ses også ved at følge den foreliggende signalstrøm, at de tredje mellemværdier og de endelige mellemværdier frembringes på tilsvarende vis.

Gitterkoefficienterne k_i repræsenterer fuldstændig den lineære prediktor og kan anvendes i stedet for de direkte prediktorkoefficienter. I virkeligheden foretrækkes de til lagring, transmission og hurtig talesyntese, eftersom de er tilvejebragt med de væsentlige fordele at være opstillet i rækkefølge, enhedsafgrænset, og de kan umiddelbart anvendes til stabilitetskontrol, effektiv kvantitering og lignende. Eftersom r_0 modsvarer signalets energi, og således vil være tilvejebragt med den største amplitude af samtlige de signaler, som behandles ved overgitteret, kan alle variable normaliseres i forhold til r_0, hvorved anvendelsen af "fastkomma"-behandling lettes med de dertil knyttede fordele, såsom nøjagtighed, hastighed og behandlingssimpelthed.

Den i fig. 1 viste opbygning kan tilvejebringe gitterkoefficienter for et system af typen $\underline{Ra} = \underline{-d}$, hvor R er en Toeplitz opbygning. En detaljeret analyse af denne teknik er omtalt i IEEE International Conference on Acoustics, Speech and Signal Processing, marts 26-29, 1985, F. Carayannis et al., "A New Look on the Parallel Implementation of the

Schur Algorithm for the Solution of Toeplitz Equations".

Selv om gitterkoefficienterne k_i og k_i^* almindeligvis foretrækkes, kan de direkte prediktorkoefficienter a_i, som er nyttige ved f.eks. spektral vurdering, afledes fra gitterkoefficienterne under anvendelse af den i fig. 2 viste behandlingsopbygning. Som vist overføres gitterkoefficienterne k_1 og k_1^* til et par multiplikationskredse 30 og 32, som virker ved at multiplicere disse gitterkoefficienter med andenrækkes gitterkoefficienter henholdsvis k_2 og k_2^*, hvorved tilvejebringes et første sæt produkter henholdsvis $a_{1,2}$ og $a_{1,2}^*$. Disse produkter multipliceres herefter henholdsvis i multiplikationskredsene 34 og 36 med k_3^* og k_3, hvorefter de adderes til henholdsvis k_2^* og k_2, hvorved tilvejebringes størrelserne $a_{2,3}^*$ og $a_{2,3}$. Tillige multipliceres størrelsen af gitterkoefficienterne k_2 og k_2^* med henholdsvis k_3^* og k_3 i multiplikationskredsene 38 og 40, og mellemværdierne $a_{1,2}^*$ og $a_{1,2}$ adderes til disse produkter for tilvejebringelse af yderligere mellemværdier henholdsvis $a_{1,3}^*$ og $a_{1,3}$. Denne proces fortsættes indtil de direkte filterkoefficienter $a_{1,8}$ til $a_{8,8}$ og $a_{1,8}^*$ til $a_{8,8}^*$ er tilvejebragt.

En særlig situation eksisterer, når det system, som analyseres, kan karakteriseres ved $\underline{Ra} = \underline{-d}$, hvor R har en symmetrisk Toeplitzopbygning, således som det er tilfældet ved autoregressiv lineær forudsigelse. I sådanne tilfælde forenkles overgitteropbygningen i fig. 1 til den symmetriske overgitterform, som er vist i fig. 3, eftersom $r_i = r_{-i}$, $k_i = k_i^*$ og $\mathfrak{J}_i^m = \xi_{-1}^m$ i det symmetriske tilfælde. Herved bliver de to trekantede områder i fig. 1 identiske, således at et af dem kan udelades, hvorved halvdelen af signalbehandlingen kan undlades. Opbygningen af signalbehandlingen i det symmetriske tilfælde, således som vist i fig. 3, tilvejebringer lineær forudsigelse, eller autoregressiv udformning, ved beregning af gitterkoefficienter, eller i det symmetriske tilfælde ved beregning af "PARCOR"-(partiel korrelations)--koefficienter.

Det skal indledningsvis bemærkes, at der ved den foreliggende signalbehandlingsopbygning (såvel som ved den i fig. 1 viste opbygning) ikke er nogen redundans. Det vil sige, at hvert $\mathcal{J}$, som optræder i processoren, kun frembringes én gang. Herudover er kun de signaler indblandede, som er nødvendige til frembringelsen af gitterkoefficienter eller PARCOR. Den i fig. 3 viste signalbehandlingsopbygning (tillige med den i fig. 1 viste) repræsenterer således optimal behandlingsplanlægning.

Den fig. 3 viste signalprocessor kan indsættes, som forklaret under henvisning til fig. 4-8, nedenfor. For enkelthedens skyld begrænses denne forklaring til det symmetriske tilfælde. Imidlertid vil behandlingen af det ikke symmetriske tilfælde blive åbenbar ud fra forklaringen af det symmetriske tilfælde.

På en tilsvarende vis, som den for det ikke symmetriske tilfælde i fig. 1 viste, overføres et systemsignal til et digitalt autokorrelationsled 10, som frembringer autokorrelationskoefficienter, som er karakteristiske for det symmetriske system, nemlig r_0 til r_8. Koefficienterne overføres til en indgangsindretning 42, såsom et digitalt register eller lager. Bemærk, at signalprocessoren i fig. 3 modtager et signal fra et system af 8. orden, medens anlægget i fig. 1 modtager et signal fra et system af 5. orden.

Som det er tilfældet med signalprocessoren i fig. 1 optager signalprocessoren fig. 3 hver autokorrelationskoefficient, undtagen den første og den sidste, nemlig r_0 og r_8, multiplicerer den med den første gitterkoefficient k_1, som er beregnet ud fra r_0 og r_1 ($\mathcal{J}_0^0, \mathcal{J}_1^0$), i overensstemmelse med den almindelige formel $k_p = -\mathcal{J}_p^{p-1}/\mathcal{J}_0^{p-1}$. Produktet af hver sådan multiplikation adderes hver for sig til de tilgrænsende to autokorrelationskoefficienter for tilvejebringelse af de første mellemværdier $\mathcal{J}_n^1$, hvor n = 0, 2 til 8 og -6 til -1. Eksempelvis multipliceres autokorrelationskoefficienten r_1, som er benævnt som $\mathcal{J}_1^0$ og $\mathcal{J}_{-1}^0$, af hensyn til overensstemmelse med mellemværdierne, med gitterkoefficienten k_1,

og autokorrelationskoefficenterne r_0 og r_2 adderes hver for sig til produktet for tilvejebringelse af et par første mellemliggende værdier, henholdsvis J_0^1 og J_2^1. På tilsvarende vis tilvejebringes de næste to første mellemværdier, nemlig J_{-1}^1 og J_3^1, ved multiplikation af autokorrelationskoefficienten r_2 med gitterkoefficienten k_1 med påfølgende addition med tilgrænsende autokorrelationskoefficienter hver for sig, nemlig r_1 og r_3 med produkterne.

Mellemværdier af anden række kan beregnes ud fra de første mellemværdier på tilsvarende vis. Først kan k_2 beregnes fra forholdet mellem J_2^1 og J_0^1 i overensstemmelse med den den ovenfor anførte formel. Herefter kan andenrækkes mellemværdier, f.eks. J_3^2 og J_0^2, beregnes ved multiplikation af første mellemværdier J_{-1}^1 og J_2^1 med gitterkoefficienten k_2, hvorefter de tilgrænsende første mellemværdier J_3^1 og J_0^1 hver for sig adderes til produkterne. Signalbehandlingen fortsættes indtil de endelige værdier J_8^7 og J_0^7 er tilvejebragt, ud fra hvilke den sidste gitterkoefficient k_8 kan beregnes ud fra den tidligere nævnte formel.

Hvis direkte forudsigelseskoefficienter ønskes kan den i fig. 2 viste behandlingsopbygning anvendes i det symmetriske tilfælde såvel som i det ikke symmetriske tilfælde. Imidlertid er behandlingsopbygningen i fig. 2 noget forenklet i det symmetriske tilfælde, eftersom $k_n = k_n^*$.

Idet der atter henvises til fig. 3 skal flere punkter her bemærkes. Selv om tegningen er vist med et trekantet område, til anskueliggørelse af de nøjagtige detaljer af signalbehandlingen, vil det kunne indses, at de forskellige materielelementer, som multiplikationskredsene, som er vist med trekanter, og additionskredsene, som er vist med cirkler, kan sammensættes af et enkelt sæt med 14 multiplikationskredsene og 14 additionsskredse, som først virker ved frembringelse af de første mellemværdier, herefter andenrækkes mellemværdier, og så fremdeles indtil de endelige vær-

dier. Hvis 14 multiplikationskredse og additionskredse ikke er til rådighed kan yderligere et mindre antal virke sammen inden for en hvilken som helst gruppe mellemværdier.

I fig. 4 er anskueliggjort en "grundcelle" 44, som indbefatter et par multiplikationskredse 46 og et par additionskredse 48. Som vist i fig. 4 virker grundcellen 44 ved at frembringe et signal $e = a + k_i \cdot b$, og et signal $f = d \mp k_i \cdot c$.

Grundcellen kan tilvejebringes med én "to-cyklus" processor, eller to "én-cyklus" processor, for frembringelse af signaler e og f. Ved anvendelse af den i fig. 4 viste grundcelle tilvejebringes en homogen opbygning med materielelementer, idet det er tilstrækkeligt, at gentage den samme grundcelle i opbygningen for frembringelse af den samlede overgitteropbygning.

I fig. 5 er vist tre grundceller angivet henholdsvis med fuldt optrukne, punkterede og prikkede linier. Ved udelukkende at gentage den i fig. 4 viste behandlingsenhed kan et overgitter af en hvilken som helst størrelse opbygges til behandling af et signal af praktisk taget en hvilken som helst kompleksitet. Eksempelvis vil det i fig. 5 viste overgitter kunne behandle et signal fra et system af 3. orden. Til behandling af et signal fra et system af 4. orden kan en første yderligere grundcelle placeres på "toppen" grundcelle nr. 2 for tilvejebringelse af første mellemværdier e_3 og f_3 (ikke vist). En anden yderligere grundcelle kan på tilsvarende vis tilføjes til modtagelse af indgangssignalerne e_2, f_2, e_3 og f_3 til frembringelse af udgangssignalerne h_2, i_2 (ikke vist). Endelig kan en trejde yderligere grundcelle anvendes til modtagelse af indgangssignalerne h_1, i_1, h_2 og i_2 og til frembringelse af udgangssignaler j_1 og l_1 (heller ikke vist).

Med de seneste fremskridt inden for VLSI-teknologien er der i tilknytning til den foreliggende opfindelse mulighed for at drage fordel af tilstedeværelsen af en eller flere processorer i omgivelser med flere processorer, hvori hver parallelprocessor (eller processorpar) virker som en "grund-

celle".

Idet der henvises til fig. 3 defineres en "fuld parallel" implementering, som en implementering, hvori alle første mellemværdier frembringes i hovedsagen samtidigt, i parallel, hvorefter på et senere tidspunkt alle mellemværdier i anden række frembringes i parallel og så fremdeles, indtil signalparametrene har gennemløbet det samlede overgitter. Ved den fuldstændige parallelle implementering af overgitteret tilvejebringes optimal signalbehandlingshastighed.

For en fuldstændig parallel implementering skal i det mindste p-1 grundceller være tilvejebragt til et system af p'ende orden. Ud fra fig. 3 kan det ses, at der udkræves syv grundceller til et system af 8. orden. Under drift overføres autokorrelationskoefficienterne til indgangsindretningen 42, såsom et digitalt register eller et bufferlager, og påtrykkes de syv grundceller hovedsagelig samtidigt, for tilvejebringelse af sættet med de første mellemværdier. Disse mellemværdier "tilbagekobles" til indgangsindretningen 42 for genoverføring til behandlingsenhederne. Eftersom der kun er fjorten første mellemværdier sammenholdt med, at der frembringes seksten indgangsværdier fra autokorrelationskoefficienterne i det andet trin, kan imidlertid kun seks af de syv grundceller anvendes, i stedet for at alle syv anvendes, til frembringelse af sættet af mellemværdier af anden række. Anden rækkes mellemværdier "tilbagekobles" på tilsvarende vis til indgangsindretningen 42 og genoverføres til fem grundceller for tilvejebringelse af sættet med de tredje mellemværdier, og så fremdeles, indtil sættet med sjette mellemværdier er "tilbagekoblet" til en enkelt grundcelle, for tilvejebringelse af de endelige værdier.

Det vil kunne ses, at en sekvens med p-1 parallelle trin udkræves til beregning af PARCOR'ene. Med andre ord er signalprocessorens kompleksitet af p'ende orden og der kræves p-1 behandlingsenheder til implementeringen af den fuldstændige parallelle teknik. I forhold til den af Kung og

Hu foreslåede teknik, som ovenfor citeret, er dette fordelagtigt, idet deres teknik kræver flere behandlingselementer. Et datamatprogram, som er skrevet i Pascal til simulering af fuld parallel implementering, er vedlagt som Appendix 1.

Ud fra fig. 5 vil det kunne indses, at den samlede overgitteroverbygning af signalprocessoren ifølge fig. 3 kan opbygges under anvendelse af en enkelt grundcelle som "byggeblok". Eksempelvis kan den i fig. 4 viste grundcelle først virke som grundcelle 1 i fig. 5 til behandling af størrelserne a1, b1, c1 og d1 til frembringelse af mellemværdier e1 og f1, hvorfra en gitterkoefficient kan beregnes. Imidlertid må grundcellen i fig. 4 for tilvejebringelse af den næste gitterkoefficient virke som grundcelle 2 i fig. 5. Som grundcelle 2 behandler den størrelserne a2, b2, c2 og d2 for tilvejebringelse af mellemværdierne e2 og f2. Størrelserne a2 og b2 er i virkeligheden de samme størrelser, som to af indgangsstørrelserne til den første grundcelle, nemlig c1 og d1.

Endelig anvendes den i fig. 4 viste grundcelle, som grundcelle 3 i fig. 5 til beregning af h1 og i1 ud fra e1, f1, e2 og f2, og den sidste gitterkoefficient kan beregnes ud fra h1 og i1.

Det vil kunne indses, at et overgitter af en hvilken som helst størrelse kan opbygges ud fra en enkelt grundcelle til fuldstændig behandling af autokorrelationskoefficienterne knyttet til et system af praktisk taget en hvilken som helst orden.

Ved implementering af ordensrekursivitetsteknikken overføres, under henvisning til fig. 3, autokorrelationskoefficienterne fra det digitale autokorrelationsled 10 til indgangsindretningen 42 og lagres heri. Den første grundcelle frembringer først J_0^1 og J_2^1 ud fra autokorrelationskoeffficienterne r_0, r_1 og r_2 og gitterkoefficienten k_1, som på sin side er beregnet ud fra r_1 og r_0. Størrelserne k_1, J_0^1 og J_2^1 lagres til senere brug. Herefter kan k_2 frembringes

fra $\mathfrak{J}_0^1$ og $\mathfrak{J}_2^1$, og k_2, $\mathfrak{J}_0^1$ og $\mathfrak{J}_2^1$ lagres herefter til yderligere brug. Den enkelte grundcelle vil herefter frembringe $\mathfrak{J}_{-1}^1$ og $\mathfrak{J}_3^1$ ud fra autokorrelationskoefficienterne r_1, r_2 og r_3. Disse størrelser tilbagekobles til indgangen på grundcellen sammen med størrelserne $\mathfrak{J}_0^1$ og $\mathfrak{J}_2^1$ for frembringelse af $\mathfrak{J}_0^2$ og $\mathfrak{J}_3^2$, ud fra disse størrelser og k_2. På dette punkt kan den næste gitterkoefficient, k_3 frembringes ud fra $\mathfrak{J}_0^2$ og $\mathfrak{J}_3^2$ og lagres sammen med de øvrige gitterkoefficienter.

Denne rekursive "tilbagekobling" af mellemvariable til grundcellen til frembringelse af yderligere mellemværdier, hvorved igen frembringes yderligere gitterkoefficienter, gentages indtil det samlede signal er behandlet i overensstemmelse med den i fig. 3 viste logiske opbygning.

Eftersom der i dette tilfælde kun er anvendt én grundcelle vil en signalprocessor, som anvender den ordensrekursive implementering i egentlig forstand være et serielt apparat, hvortil udkræves p(p-1) maskincyklus (i ordenen kvadratet på p) til behandling af et signal fra et system med p'ende orden. Eftersom den enkelte grundcelle er den eneste tilvejebragte datamatopbygning er den imidlertid meget let at indsætte. Yderligere er, i det tilfælde hvor PARCOR'ene udkræves i stedet for de direkte filterkoefficienter, den ved den foreliggende opfindelse tilvejebragte ordensrekursive teknik noget hurtigere og meget mere enkel, end Levison-Durbin-teknikken, meden LeRoux-Gueguen-teknikken er af samme kompleksitet.

Et datamatprogram udskrevet i Pascal, til simulering den ordensrekursive signalprocessor er vist i det som bilag vedlagte Appendix 2.

Som det vil kunne indses er den fulde parallelle implementering den anvendelse, hvormed tilvejebringes den hurtigste signalbehandling, således som tidligere forklaret. Imidlertid har denne teknik den ulempe, at der udkræves næsten lige så mange grundceller (p-1), som ordenen af det system, som undersøges

(p). I tilknytning til systemer af en meget høj orden vil fuld parallel implementering således ikke altid være opnåelig på grund af mangel på det tilstrækkelige antal processorer. Tillige kan en fuld parallel implementering i visse tilfælde være uønsket ud fra den højere grad af økonomi, som opnås gennem anvendelsen af færre processorer. For at gå til den anden yderlighed vil den enkleste måde at implementere signalprocessoren være, at anvende en enkelt grundcelle i tilknytning ordensrekursivitets-teknikken. Imidlertid har denne anvendelse den ulempe, at den er langsom, idet den kræver operationer i størrelsesordenen p^2 til fuldstændig behandling af signalet.

Ved den foreliggende opfindelse er der imidlertid tilvejebragt et kompromis mellem de bindinger, som tilvejebringes i tilknytning til materielkompleksitet og behandlingshastighed ved anvendelse af "opdelinger", som "skærer igennem" overgitter opbygningen i parallel, men som indsættes på tidsseriel vis.

Kort fortalt gøres der ved opdelt, parallel implementering brug af et antal grundceller, idet antallet af grundceller er mindre end p-1, hvor p er systemordenen. Ved at udnytte tilstedeværelsen af et antal parallelle processorer i et enkelt system, såsom en VLSI chip, kan signalprocessoren i overensstemmelse med den foreliggende opfindelse behandle signaler tilknyttet praktisk talt et hvilket som helst system af en hvilken som helst orden hurtigere end nogen kendt signalprocessor, med undtagelse af den af Kung og Hu foreslåede. Imidlertid kræver, som tidligere forklaret anvendelsen af Kung og Hu's fremgangsmåde brugen af lige så mange processorer, som systemets orden, et krav, som det ofte vil være umuligt at tilgodese. Yderligere er fuld parallel implementering ligeså hurtig som Kung og Hu's teknik.

Signalstrømmen gennem overgitteret ved opdelt parallel implementering forklares i det følgende under henvisning til

fig. 6, som anskueliggør en overgitteropbygning til behandling af et signal fra et system af 8. orden. Signalet behandles under anvendelse af kun tre grundceller, nemlig 50, 52 og 54, som behandler autokorrelationskoefficienterne r0 til r4 til frembringelse af sættet af første mellemværdier J_n^1, n = -2, -1, 0, 2, 3 og 4.

Disse første mellemværdier "kobles tilbage" til indgangen af to af grundcellerne, f.eks. 52 og 54, og andenrækkes mellemværdier J_n^2, n = 1, 0, 3 og 4, frembringes. Disse andenrækkes mellemvariable "tilbagekobles" til kun en enkelt grundcelle, f.eks. 54, og behandles for tilvejebringelse af tredje mellemværdier J_0^3 og J_4^3. Indtil dette tidspunkt har de tre grundceller 50, 52 og 54 virket i parallel under anvendelse af den ved den foreliggende opfindelse tilvejebragte teknik, som vist i fig. 3, men kun for en del, eller en "opdeling" af overgitteret, med en bredde modsvarende antallet af det antal parallelle grundceller, som er blevet anvendt. Denne opdeling vil i det efterfølgende blive benævnt "opdeling 1". Efter behandling af opdeling 1 tilvejebringes der til grundcellerne 50, 52 og 54 autokorrelationskoefficienterne r_3-r_7 til iværksættelse af en anden opdeling, som i fig. 6 er benævnt "opdeling 2". Grundcellerne behandler autokorrelationskoefficienterne til frembringelse af de første mellemværdier J_n^1, n = -3, -4, -5, 5, 6 og 7, hvilke størrelser "tilbagekobles" til grundcellerne 50, 52 og 54 sammen med to af de første mellemværdier J_{-2}^1 og J_4^1, som er tilvejebragt og lagret under behandlingen af den første opdeling. Grundcellerne 50, 52 og 54 behandler herefter disse størrelser i parallel for tilvejebrinelse af anden rækkes mellemværdier J_n^2, n = -4, -3, -2, 5, 6 og 7. På tilsvarende vis bliver disse størrelser "tilbagekoblet" til grundcellerne 50, 52 og 54 sammen med to af anden rækkes mellemværdier, J_{-1}^2 og J_4^2, som er frembragt og lagret under den første opdeling, og grundcellerne behandler disse størrelser i parallel for derfra at aflede de tredje mellemværdier, J_n^3, n = 3, -2,

-1, 5, 6 og 7. De tredje mellemværdier "kobles tilbage" til grundcellerne sammen med de tredje mellemværdier, J_0^3 og J_4^3, som er frembragt og lagret under opdeling 1, for tilvejebringelse af fire mellemværdier J_n^4, n = -2, -1, 0, 5, 6 og 7. Eftersom der ikke var frembragt nogle flere mellemværdier under opdeling 1, er der kun seks fjerde mellemværdier tilgængelig og disse overføres til to af grundcellerne, f.eks. 52 og 54. De seks værdier behandles heri i parallel for tilvejebringelse af femte mellemvariable J_n^5, n = -1, 0, 6 og 7, som igen "kobles tilbage" til en af grundcellerne, 54 f.eks., for således at tilvejebringe de sjette mellemvariable J_7^6 og J_0^6. Det vil kunne indses, at under behandlingen af de to første opdelinger er de gitterkoefficienter, som er blevet frembragt (k1 - k7) ligeledes lagret.

Det vil også kunne indses, at under behandlingen af opdelingerne 1 og 2 virker grundcellerne 50, 52 og 54 ved at "skære igennem" overgitteret i parallel for tilvejebringelse af gitterkoefficienterne k_2-k_4 i afdeling 1, og herefter på rekursiv vis at "skære igennem" overgitteret i parallel for tilvejebringelse af gitterkoefficienterne k_5-k_7 i opdeling 2. Herefter mangler kun behandlingen af den sidste gitterkoefficient k_8, hvilket udøves i en tredje opdeling, "opdeling 3". Eftersom systemet er af ottende orden kræves der til behandlingen af opdeling 3 kun en enkelt grundcelle, hvilken kan være en hvilken som helst af cellerne 50, 52 eller 54.

Især overføres autokorrelationskoefficienterne r_6-r_8 til den udpegede grundcelle og behandles her for tilvejebringelse af de første mellemværdier J_8^1 og J_{-6}^1. Disse værdier "kobles tilbage" til grundcellen sammen med to af de første mellemværdier, nemlig J_7^1 og J_{-5}^1, som er tilvedjebragt og lagret under opdeling 2. Disse størrelser behandles for tilvejebringelse af anden rækkes mellemværdier J_8^2 og J_{-5}^2. Den ovenfor omtalte proces gentages indtil afslutningen af opdeling 3, på hvilket tidspunkt den sidste gitterkoefficient k_8 kan frembringes.

Når således det ved den foreliggende opfindelse tilvejebragte overgitter virker under anvendelse af opdelt parallel implementering er den effektive signalstrøm fuldstændig parallel inden for en opdeling, hvorved tilvejebringes hurtig behandling skønt rekursiv eller seriel, fra opdeling til opdeling, idet der er muliggjort behandling af et signal fra et system af meget høj orden under anvendelse af et begrænset, egnet antal grundceller.

Herefter forklares et særligt eksempel på delvis opdelt implementering under henvisning til fig. 7 og 8. Som vist i fig. 7 er signalprocessoren indsat under anvendelse af tre parallelle processorer, 56, 58 og 60, benævnt "ægte processorer", hvilke indbefatter grundcellerne 50, 52 og 54 i fig. 6. Som tidligere nævnt kan hver af processorerne 56, 58 og 60 anvende en grundcelle enten som en "to-cyklus" processor eller som to "en-cyklus" processorer. Eftersom grundcellerne kun skal virke ved multiplikation og addition er de virkeligt egnede til at virke som billige og enkle "reducerede undervisningssæt" (RIS) processorer. I fig. 7 er tillige vist et antal processorer, benævnt "virtuelle processorer", hvis eksistens simuleres for at kunne behandle et indgangssignal i overensstemmelse med den i fig. 6 viste overgitteropbygning.

I fig. 8 er de parallelle processorer 56, 58 og 60 vist indbefattende grundcellerne henholdsvis 50, 52 og 54 i fig. 6. Hver processor indbefatter også et "a, b register", et "c, d register", et "e, f register" og et "k register", som virker ved at lagre de enkelte størrelser a, b, c, d, e og f, som er knyttet til hver grundcelle (se fig. 4), og størrelserne k, som er knyttet til gitterkoefficienterne. Et "r bufferlager" 62 er tilvejebragt til modtagelse af autokorrelationskoefficienterne $r_0 \ldots\ldots r_p$ og til kommunikation med processorerne 56, 58 og 60 ved en "r bus" 64. Hver af a, b registrene og c, d registrene i hver af processorerne 56, 58 og 60 modtager autokorrelationskoefficienter direkte

fra r bus 64. Et "k bufferlager" 66 tilvejebringer gitterkoefficienter k til hver af k registrene i processorerne 56, 58 og 60 over k bus 68. Divisionskredse og dertil knyttede logiske elementer 70, 72 og 74 er tilknyttet processorerne henholdsvis 56, 58 og 60 og virker ved at modtage et udgangssignal fra i, f register i den tilknyttede processor, frembringe dertil knyttede gitterkoefficienter og overføre dem til k registrene i processorerne og k bufferlageret 66 over k bus 68. Herudover er divisionskreds og logikelement 70, som er tilknyttet den nederste processor 56 også forbundet med r bus 64 over bus 76 i tilknytning til den særlige beregningsfunktion for k_1, den første gitterkoefficient, ud fra autokorrelationskoefficienterne r_0 og r_1.

Selv om det er vist, at der til hver enkelt af processorerne 56, 58 og 60 er tilvejebragt en særlig divisionskreds 70, 72 og 74, ville en enkelt divisionskrds med dertil knyttede logikelementer være tilstrækkelig til den samlede signalprocessor, og det forudses, at når en udførelsesform af den foreliggende opfindelse tilvejebringes i en kundeudformet VLSI kreds, vil der kun være tilvejebragt en enkelt divisionskreds. På den anden side kan der, i det tilfælde, hvor almindeligt lagerførte, kommercielle processorer anvendes, være tilvejebragt divisionskredse, som vist, hvilke kredse i virkeligheden kan udgøre en del af hver af processorerne 56, 58 og 60. I sidstnævnte tilfælde er anvendelsen af alle divisionskredsene ikke nødvendig, og de udkrævede divisioner kan udøves af en hvilken som helst af processorerne.

Det ses, at inden for hver processor tilvejebringer a, b registeret, c, d registeret og k registeret hver især signaler til de dertil knyttede grundceller, medens udgangssignalet fra hver grundcelle overføres til det tilknyttede e, f, register. Hvert e, f register tilvejebringer et udgangssignal til det dertil knyttede c, d register og divisionskredsen.

Mellem processorerne overføres udgangssignalet fra e, f registeret i den nederste processor 56 til a, b registeret i den midterste processor 58, medens udgangssignalet fra e, f registeret i den midterste processor 58 overføres til a, b registeret i den øverste processor 60. Udgangssignalet for e, f registeret i den øverste processor 60 overføres på tilsvarende vis til indgangen på a, b registeret i deres nederste processor 56 ved hjælp af et kantbufferlager 78.

Herefter forklares virkningen af de i fig. 8 viste materielelementer for frembringelse af den overgitterbehandlingsopbygning, som er vist i fig. 6 og fig. 7.

Fase 0 - Indledningsvis divison (k_1 beregning): Den til den nederste processor knyttede divisionskreds modtager de første to korrelationskoefficienter fra r-bufferlageret og frembringer k_1. Alle andre elementer er ubeskæftigede.

Fase 1 - Iværksættelse af opdeling 1: a, b og c, d registrene i alle processorer aktiveres gennem r-bus med de egnede autokorrelationskoefficienter.

Fase 2 - Grundcelleberegninger: Grundcellerne i alle processorer beregner mellemværdier og lagre dem i e, f registrene i hver processor.

Fase 3 - k_2 beregning - lagring - transmission: Divisionskredsen i den nederste processor beregner k_2 ud fra e, f registeret i denne processor. k_2 er lagret i k-bufferlageret og overføres til k-registeret i hver processor gennem k-bus'en.

Fase 4 - Aktivering af a, b og c, d registre, opdatering af kantbufferlager: Følgende overførsler finder sted:

e, f i den nederste processor ⟶ a, b i den midterste processor.

e, f i den midterste processor ⟶ a, b i den øverste processor.

e, f i den øverste processor ⟶ kantbufferlageret.

e, f i den midterste processor ⟶ c, d i den midterste processor.

e, f i den øverste processor ⟶ c, d i den øverste processor.

Fase 5 - Grundcelleberegninger med den nederste processor i tomgang:

Grundcellerne i det midterste og det øverste element frembringer mellemværdier, som lagres i de dertil svarende e, f registre.

Fase 6 - k_3 beregning - lagring - transmission:

Den til den midterste processor knyttede divisionskreds beregner k_3 ved tilførsel fra e, f registrene i denne processor. k_3 lagres i k-bufferlageret og overføres til k-registeret i hver processor gennem k-bus'en.

Fase 7 - Aktivering af a, b og c, d registre, opdatering af kantbufferlager:

Følgende overføringer finder sted:

e, f i den midterste processor ⟶ a, b i den øverste prosessor.

e, f i den øverste processor ⟶ kantbufferlager.

e, f i den øverste processor ⟶ c, d i den øverste processor.

Fase 8 - Beregninger i den øverste grundcelle med nederste og midterste processor ubeskæftigede:

Den øverste grundcelle beregner mellem værdier, som lagres i dens e, f registre.

Fase 9 - k_4 beregning - lagring:

Den til den øverste processor knyttede divisionskreds beregner k_4, overført fra e, f registrene i processoren. k_4 lagres i k-bufferlageret, men overføres ikke til k-registrene.

Fase 10 - k_1 retransmission:

k_1, som har været lagret i k-bufferlageret, overføres til k-registrene i alle processorer gennem k-bus'en.

Fase 11 - Iværksættelse af opdeling 2:

a, b og c, d registrene i alle processorer aktiveres gennem r-bus'en med de egnede autokorrelationskoefficienter.

Fase 12 - beregninger i grundcellerne:
Grundcellerne i alle processorer beregner mellemværdier og lagrer dem i e, f registrene i hver processor.

Fase 13 - Aktivering af a, b og c, d registre, opdatering af kantbufferlager:
Følgende overføringer finder sted:
e, f i nederste processor ⟶ a, b i den midterste processor.
e, f i den midterste processor ⟶ a, b i den øverste processor.
e, f i den nederste processor ⟶ c, d i processoren.
e, f i den midterste processor ⟶ c, d i processoren.
e, f i den øverste processor ⟶ c, d i processoren.
e, f i den øverste processor ⟶ kantbufferlager.

a, b registeret i den øverste processor skal aktiveres fra kantbufferlageret før indholdet af dette udslettes ved lagring af det nyligt beregnede indhold af e, f i den øverste processor.

Fase 14 - Beregning i grundcelle:
Som i fase 12.

Fase 15 - a, b, c, d aktivering, opdatering af kantbufferlager:
Som i fase 13.

Fase 16 - Beregning i grundcelle:
Som i fase 12.

Fase 17 - a, b, c, d aktivering opdatering af kantbufferlager:
Som i fase 13.

Fase 18 - Beregning i grundcelle:
Som i fase 12.

Fase 19 - k_5 beregning - lagring - transmission:
Som i fase 3, men k_5 beregnes - lagres - transmitteres i stedet for k_2.

Fase 20 - a, b, c, d aktivering, opdatering af kantbufferlager:
Som i fase 4, men e, f lagres i et andet register i kantbufferlageret.

Fase 21 - beregning i grundcelle med nederste processor ubeskæftiget:
Som i fase 5.

Fase 22 - k_6 beregning—lagring—transmission:
Som i fase 6, men k_6 beregnes—lagres—transmitteres i stedet for k_3.

Fase 23 - a, b, c, d aktivering, opdatering af kantbufferlager:
Som fase 7, men e, f i den øverste processor lagres i andre registre i kantbufferlageret.

Fase 24 - beregninger i den øverste grundcelle med nederste og midterste processor ubeskæftiget.
Som i fase 8.

Fase 25 - k_7 beregning-lagring:
Som i fase 9, men k_7 beregnes og lagres i stedet for k_4.

Fase 26 - k_1 retransmission:
Som i fase 10.

Fase 27 - iværksættelse af opdeling 3:
Som i fase 11.

Fortsættelsen gennem tredje opdeling vil kunne forstås af fagmanden inden for området ud fra det foregående forklarede.

Det vil kunne ses, at lagringen af mellemværdier over kantbufferlageret 78 i fig. 8 i fig. 7 er angivet med en lille cirkel i datavejen, når denne krydser den funktionsmæssige placering af kantbufferlageret 78. Det vil således kunne indses, at anvendelsen af parallelle processorer ved opdelt, parallel anvendelse frembringer en "diskontinuitet ved grænseoverskridelsen", hvilket fører til den opstilling af materiel, som er vist i fig. 7.

I den i fig. 6, 7 og 8 tilvejebragte eksempelvis udførelsesform er vist tre parallelle processorer til anskue-

liggørelse af virkningen af den opdelte, parallelle anvendelse, på grund af eksemplets generelle art. Opstillingen er tilvejebragt med manglende symmetri hvad angår processors funktionsudøvelse. Eksempelvis er den midterste processor 58 ikke forbundet med kantbufferlageret. Den øverste processor overfører data til kantbufferlageret, medens den nederste processor modtager data fra kantbufferlageret. Antallet af processorer kan således udvides til at omfatte så mange, som der er behov for, simpelthen ved at addere flere processorer af den "midterste" art. Overgitteropstillingen er egnet til effektiv materielindsættelse med kommercielt tilgængelige processorer. Et lille antal processorer, f.eks. 3 til 6, kan virke i tilknytning til mange anvendelser med særdeles gode resultater og med en væsentlig forøgelse af behandlingshastigheden i forhold til kendte signalprocessorer.

I Appendix 3 er vist et datamatprogram udskrevet i Pascal til simulering af den parallelle opdelte anvendelse, som er vist i fig. 6-8, på den vis, som er forklaret foranstående. Det vedlagte datamatprogram kan yderligere modificeres af fagfolk inden for området på simpel vis for tilvejebringelse af en multiprocessorudførelsesform af den foreliggende opfindelse, som vist i fig. 8.

Det vil således kunne indses, at der ved den foreliggende opfindelse er tilvejebragt mulighed for, at anvende et egnet antal parallelle processorer for tilvejebringelse af en overordentlig effektiv lineær forudsigelsessignalbehandling. Den "parallel-opdelte" anvendelse er fordelagtig ved parallel behandling, overskuelig materielkompleksitet og optimal signalbehandling for et udpeget antal tilgængelige processorer.

APPENDIX 1

```
(*****************************************************************************)
(*                                                                           *)
(*  NAME:  AR_PAR                                                            *)
(*                                                                           *)
(*  PURPOSE:   COMPUTER SIMULATION OF LINEAR PREDICTION,                     *)
(*              USING A PARALLEL IMPLEMENTATION OF THE                       *)
(*                        SCHUR RECURSIONS ON THE SUPERLATTICE               *)
(*                                                                           *)
(*  PARAMETERS:                                                              *)
(*       INPUT:  mo:ORDER OF THE PREDICTOR                                   *)
(*               r :AUTOCORRELATION SEQUENSE                                 *)
(*      OUTPUT:  k :LATTICE (PARCOR)  COEFFICIENTS                           *)
(*                                                                           *)
(*  PARAMETER TYPES:                                                         *)
(*     cortype:  ARRAY[0..mo] OF REAL                                        *)
(*    pcortype:  ARRAY[1..mo] OF REAL                                        *)
(*                                                                           *)
(*****************************************************************************)
procedure AR_PAR(mo:integer; r:cortype;
                          var  k: pcortype) ;
var a,b:array[0..50] of real;
    c,d:real;
    j,n:integer;

begin
   for j:=1 to mo do a[j]:=r[j];
   for j:=0 to mo do b[j]:=r[j];
   k[1]:=-r[1]/r[0];

   for n:=1 to mo-1 do
   begin
      for j:=2 to (mo-n+1) do
      begin
         c:=a[j]+k[n]*b[j-1];
         d:=b[j-2]+k[n]*a[j-1];
         a[j-1]:=c;
         b[j-2]:=d
      end;
      k[n+1]:=-a[1]/b[0]
   end
end;  (*  procedure AR_PAR  *)
```

APPENDIX 2

```
{*****************************************************************************}
{*                                                                           *}
{*  NAME    :  AR_OR                                                         *}
{*                                                                           *}
{*  PURPOSE    : COMPUTER SIMULATION  OF LINEAR PREDICTION                   *}
{*               USING AN ORDER-RECURSIVE IMPLEMENTATION                     *}
{*                      OF THE SCHUR  RECURSIONS ON THE                      *}
{*                                    SUPERLATTICE                           *}
{*  PARAMETERS:                                                              *}
{*       INPUT:  mo:ORDER OF THE PREDICTOR                                   *}
{*               r :AUTOCORRELATION SEQUENSE                                 *}
{*      OUTPUT:  k :LATTICE (PARCOR)  COEFFICIENTS                           *}
{*                                                                           *}
{*  PARAMETER TYPES:                                                         *}
{*    pcortype:   ARRAY[0..mo] OF REAL                                       *}
{*     cortype:   ARRAY[1..mo] OF REAL                                       *}
{*                                                                           *}
{*****************************************************************************}
procedure AR_OR(mo:integer; r: cortype;
                            var  k: pcortype)

var a,b:array[1..50] of real;
    c,d,e,f:real;
    j,n:integer;

begin
   a[1]:=r[1];
   b[1]:=r[0];
   k[1]:=-r[1]/r[0];
   for j:=2 to mo do
   begin
      e:=r[j];
      f:=r[j-1];
      for n:=1 to j-1 do
      begin
         c:=e+k[n]*f;
         d:=b[n]+k[n]*a[n];
         a[n]:=e;
         b[n]:=f;
         e:=c;
         f:=d;
      end;
      k[j]:=-e/f;
      a[j]:=e;
      b[j]:=f
   end;
end;
```

APPENDIX 3

```
program simulation (input,output);

type     tr =array [0..100] of real;
         tk =array [1..100] of real;

var      r :tr;
         k :tk;
         system_order,i :integer;

(*************************************************************************)

procedure partitioned_parallel(system_order :integer; r :tr; var k :tk);

(*This procedure simulates the partitioned parallel implementation of the
  superlattice, using three (two cycled) processors which work concurrent-
  ly. It accepts the autocorrelation coefficients 'r' and the sytem order
  as input and produces the reflection coefficients 'k'. *)

const    nbr_of_processors = 3;

type     tb = array [1..100,1..2] of real;

var      rest,nbr_of_partitions,level,max_level,max_level_minus_1:integer;
         a_bottom,b_bottom,a_bottom_temp,b_bottom_temp:real;
         a_middle,b_middle,a_middle_temp,b_middle_temp:real;
         bor_buffer1_temp,bor_buffer2_temp:real;
         partition,proc:integer;
         bor_buffer:tb;
         e_top,f_top:real;
         e_middle,f_middle:real;
         e_bottom,f_bottom:real;

     (*  The designation of parameters

         a_bottom,b_bottom,a_bottom_temp,b_bottom_temp
         a_middle,b_middle,a_middle_temp,b_middle_temp
         bor_buffer1_temp,bor_buffer2_temp
         e_top,f_top,e_middle,f_middle,e_bottom,f_bottom

         is made for programming convenience only and is unrelated to the
         register names appearing in the diagram of fig. 8. *)

(*************************************************************************)

   procedure initialize ( partition,proc : integer; var a,b,c,d : real );

   var   offset : integer;

   begin
      offset:=(partition-1)*3-1+proc;
      a:=r[offset];
      b:=r[offset+1];
      c:=b;
      d:=r[offset+2]
   end; (* of procedure initialize *)
```

```
(*************************************************************************)
   procedure basic_cell ( a,b,c,d,k_mult : real; var e,f : real);

   begin
      f:=d+c*k_mult;
      e:=a+b*k_mult
   end; (* of procedure initialize *)

(*************************************************************************)
   procedure bottom_processor ( partition,level : integer;
                                bb1,bb2 : real;
                                var a_bottom,b_bottom : real;
                                var e,f:real );

   var a,b,c,d,k_mult : real;

   begin
      if level=1 then
         initialize ( partition,1,a,b,c,d )
               else
         begin
            a:=bb1;
            b:=bb2;
            c:=e;
            d:=f;
         end;
      k_mult:=k[level];
      basic_cell ( a,b,c,d,k_mult,e,f );
      a_bottom:=e;
      b_bottom:=f
   end; (* of procedure bottom_processor *)

(*************************************************************************)
   procedure middle_processor ( partition,level : integer;
                                a_bottom,b_bottom : real;
                                var a_middle,b_middle : real;
                                var e,f:real );

   var a,b,c,d,k_mult : real;

   begin
      if level=1 then
         initialize ( partition,2,a,b,c,d )
               else
         begin
            a:=a_bottom;
            b:=b_bottom;
            c:=e;
            d:=f
         end;
      k_mult:=k[level];
      basic_cell ( a,b,c,d,k_mult,e,f );
      a_middle:=e;
      b_middle:=f
   end; (* of procedure middle_processor *)
```

```
(*************************************************************************)

   procedure top_processor ( partition.level : integer;
                             a_middle,b_middle : real;
                             var bor_buffer:tb;
                             var e,f:real );

   var   a,b,c,d,k_mult : real;

   begin
      if level=1 then
         initialize ( partition,3,a,b,c,d )
               else
         begin
            a:=a_middle;
            b:=b_middle;
            c:=e;
            d:=f
         end;
      k_mult:=k[level];
      basic_cell ( a,b,c,d,k_mult,e,f );
      bor_buffer[level,1]:=e;
      bor_buffer[level,2]:=f
   end; (* of procedure top_processor *)
```

```
(*****************************************************************************)

  begin

    rest:=(system_order-1) mod nbr_of_processors;
    nbr_of_partitions:=(system_order-1) div nbr_of_processors;
    if rest<>0 then nbr_of_partitions:=nbr_of_partitions+1;

    k[1]:=-r[1]/r[0];

    for partition:=1 to nbr_of_partitions do
      begin
        max_level:=partition*nbr_of_processors;
        for level:=1 to max_level-2 do
          begin
            a_bottom_temp:=a_bottom;
            b_bottom_temp:=b_bottom;

            bottom_processor ( partition,level,
                               bor_buffer1_temp,bor_buffer2_temp,
                               a_bottom,b_bottom,e_bottom,f_bottom);

            a_middle_temp:=a_middle;
            b_middle_temp:=b_middle;

            middle_processor ( partition,level,
                               a_bottom_temp,b_bottom_temp,
                               a_middle,b_middle,e_middle,f_middle);

            bor_buffer1_temp:=bor_buffer[level,1];
            bor_buffer2_temp:=bor_buffer[level,2];

            top_processor ( partition,level,
                            a_middle_temp,b_middle_temp,
                            bor_buffer,e_top,f_top);
          end; (* of level loop *)

        max_level_minus_1:=max_level-1;

        k[max_level_minus_1]:=-b_bottom/a_bottom;

        a_middle_temp:=a_middle;
        b_middle_temp:=b_middle;

        middle_processor ( partition,max_level_minus_1,a_bottom,b_bottom,
                           a_middle,b_middle,e_middle,f_middle);

        k[max_level]:=-b_middle/a_middle;

        top_processor ( partition,max_level_minus_1,
                        a_middle_temp,b_middle_temp,
                        bor_buffer,e_top,f_top );

        top_processor ( partition,max_level,
                        a_middle,b_middle,
                        bor_buffer,e_top,f_top );

        k[max_level+1]:=-bor_buffer[max_level,2]/bor_buffer[max_level,1]
      end;


  end; (* of procedure partitioned_parallel *)
```

```
begin

   (* Test data *)
   r[0] := 1;
   r[1] :=-0.2133256;
   r[2] := 0.2076102;
   r[3] :=-0.3665574;
   r[4] :=-0.2301161;
   r[5] := 0.2541363;
   r[6] :=-0.2486953;
   r[7] := 0.7792566;
   r[8] :=-0.1939328;
   r[9] := 0.2779704;
   r[10]:=-0.4803222;
   system_order:=10;

   (* The results are the following :

   k[1] = 0.2133256
   k[2] =-0.1698310
   k[3] = 0.3166915
   k[4] = 0.4394402
   k[5] =-0.3670546
   k[6] = 0.2772087
   k[7] =-0.6954674
   k[8] =-0.1860846
   k[9] =-0.0412548
   k[10]= 0.3096930  *)

   partitioned parallel(system_order,r,k);

   for i:=1 to system_order do writeln(k[i]);

end.
```

PATENTKRAV.

1. Signalprocessor, som modtager autokorrelationskoefficienter r_i, hvor i=[0-p] svarende til et system af p'ende orden, for tilvejebringelse af gitterkoefficienter k_j, hvor j=[1-p] for systemet, indbefattende en indgangsindretning (62) til modtagelse af autokorrelationskoefficienterne, et antal behandlingsenheder (50, 52, 54), som hver indbefatter en processor (44; fig. 4), som multiplicerer størrelser på en første (b) og en anden (c) indgang med en gitterkoefficient (k_j), hvorved tilvejebringes et første og et andet produkt, og adderer størrelse på en tredie og fjerde indgang (a, d) til det første henholdsvis andet produkt, en første opbygning til genoverføring af udgangssignaler fra udgangene (e, f) fra antallet af parallelle behandlingsenheder til indgangene (a, b, c, d) til i det mindste én af de parallelle behandlingsenheder, således at
den første og den anden indgang (b, c) hver modtager en første autokorrelationskoefficient i et første tidsinterval i hver processor, idet autokorrelationskoefficienterne skal multipliceres med en første gitterkoefficient (k_1), og den tredie og den fjerde indgang (a, d) modtager et udvalgt par autokorrelationskoefficienter, som er tilstødende til de første koefficienter, til frembringelse af et par første mellemværdier

$$(\zeta^1_{-(i-1)} \; ; \; \zeta^1_{i+1})$$

på udgangene (e, f) på hver processor, medens de første mellemværdipar i et andet tidsinterval genoverføres til processorindgangene ved den første opbygning, for herved at multiplicere to udvalgte første mellemværdier med en anden gitterkoefficient til frembringelse af et andet produktpar og for enkeltvis at addere et par første mellemværdier, som er tilstødende til de to udvalgte første mellemværdier, til det andet produktpar for tilvejebringelse af i det mindste et par anden mellemværdier, og divisionskredse (70, 72, 74)

til dannelse af kvotienten for et udvalgt par autokorrelationskoefficienter til frembringelse af den første gitterkoefficient, og til dannelse af kvotienten for et udvalgt par første mellemværdier til frembringelse af den anden gitterkoefficient, k e n d e t e g n e t ved, at antallet af behandlingsenheder er mindre end systemets orden p, og at der er tilvejebragt en anden opbygning (78), hvori lagres udvalgte første og anden mellemværdier, og hvori de lagrede mellemværdier, i egnede tidsintervaller efter det andet tidsinterval, fremdrages for overføring til egnede indgange på behandlingsenhederne, således at yderligere mellemværdipar frembringes, når yderligere resterende autokorrelationskoefficienter overføres til processorerne.

2. Fremgangsmåde i en signalprocessor, som modtager autokorrelationskoefficienter r_i, hvor i=[0-p], svarende til et system af p'ende orden, til frembringelse af gitterkoefficienter k_j, hvor j=[1-p] i dette system, indbefattende overføring af autokorrelationskoefficienterne til et antal behandlingsenheder (50, 52, 54), som hver især indbefatter en processor (44; fig. 4), hvori multipliceres størrelser på en første (b) og en anden (c) indgang med en gitterkoefficient (k_j), hvorved tilvejebringes et første og et andet produkt, og hvor størrelser på en tredie og fjerde indgang (a, d) adderes enkeltvis til det første henholdsvis det andet produkt,

genoverføring af signalerne på udgangene (e, f) fra antallet af parallelle behandlingsenheder til indgangene (a, b, c, d) på i det mindste én af de parallelle behandlingsenheder, således at

den første og den anden indgang (b, c) hver modtager en første autokorrelationskoefficient i et første tidsinterval i hver processor, som skal multipliceres med en første gitterkoefficient (k_1), og den tredie og den fjerde indgang (a, d) modtager et udvalgt par autokorrelationskoefficienter, som er tilstødende til de første koefficienter, for tilvejebringelse af et par første mellemværdier

$$(\zeta^1_{-(i-1)} \; ; \; \zeta^1_{i+1})$$

på udgangene (e, f) på hver processor, medens det første mellemværdipar i et andet tidsinterval genoverføres til processorindgangene, for herved at multiplicere to udvalgte første mellemværdier med en anden gitterkoefficient til frembringelse af et andet produktpar og for enkeltvis at addere et par første mellemværdier, som er tilstødende til de to udvalgte første mellemværdier, til det andet produktpar for tilvejebringelse af i det mindste et par anden mellemværdier, og

dannelse af kvotienten for et udvalgt par autokorrelationskoefficienter til frembringelse af den første gitterkoefficient, og dannelse af kvotienten for et udvalgt par første mellemværdier til frembringelse af den anden gitterkoefficient, k e n d e t e g n e t ved, at der er tilvejebragt et antal behandlingsenheder, hvor antallet er mindre end systemets orden p, og at der oplagres udvalgte værdier af første og anden mellemværdi, og at der i egnede tidsintervaller efter det første tidsinterval selektivt fremdrages de lagrede mellemværdier for overføring til egnede indgange på behandlingsenhederne således, at der frembringes yderligere par af mellemværdier, når yderligere resterende autokorrelationskoefficienter overføres til processorerne.

3. Signalprocessor, som modtager autokorrelationskoefficienter r_i, hvor i=[-(p)-p] svarende til et system af p'ende orden, for tilvejebringelse af normale og tilstødende gitterfoefficienter k_j og k^*_j, hvor j=[1-p] for systemet, indbefattende en indgangsindretning (62) til modtagelse af autokorrelationskoefficienterne, et antal behandlingsenheder (50, 52, 54), som hver indbefatter en processor (44; fig. 4), som multiplicerer størrelser på en første (c) og en anden (b) indgang med en normal og en tilstødende gitterkoefficient h.h.v. (k_j og k^*_j), hvorved tilvejebringes et første og et andet produkt, og adderer størrelser på tredie

og fjerde indgang (d, a) til det første henholdsvis andet produkt, en første opbygning til genoverføring af udgangssignaler fra udgangene (e, f) fra antallet af parallelle behandlingsenheder til indgangene (a, b, c, d) til i det mindste én af de parallelle behandlingsenheder, således at den første og den anden indgang (b, c) hver modtager en første autokorrelationskoefficient i et første tidsinterval i hver processor, idet autokorrelationskoefficienterne skal multipliceres med henholdsvis en første normal og tilstødende gitterkoefficient (k_1 og k_1^*) og den tredie og den fjerde indgang (d, a) modtager et udvalgt par autokorrelationskoefficienter, som er tilstødende til de første koefficienter, til frembringelse af et par første mellemværdier

$$(\zeta^1_{i+1} \; ; \; \zeta^1_{i-1})$$

på udgangene (f, e) på hver processor, medens de første mellemværdipar i et andet tidsinterval genoverføres til processorindgangene ved den første opbygning, for herved at multiplicere to udvalgte første mellemværdier med henholdsvis en anden normal og tilstødende gitterkoefficient til frembringelse af et andet produktpar og for enkeltvis at addere et par første mellemværdier, som er tilstødende til de to udvalgte første mellemværdier, til det andet produktpar for tilvejebringelse af i det mindste et par anden mellemværdier, og divisionskredse (70, 72, 74) til dannelse af kvotienten for et udvalgt par autokorrelationskoefficienter til frembringelse af den normale og tilstødende første gitterkoefficient, og til dannelse af kvotienten for et udvalgt par første mellemværdier til frembringelse af den anden normale og tilstødende gitterkoefficient, k e n d e t e g n e t ved, at antallet af behandlingsenheder er mindre end systemets orden p, og at der er tilvejebragt en anden opbygning (78), hvori lagres udvalgte første og anden mellemværdier, og hvori de lagrede mellemværdier, i egnede tidsintervaller efter det andet tidsinterval, fremdrages for overføring til

egnede indgange på behandlingsenhederne, således at yderligere mellemværdipar frembringes, når yderligere resterende autokorrelationskoefficienter overføres til processorerne.

4. Fremgangsmåde i en signalprocessor, som modtager autokorrelationskoefficienter r_i, hvor i=[-(p)-p], svarende til et system af p'ende orden, til frembringelse af normale og tilstødende gitterkoefficienter k_j og k_j^*, hvor j=[1-p] i dette system, indbefattende overføring af autokorrelationskoefficienterne til et antal behandlingsenheder (50, 52, 54), som hver især indbefatter en processor (44; fig. 4), hvori multipliceres størrelser på en første (c) og en anden (b) indgang med henholdsvis en normal og en tilstødende gitterkoefficient (k_j, k_j^*), hvorved tilvejebringes et første og et andet produkt, og hvor størrelser på en tredie og en fjerde indgang (d, a) adderes enkeltvis til det første henholdsvis andet produkt,

genoverføring af signalerne på udgangene (e, f) fra antallet af parallelle behandlingsenheder til indgangene (a, b, c, d) på i det mindste én af de parallelle behandlingsenheder, således at

den første og den anden indgang (c, b) i et første tidsinterval i hver processor hver modtager en første autokorrelationskoefficient, som skal multipliceres med en første normal og tilstødende gitterkoefficient (k_1, k_1^*), og den tredie og den fjerde indgang (d, a) modtager et udvalgt par autokorrelationskoefficienter, som er tilstødende til de første koefficienter, for tilvejebringelse af et par første mellemværdier

$$(\zeta^1_{i+1} \; ; \; \zeta^1_{i-1})$$

på udgangene (e, f) på hver processor, medens det første mellemværdipar i et andet tidsinterval genoverføres til processorindgangene, for herved at multiplicere to udvalgte første mellemværdier med henholdsvis en normal og en tilstødende anden gitterkoefficient til frembringelse af et

andet produktpar og for enkeltvis at addere et par første mellemværdier, som er tilstødende til de to udvalgte første mellemværdier, til det andet produktpar for tilvejebringelse af i det mindste et par anden mellemværdier, og

dannelse af kvotienten for et udvalgt par autokorrelationskoefficienter til frembringelse af en første normal og tilstødende gitterkoefficient, og dannelse af kvotienten for et udvalgt par første mellemværdier til frembringelse af den normale og tilstødende anden gitterkoefficient,

k e n d e t e g n e t ved, at der er tilvejebragt et antal behandlingsenheder, hvor antallet er mindre end systemets orden p, og at der oplagres udvalgte værdier af første og anden mellemværdi, og at der i egnede tidsintervaller efter det første tidsinterval selektivt fremdrages de lagrede mellemværdier for overføring til egnede indgange på behandlingsenhederne således, at der frembringes yderligere par af mellemværdier, når yderligere resterende autokorrelationskoefficienter overføres til processorerne.

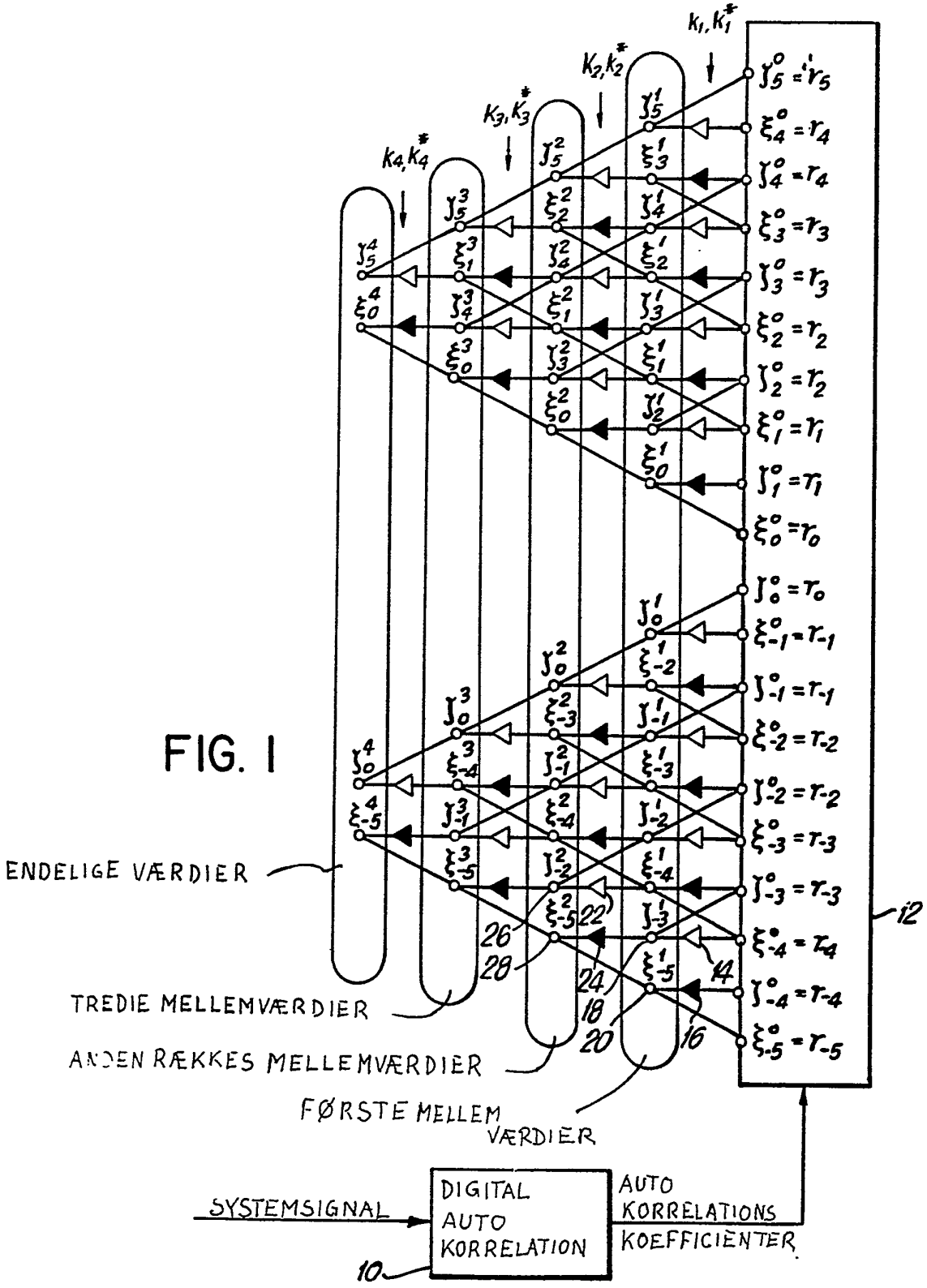

ENDELIGE VÆRDIER
TREDIE MELLEMVÆRDIER
ANDEN RÆKKES MELLEMVÆRDIER
FØRSTE MELLEM VÆRDIER
SYSTEMSIGNAL
DIGITAL AUTO KORRELATION
AUTO KORRELATIONS KOEFFICIENTER
10
12
14
16
18
20
22
24
26
28

FIG. 1

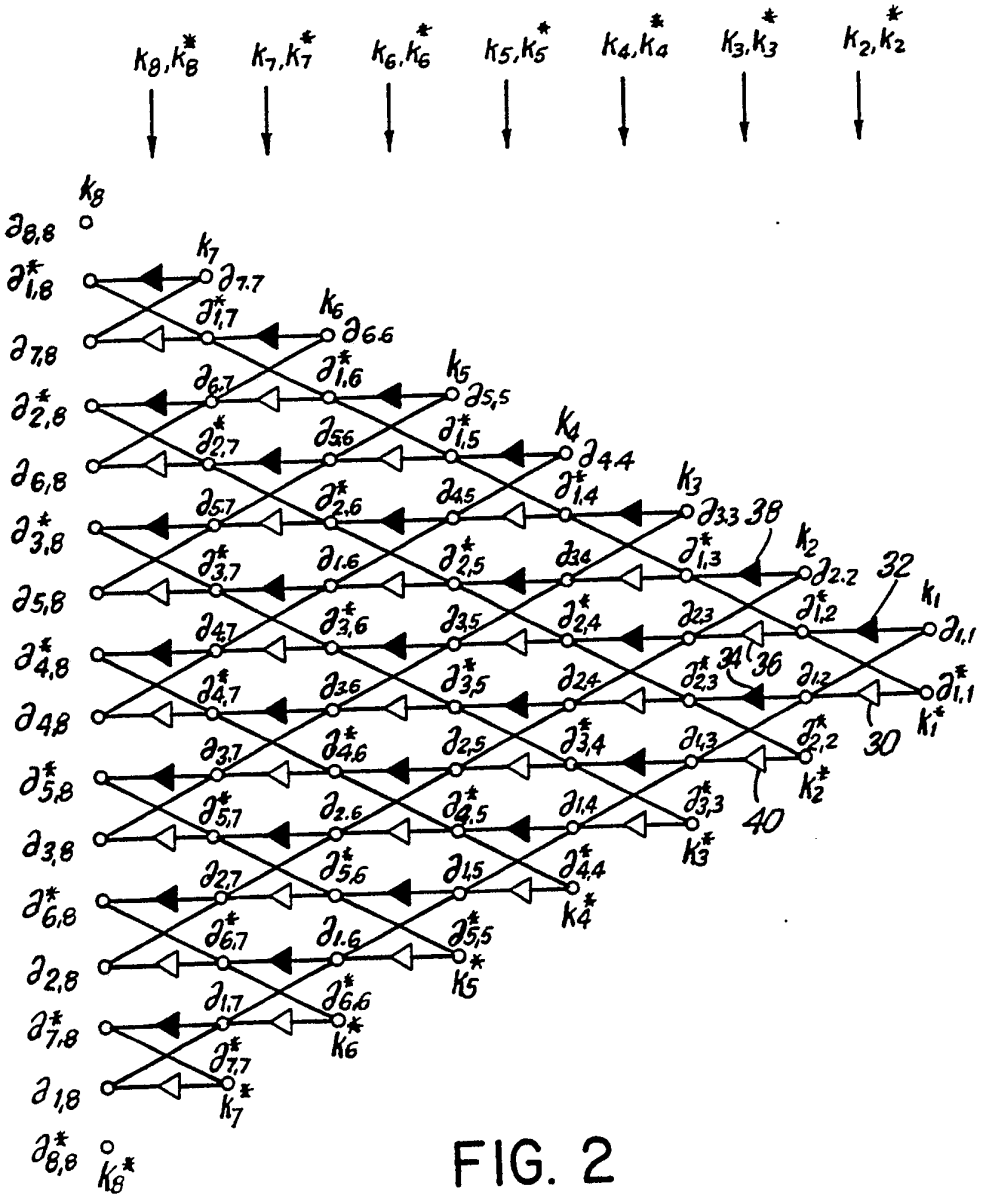


FIG. 2

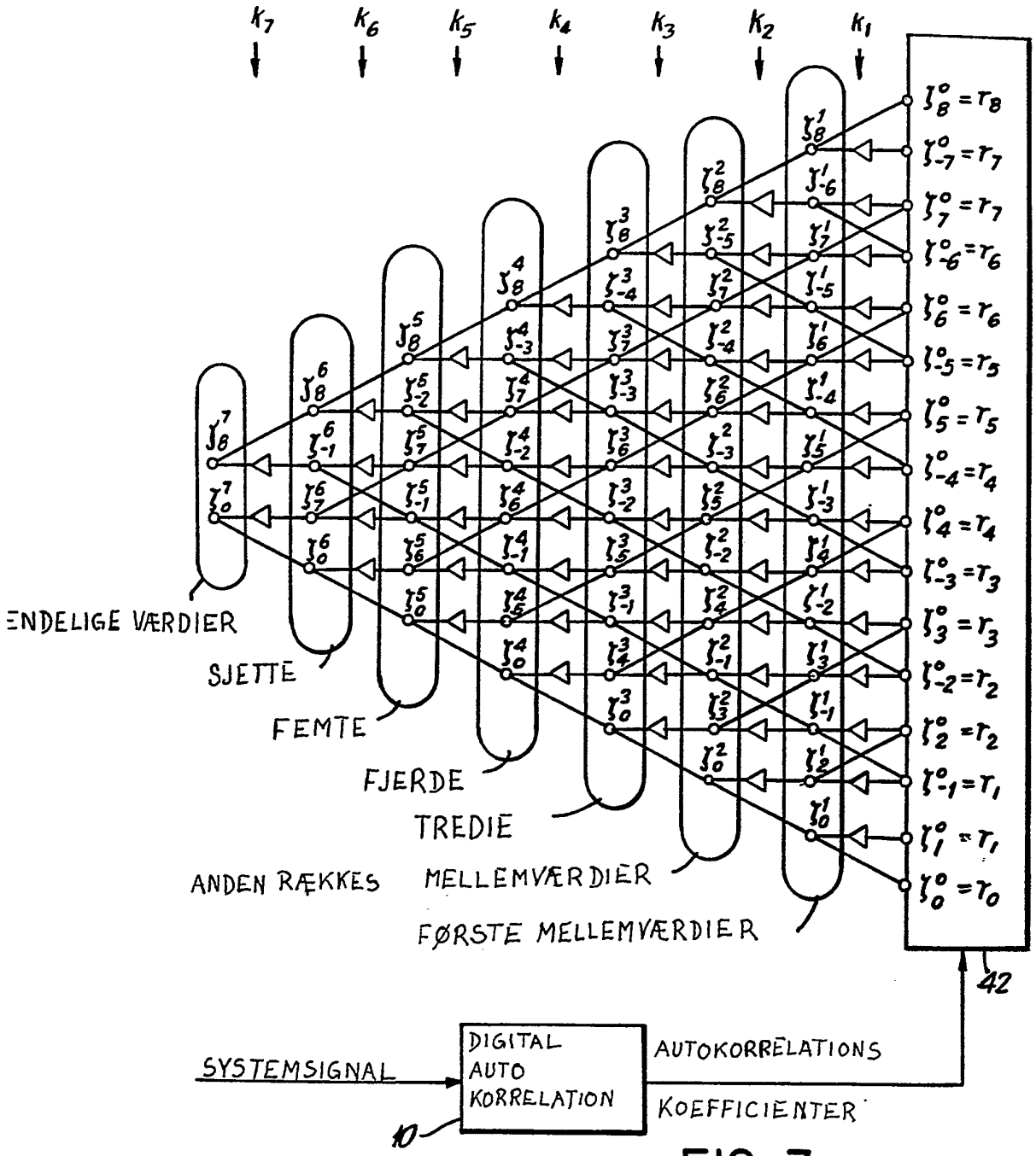

k_7 k_6 k_5 k_4 k_3 k_2 k_1
$r_8^0 = r_8$
$r_{-7}^0 = r_7$
$r_7^0 = r_7$
$r_{-6}^0 = r_6$
$r_6^0 = r_6$
$r_{-5}^0 = r_5$
$r_5^0 = r_5$
$r_{-4}^0 = r_4$
$r_4^0 = r_4$
$r_{-3}^0 = r_3$
$r_3^0 = r_3$
$r_{-2}^0 = r_2$
$r_2^0 = r_2$
$r_{-1}^0 = r_1$
$r_1^0 = r_1$
$r_0^0 = r_0$
ENDELIGE VÆRDIER
SJETTE
FEMTE
FJERDE
TREDIE
ANDEN RÆKKES MELLEMVÆRDIER
FØRSTE MELLEMVÆRDIER
42
SYSTEMSIGNAL
DIGITAL AUTO KORRELATION
10
AUTOKORRELATIONS
KOEFFICIENTER

FIG. 3

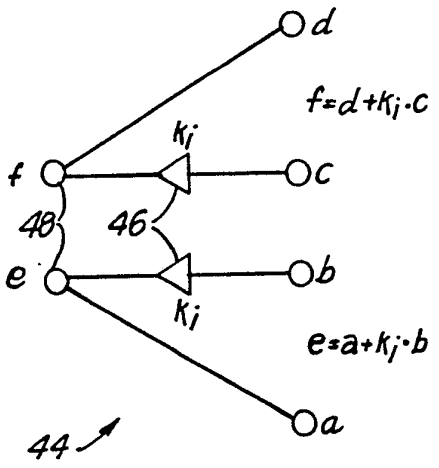

d
f=d+kj·c
kj
f
c
48
46
e
b
kj
e=a+kj·b
a
44

FIG. 4

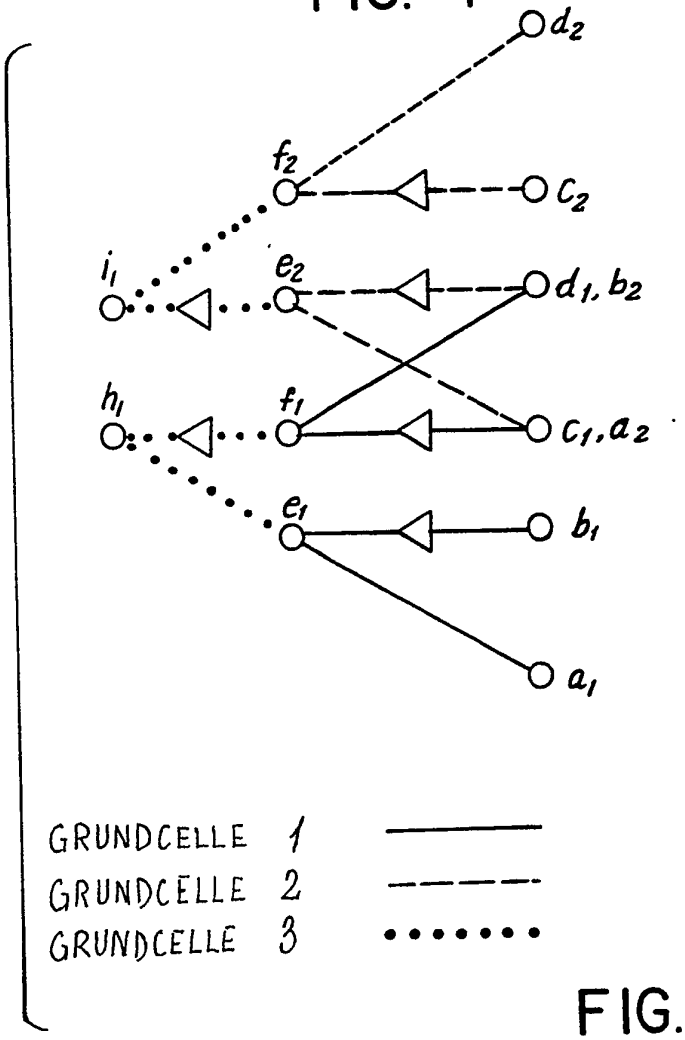

d2
f2
c2
i1
e2
d1, b2
h1
f1
c1, a2
e1
b1
a1
GRUNDCELLE 1
GRUNDCELLE 2
GRUNDCELLE 3

FIG. 5

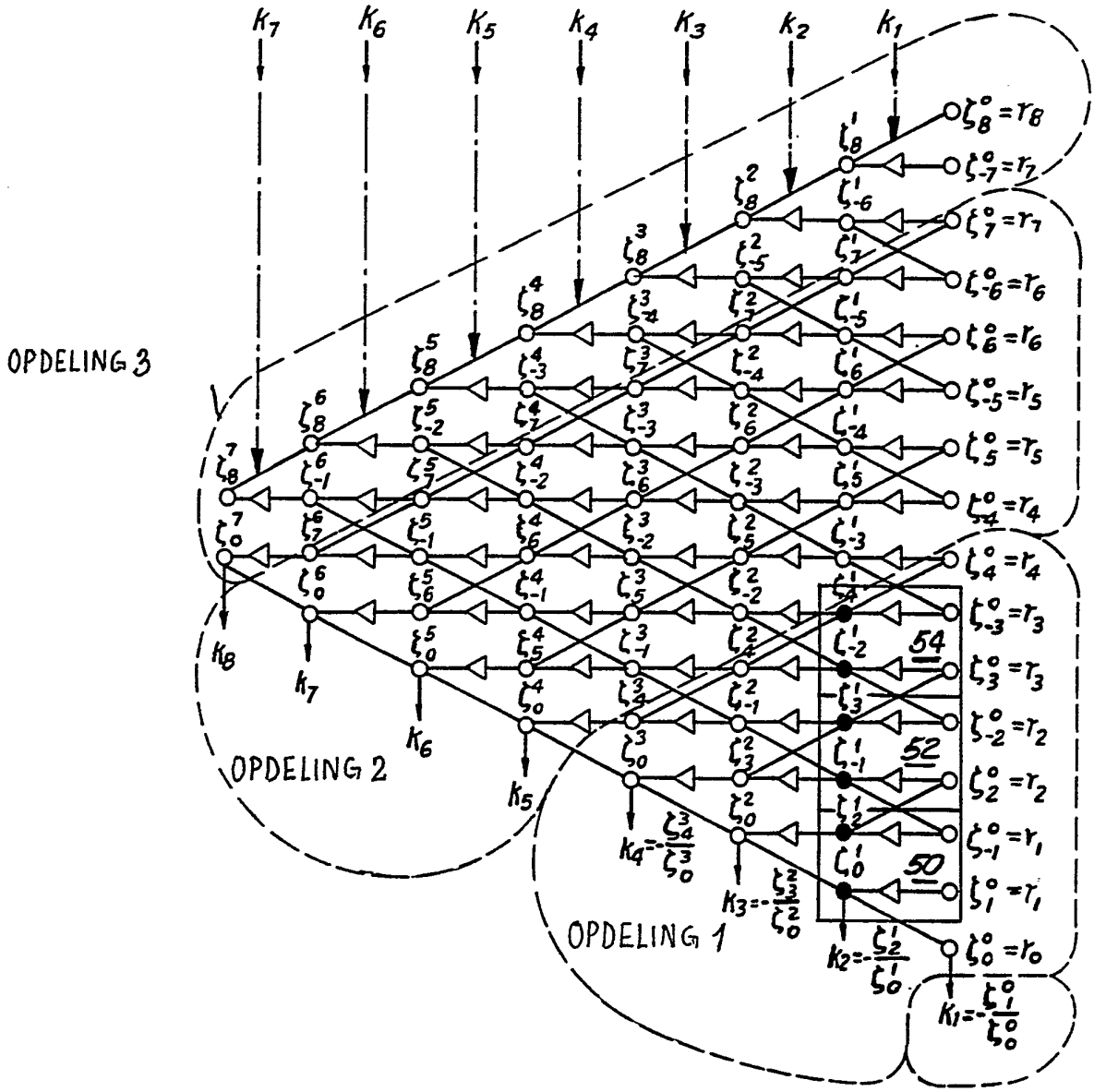


FIG. 6

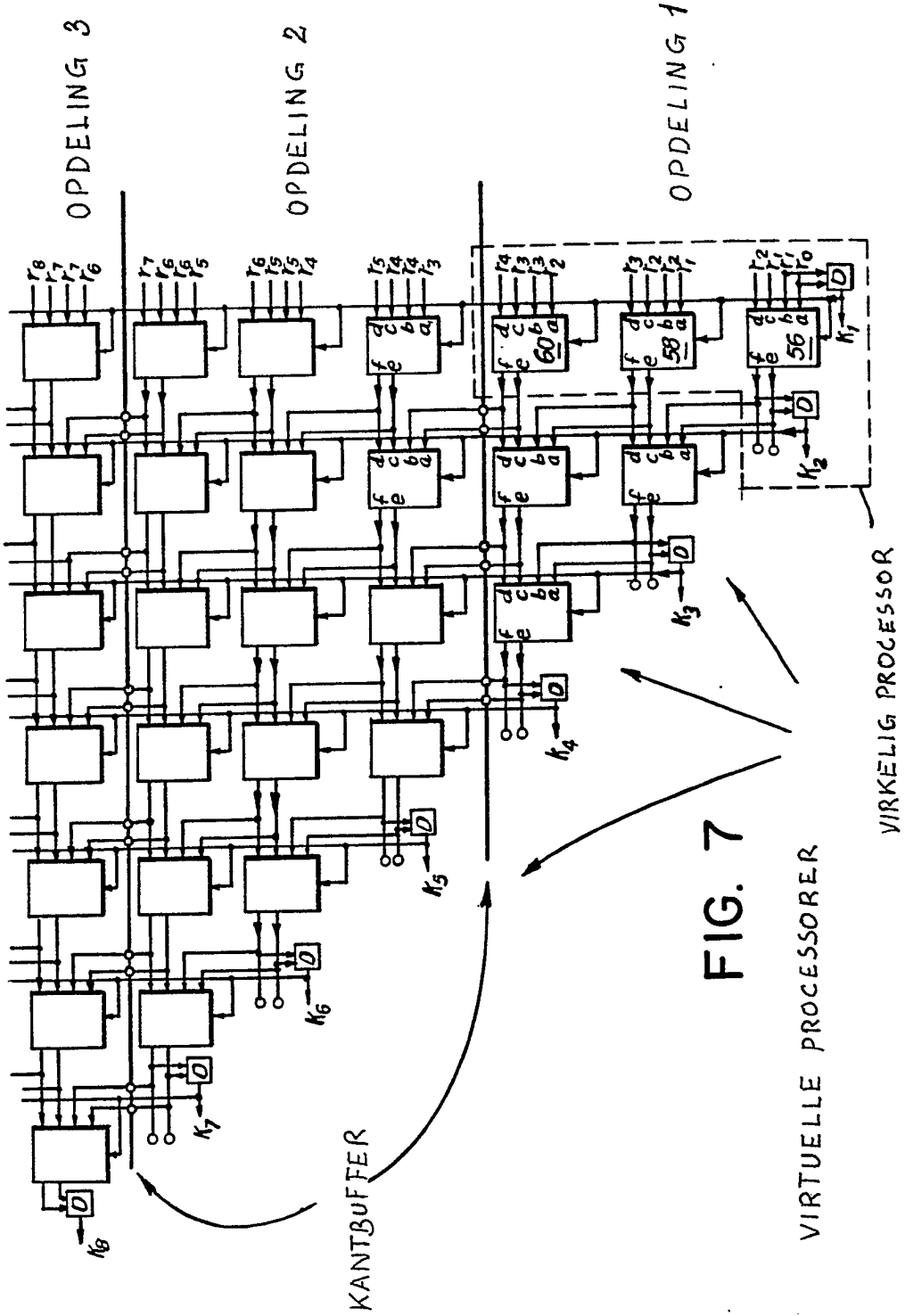

OPDELING 3
OPDELING 2
OPDELING 1
KANTBUFFER
VIRTUELLE PROCESSORER
VIRKELIG PROCESSOR

FIG. 7

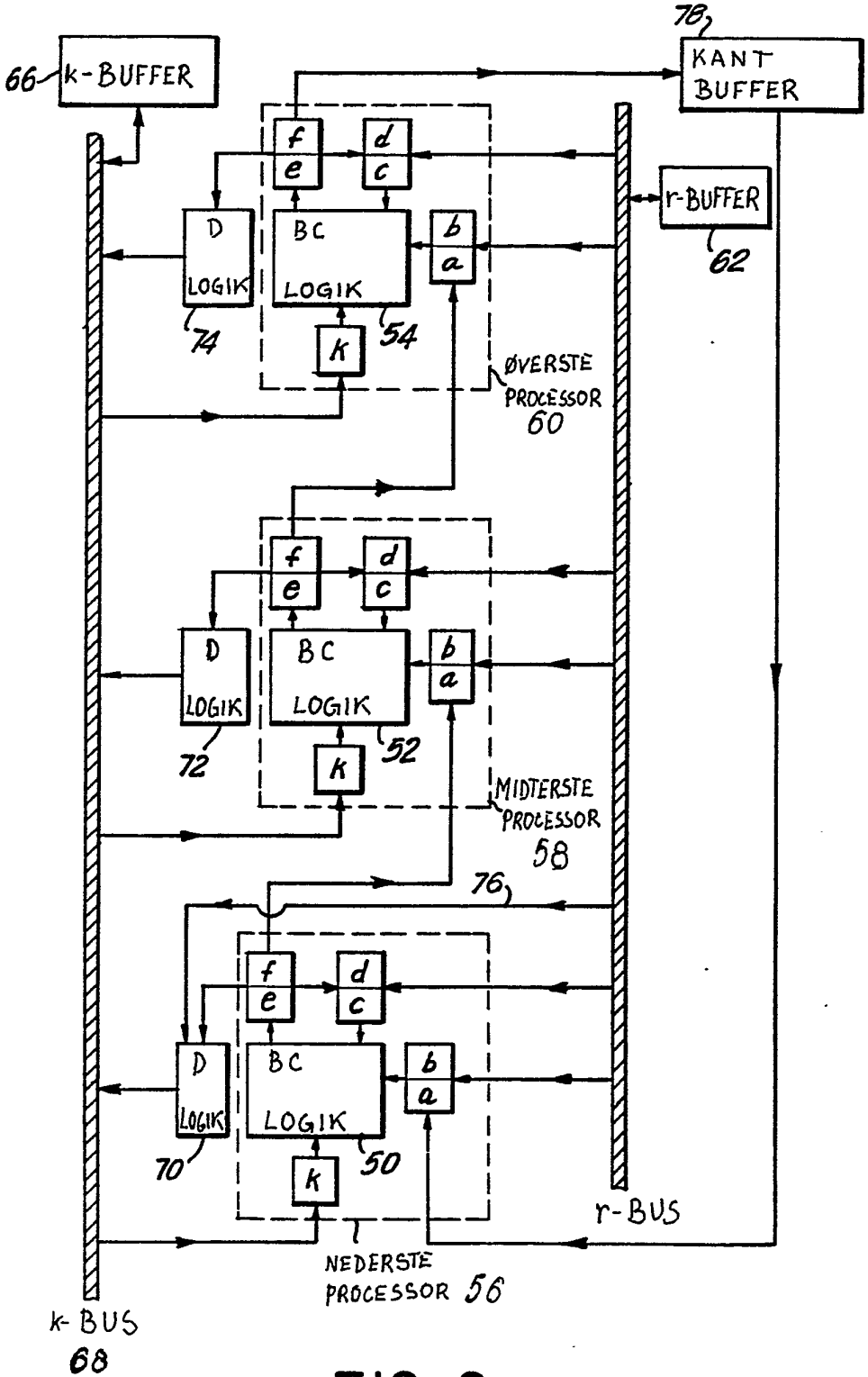

66
k-BUFFER
78
KANT BUFFER
r-BUFFER
62
f
e
d
c
b
a
D LOGIK
74
BC LOGIK
54
K
ØVERSTE PROCESSOR 60
72
52
MIDTERSTE PROCESSOR 58
76
70
50
r-BUS
NEDERSTE PROCESSOR 56
k-BUS
68

FIG. 8