

19



Octrooiraad
Nederland

11 192637

12 C OCTROOI

21 Aanvraag om octrooi: 8500537

51 Int.Cl.⁸
G06F15/173, G06F13/12

22 Ingediend: 26.02.85

30 Voorrang:
27.02.84 JP 0034450/84
16.05.84 JP 0096656/84

43 Ter inzage gelegd:
16.09.85 I.E. 85/18

44 Openbaargemaakt:
01.07.97 I.E. 97/07

47 Dagtekening:
04.11.97

45 Uitgegeven:
05.01.98 I.E. 98/01

73 Octrooihouder(s):
Nippon Telegraph and Telephone Corporation te
Tokio, Japan (JP).

74 Gemachtigde:
Ir. Th.A.H.J. Smulders c.s. te 2587 BN Den
Haag.

54 Stelselprocessor.

Stelselprocessor

- De uitvinding heeft betrekking op een stelselprocessor voorzien van n verwerkingselementen, welke onderling in de vorm van een ring gekoppeld zijn met behulp van n multiplexinrichtingen, waarbij elk van de
- 5 n verwerkingselementen omvat: invoerorganen voor het invoeren van één van de ingangsvectorgegevens van een eerste en een tweede reeks van ingangsvectorgegevens; organen voor het uitvoeren van voorafbepaalde berekeningen, zoals optellen, aftrekken, vergelijkingsberekeningen en product-somberekeningen tussen de ingangsvectorgegevens van de eerste en de tweede reeks; opslagorganen voor het opslaan van de ingangsvectorgegevens van de eerste reeks, van de tweede reeks, en van de
- 10 resultaten van de berekeningen; een gegevensoverdraagorgaan voor het overdragen van één van de ingangsvectorgegevens en de berekeningsresultaten naar een naastgelegen verwerkingselement via een van de multiplexinrichtingen; een met een I/O-lijn verbonden uitvoerorgaan voor het uitvoeren als eind-berekening van het resultaat naar de I/O-lijn; en een stuurorgaan voor het besturen van de invoerorganen, de opslagorganen, het berekeningsorgaan, het overdrachtsorgaan en het uitvoerorgaan.
- 15 Een dergelijke inrichting is bekend uit het artikel "VLSI structures for speech analysis and pattern recognition" door L. Ciminiera et al in "Proceedings, IEEE Computer Society Conference on pattern recognition and Image Processing, 14-17 juni 1982, Las Vegas, Nevada, bladzijden 692-697, IEEE, New York, US. Dit artikel toont een processor voorzien van een aantal PE's (PE is processing element) in een ring, en is gebaseerd op parallelle verwerking van de informatie.
- 20 De informatie wordt vanuit een externe bron via een enkel type informatiebus toegevoerd aan de PE's en wordt via een poort van elke PE afgevoerd.
- Voorts wordt een zogenaamde kruisstangkiezer toegepast om een soort van deeltijdregeling aan te brengen, maar een nadeel hiervan is dat een snelle modulaire werking niet meer mogelijk is.
- Dit artikel beoogt te voorzien in een systeem om met behulp van een processor een hoge verwerkings-
- 25 snelheid te realiseren tussen uitsluitend één soort referentiepatroon en een bepaald onbekend object dat moet worden verwerkt, terwijl volgens de aanvraag een lage snelheid wordt verkregen tussen een aantal referentiepatronen en een bepaald onbekend object. Het verschil zit derhalve in het aantal referentiepatronen.
- Zou men met de schakeling volgens genoemd artikel werken met een aantal patronen in ware tijd, zal elk
- 30 referentiepatroon een afzonderlijke processor moeten hebben, met als gevolg dat het aantal soorten referentiepatronen enorm groot wordt, en dus een groot aantal hardware componenten nodig is.
- De uitvinding ondervangt deze bezwaren.
- De stelselprocessor volgens de uitvinding heeft daartoe het kenmerk, dat elk van de n multiplexinrichtingen een tweede ingang heeft, die verbonden is met een ingangsvectorgegevenslijn, en voorzien is
- 35 van organen voor het kiezen van informatie-overdracht hetzij tussen twee verwerkingselementen ($PE_1 \dots PE_n$), hetzij tussen de ingangsvectorgegevenslijn en een verwerkingselement; dat de invoerorganen bestaan uit een eerste invoerorgaan voor het invoeren van één van de ingangsvectorgegevens (c_j) van de eerste reeks ($C = \{c_j\} \ (j = 1, 2, \dots, I)$) vanaf de ingangsvectorgegevenslijn via de multiplexinrichting, en uit een
- 40 tweede invoerorgaan voor het invoeren van één van de ingangsvectorgegevens (r_j) van de tweede reeks ($R = \{r_j\} \ (j = 1, 2, \dots, N)$) uit de I/O-lijn; en dat het gegevensoverdraagorgaan is ingericht om respectieve verwerkingsresultaten gelijktijdig over te dragen naar respectieve naastgelegen verwerkingselementen, en wel mod_n^N maal (mod_n^N is de rest als N door n wordt gedeeld), parallel met een berekeningsverwerking in elk verwerkingselement.
- Op deze wijze kan men toe met minder PE's en bovendien zijn er minder toe- en afvoerdraden nodig om
- 45 toch ten minste dezelfde hoeveelheid informatie te verwerken als bij het stelsel volgens de stand van de techniek. Voorts kan het aantal PE's bepaald worden ongeacht het aantal en de lengte van een referentiepatroon en de lengte van een onbekend patroon.
- Opgemerkt wordt dat stelselprocessoren kunnen worden gebruikt voor het uitvoeren van berekeningen voor alle combinaties van twee typen variabelen en voor recursieve formules met lokale afhankelijkheid van
- 50 informatie, waarbij deze berekeningsresultaten worden gebruikt, welke worden gekenmerkt door aanpassingsberekeningen, gebaseerd op een dynamische tijdvervormings- of dynamische programmeringstheorie, welke wordt gebruikt in het geval van patroonaanpassing bij spraakherkenning en symboolherkenning, en een besturingsmethode van de stelselprocessor.
- Dynamische tijdvervorming staat bekend als een doeltreffende methode bij patroonaanpassing in
- 55 spraakherkenning of symboolherkenning. De dynamische tijdvervorming vereist een tijdsgetrouwe verwerking van een bijzonder groot volume aan informatie, zodat een snelle verwerking nodig is. Verwerking van de dynamische tijdvervorming is één type berekening, dat bestaat uit de berekening van alle combinaties

van twee typen variabelen en die van recursieve formules met een lokale afhankelijkheid van informatie, waarbij de resultaten van een dergelijke berekening worden gebruikt.

- Ter illustratie zal in het onderstaande een voorbeeld van een aanpassingsberekening, gebaseerd op dynamische tijdvervorming, welke de berekening van twee typen vectorvariabelen en die van recursieve formules van cumulatieve operaties onder gebruik van dergelijke berekeningsresultaten omvat, worden aangegeven.

$$D_{ij} | C_i - r_j |^2 = \sum_{k=1}^m | c_k^i - r_k^j |^2 \quad (1)$$

$$S_{ij} = D_{ij} + \min \begin{pmatrix} S_{i,j-1} \\ S_{i-1,j-1} \\ S_{i-1,j} \end{pmatrix} \quad (2)$$

waarbij: $S_{0j} = \infty, S_{i0} = \infty, S_{00} = 0$ (3)

- c_i en r_j respectievelijk i -de en j -de elementen van een vector $C = (c_1, c_2, \dots, c_l)$ met l elementen en een vector $R = (r_1, r_2, \dots, r_N)$ met N elementen zijn. Voorts drukt m een orde van elke vector uit. c_i stelt $(c_1^i, c_2^i, \dots, c_m^i)$ en r_j stelt $(r_1^j, r_2^j, \dots, r_m^j)$ voor. D_{ij} en S_{ij} drukken respectievelijk een afstand tussen vectoren en cumulatieve afstand uit.

- Als een stelselprocessor, welke dit type berekening parallel kan verwerken is op zich een conventioneel stelsel bekend, waarin, wanneer de aantallen gegevens in twee typen vectoren respectievelijk l en N zijn ($l \times N$) verwerkingselementen (hierna betiteld als PE) in een orthogonaal netwerk aanwezig zijn.

Een dergelijke orthogonale constructie van een stelselprocessor is beschreven in het Amerikaanse octrooischrift 4.384.273.

- De uitvinding zal nu aan de hand van de tekening en de beschrijving nader worden toegelicht. Daarbij toont: figuur 1 een schema van een voorbeeld van een constructie van een conventionele orthogonale stelselprocessor;

figuren 2A en 2B; 3A en 3B; 4A en 4B; en 5A en 5B diagrammen ter toelichting van een voorbeeld van de verwerkingshandelingen van de conventionele stelselprocessor;

- figuur 6 een blokschema van een uitvoeringsvorm van een stelselprocessor volgens een verbeterd concept;

figuur 7 een blokschema van een uitvoeringsvorm van het verwerkingselement in de stelselprocessor van figuur 6;

figuur 8 een diagram ter toelichting van een voorbeeld van de verwerkingshandelingen in de in figuur 6 afgebeelde stelselprocessor;

- figuur 9 een diagram ter toelichting van de informatie-invoer vanuit uitwendige bronnen en toestanden van informatieoverdracht tussen verwerkingselementen in de in figuur 8 afgebeelde stelselprocessor;

figuur 10 een diagram ter toelichting van een voorbeeld van de verwerkingshandelingen van elk verwerkingselement;

- figuur 11 een blokschema van een andere uitvoeringsvorm van een stelselprocessor, waarin de verwerkingselementen zijn opgesteld en verbonden in de vorm van een ring;

figuur 12 een schema ter toelichting van een voorbeeld van een aan de ingangsvectorinformatie toegevoegde vlag;

figuur 13 een blokschema ter toelichting van een voorbeeld van regelhandelingen van een conditievlag-register in het verwerkingselement;

- figuren 14A–14E diagrammen ter toelichting van een voorbeeld van regelhandelingen wanneer in de in figuur 11 afgebeelde stelselprocessor een dynamische tijdvervormingsberekening wordt uitgevoerd;

figuur 15 een blokschema van een uitvoeringsvorm van een constructie, waarbij een regeleenheid als een deel van de stelselprocessor in de verwerkingselementen is opgenomen;

- figuren 16A–16I diagrammen ter toelichting van een voorbeeld van regelhandelingen wanneer in een stelselprocessor met een orthogonale constructie een matrix vermenigvuldigd wordt uitgevoerd;

figuur 17 een stroomdiagram van een voorbeeld van een besturingsstroom van een logische regelketen in de regeleenheid van elk verwerkingselement;

figuur 18 een stroomdiagram van een voorbeeld van een regelstroom van de regeleenheid in elk verwerkingselement.

Onder verwijzing nu naar figuren 1–5 toont figuur 1 een illustratieve constructie van een stelselprocessor, die in een orthogonaal netwerk is opgesteld en figuren 2A en 2B; 3A en 3B; 4A en 4B; en 5A en 5B tonen

een voorbeeld van de werking van de stelselprocessor. In figuur 1 is 100 een PE, 200 een lokale informatiebaan en 300 een instructielijn. Met 400 is eeningangsklem aangegeven en met 500 een uitgangsklem.

Elke PE 100 bezit organen voor het uitvoeren van een intervectorafstandsberekeningsvergelijking (1), welke bestaat uit optel- en vermenigvuldigingsberekeningen en een vergelijkings- en cumulatieve berekeningsvergelijking (2), en organen voor het ontvangen en overdragen van resultaten van de vergelijkingsberekeningen vanuit de naast gelegen PE's en de resultaten van cumulatieve berekeningen S_{ij} , en de vectorinformatie c_i en r_j . Voorts is elke PE aangegeven met een getal, dat de positie daarvan in het orthogonale stelsel aangeeft. Wanneer de PE in het snijpunt van de kolom i en de rij j wordt omschreven als PE_{ij} , kunnen de bovenstaande vergelijkingen (1), (2) en (3) worden uitgevoerd met de volgende handelingen.

- (i) Vanuit de links naastgelegen $PE_{i-1,j}$, welke naast PE_{ij} ligt aan de linkerzijde en de naar beneden gericht naastgelegen $PE_{i,j-1}$, welke bij PE_{ij} in neerwaartse richting ligt (alternatief uit eeningangsklem aan de linkerrand en eeningangsklem aan de bodemrand) worden twee typen vectorgegevens c_i , r_j ingevoerd en wordt de afstand tussen deze vectoren verkregen door vergelijking (1) uit te voeren.
- (ii) De vectorinformatie c_i en r_j wordt respectievelijk naar de rechts naast gelegen $PE_{i+1,j}$ en naar de boven naast gelegen $PE_{i,j+1}$ overgedragen.
- (iii) Vanuit de links naast gelegen $PE_{i-1,j}$ wordt een cumulatief berekeningsresultaat $S_{i-1,j}$ ingevoerd en vanuit de onder naast gelegen $PE_{i,j-1}$ wordt een vergelijkingsberekeningsresultaat van $\min(S_{i,j-1}, S_{i-1,j-1})$ ingevoerd en wordt een vergelijkingsberekening $\min\{S_{i-1,j}, \min(S_{i,j-1}, S_{i-1,j-1})\}$ uitgevoerd en wordt D_{ij} verkregen bij stap (i) bij het resultaat van deze berekeningen opgeteld voor het verkrijgen van S_{ij} .
- (iv) Een vergelijkingsberekening $\min(S_{ij}, S_{i-1,j})$ wordt uitgevoerd en het resultaat van deze berekening wordt overgedragen naar de boven naast gelegen $PE_{i,j+1}$ terwijl een cumulatief berekeningsresultaat S_{ij} wordt overgedragen naar de rechts naast gelegen $PE_{i+1,j}$.

De stappen (iii) en (iv) illustreren het proces voor het uitvoeren van de vergelijking en cumulatieve berekeningsvergelijking (2). Dat wil zeggen, van de drie typen cumulatieve resultaten $S_{i,j-1}$, $S_{i-1,j}$, $S_{i-1,j-1}$, welke nodig zijn voor het uitvoeren van de cumulatieve berekeningsvergelijking (2) in PE_{ij} , zijn $S_{i,j-1}$ en $S_{i-1,j}$ respectievelijk aanwezig in $PE_{i,j-1}$, welke rechts naast PE_{ij} ligt en links naast $PE_{i-1,j}$, terwijl $S_{i-1,j-1}$ aanwezig is in een $PE_{i-1,j-1}$, welke diagonaal naast PE_{ij} ligt.

In verband hiermede vereisen de bovengenoemde twee gegevens een enkele overdracht en deze laatste vereist twee overdrachten via $PE_{i,j-1}$. Indien evenwel in de $PE_{i,j-1}$ via welke $S_{i-1,j-1}$ wordt overgedragen, $S_{i,j-1}$ en $S_{i-1,j-1}$ vooraf worden vergeleken en het resultaat van deze vergelijking wordt overgedragen naar PE_{ij} , zodat een vergelijkingsberekening kan worden uitgevoerd met deze informatie en de informatie $S_{i-1,j}$, welke is overgedragen uit $PE_{i-1,j}$, is het resultaat gelijk aan het uitvoeren van een vergelijkingsberekening van de drie gegevens overeenkomstig vergelijking (2) in PE_{ij} .

Deze handelingen maken het mogelijk een cumulatief eindresultaat $S_{i,N}$ te verkrijgen bij het berekenen van de afstand D_{ij} tussen de vectoren en het cumulatieve resultaat S_{ij} door een wijze van uitvoeren van de handelingen in stappen (i)–(iv) voor alle PE's op de diagonalen, aangegeven door getrokken lijnen in figuur 1, als de inhoud van het verwerken in alle PE's, of een werkwijze voor het uitvoeren van de twee typen handelingen in (i) en (iii) en in (ii) en (iv) als de inhoud van een parallelle verwerking door handelingen bij naast elkaar gelegen diagonalen.

In het geval van de laatstgenoemde uitvoeringsmethode verschilt het aantal effectieve dynamische stappen uit de twee parallelle processoreenheden, zodat het aantal uitvoeringsstappen moet worden ingesteld met NOP-instructie (geen bedrijfsinstructie). Een gedetailleerde toelichting hiervan wordt weggelaten.

Figuren 2A en 2B; 3A en 3B; 4A en 4B; en 5A en 5B tonen de handelingen in het orthogonale stelsel in het geval van deze laatste uitvoeringsmethode over een periode vanaf het tijdstip t tot het tijdstip $t+3$. Hier wordt de tijd gemeten in eenheden van tijd, welke voor elke PE nodig is om de gehele verwerking in stappen (i) en (iii), en (ii) en (iv) uit te voeren. Figuren 2A, 3A, 4A en 5A, en 2B, 3B, 4B en 5B tonen respectievelijk de omstandigheden onder welke de informatie, omsloten in de rechthoeken, tijdens de bovenstaande eenheidstijden in elke PE wordt berekend.

Bij een dergelijk orthogonaal stelsel worden het lokalisme en de uniformiteit van de berekening gebruikt om een parallelverwerking mogelijk te maken. Wanneer echter bijvoorbeeld de bovenstaande dynamische tijdvervormingsvergelijking (2) een gecompliceerde berekeningsvergelijking is, zoals die, weergegeven in (4), moet de overdracht van de cumulatieve resultaten $S_{i-1,j-1}$, $S_{i-1,j-2}$, $S_{i-2,j-1}$ van vergelijking (4) en het accepteren van de drie gegevens, welke de onderwerpen zijn van de vergelijkingsberekening, via twee PE's worden uitgevoerd.

De betreffende methoden kunnen de vergelijkingsberekening in PE_{ij} uitvoeren nadat deze drie gegevens

in PE_{ij} zijn ingevoerd. Derhalve wordt de inhoud van de verwerking in elke PE, welke parallel moet worden uitgevoerd, gecompliceerd en bovendien is het niet mogelijk een parallelle verwerking uit te voeren waarbij de PE's met een voldoende rendement worden benut.

$$5 \quad S_{ij} = D_{ij} + \min \begin{pmatrix} S_{i-2,j-1} + 2D_{i-1,j} \\ S_{i-1,j-1} + D_{ij} \\ S_{i-1,j-2} + 2D_{i,j-1} \end{pmatrix} \quad (4)$$

Voorts moet bij de hier beschouwde dynamische tijdvervormingsberekening het aantal PE's worden bepaald in overeenstemming met beide positieve getallen N en I , welke het aantal gegevens in twee typen vectorreeksen voor verwerking uitdrukken, zodat voor het uitvoeren van de dynamische tijdvervormings-
 10 berekeningen van de eenvoudige vectorreeks C_u ($C_u = \{c_1^u, c_2^u, \dots, c_{I_u}^u\}$; waarbij $u = 1, 2, \dots, 1_r$) en een meervoudige vectorreeks R_v ($R_v = \{r_1^v, r_2^v, \dots, r_{N_v}^v\}$; waarbij $v = 1, 2, \dots, 1_r$), de waarden $N_{\max} = (1 \leq v \leq 1_r, N_v)$ en $I_{\max} = (1 \leq u \leq 1_r, I_u)$ moeten worden gekozen voor de positieve gehele aantallen N en I en is het aantal vereiste PE's gelijk aan ($N_{\max} \times I_{\max}$).

Wanneer derhalve de verwerking van de vectorreeksen C_u en R_v wordt uitgevoerd, zal een aantal PE's
 15 aanwezig zijn, dat onnodig is voor het uitvoeren van bewerkingen bij de dynamische tijdvervormingsberekeningsverwerking ten aanzien van alle combinaties van vectorreeksen behalve de combinatie van $C_{\max}$ en $R_{\max}$. Dientengevolge wordt geen doeltreffend gebruik van de onderdelen verkregen.

Bovendien vormt de hoofdzak tot het bepalen van het aantal vereiste PE's overeenkomstig het
 20 maximale aantal gegevens, dat moet worden verwerkt, een grote belemmering voor de reductie in de stelselschaal onder gebruik van LSI-methoden. Het aantal PE's, dat kan worden ondergebracht in een enkel LSI-plaatje, varieert overeenkomstig de functies van de PE's. Wanneer bijvoorbeeld ongeveer vier PE's op een enkel LSI-plaatje zijn gemonteerd en $N_{\max}$ is 60 en $I_{\max}$ is 60, moeten 900 LSI-plaatjes in een orthogonaal netwerk worden ondergebracht en verbonden.

Om dit type conventionele stelselprocessor te regelen, voert elk processorelement (PE) een gelijktijdige
 25 verwerking uit in overeenstemming met instructies, welke worden verstrekt door een gastheer-rekeninrichting.

Bij het bovenbeschreven stelsel zal wanneer de verwerking wordt uitgevoerd voor een bepaalde
 vectorreeks R_v , welke voldoet aan de voorwaarde $N_v < N_{\max}$ ($v=1, 2, \dots, 1_r$) zal uit elk verwerkingselement van PE_{ij} , N_v (een nummer wordt gegeven aan elke PE en de PE in het snijpunt van een kolom i en een rij j)
 30 gelijktijdig een eindresultaat optreden. Aangezien elk van deze PE's zich evenwel op één bepaalde diagonaal bevindt, is de verwerking, welke wordt uitgevoerd overeenkomstig vergelijking (1) door de PE's behoudens de zojuist genoemde PE, die zich op dezelfde diagonaal bevindt, dezelfde als die, welke door deze PE op dezelfde diagonaal wordt uitgevoerd en men kan geen onderscheid maken tussen de inhoud van verwerking bij deze PE's op de diagonalen. Derhalve moet ook het eindresultaat voor deze vectorreeks
 35 R_v uit elk verwerkingselement $PE_{i,N_{\max}}$ worden uitgevoerd en moeten de PE's van kolom ($N_v + 1$) tot kolom $N_{\max}$ naar de functie van het overdragen van deze behandelingsresultaten in hun eigenlijke vorm vervullen. Dit type regeling is nodig voor elke vectorreeks R_v . Aangezien de verdeling van PE's, welke de verwerkingsresultaten, in hun eigen vorm naar elke vectorreeks overdragen, varieert, moet een geschikte regeling plaatsvinden wanneer de gastheer-rekeninrichting de verwerkingsomstandigheden van elke PE regelt. Bij
 40 deze methode worden alle ($I \times N_{\max}$) PE's geregeld en wordt de uitwisseling van regelsignalen tussen elke PE en de gastheer-rekeninrichting en de invoer van instructies naar elke PE bijzonder gecompliceerd.

Figuur 6 toont een verbeterde uitvoeringsvorm van een stelselprocessor voor de bovengenoemde
 vergelijkingen (1), (2) en (3) als een voorbeeld van een aanpassingsberekening, gebaseerd op dynamische tijdvervorming, worden uitgevoerd voor twee typen vectorreeksen C_u en R_v ($u=1, 2, \dots, 1_r, v=1, 2, \dots, 1_r$).

In figuur 6 is aangenomen dat het aantal PE's gelijk is aan n . Met 1 is de enkele PE aangegeven, welke
 45 een rekenkundige eenheid bevat voor het uitvoeren van optellen en aftrekken, vergelijkingsberekeningen en vermenigvuldigings- en optelberekeningen teneinde de dynamische tijdvervormingsberekeningsvergelijkingen (1), (2), en (3) uitvoeren en welke ook is voorzien van registers voor het uitwisselen van informatie met naast gelegen PE's en voor het uitwisselen van informatie met uitwendige bronnen, en een
 50 geheugen voor het opzamelen van berekeningsresultaten en overgedragen informatie. Deze PE's 1 zijn opgesteld in de vorm van een ring.

Met 2-1 tot 2- n zijn multiplexinrichtingen aangegeven, welke worden bestuurd door een hierna te beschrijven regeleenheid 10 voor het kiezen van één van twee situaties. Dat wil zeggen, in één situatie
 55 worden $\underline{n}$ (het aantal PE's) ingangsvectorgegevens vanuit uitwendige bronnen c_i^u ($i=1, 2, \dots, I_u$) op een bepaald moment in het stelsel ingevoerd, en in een andere situatie wordt de ingangsvectorinformatie c_i^u ($i=1, 2, \dots, I_u$) circulair uit een naast gelegen PE overgedragen. Wanneer bij voorbeeld $\underline{n}$ ingangsvector-

informatiereeksen $c_1^u, c_2^u, \dots, c_n^u$ in elke PE in het stelsel vanuit PE₁ worden ingevoerd, kiest slechts de multiplexinrichting 2-1 een ingangsvectorinformatielijn 3 uit een (niet afgebeelde) uitwendige bron, en deze wordt als een ingangspoort voor de ingangsvectorinformatiereeks $c_1^u, c_2^u, \dots, c_n^u$ uit de uitwendige bron gebruikt. Uitgaande van PE₁ als een oorsprongspunt, wordt elk van de n ingangsvectorgegevens $c_1^u, c_2^u, \dots, c_n^u$ aan elk PE toegewezen door de ingangsvectorgegevens c_i^u achtereenvolgens naar de naast gelegen PE's over te dragen. In alle gevallen kiezen alle multiplexinrichtingen 2-1 tot 2-n een informatieoverdrachtsbaan 5 tussen PE's, welke ingangsvectorinformatiereeksen $c_1^u, c_2^u, \dots, c_n^u$ circulair tussen PE's overdraagt. Elke PE is zodanig opgesteld, dat elk verwerkingsresultaat $\text{mod } N$ maal gelijktijdig naar de naast gelegen PE wordt overgedragen parallel met een gebruikelijke parallelverwerking in alle PE's op elk moment, dat de bovenbeschreven n ingangsvectorinformatiepatronen op een hierna te beschrijven wijze worden uitgewisseld. Hierbij drukt de rest uit wanneer N wordt gedeeld door n.

Met 4 I/O-lijn aangegeven voor het achtereenvolgens invoeren van vectorgegevens van andere ingangsvectorinformatiereeksen $r_v = \{r_1^v, r_2^v, \dots, r_{N_v}^v\}$ ($k=1, 2, \dots, 1_n$) in elke PE, naast het uitvoeren van eindresultaten van berekeningen $S_{1_i}, N_1, S_{1_2}, \dots, S_{1_n}, N_v, \dots, S_{1_n}, N_1$. Met 5 is een informatieoverdrachtsbaan aangegeven voor het uitvoeren van een circulaire overdracht van ingangsvectorinformatie c_i^u tussen PE's en voor het uitvoeren van een overdracht van een cumulatief berekeningsresultaat S_{ij} . Met 6 is I/O-klem van elke PE aangegeven, welke met een I/O-lijn is verbonden. Voorts geven 7, 8 en 9 ingangsvectorgegevens c_i^u en r_j^v ($i=1, 2, \dots, I_u; j=1, 2, \dots, N_v; u=1, 2, \dots, 1_c; v=1, 2, \dots, 1_n$) en eindberekeningsresultaten $S_{1_i}, N_1, S_{1_2}, N_2, \dots, S_{1_n}, N_v, \dots, S_{1_n}, N_1$ aan. Met 10 is een regeleenheid aangegeven voor het uitvoeren van regelhandelingen voor het totale stelsel, inclusief het bepalen van het tijdstip van het uitwisselen van de bovengenoemde ingangsvectorinformatie en het tellen van het aantal overdrachten van de verwerkingsresultaten.

Figuur 7 toont een uitvoeringsvorm van een constructie van elke PE. Het door een streep-stippellijn 1 omgeven gedeelte geeft een enkele PE 1 aan. Met 4 is een uitwendige I/O-lijn voor ingangsvectorinformatie r_j^v ($J=1, 2, \dots, N_v$) voor elke PE en voor het uitvoeren van eindberekeningsresultaten S_{1_n}, N_v aangegeven. 12 is een I/O-klem voor het ontvangen/overdragen van informatie uit/naar een uitwendige I/O-lijn 4. Met 13 is een klem voor informatieoverdracht vanuit een links naast gelegen PE aangegeven, terwijl 14 een klem is voor informatieoverdracht naar een rechts gelegen PE.

15 is een bufferregister voor het opslaan van een ingangssignaal van een vectorinformatie r_j uit de uitwendige I/O-lijn 4. 16 is een bufferregister voor het opslaan en uitvoeren van eindberekeningsresultaten S_{1_n}, N_v naar de uitwendige I/O-lijn 4. Met 17 is een register aangegeven voor het opslaan van een ingangssignaal van de vectorinformatie c_{1_u} ($i=1, 2, \dots, I_u$) vanuit een naast gelegen PE en een ingangsgegeven, dat nodig is bij het berekenen van een cumulatieve berekening S_{ij} , welke wordt uitgevoerd in de hierna te beschrijven verwerkingshandelingen (b) en (c). Met 18 is een register aangegeven voor het overdragen van een vectorgegevens c_{1_u} ($i=1, 2, \dots, I_u$) naar een naast gelegen PE en het uitvoeren van de overdracht van informatie, welke nodig is voor het berekenen van de cumulatieve berekening S_{ij} . Met 19 is een inwendige lijn aangegeven.

Met 20 en 21 zijn respectievelijk buffergeheugens aangegeven, welke alle componenten r_k^{vj}, c_k^{ui} ($K=1, 2, \dots, m$) van de vectorinformatie r_j^v en c_i^u , welke in de respectieve PE's moeten worden ingevoerd, opzamelen. Met 22-1 is een optel- en aftrekeenheid aangegeven, welke optel- en aftrekfunctie vervult, terwijl met 22-2 een accumulator is aangegeven om het resultaat uit de optel- en aftrekeenheid 22-1 te accumuleren. Met 22-3 is een product-someenheid aangegeven, welke product-somberekeningsfuncties vervult. Een rekenkundige eenheid 22, gevormd door eenheid 22-1 en 22-3 en de accumulator 22-2 voert de door de vergelijkingen (1) en (2) uitgedrukte berekeningen uit. De optel- en aftrekeenheid 22-1 kan bestaan uit een 54F385 of 74F385, de product-someenheid 22-3 uit een ADSP-1010. Met 23 is een bedrijfsgeheugen aangegeven voor het opzamelen van informatie, welke nodig is voor het uitvoeren van de vergelijkingen (2) en (3). Dit bedrijfsgeheugen 23 bezit twee gebieden 23-1 en 23-2 overeenkomstig de status van de opgeslagen informatie. Dat wil zeggen, dat het gebied 23-1 de informatie opslaat welke nodig is voor het uitvoeren van de verwerkingshandelingen (a), (b) en (c) gedurende een circulaire overdracht van ingangsvectorinformatie c_i^u , welke later zal worden beschreven, terwijl het gebied 23-2 de informatie opslaat, welke nodig is voor het uitvoeren van de verwerkingshandelingen (b) en (c) direct naar het uitwisselen van n vectorreeksen van de vectorreeksen $c_1, c_2, \dots, c_{1_c}$.

Met 24 is een PE-regeleenheid aangegeven, welke een regeling overeenkomstig een inwendig microprogramma of instructies uit een uitwendige bron uitvoert. De PE regeleenheid 24 omvat een logische regelketen 24-1, een programmeergeheugen 24-2 voor het opslaan van het inwendige microprogramma, een programmateller 24-3 voor het regelen van het geheugen 24-2, een instructieregister 24-4 voor het opslaan van informatie, die uit het geheugen 24-2 is uitgelezen, en een instructiedecodeerinrichting 24-5 voor het

decoderen van de in het instructieregister 24-4 opgeslagen informatie. De logische regelketen 24-1 ontvangt regelsignalen uit de regeleenheid 10, weergegeven in figuur 6, via een ingangsklem 25 om de gedeelten 24-2 tot 24-5 te regelen.

- 5 Met 26 en 27 zijn adreslijnen naar het bedrijfsgeheugen 23 aangegeven. De adreslijn 26 ontvangt het uitgangssignaal uit een teller 28, welke toegang heeft tot het gebied 23-2, waarin tussen berekeningsresultaten zijn opgeslagen. De adreslijn 27 komt bij voorbeeld overeen met een direct adres uit het microprogramma, dat uit het instructieregister 24-4 wordt geleverd, en heeft toegang tot het opzamelgebied 23-1 voor het opslaan van informatie, welke nodig is voor de individuele verwerking bij de bovenbeschreven verwerkingshandelingen (b) en (c).
- 10 Met 29, 30, 31 en 32 zijn kiezers aangegeven, welke door de PE-regeleenheid 24 wordt geregeld. Zoals boven is toegelicht, wordt het in de rekenkundige eenheid 22 verschaft berekeningsresultaat in het bedrijfsgeheugen 23 opgeslagen. De registers 17 en 18 dienen voor het overdragen van informatie tussen naast elkaar gelegen PE's, op een zodanige wijze, dat wanneer het bovengenoemde berekeningsresultaat in het register 18 vanuit het bedrijfsgeheugen 23 wordt opgeslagen en daarna het resultaat vanuit
- 15 dit register 18 naar het register 17 van de naast gelegen PE wordt overgedragen, de volgende berekening in de rekenkundige eenheid 22 wordt uitgevoerd. Derhalve is het parallel aan de regelmatige behandelings-eenheid voor het overdragen van ingangsvectorinformatie c_i , welke wordt bepaald door de alinea's (a), (b) en (c), welke later zullen worden beschreven, voor het berekenen van D_{ij} en S_{ij} en voor het overdragen van het resultaat van de berekening, mogelijk de verwerkingsresultaten $\text{mod } N$ maal gelijktijdig op een later te
- 20 beschrijven wijze over te dragen naar de naast gelegen PE's, onder gebruik van het interval gedurende hetwelk elke PE de bovenbeschreven berekeningen uitvoert, zonder dat de normale verwerkingsstroom in zijn geheel wordt gestoord, wanneer van ingangsvectorinformatiepatroon wordt gewisseld.

De tabel toont een uitvoeringsvorm van een programmeergeheugen.

TABEL

Infr. overdracht				Rek. behandeling	
(1)	M21 → R18				
	M21 → R18	T13 → R17			
	M21 → R18	T13 → R17			
	M21 → R18	T13 → R17			
	M21 → R18	T13 → R17			
(2)	M21 → R18	T13 → R17			

M = geheugen T = klem R = Register U = eenheid A = Accumulator

De delen (1) en (2) geven de inhoud van de programma's voor vectorafstandsberekeningen respectievelijk cumulatieve afstandsberekeningen aan. Het deel (1) toont de wijze waarop de vectorafstandsberekeningen parallel worden uitgevoerd, terwijl de respectieve componenten c_i en r_i gelijktijdig worden ingevoerd. Er zal een toelichting worden gegeven op de programma-inhoud van de cumulatieve afstandsberekeningen bij deel (2). De instructies in de eerste en vierde velden van deel (2) komen overeen met cumulatieve afstandsberekeningen voor elk roosterpunt. Het tweede veld komt overeen met de hertoewijzingsverwerking van de cumulatieve resultaten. Het derde veld komt overeen met de verwerking voor het uitvoeren van het cumulatieve eindresultaat naar een uitwendige bestemming.

Er zal een gedetailleerd toelichting worden gegeven door de volgorde van instructies in de eerste en vierde velden voor de verwerkingshandelingen (a), (b) en (c) te vergelijken. ① betekend het invoeren van D_{ij} , terwijl ② en ③ overeenkomen met de overdracht van het vergelijkingsberekenningsresultaat van $\min(S_{i-2, j}, S_{i-1, j})$ en hetingangssignaal van $\min(S_{i-1, j-1}, S_{i, j-1})$. ④ – ⑥ stellen de uitvoeringsvolgorde van de vergelijkingsberekening van $\min\{S_{i-1, j}, \min(S_{i-1, j-1}, S_{i, j-1})\}$ voor. ⑥ – ⑧ stellen de volgorde voor het optellen van D_{ij} bij dit vergelijkingsresultaat voor het verkrijgen van een S_{ij} voor. ⑧ – ⑪ komen overeen met het invoeren van S_{ij} op hetzelfde moment, dat het vergelijkingsresultaat van $\min(S_{i-1, j}, S_{ij})$ wordt verkregen om te worden overgedragen naar de naast gelegen PE.

De logische regelketen 24-1 in de PE-regeleenheid 24 is zodanig uitgevoerd, dat een regelstroom plaatsvindt, als aangegeven in figuur 17. In figuur 17 stellen q_1, q_2, q_3 de namen van regelsignalen voor, die naar de respectieve PE's worden gezonden vanuit een inwendige regeleenheid, zoals een gastheerrekeninrichting, en stellen respectievelijk de ingangstatus van de vectorinformatie r_j , de uitvoeringsstatus van de hertoewijzingsverwerking en de uitgangstatus van het uiteindelijke berekeningsresultaat voor. Wanneer elke PE een programmastartregelsignaal ontvangt, voert de logische regelketen 24-1 de regeling van het invoeren van het initiële adres van het programmeergeheugen 24-1 in de programmateller 24-3 uit. Daarna ontvangt de logische regelketen 24-1 de regelsignalen q_1, q_2 en q_3 voor elke status om deze regelsignalen te controleren voordat de statusvlaggen $f_{5,6}, f_2$ en f_3 worden ingesteld. Hier activeren deze vlaggen $f_{5,6}, f_2$ en f_3 en vlaggen, welke vijfde en zesde activeren, respectievelijk het tweede veld en het derde veld van de programmareeksen (1) en (2). Wanneer deze vlaggen worden ingesteld, wordt de programmateller 24-3 gestart en worden de programma's (1) en (2) uitgevoerd. Tijdens het uitvoeren van deze programma's worden de vlaggen $f_{5,6}, f_2$ en f_3 gecontroleerd en zendt de logische regelketen 24-1 naar het instructieregister 24-4 een regelsignaal voor het maskeren van eventuele velden, welke niet moeten worden uitgevoerd. Vervolgens bepaald de logische regelketen 24-1 het eind van het programma en zendt een regelsignaal naar de uitwendige regeleenheid om de uitwendige regeleenheid te informeren omtrent het eind van de behandelingseenheid.

De logische regeleenheid 24-1 herhaald deze reeks regelhandelingen op elk verwerkingstijdstip.

Vervolgens zal de wijze van het uitvoeren van de dynamische tijdvervormingsberekeningen, aangegeven in de bovengenoemde berekeningsvergelijkingen (1), (2) en (3) met behulp van de in figuur 7 afgebeelde inrichting worden toegelicht. De aanpassingsberekening overeenkomstig de dynamische tijdvervormingsberekening komt overeen met de berekening van de vergelijkingen (1) en (2) voor roosterpunten in twee dimensionale roostervlakken, verkregen door twee typen vectorinformatiereeksen C_U respectievelijk R_V .

Figuur 8 toont een reeks uitvoeringen van dynamische tijdvervormingsberekeningsvergelijkingen (1) en (2) voor de twee typen vectorinformatiereeksen bij deze inrichting, dat wil zeggen de 1_c -vectorinformatiereeksen $C_U = \{c_1^U, c_2^U, \dots, c_n^U\}$ ($U=1, 2, \dots, 1_c$) en de 1_r -vectorinformatiereeksen $R_V = \{r_1^V, r_2^V, \dots, r_{N_V}^V\}$ ($V=1, 2, \dots, 1_r$). De sequentiële uitvoeringen zijn uitgedrukt in de algemene grafische illustratie van dynamische tijdvervorming. In figuur 8 drukken onderbroken diagonalen en ononderbroken diagonalen in het rooster een moment uit in termen van een PE, welke als een tijdeenheid werkt, en geeft een pijl A de richting van het verstrijken van de tijd bij de momenten $t_1, t_2, t_3, \dots$ aan. Met andere woorden betekenen roosterpunten op een zelfde onderbroken of ononderbroken lijn, dat informatie gelijktijdig in de verschillende PE's op de lijn wordt verwerkt. Aangezien er n PE's zijn, worden de n roosterpunten op de diagonaal steeds tijdens het uitvoeren van de verwerking op hetzelfde moment verwerkt. De verwerking op elk verwerkingstijdstip beweegt zich naar boven met het verstrijken van de tijd in een gebied, aangegeven door X_1 en beweegt zich dan continue naar het onderste gedeelte van een gebied X_2 . Tenslotte beweegt de verwerking zich naar het bovenste gedeelte van het gebied X_2 . Derhalve toont figuur 8 een sequentiële verwerking op de wijze van de algemene grafische uitdrukken van dynamische tijdvervorming. Met I en II zijn groepen onderbroken lijnen aangegeven, welke later zullen worden toegelicht.

Bij het in figuur 8 afgebeelde voorbeeld bestaan slechts n PE's fysisch. Dat wil zeggen, dat in figuur 8 slechts de n PE's $PE_1 - PE_n$ aanwezig zijn (welke in de tekening door de omcirkelde getallen 1 – n zijn

aangegeven), die in een schuine richting bij dezelfde tijdpositie zijn gelegen.

Figuur 9 toont de wijze van informatie invoerhandelingen bij een dergelijke inrichting. In figuur 9 is aangenomen, dat $n = 6$. Met 41 is een PE aangegeven en met 42 een informatieoverdrachtsbaan om vectorinformatie C_i ($i=1, 2, \dots, I$) en het cumulatieve resultaat S_{ij} naar een naast gelegen PE over te dragen. Met 43 is een stroom van vectorinformatie c_i ($i=1, 2, \dots, I$) op een informatieoverdrachtsbaan op elk
5 behandelingstijdstip aangegeven. Met 44 is vectorinformatie r_j ($j=1, 2, \dots, N$) op de I/O-lijn aangegeven, welke op elk verwerkingstijdstip in elke PE moet worden ingevoerd. De verwerkingstijd beweegt zich langs een tijdkolom 45 voort, welke het verstrijken van de tijd aangeeft. De ingangsvectorinformatiereeksen $c_1, c_2, \dots, c_6$ voor de zes PE's worden respectievelijk uit PE_1 ingevoerd. Wanneer de verwerking van elk vector-
10 gegeven bij elke PE is voltooid, wordt deze vectorinformatie achtereenvolgens naar een rechts naast gelegen PE overgedragen. Gedurende de periode tot de eerste ingangsvectorinformatie c_1 naar de PE_1 terugkeert, incrementeerd de informatieoverdrachtsbaan, welke de ingangsvectorinformatie c_i ($i=1, 2, \dots, 6$) voert, steeds met één bij het voortschrijden van de verwerkingstijd. Na een moment waarop de ingangsvectorinformatie c_i vanuit PE_6 naar PE_1 is overgedragen, worden de ingangsvectorgegevens $c_1 - c_6$, die in
15 elke PE aanwezig zijn, gelijktijdig op elk verwerkingstijdstip naar een naast gelegen PE overgedragen. Anderzijds wordt ingangsvectorinformatie r_j ($j=1, 2, \dots, N$) achtereenvolgens ingevoerd in elke PE, in synchronisatie met de overdrachtshandelingen voor de ingangsvectorinformatie c_i ($i=1, 2, \dots, 6$) tussen elke PE. Vervolgens wordt de berekening van de vergelijkingen (1), (2) en (3) uitgevoerd voor alle roosterpunten, terwijl informatie regelmatig tussen PE's wordt uitgewisseld.

De stippellijnen in groep I in figuur 8 geven aan, dat slechts de multiplexinrichting 2-1 is ingesteld voor het kiezen van een ingangsvectorinformatielijn uit een uitwendige bron dat een aantal ingangsvectorinformatiereeksen $c_1^1, c_2^1, \dots, c_n^1$ equivalent aan het aantal PE's sequentieel is ingevoerd, en dat PE_2-PE_n een
20 vectorinformatie c_x^1 ($x=1, 2, \dots, n-1$) gelijktijdig met de naast gelegen PE's uitwisselen op elk tijdstip, dat de verwerking in elke PE is voltooid. De getrokken lijnen in groep I, die zich vanuit de stippellijnen in deze groep I voortzetten, geven aan, dat nadat de informatie c_i^1 in de PE_n is ingevoerd, alle multiplexinrichtingen 2-1 tot 2-n als informatieoverdrachtsbanen tussen PE's worden gekozen en dat berekeningsvergelijkingen (1), (2) en (3) worden uitgevoerd, terwijl ingangsvectorinformatie $c_1^1, c_2^1, \dots, c_n^1$ circulair vanuit PE naar PE
25 wordt overgedragen. Voorts geven de stippellijnen in de volgende groep II het proces van een voortgezette berekening aan, dat wordt uitgevoerd wanneer de ingangsvectorinformatie $c_1^1, c_2^1, \dots, c_n^1$ wordt vervangen door de volgende n vectorinformatiereeksen $c_{n+1}^1, \dots, c_1^2, \dots, c_n^2$.

Ofschoon twee typen vectorinformatie c_i^u en r_j^v gedurende elk verwerkingstijdstip in elke PE worden ingevoerd, zodat de berekeningsvergelijking (1) onafhankelijk van en parallel aan elke PE wordt uitgevoerd, wordt de berekeningsvergelijking (2) uitgevoerd wanneer informatie met de naast gelegen PE's wordt
30 uitgewisseld.

Zo is in figuur 10 het aantal PE's, gelijk aan $n=5$. Figuur 10 toont de verwerkingsprocedure in elke PE en de verdeling van de roosterpunten, onder controle van elke PE, wanneer een continue verwerking wordt uitgevoerd voor alle combinaties van vectorinformatiereeksen C_1 en C_2 en vectorinformatiereeksen R_1 en R_2 . In figuur 10 worden de groepen roosterpunten, die in een langwerpige cirkel zijn opgesloten, door
40 dezelfde PE verwerkt en het nummer bij de linker bovenhoek van deze cirkel geeft het PE-nummer aan.

In het geval van het verkrijgen van bij voorbeeld $S_{7,8}$ in figuur 10 is de informatie, welke nodig is voor het berekenen van $S_{7,8}$ op het tijdstip t_1 $S_{6,7}$, $S_{7,7}$ en $S_{6,8}$, verkregen door PE_4 en PE_5 . Aangezien de tijdstippen t_2 en t_3 in het verleden liggen ten opzichte van het tijdstip t_1 is de informatie $S_{6,8}$ in PE_5 aanwezig, welke $S_{7,8}$ berekend, en is de informatie $S_{6,7}$ en $S_{7,7}$ aanwezig in PE_4 . Met andere woorden is de vereiste
45 informatie steeds aanwezig in de naast gelegen PE, zodat in het geval van het uitvoeren van een vergelijkingsberekening van vergelijking (2) voor $S_{7,8}$, $\min(S_{6,7}, S_{7,7})$ wordt uitgevoerd in PE_4 en het resultaat wordt overgedragen naar PE_5 , waar $\min\{S_{6,8}, \min(S_{6,7}, S_{7,7})\}$ wordt uitgevoerd.

In dit geval moet, zoals boven is toegelicht, op dat de ingangsvectorinformatiereeksen $C_1, C_2, \dots, C_{16}$ worden gesplitst in een aantal groepen, equivalent aan het aantal PE's (n) en in het stelsel worden
50 uitgevoerd voor verwerking, S_{ij} , welke overeenkomt met de roosterpunten, gemarkeerd door een schuine lijn als aangegeven in figuur 10, worden overgedragen naar een vooraf bepaalde PE voordat de substitutie van de ingangsvectorinformatie begint. Zo is het nodig, dat de $S_{5,1}$, welke aanwezig is in PE_1 , naar PE_3 wordt overgedragen, $S_{5,2}$ in PE_2 naar PE_4 , $S_{5,3}$ in PE_3 naar PE_5 , $S_{5,4}$ in PE_4 naar PE_1 respectievelijk $S_{5,5}$ in PE_5 naar PE_2 .

In het algemeen beginnen vanaf het tijdstip, dat met de tijd overeenkomende $(\text{mod } v \sum_{v=1}^1 N_v)$
55 overeenkomt tot het tijdstip waarop het vervangen van n vectorinformatiereeksen begint, dat wil zeggen bij het in figuur 10 afgebeelde voorbeeld vanaf het tijdstip voorafgaande aan het tijdstip, overeenkomende

met mod^{17} tijdeenheden, alle PE's handelingen voor het gelijktijdig naar de respectieve naast gelegen PE's overdragend van de cumulatieve resultaten $S_{pn,j}$ ($p=1, 2, \dots$), die steeds in de respectieve PE zijn opgeslagen. De overdracht van deze informatie geschiedt één voor één voor elke verwerking van de PE's onder gebruik van de periode gedurende welke de PE's slechts de berekeningsverwerking, als boven beschreven, parallel met de normale verwerkingshandelingen volgens de alinea's (a), (b) en (c) welke later zullen worden beschreven, uitvoeren, zodat het mogelijk is de vereiste informatie $S_{pn,j}$ naar een vooraf bepaalde PE over te dragen en wel onmiddellijk vóór de substitutie van n ingangsvectorinformatiereeksen.

Bij het in figuur 10 afgebeelde voorbeeld worden parallel met de verwerking van de roosterpunten (c_1^1, r_6^1) in de PE_1 , de informatie $S_{5,1}, S_{5,2}, S_{5,3}, S_{5,4}, S_{5,5}$, welke aanwezig is in dezelfde adressen in de bedrijfsgeheugens 23-2 voor PE_1, PE_2, PE_3, PE_4 en PE_5 overgedragen naar de naast gelegen PE's en toegewezen aan $PE_2, PE_3, PE_4, PE_5, PE_1$. Op een soortgelijke wijze wordt bij de verwerking van het roosterpunt (c_2^1, r_6^1) van PE_1 de informatie toegewezen aan $PE_3, PE_4, PE_5, PE_1, PE_2$ en vervolgens is de informatieoverdracht voltooid.

Bij de verwerking onmiddellijk na de substitutie met de volgende ingangsvectorinformatiereeksen $c_3^1, c_6^1, c_1^1, c_2^1$ welke plaatsvindt op het verwerkingstijdstip voor het volgens roosterpunt (c_3^1, r_6^1) van PE_1 , worden de verwerkingshandelingen (a), (b) en (c) uitgevoerd door PE_3, PE_4, PE_5, PE_1 en PE_2 onder gebruik van de informatie $S_{5,1}, S_{5,2}, S_{5,3}, S_{5,4}$ en $S_{5,5}$, welke door de bovengenoemde twee overdrachten wordt verkregen. Door een herhaald uitvoeren van deze wijze van verwerking, kan elke PE de dynamische tijdsvervormingsberekeningsvergelijkingen (1), (2) en (3) regelmatig en sequentieel uitvoeren.

Zoals zojuist is beschreven voert terwijl de invoer of substitutie en circulaire overdracht van n vectorinformatiereeksen van de ingangsvectorinformatiereeksen C_u ($u=1, 2, \dots, 1_0$) en vectorinformatie r_j^v ($j=1, 2, \dots, N_0$) in elke PE wordt ingevoerd in synchronisme met de invoer en circulaire overdracht van bovengenoemde vectorinformatie c_i^u ($i=1, 2, \dots, 1_0$) steeds wisselt, elke PE een aantal malen de berekeningsvergelijkingen (1), (2) en (3) voor elk roosterpunt uit, zodat de verwerking voor alle roosterpunten wordt voltooid.

Het bovenstaande resumerende zijn de normale verwerkingshandelingen (de gebruikelijke verwerking) voor een PE, bij het uitvoeren van de vergelijkingen (1) en (2) de volgende:

(a) Op hetzelfde moment dat de vectorinformatie c_i ($i=1, 2, \dots, 1$) vanuit een links naast gelegen PE of vanuit een uitwendige bron via een ingangsvectorinformatielijn wordt ingevoerd, wordt de vectorinformatie c_{i-1} in de rechts naast gelegen PE overgedragen, wordt de vectorinformatie r_j ($j=1, 2, \dots, N$) vanuit de I/O-lijn synchroon met de overdracht van de vectorinformatie ingevoerd, zodat de bovenstaande berekeningsvergelijking (1) wordt uitgevoerd voor het verkrijgen van D_{ij} .

(b) Vergelijkingsberekening van $\min \{S_{i-1,j}, \min (S_{i-1,j-1}, S_{i,j-1})\}$ wordt uitgevoerd en D_{ij} wordt bij dit resultaat opgeteld voor het verkrijgen van S_{ij} .

(c) De vergelijkingsberekening van $\min (S_{i-1,j}, S_{ij})$ wordt uitgevoerd en het resultaat wordt overgedragen naar de naast gelegen PE. Tegelijkertijd wordt het resultaat van de vergelijkingsberekening $(S_{i,j-1}, S_{i+1,j-1})$ vanuit de links naast gelegen PE ingevoerd.

De handelingen in (a) komen overeen met het uitvoeren van de berekeningsvergelijking (1) en die in de alinea's (b) en (c) komen overeen met het uitvoeren van de berekeningsvergelijkingen (2) en (3). Elke PE voert de verwerkingshandelingen gelijktijdig in de reeks van (a), (b) en (c) uit. Met andere woorden voeren wanneer (a) wordt uitgevoerd alle PE's (a) uit. Wanneer (b) wordt uitgevoerd, voeren alle PE's (b) uit.

Het fundamentele verschil tussen deze werkingen en de werkingen van een orthogonaal stelsel zijn gelegen in de informatieoverdrachthandeling bij het uitvoeren van vergelijking (2). Bij de werking van een orthogonaal stelsel vindt de vergelijkingsberekening van $\min (S_{i,j-1}, S_{i,j}, S_{i-1,j-1})$ plaats nadat het cumulatieve resultaat $S_{i-1,j-1}$ naar de links naast gelegen PE is overgedragen, terwijl bij deze werkingen $S_{i-1,j-1}$ en $S_{i,j-1}$, welke op het volgende verwerkingstijdstip moeten worden verkregen, aanwezig zijn in dezelfde PE, zodat de vergelijkingsberekening kan worden uitgevoerd zelfs onder dat informatie wordt overgedragen.

Wanneer vergelijking (4) wordt uitgevoerd, kunnen de cumulatieve resultaten normaliter op dezelfde wijze als boven is beschreven worden verkregen door het uitvoeren van twee typen eenvoudige handelingen. Namelijk een eenvoudige werking van (d) het invoeren van het cumulatief resultaat vanuit de naast gelegen PE, het bij deze informatie optellen van het dubbele van de waarde van de intervectorafstand, welke in die PE wordt uitgevoerd, en het uitvoeren van dit opgetelde resultaat naar de naast gelegen PE, en een werking van (e) het invoeren van het cumulatieve resultaat uit de naast gelegen PE, het optellen van de intervectorafstand bij deze informatie en het opslaan van het resultaat. Zoals boven is toegelicht, maakt deze uitvoering het aantal PE's volledig onafhankelijk van de positieve getallen 1_0 en N_0 , welke aantal vectorgegevens uitdrukken, dat moet worden verwerkt, en maakt het mogelijk, dat het aantal PE's op een

geschikte wijze wordt ingesteld overeenkomstig het geschatte volume van de informatieverwerking, en maakt het mogelijk een dynamische tijdvervormingsberekening uit te voeren via "pipelineparallel verwerking", waarbij een maximaal doeltreffend gebruik wordt gemaakt van de onderdelen doordat de PE's, welke de normale verwerkingshandeling een aantal malen uitvoeren, volledig in bedrijf zijn.

- 5 Wanneer derhalve LSI-methoden worden gebruikt voor het realiseren van een stelselprocessor, kan de stelselschaal bijzonder klein worden gemaakt vergeleken met het conventionele orthogonale stelsel, waarin het aantal PE's wordt bepaald in overeenstemming met de positieve gehele getallen I_u en N_v , terwijl bovendien de onderdelen op een effectieve wijze kunnen worden gebruikt. Voorts is het mogelijk het aantal PE's te vergroten. Dat wil zeggen, dat de verwerking van vectorinformatiereeksen met willekeurig gewenste

10 aantal N_v en I_u kunnen worden verwerkt, onafhankelijk van het aantal PE's.
Nu zullen de rendementen van het orthogonale stelsel en de stelselprocessor voor de bit per PE en per eenheid van tijd worden vergeleken waarbij rekening wordt gehouden met een gemiddeld gebruik van de PE's.

- 15 Van de twee typen verwerkingsinhouden in het orthogonale stelsel bestaande uit de bovenbeschreven verwerkingshandelingen (i) en (iii), en (ii) en (iv) wordt aangenomen, dat het aantal stappen, dat van deze twee typen het grootst is, U_{square} is, en voorts wordt aangenomen, dat het aantal dynamische stappen bij de verwerking van elke PE in het stelsel bestaande uit de verwerkingshandelingen (a), (b) en (c) gelijk is aan U_{ring} . Bij het orthogonale stelsel zijn voor het voltooiën van de aanpassingsberekening, gebaseerd op de

- 20 afwisselend uitvoeren van de twee verwerkingseenheden (i) en (iii), en (ii) en (iv), $2U_{\text{square}}$ stappen nodig.
Bij de aanpassingsberekening, gebaseerd op de betreffende dynamische tijdvervorming, behoeft indien PE_{ij} het uitvoeren van de berekeningsvergelijkingen (1), (2) en (3) voor een enkele vectorinformatiereeks R voltooit voor het verkrijgen van het cumulatieve resultaat S_{ij} , PE_{ij} niet de verwerking voor deze vectorinformatiereeks R uit te voeren wanneer $PE_{i'j'}$ ($i' > 1$, $j' > j$) de bovengenoemde berekeningsvergelijkingen
- 25 uitvoert. Derhalve kan tijdens het uitvoeren van een verwerking voor een vectorinformatiereeks R_v , een PE, welke niet tot die verwerking bijdraagt, worden toegewezen aan de verwerking van een andere vectorinformatiereeks R_v . Met andere woorden kan wanneer het cumulatieve resultaat S_{ij} wordt berekend voor een eerste vectorinformatiereeks R_1 de berekening van het cumulatieve resultaat S_{ij} ook worden uitgevoerd voor een tweede vectorinformatiereeks R_2 via het faseverschil van $2U_{\text{square}}$ stappen. Totdat het uiteindelijke

- 30 berekeningsresultaat S_{I_u, N_v} voor de vectorinformatiereeks I_u en voor de vectorinformatiereeks R_v is verkregen, is het aantal stappen $(N_{\text{max}} + I_{\text{max}})$, gebaseerd op een eenheid van het aantal dynamische stappen $2U_{\text{square}}$, dat nodig is voor het verkrijgen van S_{ij} vereist, zodat $(N_{\text{max}} + I_{\text{max}})$ typen uiteindelijke cumulatieve resultaten S_{I_u, N_v} in de periode van $(N_{\text{max}} + I_{\text{max}})$ stappen kunnen worden verkregen.

- 35 Anderzijds wordt de verwerking herhaald, hetgeen de aanpassingsverwerking is tussen de ingangsvectorinformatiereeks $R_1, R_2, \dots, R_1$, en elke n vectorinformatiereeks van de ingangsvectorinformatiereeksen $C_1, C_2, \dots, C_{1_c}$, zodat het uiteindelijke cumulatieve resultaat S_{I_u, N_v} wordt verkregen.

- Op basis van de bovenbeschreven verwerkingshandelingen van het totale stelsel zal het rendement van de PE's, bij het uitvoeren van een verwerking voor alle combinaties van de vectorinformatiereeksen $C_1, C_2, \dots, C_{1_c}$ en de vectorinformatiereeksen $R_1, R_2, \dots, R_1$, als volgt zijn:

- 40 In het geval van het orthogonale stelsel:
Voor het verkrijgen van de eindresultaten van $1_r.1_c$ zijn $(N_{\text{max}} + I_{\text{max}} + 1_r.1_c)$ stappen met een eenheid van $2U_{\text{square}}$ nodig. Aangezien het aantal PE's gelijk is aan $N_{\text{max}}.I_{\text{max}}$, wordt het rendement van PE gegeven door de volgende vergelijking:

$$45 \quad \eta_{\text{sq}} = \frac{1_r.1_c}{(N_{\text{max}} + I_{\text{max}} + 1_r.1_c).2U_{\text{square}}} \cdot N_{\text{max}}.I_{\text{max}} \quad (5)$$

In het geval van de verbeterde stelselprocessor:

- 50 Gebaseerd op U_{ring} als eenheid van verwerkingsstap is het aantal verwerkingsstappen tijdens de werking voor de substitutie van n ingangsvectorinformatiereeksen n stappen en is het aantal verwerkingsstappen, dat moet worden uitgevoerd wanneer de ingangsvectorinformatie circulair wordt overgedragen ($\sum_{v=1}^n N_v.n$) stappen. Het uitvoeren van de verwerking van de ingangsvectorinformatiereeksen $C_1, C_2, \dots, C_{1_c}$ geschiedt op een wijze equivalent aan de verwerking van een enkele ingangsvectorinformatiereeks, zodat het aantal
- 55 stappen, dat nodig is voor het verkrijgen van de uiteindelijke resultaten van $1_r.1_c$ wordt gegeven door de volgende vergelijking:

$$\left(\sum_{v=1}^{1_r} N_{v-n} \right) \left(\frac{\left(\sum_{u=1}^{1_c} I_u - r_1 \right)}{n} + 1 \right) + n \left(\frac{\left(\sum_{u=1}^{1_c} I_u - r_1 \right)}{n} - 1 \right) + 2n \quad (6)$$

De eerste term van vergelijking (6) stelt het aantal stappen tijdens de circulaire overdracht voor. De tweede term stelt het aantal stappen tijdens de ingangsvectorinformatie substitutie voor. De derde term stelt het aantal stappen bij de eerste en laatste stadia van de gehele aanpassingsverwerking voor. Voorts is

$r_1 = \text{mod} \sum_{u=1}^{1_c} I_u$. Aangezien er n PE's zijn, is het rendement η_{ring} van de PE:

$$\eta_{\text{ring}} = \frac{I_r \cdot I_c}{\left(\sum_{v=1}^{1_r} N_v \right) \left(\frac{\left(\sum_{u=1}^{1_c} I_u - r_1 \right)}{n} + 1 \right) + n \left(\frac{\left(\sum_{u=1}^{1_c} I_u - r_1 \right)}{n} - 1 \right) + 2n} \cdot U_{\text{ring}} \quad (7)$$

Indien N_{av} het gemiddelde aantal van $N_1, N_2, \dots, N_{1_c}$ is en I_{av} het gemiddelde aantal van $I_1, I_2, \dots, I_{1_c}$ is,

geldt $\sum_{v=1}^{1_r} N_v = 1_r \cdot N_{av}$, $\sum_{u=1}^{1_c} I_u = 1_c \cdot I_{av}$, zodat vergelijking (7) als volgt kan worden herschreven:

$$\frac{\eta_{\text{ring}}}{\eta_{\text{sq}}} = \frac{N_{\text{max}}}{N_{av}} \cdot \frac{I_{\text{max}}}{I_{av}} \cdot \frac{1 + \frac{N_{\text{max}} + I_{\text{max}}}{1_r \cdot 1_c}}{1 + \frac{n - r_1}{1_c I_{av}} + \frac{n^2}{1_c I_{av}}} \cdot \frac{2U_{\text{square}}}{U_{\text{ring}}} \quad (8)$$

De noemers en tellers afgezien van 1 in de derde term van vergelijking (8) hebben betrekking op het rendement van elk stelselconfiguratie bij de eerste en laatste stadia van de gehele aanpassingsverwerking. Derhalve wordt de verhouding van de rendementen van de PE's tijdens het uitvoeren van de verwerking als volgt uitgedrukt:

$$\frac{\eta_{\text{ring}}}{\eta_{\text{sq}}} = \frac{N_{\text{max}}}{N_{av}} \cdot \frac{I_{\text{max}}}{I_{av}} \cdot \frac{2U_{\text{square}}}{U_{\text{ring}}} \quad (9)$$

Indien wordt aangenomen, dat elke PE in een orthogonaal stelsel is voorzien van organen om een gelijktijdig uitvoeren van in- en uitvoerhandelingen mogelijk te maken, dan is $2U_{\text{square}} = U_{\text{ring}}$ en $N_{\text{max}} > N_{av}$, $I_{\text{max}} > I_{av}$, zodat het beschreven stelsel steeds voorziet in een verbeterd rendement, dat groter is dan dat van het orthogonale stelsel. Wanneer bij voorbeeld $N_{av} = \frac{3}{4}N_{\text{max}}$ en $I_{av} = \frac{3}{4}I_{\text{max}}$, dan heeft het beschreven stelsel een rendement, dat bij benadering 1,8 maal zo groot is dan dat van het orthogonale stelsel. Indien bovendien het conventionele stelsel is voorzien van organen voor het slechts uitvoeren van in- en uitvoerhandelingen afwisselt gedurende elke verwerking in elke PE, dan is $2U_{\text{square}} < U_{\text{ring}}$. Dit betekent, dat het relatieve rendement van de stelselprocessor nog groter is vergeleken met dat van het orthogonale stelsel.

De stelselschaalafmeting in het geval van het orthogonale stelsel is bijzonder groot geweest ten gevolge van de noodzaak om ten minste ($N_{\text{max}} \times I_{\text{max}}$) PE's aan te brengen en met elkaar te verbinden. Derhalve is bij de conventionele stelsels een serie bitmethode toegepast voor de invoer en uitvoer van elke PE teneinde de afmetingen van de PE compact te maken. Omdat echter de bij de aanpassingsberekening in hier beschouwde dynamische tijdvervorming te verwerken informatie vectorinformatie is, waarin de informatiereeks van een bepaalde afmeting een enkel gegeven is, aangegeven in vergelijking (1) neemt het aantal overdrachtsschakelingen tussen PE's extreem toe indien de informatie invoer/uitvoer wordt uitgevoerd in de vorm van serie bitstromen en derhalve vereisen de totale berekeningen een langere tijd. Anderzijds kan bij het hier beschreven stelsel het aantal PE's op een significante wijze worden gereduceerd, zodat zich geen problemen voordoen ten aanzien van de afmetingen van de stelselprocessor, zelfs indien tussen PE's parallel informatieoverdracht wordt toegepast. Het beschreven stelsel is zeer geschikt voor het verwerken van vectorinformatie, zoals die, welke wordt verwerkt bij de aanpassingsberekening, gebaseerd op de hier beschouwde dynamische tijdvervorming.

Ofschoon de bovenstaande toelichting in hoofdzaak is gericht op de dynamische tijdvervormingsberekeningen, aangegeven in vergelijkingen (1), (2) en (3), is de stelselprocessor niet tot deze berekeningen beperkt, doch kan deze op dezelfde wijze worden toegepast op het geval als boven beschreven, waarbij bij voorbeeld vergelijking (2) wordt gesubstitueerd door vergelijking (4) of anderzijds, voor het uitvoeren van berekeningen voor alle combinaties van twee typen variabelen en voor een recursieve formulevergelijking met lokale afhankelijkheid van informatie, waarbij deze berekeningsresultaten worden gebruikt.

Bij een stelselprocessor volgens bovenstaande beschrijving zijn verwerkingselementen met vooraf bepaalde I/O-organen en berekeningsorganen in de vorm van een ring verbonden over multiplexinrichtingen, die één informatieoverdrachtsbaan voor het uitwisselen van informatie met naast gelegen verwerkings-

elementen en een uitwendige ingangslijn kiezen. Voorts zijn deze verwerkingselementen zodanig opgesteld, dat de gelijktijdige overdracht van de respectieve verwerkingsresultaten door alle verwerkingselementen naar naast gelegen verwerkingselementen een vooraf bepaald aantal malen parallel plaatsvindt met de gebruikelijke verwerking in elk verwerkingselement. Deze constructie maakt het mogelijk berekeningen voor
 5 alle combinaties van twee typen variabelen en voor een willekeurige recursieve formulevergelijking met lokale afhankelijkheid van informatie, waarbij deze berekeningsresultaten worden gebruikt, te realiseren, welke laatste typerend zijn voor de aanpassingsberekening van dynamische tijdsvervorming met een stelselconstructie, opgebouwd uit geschikte aantal verwerkingselementen, overeenkomende met de hoeveelheid berekeningen, die moeten worden uitgevoerd door een parallelverwerking met hoog rendement,
 10 waarbij alle verwerkingselementen met een hoog rendement werken.

Bij een berekeningsverwerking zoals uitgedrukt door de vergelijkingen (1) – (3) is S_i, N als het eindverwerkingsresultaat nodig. Een regelmethode van een stelselprocessor zal nu worden toegelicht onder verwijzing naar het geval, waarin de bovengenoemde dynamische tijdsvervormingsberekeningen worden uitgevoerd ten aanzien van alle combinaties van een enkele vectorreeks C en 1 vectorreeks R_p ($p=1, 2, \dots$,
 15 1) om als een eindresultaat S_i^p, N_p te verkrijgen.

Figuur 11 toont een uitvoeringsvorm van een constructie van een stelselprocessor waarin de verwerkingselementen (PE's) in de vorm van een ring zijn opgesteld en met elkaar zijn verbonden. In figuur 11 is aangenomen, dat aantal PE's gelijk is aan n ($n \leq N_p$).

In figuur 11 is 51 een PE, welke een rekenkundige eenheid bevat voor het uitvoeren van optellen en
 20 aftrekken, vergelijkingsberekeningen en product-somberekeningen, welke nodig zijn om de bovengenoemde vergelijking (1) op te lossen. Deze PE 51 bezit een register voor het uitvoeren van een informatie-uitwisseling met een naast gelegen PE en een informatieuitwisseling met een uitwendige bron, een geheugen voor het opslaan van resultaten van berekeningen en overgedragen informatie en een regelaar voor het regelen van elke sectie van een microprogrammegeheugen, dat PE verwerkingshandelingen
 25 omvat. Met 52-1 tot 52- n zijn multiplexinrichtingen aangegeven, die een uitwendige lijn 53 kiezen, welke n (overeenkomende met het aantal PE's) ingangsvectorgegevens C_i vanuit een uitwendige bron op een bepaald moment invoert, en een informatieoverdrachtsbaan 55, welke ingangsvectorinformatie (C_i) vanuit een naast gelegen PE overdraagt. Met 54 is I/O-lijn aangegeven, welke wordt gebruikt bij het invoeren van ingangsvectorinformatie r_j^p en bij het uitvoeren van de eindresultaten van de berekeningen S_i^p, N_p naar een
 30 uitwendige bestemming. De informatieoverdrachtsbaan 55 draagt de informatie C_i en tussenresultaten van de berekeningen S_{ij}^p ($j=1, 2, \dots, N_p-1$) tussen de PE's over. Met 56 zijn I/O-klemmen van elke PE aangegeven, die met I/O-lijn 54 zijn verbonden. Met 57 is een totaal regelstelsel, zoals een gastheerrekeninrichting aangegeven, welke aan elke PE-regelsignalen 58 toevoert. De verwijzingen 59, 60 en 61 hebben respectievelijk betrekking op een ingangsvectorinformatiereeks C , een ingangsvectorinformatiereeks R_p ($p=1, 2, \dots$,
 35 1) en eindberekeningsresultaat S_i^p, N_p ($p=1, 2, \dots, 1; i=1, 2, \dots, l$).

Bij een stelselprocessor met een dergelijke constructie is de volgorde voor het uitvoeren van de op de bovenstaande vergelijkingen (1) tot (3) gebaseerde dynamische tijdsvervormingsberekeningen als volgt.

Een informatiereeks, verkregen door een ingangsvectorinformatiereeks C te splitsen in reeksen van n gegevens (gelijk aan het aantal PE's) wordt uitgedrukt als volgt:

$$40 \quad S_k = (C_{kn+1}, C_{kn+2}, \dots, C_{kn+q}, \dots, C_{(k+1)n})$$

waarbij $k=0, 1, 2, \dots; q=1, 2, \dots, n-1$ en $kn + q \leq l$. De verwerkingsvolgorde omvat het herhaaldelijk uitvoeren van de volgende (I) – (V) procedures voor $k=0, 1, \dots$

- (I) De multiplexinrichting voor de PE, welke zich bevindt in de $\text{mod}(\sum_{p=1}^l N_p)$ -de volgorde telling vanaf de PE
 45 waarin de verwerking tussen de ingangsvectorinformatiereeks S_{k-1} en alle ingangsvectorinformatiereeksen R_p ($p=1, 2, \dots, 1$) begint, wordt naar de uitwendige lijn geschakeld om het ingangssignaal van een volgende ingangsvectorinformatiereeks S_k op te nemen. Gesynchroniseerd met deze informatie invoer begint de berekeningsverwerking voor alle ingangsinformatiereeksen R_p , terwijl de ingangsvectorinformatie r_j^p achtereenvolgens in elke PE wordt ingevoerd. Wanneer $k=0$ wordt de multiplexinrichting voor een PE naar
 50 de uitwendige lijn omgeschakeld en deze verwerking wordt uitgevoerd.
- (II) De ingangsvectorinformatiereeks S_k wordt in het stelsel ingevoerd en wanneer de informatie, waaruit S_k bestaat, aan de verschillende PE's wordt toegewezen, kiezen alle multiplexinrichtingen de baan voor de informatieoverdracht tussen de PE's. Tegelijkertijd wordt informatie circulair tussen PE's in alle PE's overgedragen en terwijl de ingangsinformatie r_j^p achtereenvolgens in elke PE wordt ingevoerd, gesynchroniseerd met de overdracht van deze informatie, wordt de berekeningsverwerking, gebaseerd op de bovenstaande vergelijkingen (1) – (3) voor alle ingangsinformatiereeksen R_p uitgevoerd.
- (III) Bij de verwerking volgens de procedures (I) en (II) wordt het berekeningsresultaat S_i^p, N_p voor elke

ingangsvectorinformatiereeks R_p naar een uitwendige bestemming uitgevoerd.

(IV) Bij de verwerking volgens de procedures (I) en (II) wordt het berekeningsresultaat $S_{(k+1)n,j}$ voor de laatste informatie $c_{(k+1)n}$ in elke ingangsvectorinformatiereeks S_k opgeslagen in het inwendige geheugen van de PE, welke die berekening uitvoert.

- 5 (V) Bij de verwerking volgens de procedures (I) en (II) $\sum_{p=1}^{mod} N_p$ verwerkingstijden voor (de verwerkingstijd, welke indicatief is voor de tijd, die elke PE nodig heeft om de verwerking in elke PE uit te voeren) het tijdstip waarop de substitutie van ingangsinformatiereeks S_k aanvangt om te worden uitgevoerd, draagt elke PE tegelijkertijd een enkel deel van het berekeningsresultaat, dat in het inwendige geheugen daarvan is
10 opgeslagen, een maal op elk verwerkingstijdstip over. Dat wil zeggen, dat de overdracht in zijn totaal $\sum_{p=1}^{mod} N_p$ maal geschiedt. Voorts wordt tijdens de circulaire overdracht van de ingangsvectorinformatie c_i de overdracht van de berekeningsresultaten tussen de PE's in alle PE's bij elke $\{n - \sum_{p=1}^{mod} N_p\}$ behandelings-
15 tijdstip herhaald. Een enkel berekeningsresultaat voor elke PE wordt éénmaal per behandelingstijd overgedragen dat wil zeggen in zijn totaal $\sum_{p=1}^{mod} N_p$ overdrachten.

Een regelmethode om een dergelijke verwerking te regelen zal worden uiteengezet in de onderstaande alinea (A) en (B).

- 20 Figuur 12 toont een voorbeeld van vlaggen, welke aan de ingangsvectorinformatie kunnen worden toegevoegd om de regelmethode te realiseren. Met 61 en 62 zijn vlaggen aangegeven, die de kop en de staart van elke informatiereeks aangeven. Met 63 zijn vlaggen aangegeven voor het instellen van de tempering voor een gelijktijdige overdracht van de berekeningsresultaten van alle PE's onmiddellijk vóór de substitutie van de ingangsvectorinformatiereeks S_k en 64 geeft een ingangsvectorinformatie r_j^p aan.

- 25 (A) Zoals aangegeven in figuur 12, worden voor elke ingangsvectorinformatiereeks R_p , de drie bits van vlaggen 61 en 62 voor het aangeven van de kop en de staart van elke informatiereeks en de vlag 63 voor het instellen van de tempering voor de gelijktijdige overdracht van de berekeningsresultaten van alle PE's onmiddellijk vóór de substitutie van de ingangsvectorinformatiereeks S_k opgeteld bij de overeenkomstige ingangsvectorinformatie r_j^p .

- 30 (B) Elke PE heeft een statusvlagregister voor het opslaan van de informatie om de functie te beoordelen, welke elke PE moet vervullen bij het verwerken van het totale stelsel, zoals de ingangstoestand van de ingangsvectorinformatie r_j^p en de telhandelingen voor het aantal gelijktijdige overdrachten (hierna betiteld als hertoewijzingsverwerking) van de berekeningsresultaten over alle PE's.

- 35 Figuur 13 toont een voorbeeld van een regelwerking van een statusvlagregister in een PE. De figuur toont een werking voor het achtereenvolgens in elke PE invoeren van ingangsvectorinformatie r_j^p . Met 66 is een statusvlagregister aangegeven, dat bestaat uit een ingangsvlag 67, welke een ingangstoestand van ingangsvectorinformatie r_j^p specificeert en een telvlag 68 voor het tellen van het aantal overdrachten, dat bij de hertoewijzingsverwerking nodig is.

- 40 Met 51 is de PE aangegeven. Met 54 is de uitwendige I/O-lijn. Met 55 is de baan voor de informatieoverdracht tussen de PE's aangegeven.

- 45 De inhoud van het statusvlagregister 66 wordt op elk verwerkingstijdstip naar een naast gelegen PE overgedragen en vervult de functie van het instellen van de respectieve status in elke PE voor het begin van de verwerking op het volgende tijdstip. De ingangsvectorinformatie r_j^p wordt ingevoerd in de PE waarvan de vlag 67 gelijk is "1". De inhoud van het statusvlagregister 66 wordt op elk verwerkingstijdstip achtereenvolgens overgedragen naar een naast gelegen PE, zodat de ingangsvectorinformatie r_j^p achtereenvolgens in elke PE wordt ingevoerd. Op dit moment wordt de ingangsvectorinformatie c_i circulair overgedragen tussen PE's van alle PE's, welke zijn gesynchroniseerd met de achtereenvolgende invoer van de ingangsinformatie r_j^p .

- 50 Bij de initiële instelling wordt de vlag 67 in het statusvlagregister van een de PE's ingesteld op "1" en de vlaggen 67 in de resterende PE's worden ingesteld op "0".

- De twee typen regelmiddelen, vermeld in de alinea's (A) en (B) worden gecombineerd, zodat de procedure voor het uitvoeren van de regeling van handelingen van een stelselprocessor als boven beschreven zal worden toegelicht in overeenstemming met de bovengenoemde handelingen (I)–(V).

- 55 De figuren 14A–14E tonen een voorbeeld van een regelhandeling wanneer de stelselprocessor in figuur 11 de aanpassingsberekening uitvoert, welke is gebaseerd op de dynamische tijdvervorming in het geval van $n = 5$, $1 = 2$, $N_1 = 6$, $N_2 = 7$.

In deze tekeningen is 51 de PE, 52 de multiplexinrichting voor het kiezen van een informatieoverdrachts-

baan 55 en de uitwendige lijn 53 en 54 de uitwendige I/O-lijn. Met 66 is het statusvlagregister aangegeven, dat bestaat uit de vlag 67 voor het specificeren van de ingangsvectorinformatie r_1^p en de vlag 68 voor het uitvoeren van de telwerking van het aantal overdrachten, dat tijdens de hertoewijzingsverwerking wordt uitgevoerd. Met 69 is de informatie aangegeven, welke op elk verwerkingstijdstip door de hertoewijzingsverwerking opnieuw moet worden toegewezen. In de hierna volgende toelichting zullen de bovengenoemde verschillende vlaggen als volgt worden aangegeven.

Vlaggen toegevoegd aan de ingangsvectorinformatie r_1^p :

Vlag 61 "kop"-vlag

Vlag 62 "staart"-vlag

10 Vlag 63 "overdrachts"-vlag

Vlaggen in het statusvlagregister 66:

Vlag 67 "ingangs"-vlag

Vlag 68 "tel"-vlag

De figuren 14A–14E tonen de verwerkingshandelingen van de PE op elk verwerkingstijdstip vanaf het verwerkingstijdstip $(t + 1)$, waarop de ingangsvectorinformatiereeks $S_1 = (c_1, c_2, c_3, c_4, c_5)$ wordt vervangen door de ingangsvectorinformatiereeks $S_2 = (c_6, c_7, c_8, c_9, c_{10})$ tot het verwerkingstijdstip $(t + 20)$ waarop de volgende ingangsvectorinformatiereeks $S_3 = (c_{11}, c_{12}, c_{13}, c_{14}, c_{15})$ wordt gesubstitueerd voor de ingangsvectorinformatiereeks S_2 , en de regeltoestanden op elk moment van deze verwerkingstijden. De eenheid van tijd is de tijd, welke elke PE nodig heeft om de verwerking overeenkomstig de bovenstaande vergelijkingen (1) – (3) in elke PE uit te voeren.

Het verwerkingstijdstip $(t + 1)$ komt overeen met het tijdstip dat met $\text{mod}(N_1 + N_2) = \text{mod}(6 + 7) = 3$ tijdintervallen eerder is dan het verwerkingstijdstip $(t + 4)$, waarop de verwerking begint met het substitueren van de ingangsvectorinformatiereeks S_2 voor S_1 . Onmiddellijk voor dit verwerkingstijdstip $(t + 1)$, wordt een regelsignaal (hierna betiteld als een "substituut"-signaal), dat het uitvoeren van een substitutie van de ingangsinformatiereeks S_k instrueert, naar elke PE overgedragen. Slechts PE_1 , welke informatie opzamelt, die de eerste PE aangeeft, waarin het verwerken voor de invoer van de informatiereeks S_1 wordt ingeleid, accepteert dit "substituut"-signaal en maakt zich gereed om de hertoewijzingsverwerking te beginnen.

De boven in alinea (II) beschreven regelmiddelen voegen een "overdrachts"-vlag "1" aan de laatste vijf gegevens $r_3^p, r_4^p, r_5^p, r_6^p, r_7^p$ van de ingangsvectorinformatiereeksen R_1 en R_2 toe. Derhalve accepteert de PE_1 , waarvan de "ingangs"-vlag in het statusvlagregister op het verwerkingstijdstip $(t + 1)$ een "1" is, de informatie r_3^p van deze informatie om te beoordelen, dat de "overdrachts"-vlag een "1" is voor het onderkennen van het begin van de hertoewijzingsverwerking, en stelt de "tel"-vlag op "1" in. Op hetzelfde moment wordt "tel"-vlag beoordeelt als een "1" te zijn en alle PE's worden achtereenvolgens geïnformeerd ten aanzien van het begin van de hertoewijzingsverwerking (in de richting van de gestippelde pijl in de tekeningen).

Alle PE's worden dan in een overdrachtstoestand voor de hertoewijzingsverwerking ingesteld. Vervolgens wordt tijdens de verwerkingstijd het resultaat van deze berekening ook door alle PE's parallel met de op de bovengenoemde vergelijkingen (1) – (3) gebaseerde berekeningsverwerking overgedragen. Op het verwerkingstijdstip $(t + 1)$ worden de berekeningsresultaten $S_{5,1}^1, S_{5,2}^1, S_{5,3}^1, S_{5,4}^1, S_{5,5}^1$ (welke respectievelijk zijn verkregen in PE_1, PE_2, PE_3, PE_4 en PE_5) in elke PE uitgelezen, welke de eerste overdrachtsgegevens vanuit het inwendige geheugen daarvan zijn, en door de verwerking op dit tijdstip naar een naast gelegen PE overgedragen. Derhalve worden deze gegevens $S_{5,5}^1, S_{5,1}^1, S_{5,2}^1, S_{5,3}^1, S_{5,4}^1$ opnieuw toegewezen in respectievelijk PE_1, PE_2, PE_3, PE_4 en PE_5 tot het begin van de verwerking op het verwerkingstijdstip $(t + 2)$. De ingangsvectorinformatie r_6^p , welke in PE_2 op het verwerkingstijdstip $(t + 2)$ moet worden ingevoerd bevat geen informatie voor het terugstellen van de status van de hertoewijzingsverwerking, zodat de hertoewijzingsverwerking ook op dit tijdstip voortgaat. Daarna wordt voordat de verwerking op het verwerkingstijdstip $(t + 3)$ begint, deze informatie opnieuw toegewezen in de volgorde van $S_{5,4}^1, S_{5,5}^1, S_{5,1}^1, S_{5,2}^1$ en $S_{5,3}^1$.

Door deze wijze van verwerking te herhalen, worden deze gegevens opnieuw in vooraf bepaalde PE's toegewezen in de volgorde van $S_{5,3}^1, S_{5,4}^1, S_{5,5}^1, S_{5,1}^1$ en $S_{5,2}^1$ voordat de verwerking op het verwerkingstijdstip $(t + 4)$ begint. Zo wordt bij voorbeeld $S_{5,1}^1$, welke opnieuw in PE_4 is toegewezen, onmiddellijk gebruikt bij de verwerking op verwerkingstijdstip $(t + 4)$ om voor deze PE_4 $S_{6,1}^1$ te verkrijgen. Hetzelfde geldt voor de andere gegevens, zodat $S_{5,2}^1$ opnieuw toegewezen in PE_5 wordt gebruikt om $S_{6,2}^1$ op het verwerkingstijdstip $(t + 5)$ te verkrijgen. Op het verwerkingstijdstip $(t + 4)$ wordt de informatie r_1^p waarvan de "kop"-vlag bestaat uit een "1" in PE_4 opgenomen, zodat deze vlag wordt beoordeeld en onderkent, dat de hertoewijzingsverwerking is voltooid, en wordt de "overdrachts"-vlag in het statusvlagregister teruggesteld. Op hetzelfde moment wordt deze terugstelling bevestigd en worden alle PE's (in de richting van de gestippelde pijlen in de tekeningen) ervan op de hoogte gesteld, dat de hertoewijzingsverwerking is voltooid, en wordt de hertoewijzingsverwer-

kingsstatusinstelling in alle PE's vrijgegeven. Derhalve wordt op het verwerkingstijdstip $(t + 4)$ slechts de verwerking overeenkomstig de bovenstaande vergelijkingen (1) – (3) in elke PE uitgevoerd.

Voorts onderkent nadat een "substituut"-signaal in elke PE vanuit het totale regelstelsel is ingevoerd, de eerste PE, welke ingangsvectorgegevens heeft geaccepteerd, waarin de "kop"-vlag een "1" is, dat deze PE de eerste PE is, omgeroepen door de uitgangs-PE, welke de verwerking uitvoert voor die ingangsvectorinformatiereeks S_k , en de door deze PE beïnvloede multiplexinrichting wordt naar de uitwendige lijn omgeschakeld om de ingangspoort voor de ingangsvectorinformatiereeks S_k te worden. Vervolgens wordt de informatie, welke de eerste PE voor de verwerking aangeeft, in deze PE opgeslagen. Deze wijze van regeling leidt de substitutie van de ingangsvectorinformatiereeksen in. In figuur 14A voert PE_4 de werking op het verwerkingstijdstip $(t + 4)$ uit.

De PE's welke de uitgangs-PE of de eind-PE waren, die de berekening uitvoeren voor de eerste ingangsvectorinformatie in de gehele vectorinformatiereeksen $R_1, R_2, \dots, R_1$ van de tellingshandelingen in de hertoewijzingsverwerking zamelen de informatie voor deze uitgangs-PE of eind-PE binnen de PE's op totdat de uitgangs-PE of eind-PE van de hertoewijzingsverwerking voor de volgende ingangsvectorinformatiereeks S_{k+1} optreedt. Daarna worden de handelingen opnieuw ingeleid op elk moment, dat een "ingangs"-vlag van het statusvlagregister in deze PE's optreedt en vindt de bovengenoemde hertoewijzingsverwerking plaats. Bij het in figuur 14B afgebeelde voorbeeld vindt de tweede hertoewijzingsverwerking plaats gedurende de periode vanaf de verwerkingstijdstippen $(t + 6)$ tot $(t + 8)$. De berekeningsresultaten $S_{5,6}^1, S_{5,6}^2, S_{5,6}^3, S_{5,6}^4$ en $S_{5,6}^5$, welke de tweede overdrachtsinformatie voor elke PE zijn, worden in vooraf bepaalde PE's opnieuw toegewezen door de tweede hertoewijzingsverwerking gedurende de periode vanaf de verwerkingstijdstippen $(t + 6)$ tot $(t + 8)$. De berekeningsresultaten $S_{5,6}^5, S_{5,6}^6$ en $S_{5,6}^7$ worden in vooraf bepaalde PE's opnieuw toegewezen door de derde hertoewijzingsverwerking gedurende de periode vanaf de verwerkingstijdstippen $(t + 11)$ tot het moment $(t + 13)$.

Bij de verwerkingstijdstippen $(t + 14)$ is de hertoewijzingsverwerking voor de ingangsvectorinformatiereeks S_1 , gebaseerd op de telhandelingen tussen de PE's welke de uitgangs-PE en de eind-PE van de hertoewijzingsverwerking zijn, voltooid. De informatie, welke de uitgangs-PE van de verwerking aangeeft voor de ingangsvectorinformatiereeks S_2 ingesteld in PE_4 en het "substituut"-signaal, uitgezonden uit het totale regelstelsel onmiddellijk voor dit verwerkingstijdstip, worden gecontroleerd en de "overdrachts"-vlag, welke aan de ingangsvectorinformatie r_2^1 wordt toegevoegd, wordt beoordeeld als een "1" te zijn. De "tel"-vlag van het statusvlagregister wordt dan onmiddellijk ingesteld.

Daarna wordt de bovenbeschreven bewerkingprocedure na het verwerkingstijdstip $(t + 15)$ herhaald.

Anderzijds beoordeeld de PE, welke de verwerking van de ingangsvectorinformatie r_1^1 en r_2^1 heeft uitgevoerd, welke "staart"-vlaggen hebben, dat deze vlaggen gelijk zijn aan "1" en voert dan naar een uitwendige bestemming het berekeningsresultaat $S_{N,P}^1$ parallel met de berekeningsbewerking volgens de bovenstaande vergelijkingen (1) – (3) uit.

In de figuren 14A–14E wordt het resultaat $S_{1,7}^1$ op het bewerkingstijdstip $(t + 3)$ uitgevoerd, $S_{2,7}^1$ op het verwerkingstijdstip $(t + 4)$, $S_{3,7}^1$ op het verwerkingstijdstip $(t + 5)$, $S_{4,7}^1$ op het verwerkingstijdstip $(t + 6)$, $S_{5,7}^1$ op het verwerkingstijdstip $(t + 7)$, $S_{6,7}^1$ op het verwerkingstijdstip $(t + 9)$, $S_{7,7}^1$ op het verwerkingstijdstip $(t + 10)$, $S_{8,7}^1$ op het verwerkingstijdstip $(t + 11)$, $S_{9,7}^1$ op het verwerkingstijdstip $(t + 12)$, $S_{10,7}^1$ op het verwerkingstijdstip $(t + 13)$, $S_{11,7}^1$ op het verwerkingstijdstip $(t + 16)$, $S_{12,7}^1$ op het verwerkingstijdstip $(t + 17)$, $S_{13,7}^1$ op het verwerkingstijdstip $(t + 18)$, $S_{14,7}^1$ op het verwerkingstijdstip $(t + 19)$ en $S_{15,7}^1$ op het verwerkingstijdstip $(t + 20)$. Derhalve worden de resultaten van de berekeningen voor het ingangsvectorinformatiereeks S_2 naar de uitwendige bestemmingen uitgevoerd in de volgorde van $S_{6,6}^1, S_{7,6}^1, S_{8,6}^1, S_{9,6}^1, S_{10,6}^1, S_{11,6}^1, S_{12,6}^1, S_{13,6}^1, S_{14,6}^1$ en $S_{15,6}^1$.

De informatie, welke de beginnende PE van de verwerking voor elke ingangsvectorinformatiereeks S_k aangeeft, wordt teruggesteld nadat is vastgesteld, dat de "overdrachts"-vlag een "1" is en het is bevestigd, dat de "tel"-vlag is ingesteld op "1". Bovendien wordt wanneer de "overdrachts"-vlag "1" is, de informatie, welke de beginnende PE van de hertoewijzingsverwerking aangeeft ingesteld op "1" indien de "tel"-vlag gelijk is aan "0" en de informatie niet is gewijzigd, indien deze "tel"-vlag gelijk is "1". Wanneer de "overdrachts"-vlag gelijk is aan "1", wordt de informatie, welke de eind-PE van de hertoewijzingsverwerking aangeeft, ingesteld op "0", indien de "tel"-vlag gelijk is "1" en de informatie wordt niet gewijzigd indien deze "tel"-vlag gelijk is "0". Nadat de multiplexinrichting de uitwendige lijn kiest, onderhoud de multiplexinrichting de keuze van de uitwendige lijn totdat de "ingangs"-vlag van het statusvlagregister veranderd in "1" en nadat is vastgesteld, dat de "ingangs"-vlag gelijk is "1", kiest de multiplexinrichting de baan voor de overdracht van informatie tussen PE's.

Zoals boven is toegelicht, vervullen de inhouden van het statusvlagregister en de vlag, welke is toegevoegd aan de ingangsvectorinformatie r_1^1 de functie van het bepalen van de verwerkingsstatus van

elke PE onmiddellijk voor het uitvoeren van de berekeningsverwerking overeenkomstig de bovenstaande vergelijking (1) – (3). Dientengevolge vereist deze regelmethode geen regelsignalen uit het totale regelsel, welke de verwerkingshandelingen in de respectieve PE specificeren. Het is voldoende, dat een regelsignaal, dat de tempering voor de substitutie van de ingangsvectorinformatiereeks S_k aangeeft, cyclisch
 5 aan elke PE wordt toegevoerd. Deze inrichting maakt het mogelijk dynamische tijdvervormingsberekeningen continue overeenkomstig de bovenstaande vergelijkingen (1) – (3) uit te voeren, terwijl een autonome regeling in het totale stelselprocessor plaatsvindt.

Figuur 15 toont een uitvoeringsvorm van een inrichting wanneer een middel voor het uitvoeren van de regelmethode is opgenomen in een PE, welke in deze stelselprocessor vormt.

10 In figuur 15 zijn de uitwendige ingangsvectorinformatielijn 33, de uitwendige I/O-lijn 34 en de klem 36, welke de PE en de uitwendige I/O-lijn 34 verbindt, dezelfde als die in figuur 11. Met 71 is een regelingangsklem van een regeleenheid 88 vanuit een regelsignaalbron, zoals een gastheerrekeninrichting aangegeven. Met 72 en 73 zijn ingangs- en uitgangsklemmen van de banen voor de overdracht van informatie tussen respectieve PE's aangegeven. Met 74 en 75 zijn ingangs- en uitgangsklemmen van een statusvlagregister
 15 86 aangegeven. Met 76 en 77 zijn ingangs- en uitgangsbuffers voor de informatieoverdracht aangegeven. Met 78 en 79 zijn buffergeheugens voor de ingangsvectorinformatie c_i en r_i aangegeven. Met 80 is een bufferregister voor de uitgangsinformatie naar de uitwendige I/O-lijn 34 aangegeven. Met 81 is een teller voor de geheugentoegang aangegeven. Met 82 is een rekenkundige eenheid weergegeven, voorzien van een optel- en aftrekeenheden 82-1, een accumulator 82-2 en een productsomeenheid 82-3. Met
 20 83 is een bedrijfsgeheugen aangegeven. Met 84 en 85 zijn bedrijfsgeheugentoegangslijnen aangegeven. Met 87 is een bufferregister voor aan de ingangsvectorinformatie toegevoegde vlaggen aangegeven. Met 89 is instructie-eenheid aangegeven, voorzien van een microprogrammegeheugen 89-1, dat de PE-verwerkingshandelingen bevat, een programmateller 89-2, een instructieregister 89-3 en een instructie-decodeerinrichting 89-4. Met 90 is een bufferregister aangegeven voor het opslaan van de ingangsvectorinformatie r_i wanneer de informatie uit de uitwendige I/O-lijn 34 wordt ingevoerd. Met 91, 92, 93 en 94 zijn
 25 kiezers aangegeven, die door de instructie-eenheid 89 worden geregeld.

De regeleenheid 88 in elke PE ontvangt een verwerkingsbeginactiveringssignaal uit een uitwendige regeleenheid en wordt geregeld door een regelstroom, als aangegeven in figuur 17. De regeleenheid 88 wekt een regelsignaal op voor het instellen van het initiële adres van het programma in de programmateller
 30 89-2. Vervolgens regelt de regeleenheid 88 onder gebruik van overdracht van "ingangs"-vlag en de "tel"-vlag, en controleert de vlag $f_{5,6}$ voor de vijfde en zesde velden in het programma (1) door te beoordelen of de "ingangs"-vlag al dan niet een "1" is, en stelt daarna deze vlag in. In deze situatie werkt de programmateller 89-2, zodat het uitvoeren van het programma (1) wordt geregeld.

Wanneer het programma (1) is beëindigd, wordt de "ingangs"-vlag beoordeeld en indien de "ingangs"-vlag gelijk is aan "1", wordt de vlag toegevoegd aan de vectorinformatie r_i , die in PE wordt ingevoerd, beoordeeld, zodat de nodige status instelling voor het uitvoeren van het programma (2) plaatsvindt.
 35 Wanneer het blijkt, dat de "kop"-vlag gelijk is aan "1", wordt de "tel"-vlag beoordeeld. Indien de "tel"-vlag gelijk is aan "0", blijft de eerder ingestelde status ongewijzigd en begint de regeleenheid 88 het uitvoeren van het programma (2) te regelen.

40 Wanneer de "tel"-vlag gelijk is aan "1", wordt de hertoewijzingsverwerkingsstatus uitschakelvlag ingesteld op "1" teneinde het opwekken van een terugstelsignaal van de hertoewijzingsbewerkingsstatus voor te bereiden. De "tel"-vlag wordt ingesteld op "0" en er wordt een terugstelsignaal van de hertoewijzingsverwerkingsstatus uitgezonden. Daarna wordt voor het maskeren van het tweede veld van het programma (2), f_2 ingesteld op "0" en wordt de informatie voor de startende PE ingesteld op "1". In het
 45 geval, dat de "staart"-vlag wordt gecontroleerd, wordt de vlag f_3 , welke het uitvoeren van het derde veld van het programma (2) regelt, ingesteld op "1" of "0" afhankelijk van het feit dat de "staart"-vlag gelijk is aan "1" of niet. In het geval, dat "overdrachts"-vlag wordt gecontroleerd, wordt de informatie voor de startende PE beoordeeld en indien de informatie niet gelijk is "1", begint de regeleenheid 88 te regelen voor het uitvoeren van het programma (2) zonder dat de voorafgaande status wordt gewijzigd.

50 Indien de informatie voor de startende PE gelijk is "1", wordt de hertoewijzingsverwerkingsstatus inschakelvlag ingesteld op "1" om het verschaffen van een hertoewijzingsverwerkingsstatus in te leiden, en wordt de "tel"-vlag ingesteld op "1". Daarna wordt het instelsignaal van de hertoewijzingsverwerkingsstatus uitgezonden en wordt de vlag f_2 voor het regelen van het uitvoeren van het tweede veld in het programma (2) ingesteld op "1". Bovendien wordt de informatie voor de startende PE ingesteld op "0" en begint de
 55 regeleenheid 88 het uitvoeren van het programma (2) te regelen.

Wanneer de aan de ingevoerde vectorinformatie r_i toegevoegde vlag een vlag is, welke niet deze drie typen vlaggen omvat, dat wil zeggen NOP-vlag is, wordt de hertoewijzingsverwerkingsstatus inschakelvlag

gecontroleerd. Indien deze vlag "0" is, begint de regeleenheid 88 het uitvoeren van het programma (2) te regelen zonder dat de voorafgaande status wordt gewijzigd. Wanneer deze vlag gelijk is aan "1", wordt de "tel"-vlag ingesteld op "1" om de hertoewijzingsverwerking opnieuw in te leiden, wordt het instelsignaal van de hertoewijzingsverwerkingsstatus uitgezonden, en wordt de vlag f_2 ingesteld op "1". Bovendien wordt de informatie voor de startende PE ingesteld op "0".

Wanneer de "ingangs"-vlag niet gelijk is "1", worden de berekeningen uitgevoerd onder gebruik van vectorinformatie r_j , welke reeds in de PE is opgeslagen tijdens het uitvoeren van het programma (1), zodat de statusinstellingen, welke nodig zijn voor het uitvoeren van het programma (2) plaatsvinden overeenkomstig het feit of het instel- of terugstelsignaal voor de hertoewijzingsverwerkingsstatus al dan niet is ontvangen. Wanneer het instelsignaal is ontvangen wordt de vlag f_2 ingesteld op "1". Wanneer in plaats van instelsignaal het terugstelsignaal is ontvangen, wordt de vlag f_2 ingesteld op "0". Wanneer geen signaal is ontvangen, begint de regeleenheid 88 het uitvoeren van het programma (2) te regelen zonder dat de voorafgaande status wordt gewijzigd.

Wanneer de statusinstellingen voor de bovenbeschreven verschillende gevallen zijn voltooid, wordt de programmateller 89-2 geactiveerd, zodat het programma (2) wordt uitgevoerd. De regeleenheid 88 herhaalt de bovenstaande regelprocedure op elk verwerkingstijdstip.

De bovenbeschreven uitvoeringsvorm maakt een autonome regeling in het totale stelsel mogelijk door slechts vlaggen van drie bits toe te voegen en in elke PE statusvlaggen van twee bits op te nemen. Deze inrichting maakt niet slechts een reductie van de belasting van het totale regelstelsel 57 mogelijk doch eveneens een vereenvoudiging van de regelmethode.

Er zal een verder uitvoeringsvorm van de regelmethode volgens de uitvinding worden toegelicht voor het geval, dat een productmatrix C wordt verkregen uit $(l \times k)$ matrix A en $(k \times m)$ matrix B. Indien $A = \{a_{ij} \mid 1 \leq i \leq l, 1 \leq j \leq k\}$, $B = \{b_{ij} \mid 1 \leq i \leq k \text{ en } 1 \leq j \leq m\}$ en $C = \{c_{ij} \mid 1 \leq i \leq l, 1 \leq j \leq m\}$ is,

$$c_{ij} = \sum_{p=1}^k a_{ip} b_{pj} \quad (5)$$

Indien $a_i = (a_{i1}, a_{i2}, \dots, a_{ik})$ en $b_j = (b_{1j}, b_{2j}, \dots, b_{kj})$ wordt de bovenstaande vergelijking (5) uitgedrukt door:

$$c_{ij} = a_i \cdot b_j \quad (6).$$

In deze vergelijking stelt "." de inwendige productoperator tussen de vectoren voor.

Een vermenigvuldiging van matrix A en matrix B betekend het verkrijgen van c_{ij} in de bovenstaande vergelijking (6) voor alle gevallen van $1 \leq i \leq l, 1 \leq j \leq m$. Deze regelmethode zal worden toegelicht voor het geval, dat een stelselprocessor met l ($l \leq m$) PE's in een lineaire stelselconstructie deze wijze van berekening uitvoert voor verschillende m , in een specifiek geval van $l = 4$ en $m = 6$. Thans wordt hier aangenomen, dat elke PE berekeningsfuncties voor het uitvoeren van bovenstaande vergelijking (6), functies voor het invoeren van het informatie a_i , voor het invoeren en uitvoeren van informatie b_j en het opzamelen en uitvoeren van het resultaat van de berekeningen c_{ij} , en ingebouwde verwerkingsinstructies voor het uitvoeren van deze verwerkingshandelingen omvat.

De figuren 16A-16I tonen een voorbeeld van de regelhandelingen wanneer de matrix vermenigvuldiging plaatsvindt in een stelselprocessor met een lineaire opbouw.

Met 54 is I/O-lijn voor het invoeren van de informatie a_i in de PE en voor het uitvoeren van het berekeningsresultaat c_{ij} uit de PE aangegeven. Met 55 is de baan voor het overdragen van informatie tussen PE's aangegeven. Met 66 is het statusvlagregister aangegeven, bestaande uit de vlag 67, welke de ingangstatus van de informatie uit I/O-lijn 54 aangeeft, en de vlag 68 welke de uitgangstatus van het berekeningsresultaat aangeeft. Met 51 is de PE aangegeven. a_i is kolomvector en b_j is een rijvector van de matrixes A en B.

Bij de hierna volgende toelichting van de handelingen, zullen de vlaggen 67 en 68 als volgt worden betiteld:

Vlag 67 "ingangs"-vlag

Vlag 68 "uitgangs"-vlag

Voorts wordt aan de vector b_1, b_2, b_3, b_4, b_5 en b_6 de "eind"-vlag toegevoegd. Slechts de "eind"-vlag van de vector b_6 is gelijk aan "1".

Initieel wordt slechts de "ingangs"-vlag van PE_1 ingesteld op "1". Op het verwerkingstijdstip t_1 , controleert PE_1 deze vlag en wordt de vector a_1 via de I/O-lijn 54 in PE_1 ingevoerd. Tegelijkertijd wordt de vector b_1 vanuit de informatieoverdrachtsbaan 55 ingevoerd. Vervolgens wordt $a_1 \cdot b_1$ berekend en inwendig opgeslagen als de informatie c_{11} . Voor het inleiden van de berekeningsverwerking op het verwerkingstijdstip t_2 , wordt de inhoud van de status van het vlagregister 66 overgedragen naar de naast gelegen PE, en worden "ingangs"-vlag van PE_2 ingesteld op "1". PE_2 beoordeelt deze vlag op het verwerkingstijdstip t_2 en daarna wordt op hetzelfde moment, dat de vector a_2 vanuit de I/O-lijn 54 in PE_2 wordt ingevoerd, de vector

b_1 vanuit PE_1 in PE_2 ingevoerd, zodat a_2, b_1 wordt berekend en als informatie c_{21} wordt opgeslagen. Op dit moment wordt parallel aan de verwerking in PE_2 de vector b_2 in PE_1 ingevoerd en wordt a_1, b_2 berekend en als de informatie c_{12} opgeslagen.

- Daarna wordt dezelfde verwerkingsprocedure herhaald tot het verwerkingstijdstip t_4 . Voordat de berekeningsverwerking op het verwerkingstijdstip t_5 begint, wordt de inhoud van het statusvlagregister 66 overgedragen, zodat de "ingangs"-vlaggen in alle PE's worden ingesteld op "0" en derhalve wordt beoordeeld, dat het ingangssignaal uit I/O-lijn 54 is voltooid. De vectoren b_j worden op elk verwerkingstijdstip naar naast gelegen PE's overgedragen, zodat op het verwerkingstijdstip t_5 alle PE's de inwendige productberekeningen uitvoeren en de resultaten van de berekeningen accumuleren. Voor het begin van de berekeningsverwerking op het tijdstip t_6 , beoordeeld PE_1 , dat de aan de ingevoerde vector b_6 toegevoegde "eind"-vlag gelijk is "1" en daarna stelt PE_1 de uitgangsverwerkingsstatus voor alle berekeningsresultaten, welke in die PE_1 zijn opgezameld, inclusief het resultaat, dat op dit tijdstip wordt verkregen, in en stelt de "uitgangs"-vlag op "1" in. Op hetzelfde moment, dat PE_1 de "uitgangs"-vlag op "1" instelt, informeert PE_1 , PE_2 , PE_3 en PE_4 sequentieel omtrent de eigen verwerkingsstatus daarvan. PE_2 , PE_3 en PE_4 controleren deze status en beoordelen, dat er een verschil in het aantal verwerkingsstappen tussen PE's is door de uitgangsverwerking van de berekeningsresultaten. Zelfs in het geval, dat de eigen verwerking daarvan is voltooid, worden deze PE's in een verwerkingsparaatstatus ingesteld totdat zij een regelsignaal ontvangen, dat aangeeft, dat de status in PE_1 is vrijgegeven. PE_1 stelt "uitgangs"-vlag terug wanneer de uitvoer van de eindresultaten van de berekening is voltooid, en na het bevestigen van dit terugstellen van de vlag, informeert PE_1 alle PE's, dat de uitgangstoestand daarvan is vrijgegeven. Vervolgens wordt op verwerkingstijdstip t_7 de vector b_j naar de naast gelegen PE overgedragen en voert PE_2 dezelfde verwerkings- en regelhandelingen uit als die, welke door PE_1 op het verwerkingstijdstip t_6 zijn uitgevoerd. Op deze wijze wordt wanneer de verwerking op het verwerkingstijdstip t_9 is voltooid, het resultaat van de vermenigvuldiging van de matrixes A en B verkregen.
- Zoals boven is toegelicht heeft de beschreven regelmethode het voordeel, dat deze het mogelijk maakt autonome matrix vermenigvuldigingsverwerkingen in een stelselprocessor uit te voeren door slechts een vlag van één bit bij een ingangsvectorgegeven op te tellen en in een PE een vlag van twee bits op te nemen.

30

Conclusies

1. Stelselprocessor voorzien van n verwerkingselementen, welke onderling in de vorm van een ring gekoppeld zijn met behulp van n multiplexinrichtingen, waarbij elk van de n verwerkingselementen omvat:
 - invoerorganen voor het invoeren van één van de ingangsvectorgegevens van een eerste en een tweede reeks van ingangsvectorgegevens;
 - organen voor het uitvoeren van voorafbepaalde berekeningen, zoals optellen, aftrekken, vergelijkingsberekeningen en product-somberekeningen tussen de ingangsvectorgegevens van de eerste en de tweede reeks;
 - opslagorganen voor het opslaan van de ingangsvectorgegevens van de eerste reeks, van de tweede reeks, en van de resultaten van de berekeningen;
 - een gegevensoverdraagorgaan voor het overdragen van één van de ingangsvectorgegevens en de berekeningsresultaten naar een naastgelegen verwerkingselement via een van de multiplexinrichtingen;
 - een met een I/O-lijn verbonden uitvoerorgaan voor het uitvoeren als eindberekening van het resultaat naar de I/O-lijn; en
 - een stuurorgaan voor het besturen van de invoerorganen, de opslagorganen, het berekeningsorgaan, het overdrachtsorgaan en het uitvoerorgaan,
 - met het kenmerk,
 - dat elk van de n multiplexinrichtingen ($2-1 \dots 2-n$) een tweede ingang heeft, die verbonden is met een ingangsvectorgegevenslijn (3), en voorzien is van organen voor het kiezen van informatie-overdracht hetzij tussen twee verwerkingselementen ($PE_1 \dots PE_n$), hetzij tussen de ingangsvectorgegevenslijn en een verwerkingselement;
 - dat de invoerorganen bestaan uit een eerste invoerorgaan (13, 17) voor het invoeren van één van de ingangsvectorgegevens (c_i) van de eerste reeks ($C = \{c_i\} (i = 1, 2, \dots, I)$) vanaf de ingangsvectorgegevenslijn (3) via de multiplexinrichting, en uit een tweede invoerorgaan (12, 15) voor het invoeren van één van de ingangsvectorgegevens (r_j) van de tweede reeks ($R = \{r_j\} (j = 1, 2, \dots, N)$) uit de I/O-lijn (4); en
 - dat het gegevensoverdraagorgaan (18, 14) is ingericht om respectieve verwerkingsresultaten gelijktijdig

over te dragen naar respectieve naastgelegen verwerkingselementen, en wel $\frac{\text{mod}N}{n}$ maal ($\frac{\text{mod}N}{n}$ is de rest als N door n wordt gedeeld), parallel met een berekeningsverwerking in elk verwerkingselement.

2. Stelselprocessor volgens conclusie 1, met het kenmerk, dat elk verwerkingselement verder is voorzien van: organen voor het ontvangen van een statusvlag uit een naast gelegen verwerkingselement teneinde de
- 5 statusvlag naar het andere naastgelegen verwerkingselement over te dragen; van een tweede register met een gedeelte voor het opslaan van een vlag, welke is toegevoegd aan de vanuit de I/O-lijn ingevoerde ingangsvectorinformatie; van regelorganen die zijn voorzien van een verwerkingselementregeleenheid, welke het regelsignaal uit een uitwendige bron, het vlagsignaal uit het registergedeelte en de statusvlag ontvangt voor het vormen van een instructieregelsignaal; en van een verwerkingsinstructie-eenheid, welke
- 10 het instructieregelsignaal ontvangt voor het vormen van de adresinformatie en regelsignalen om eerste, tweede en derde kiesorganen te besturen.

Hierbij 22 bladen tekening

FIG. 1

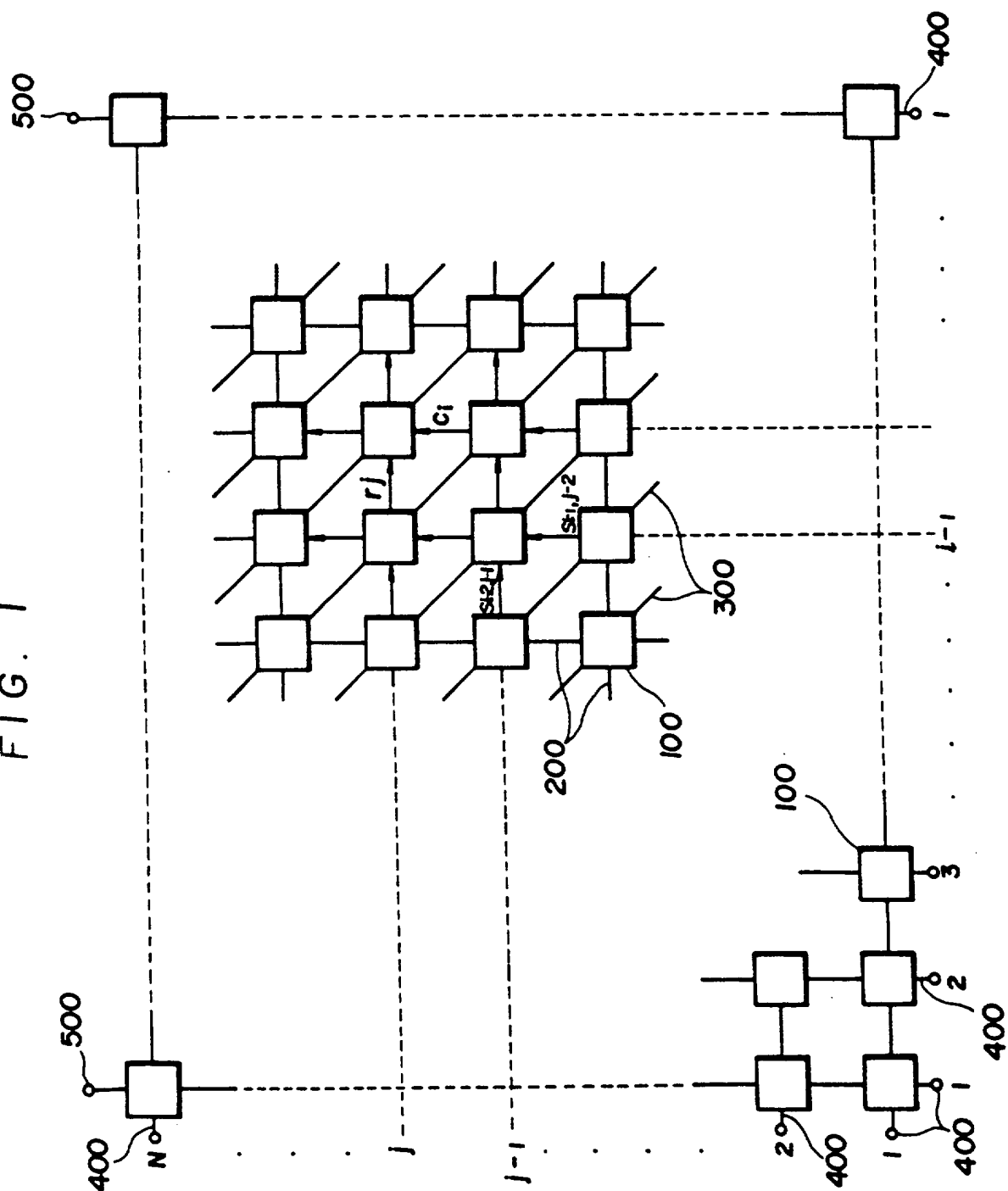


FIG. 3A

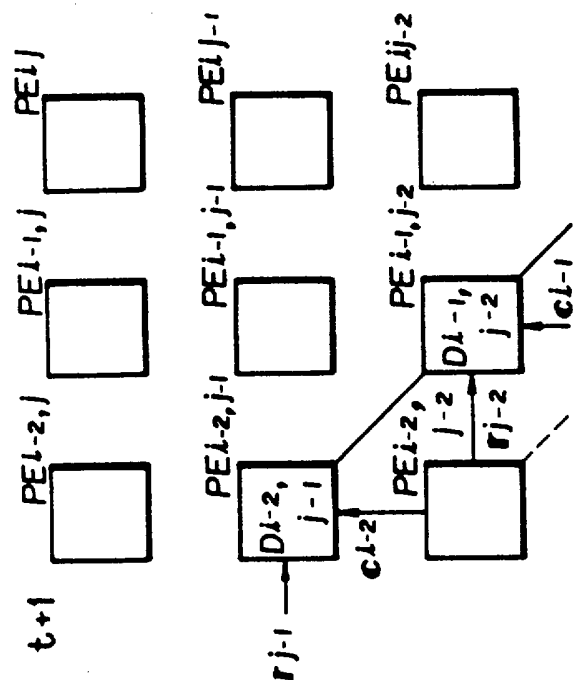


FIG. 3B

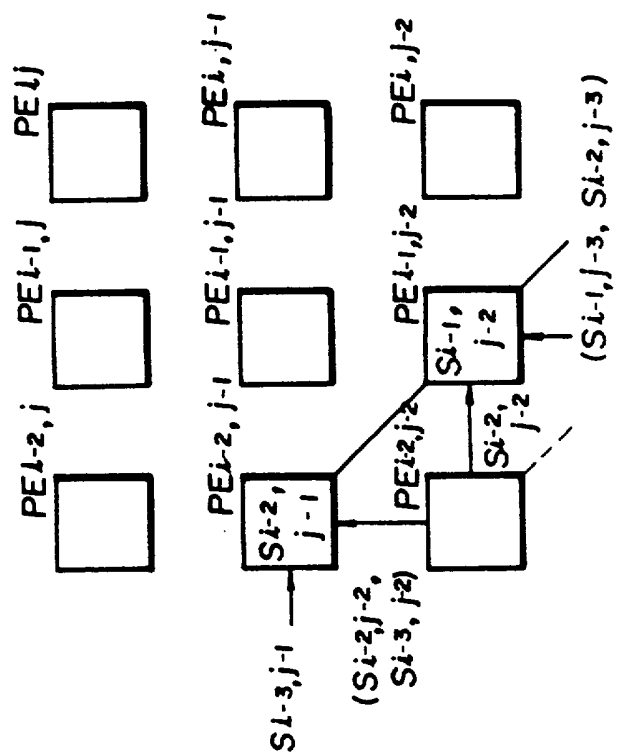


FIG. 2A

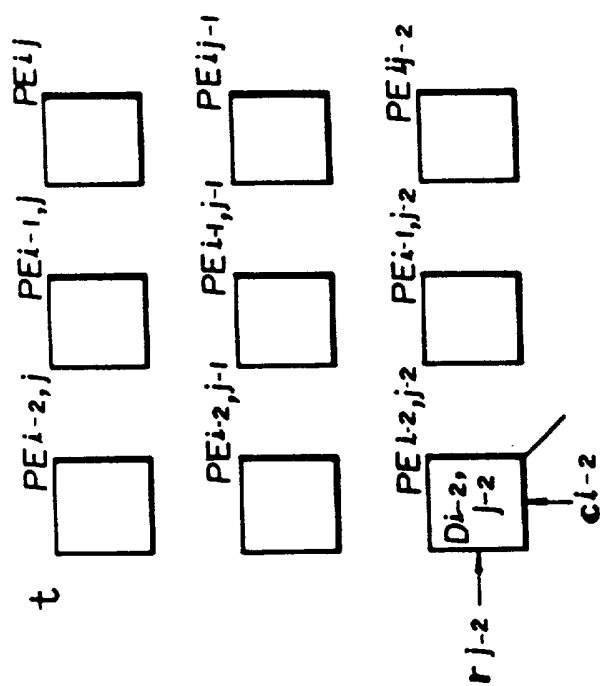


FIG. 2B

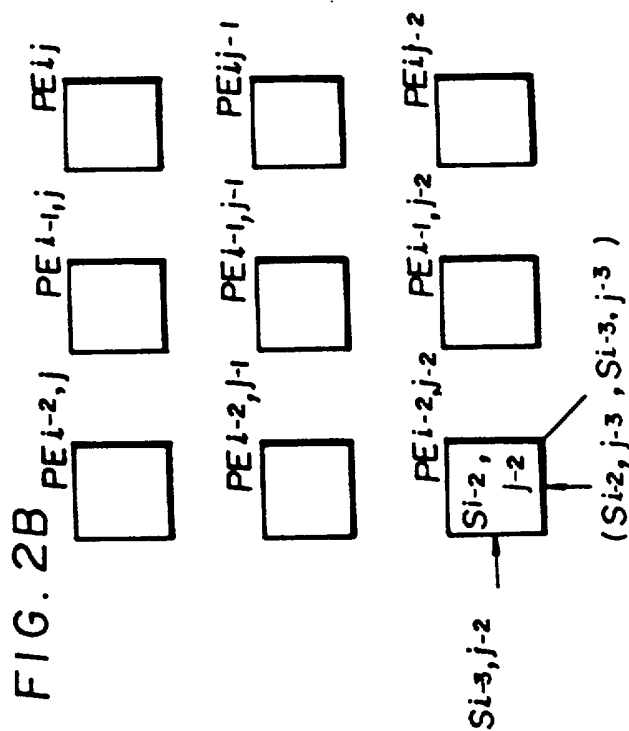


FIG. 5A

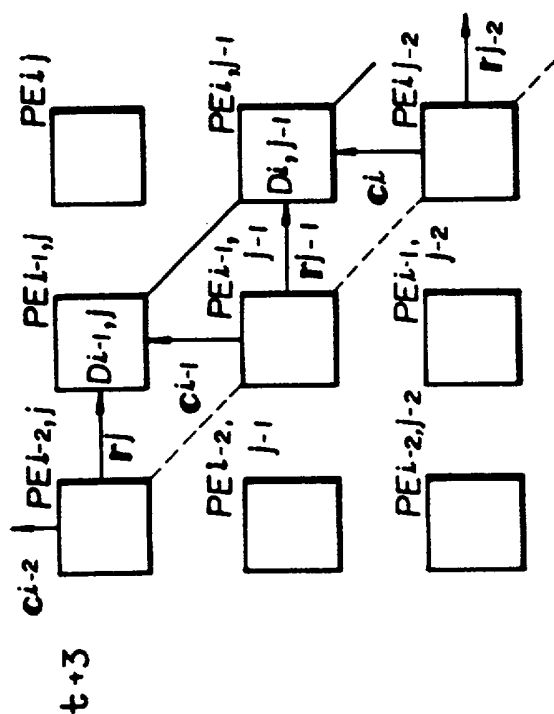


FIG. 5B

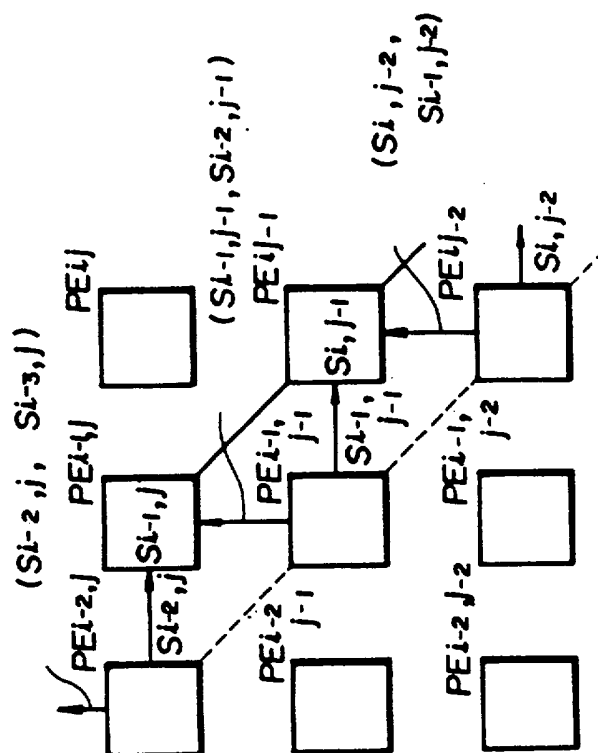


FIG. 4A

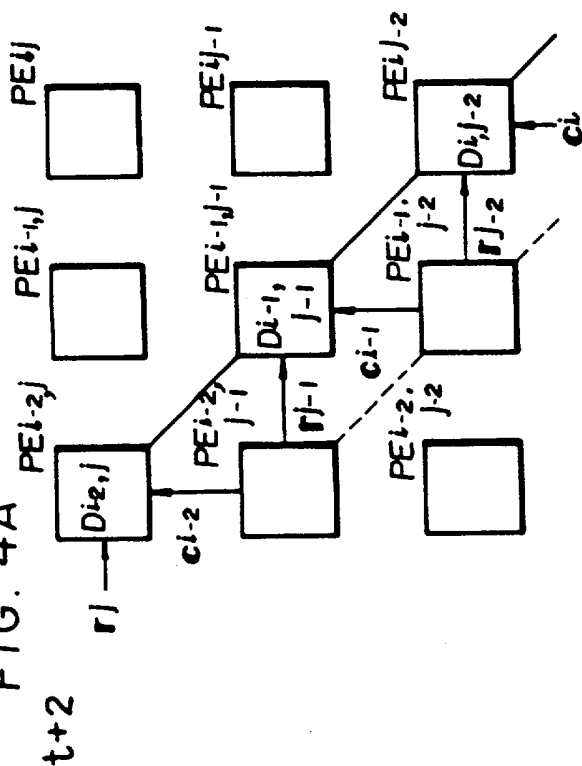


FIG. 4B

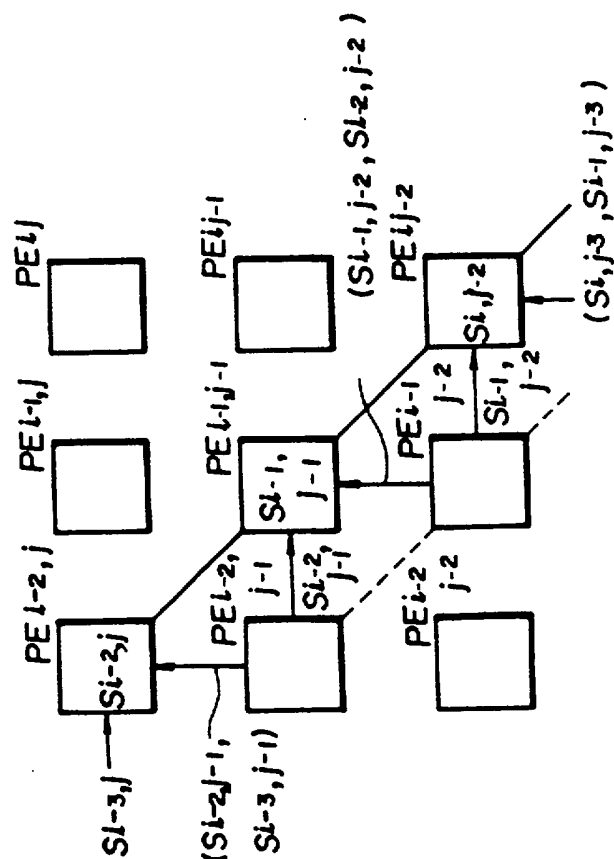


FIG. 6

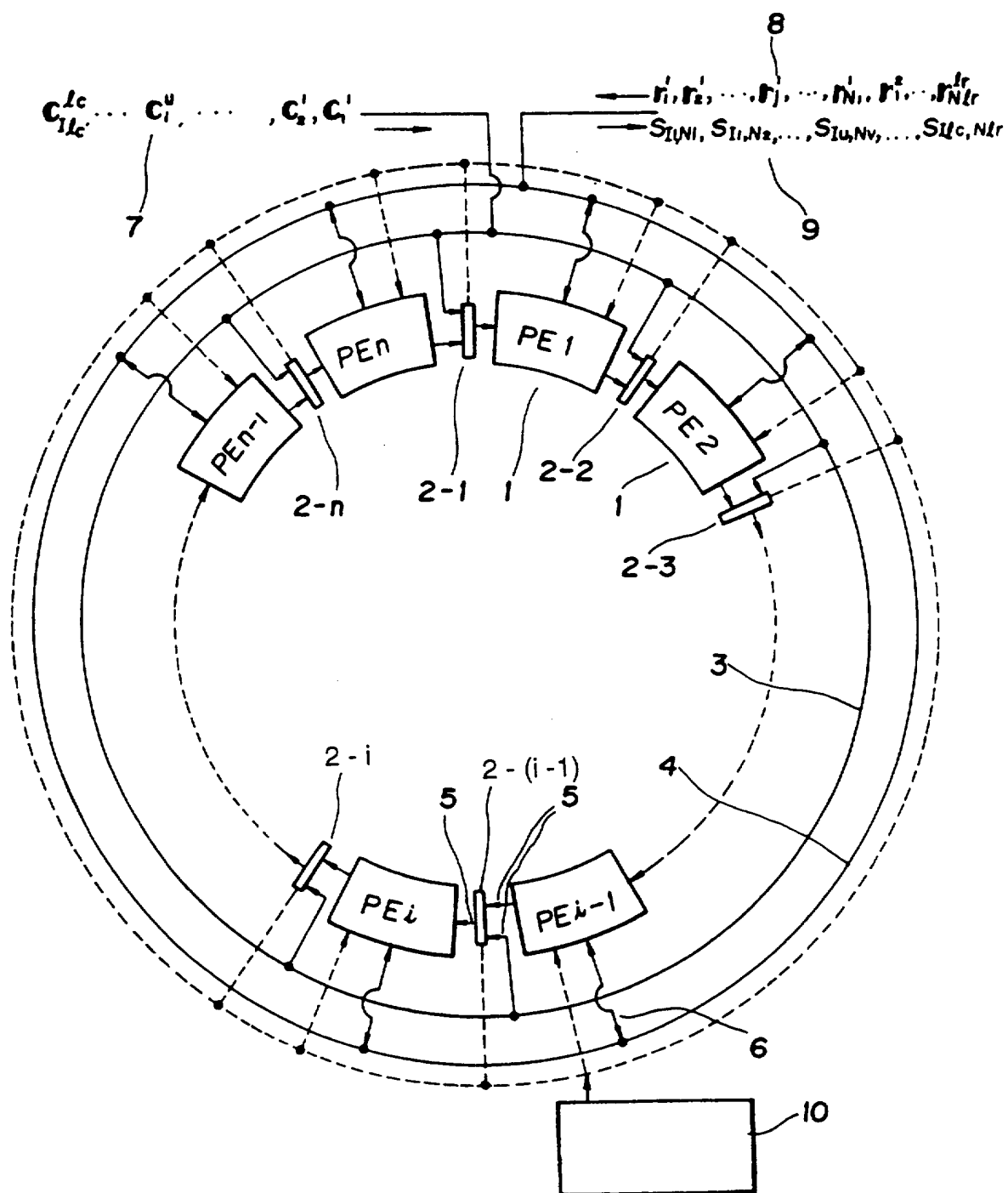


FIG. 7

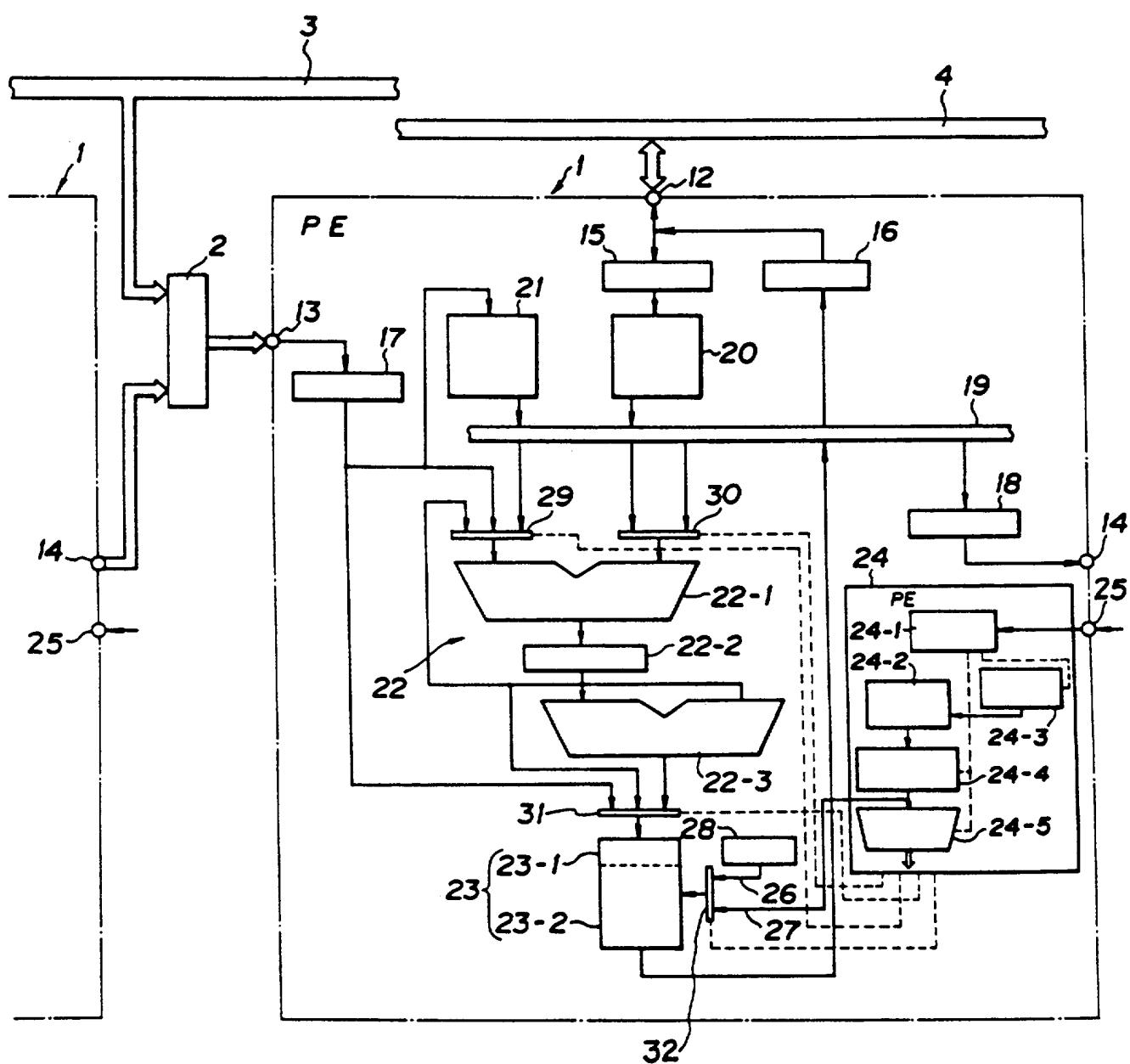


FIG. 8

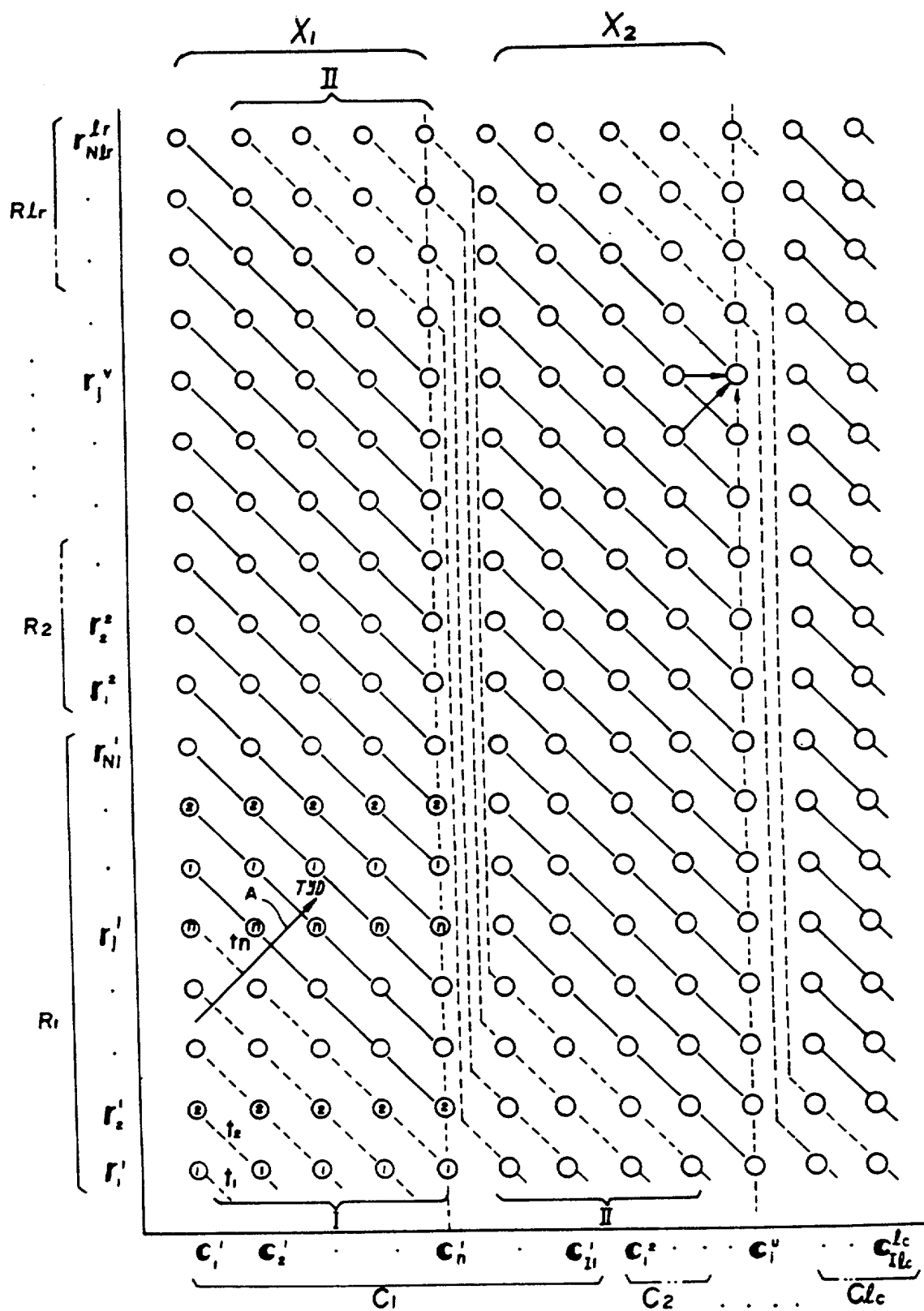


FIG. 9

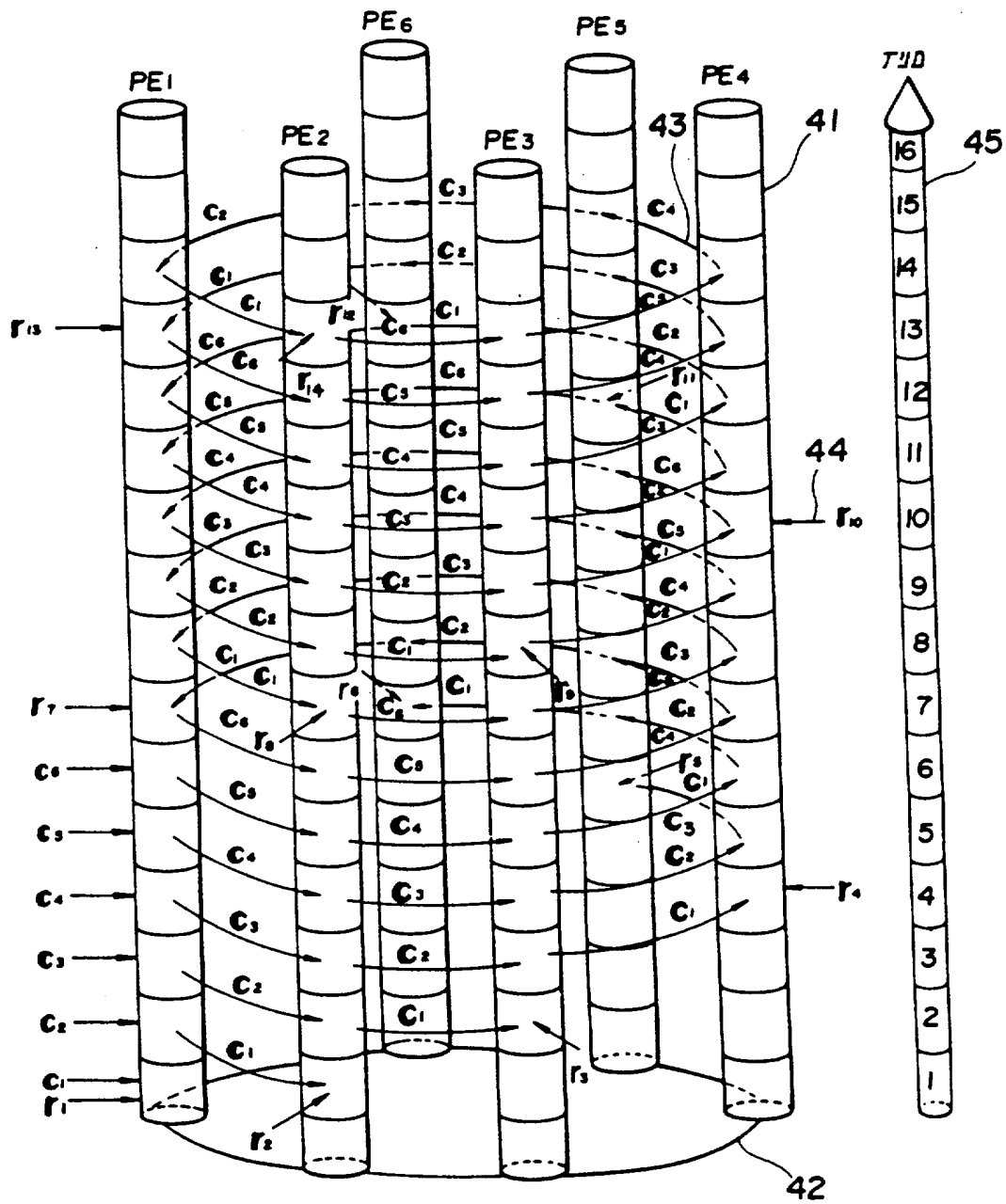


FIG. 10

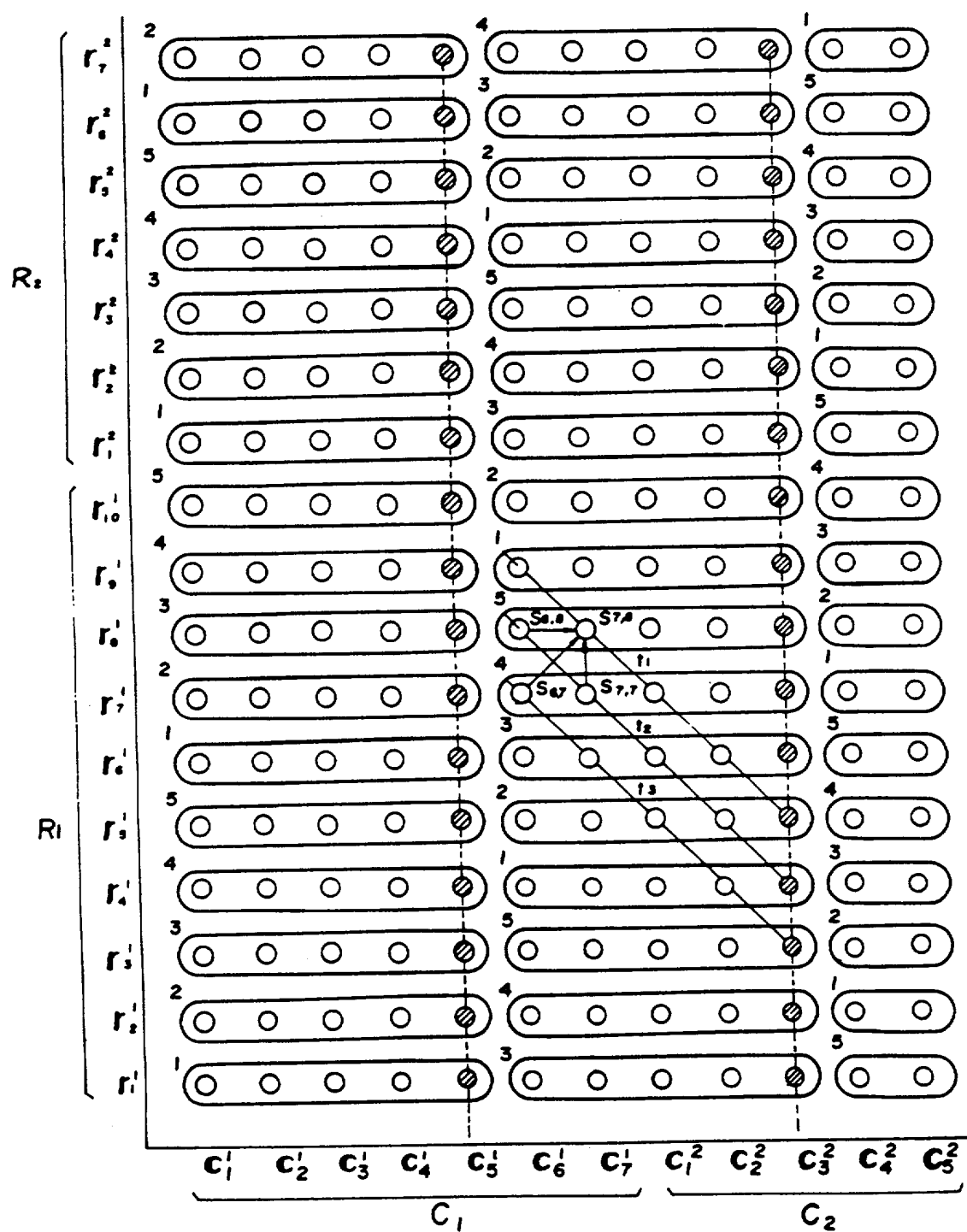


FIG. 11

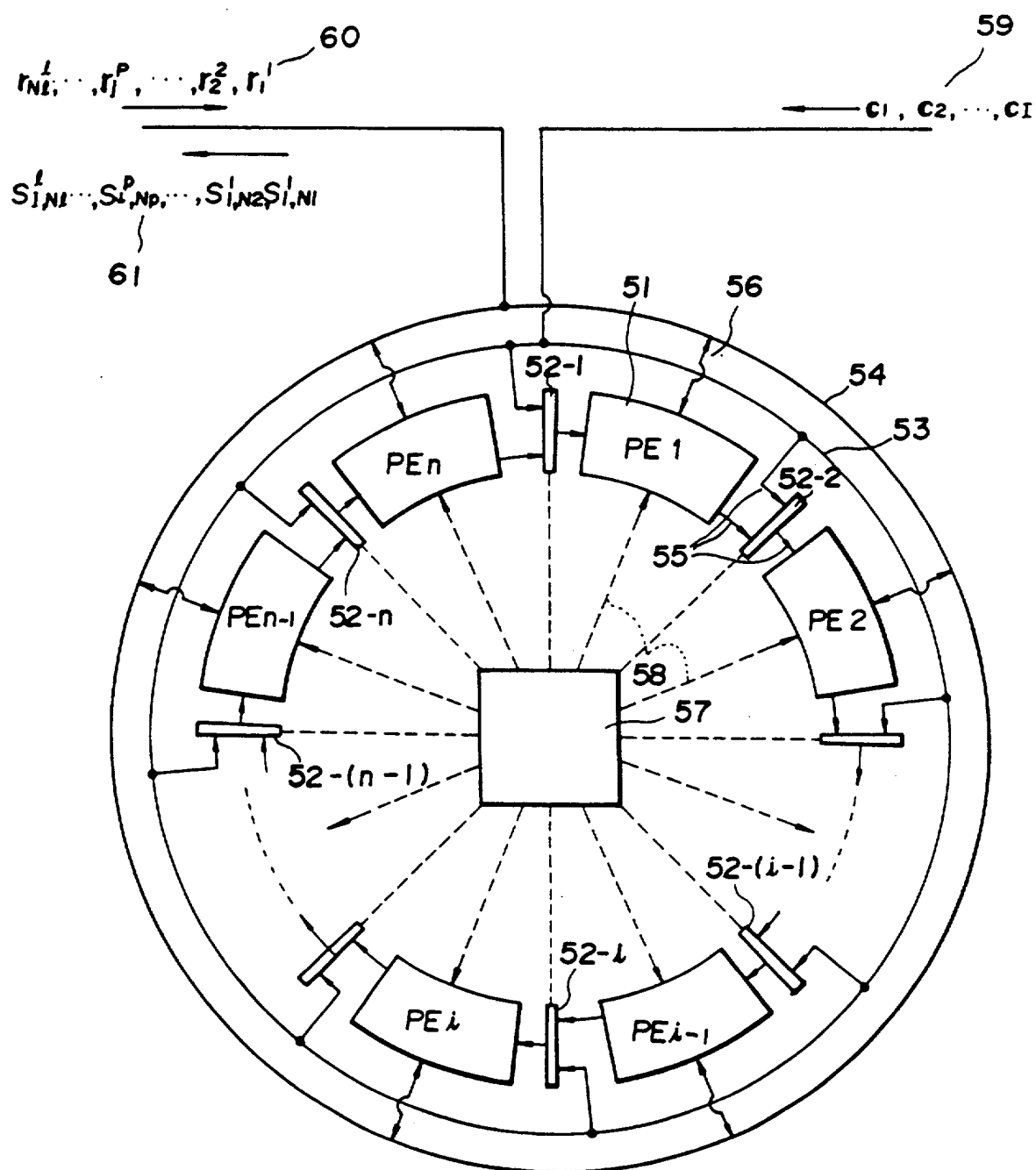


FIG. 12

	ING. INF.	KDP	TAIL	OVERDR. TEMP.
R1	r_1^1	1	0	0
	r_2^1	0	0	0
	$\vdots$	0	0	0
	r_{N1}^1	0	1	0
R2	r_1^2	1	0	0
	$\vdots$			
	$\vdots$	0	0	0
	$\vdots$			
Rl	r_{Nl}^{l-1}	0	1	0
	r_1^l	1	0	0
	$\vdots$	0	0	0
	r_{Nl-n+1}^l	0	0	1
	$\vdots$	0	0	1
	r_{Nl}^l	0	1	1

64 61 62 63

n

FIG. 13

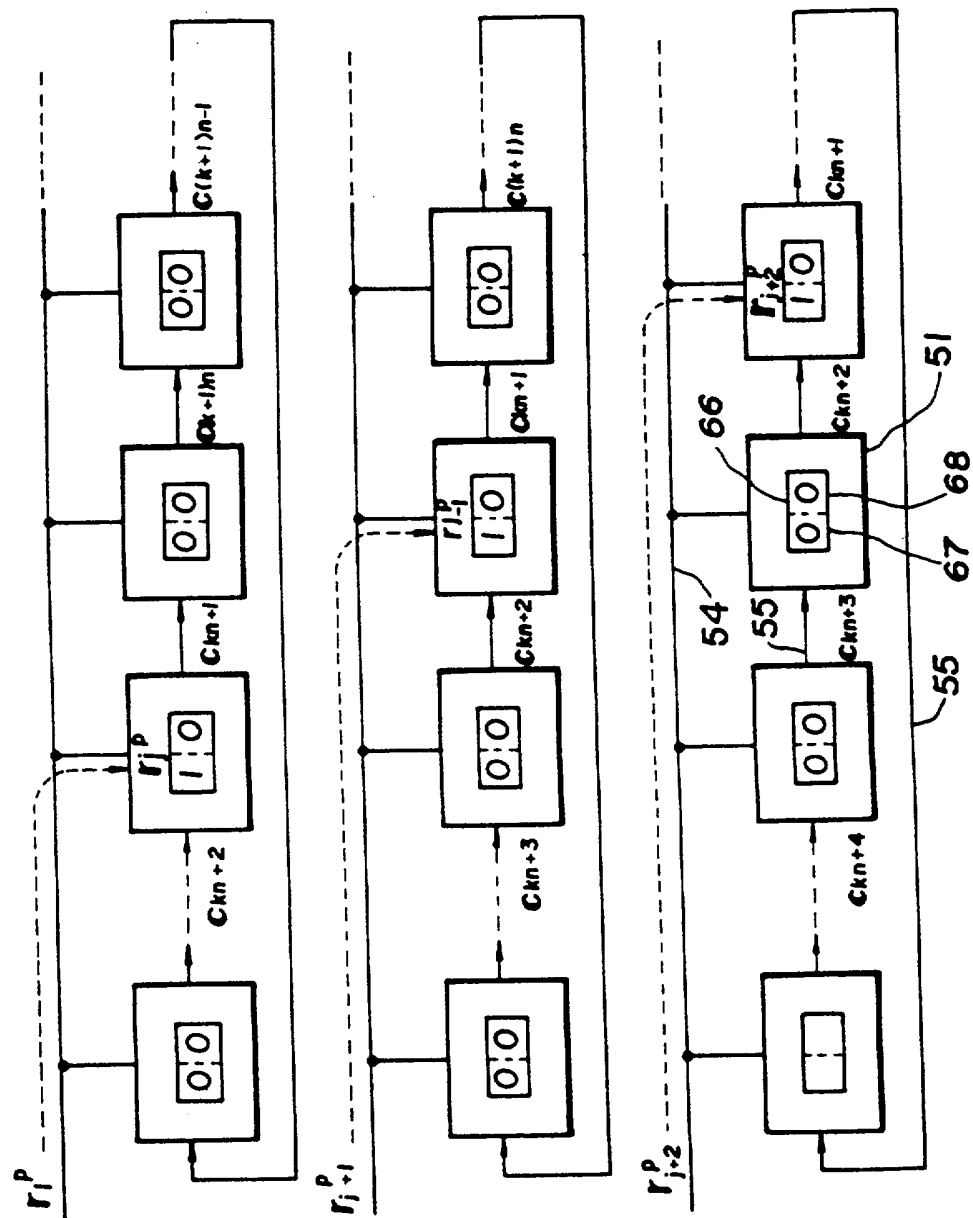


FIG. 14C

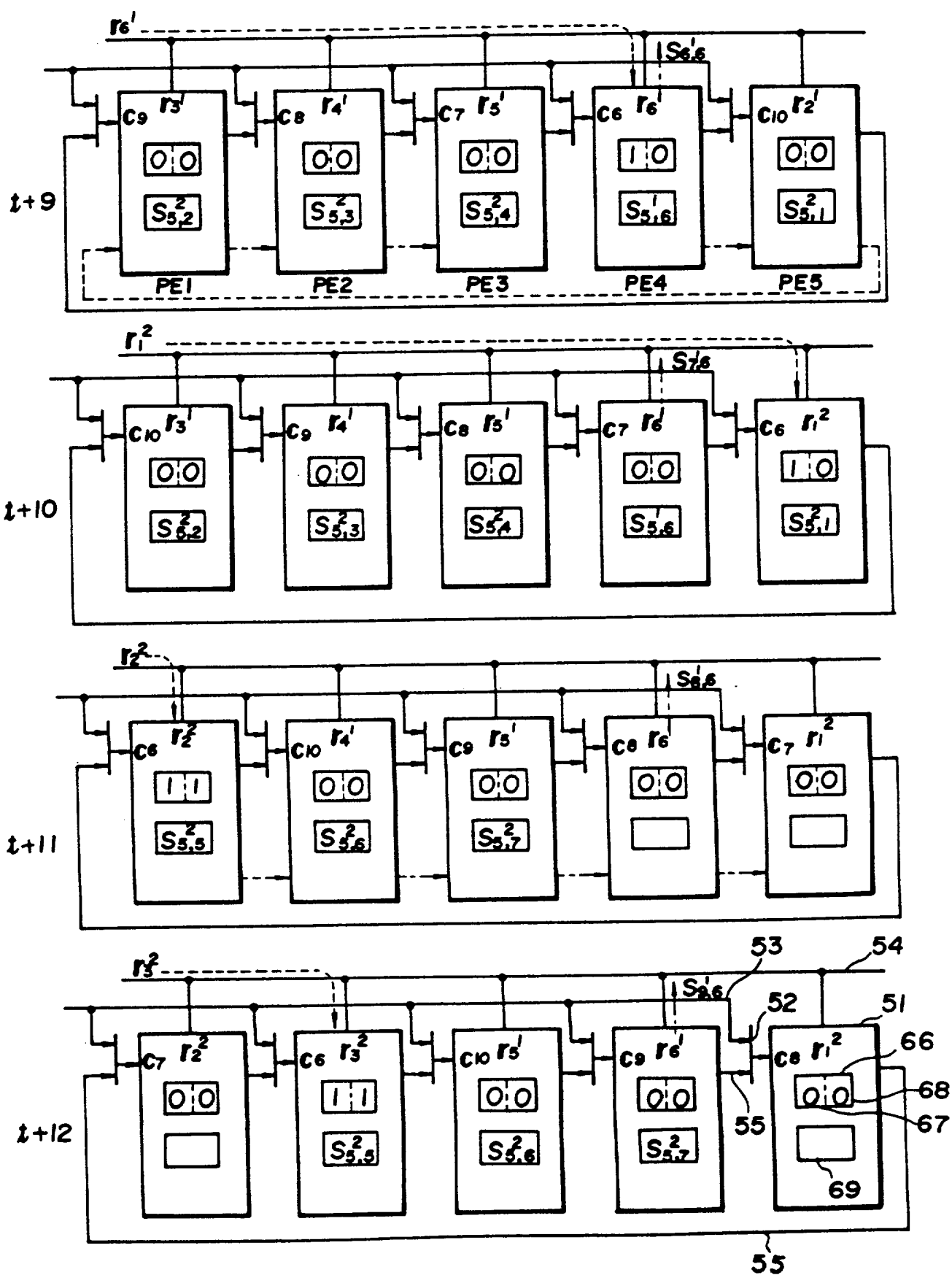


FIG. 14D

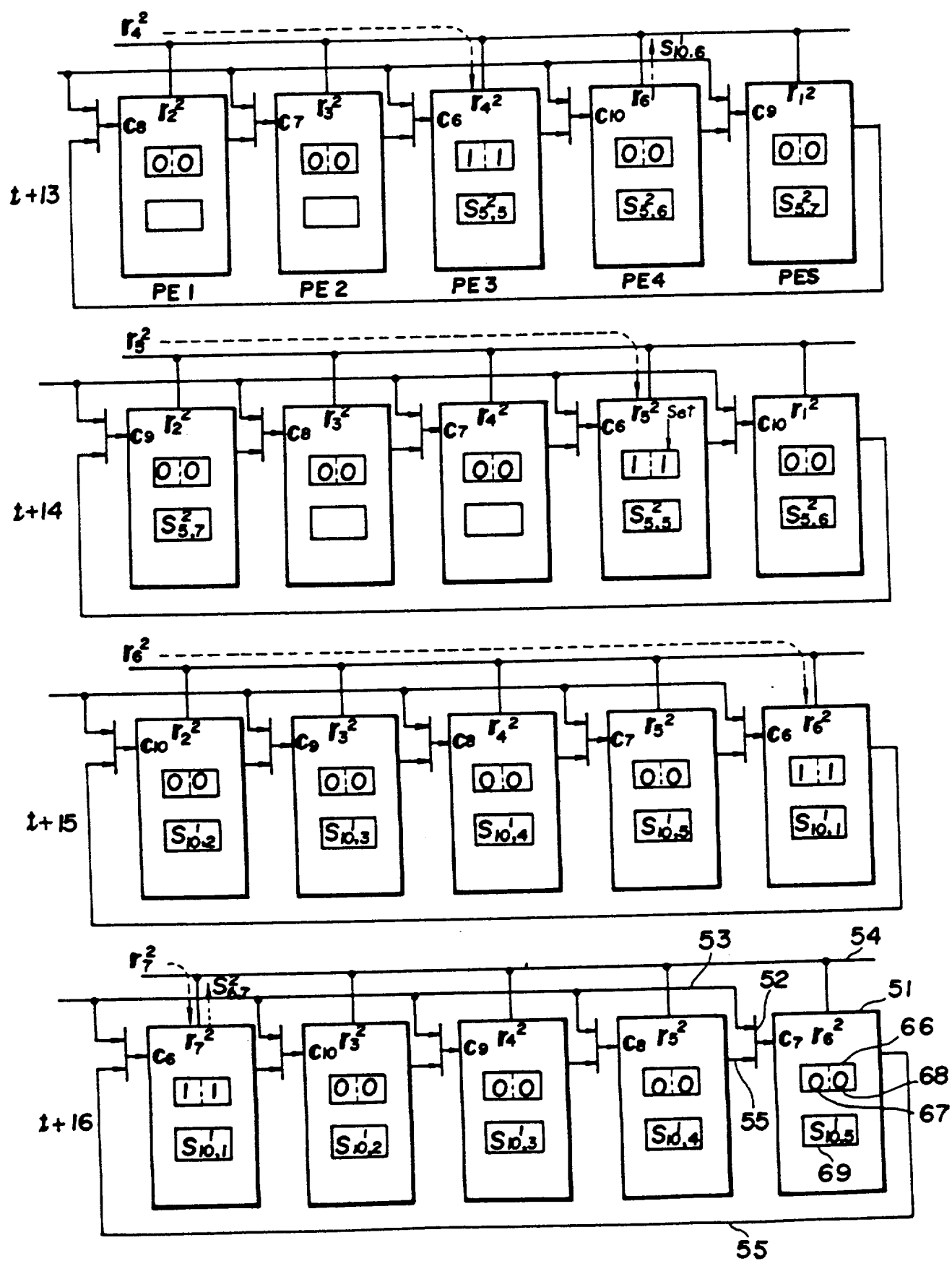
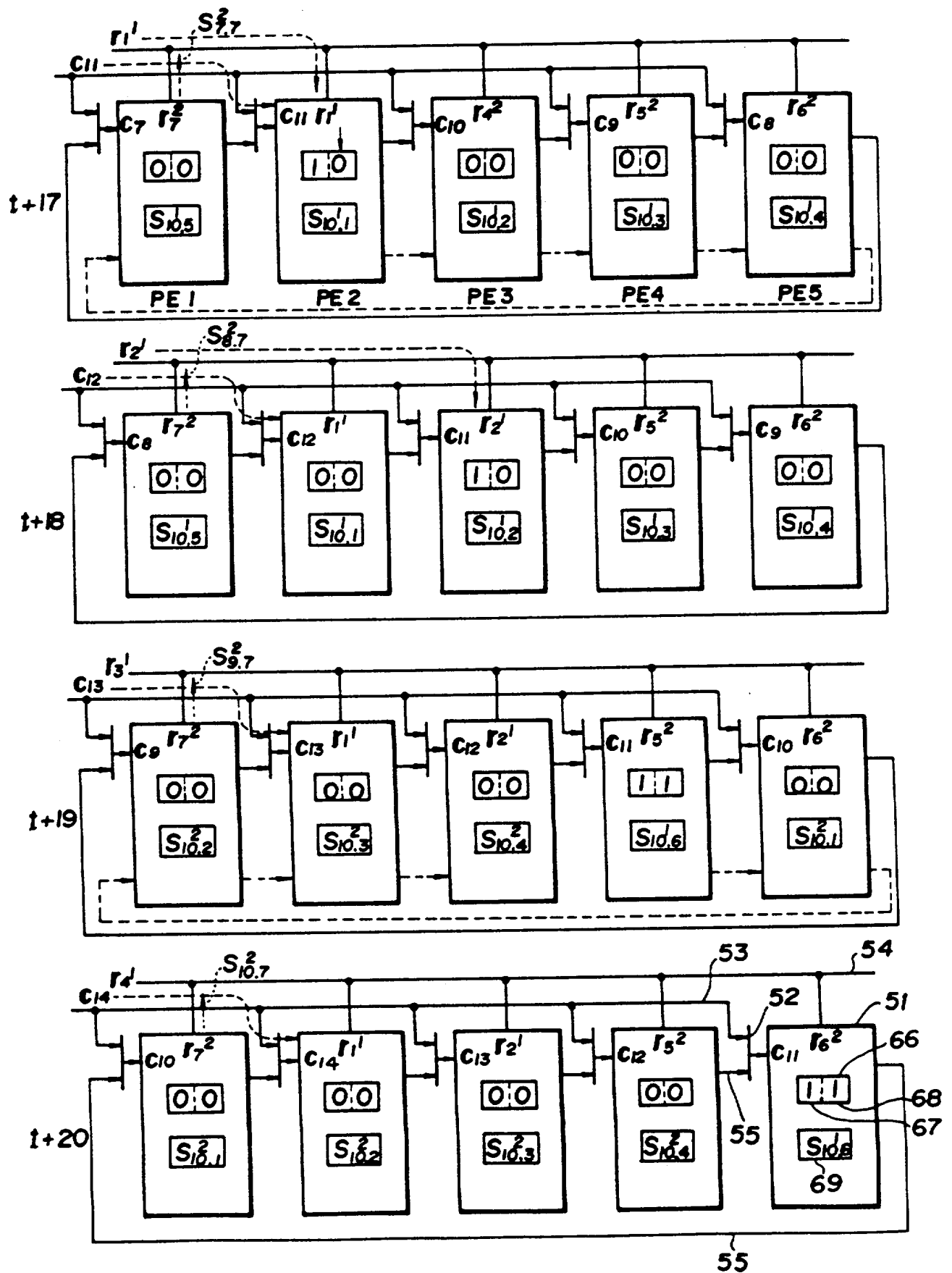


FIG. 14E



F I G. 15

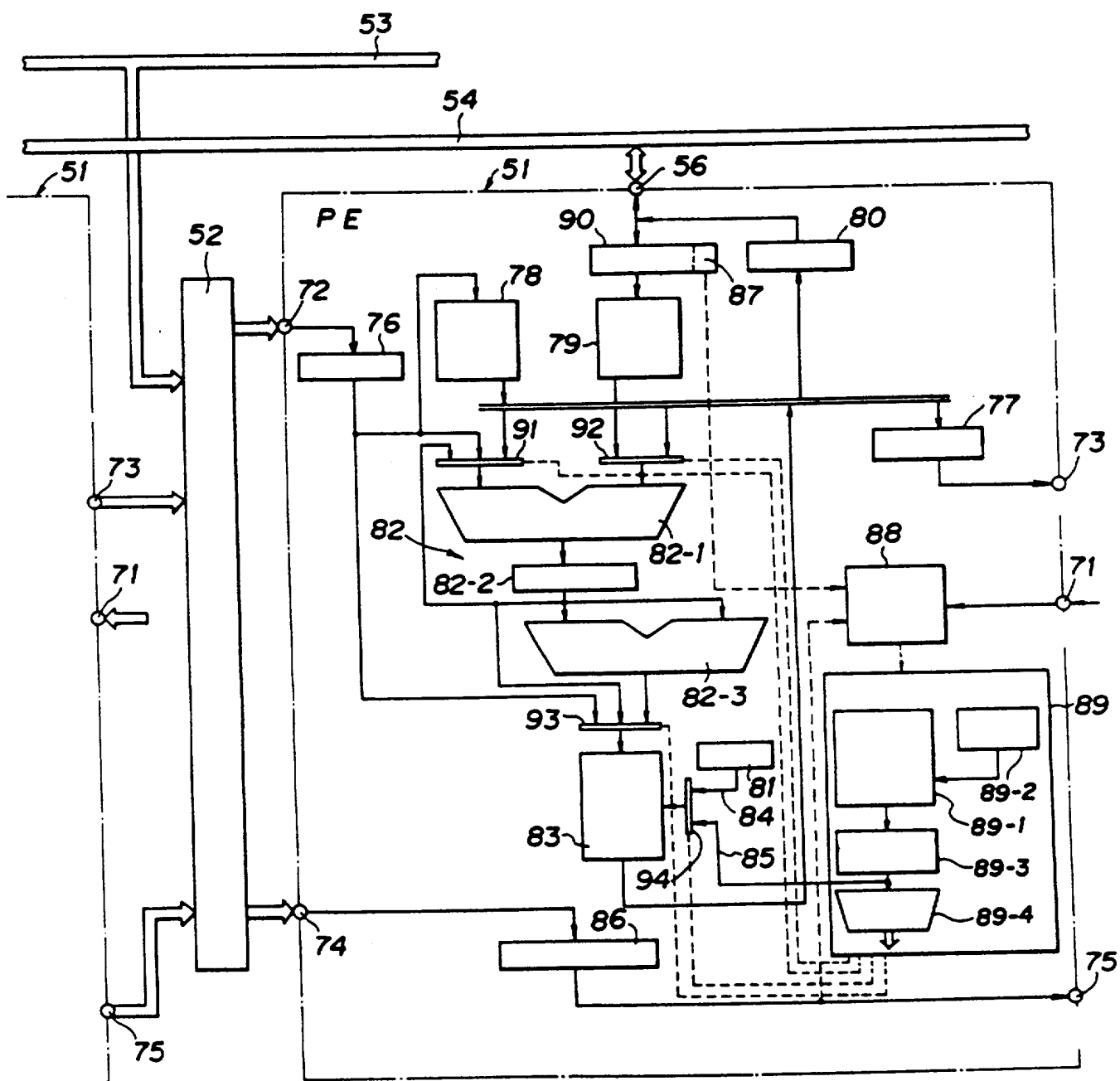


FIG. 16A

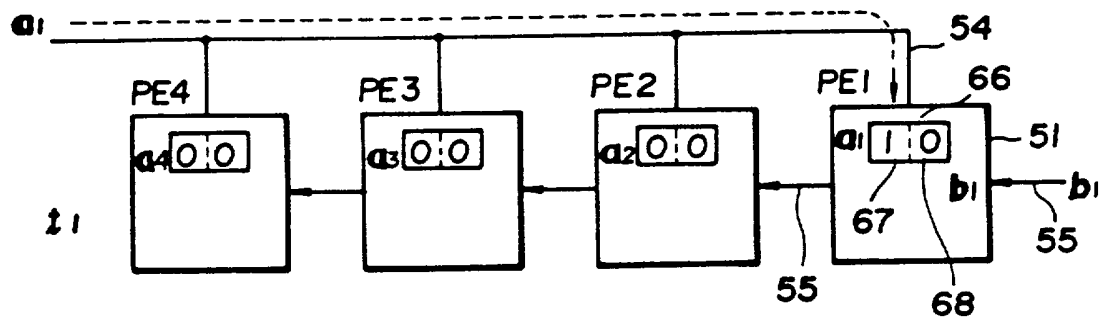


FIG. 16B

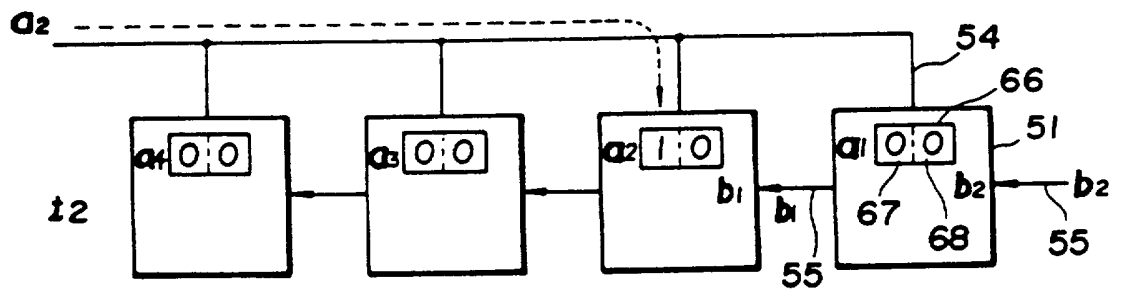


FIG. 16C

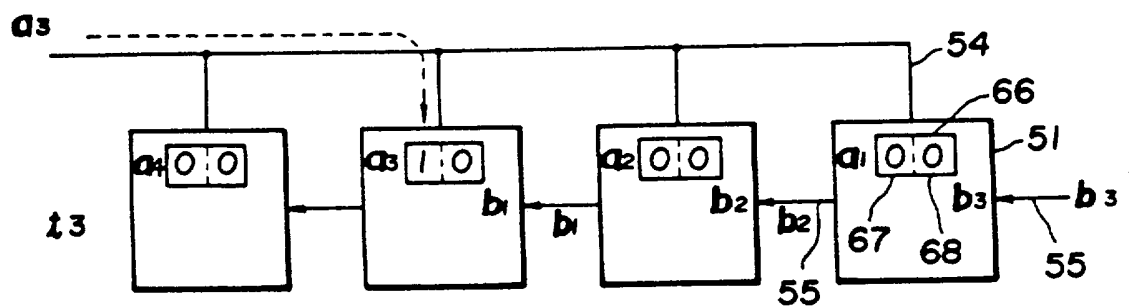


FIG. 16D

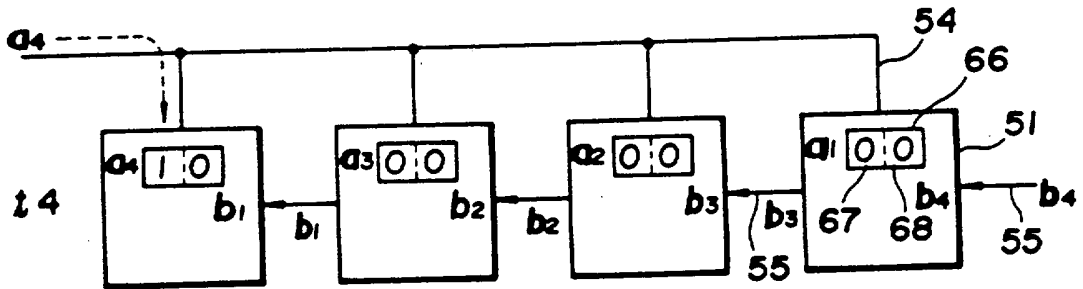


FIG. 16E

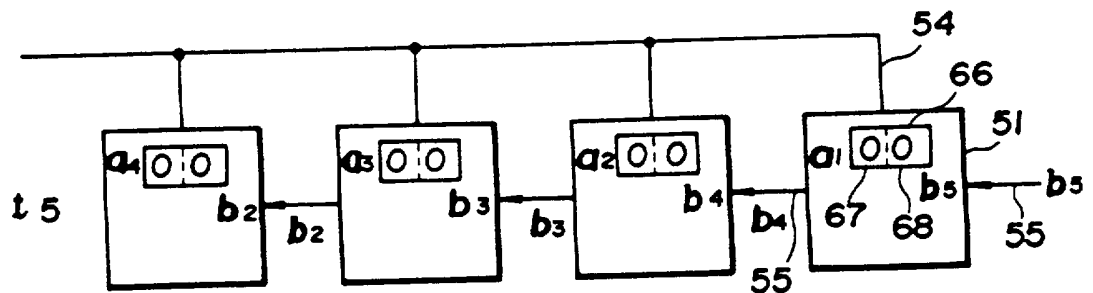


FIG. 16F

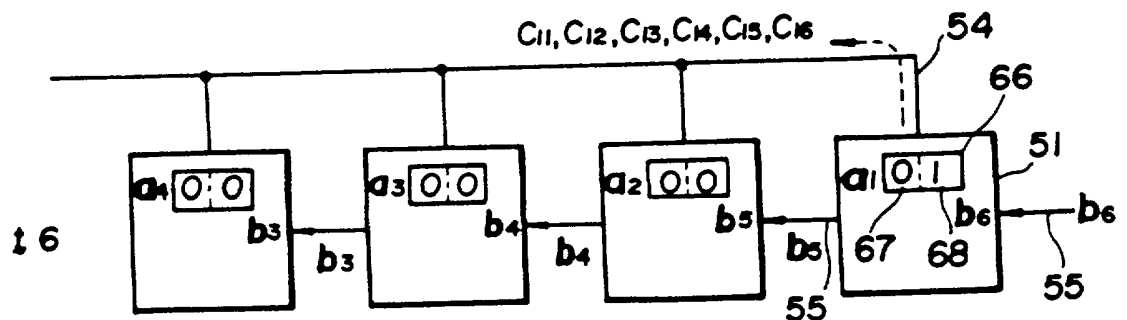


FIG. 16G

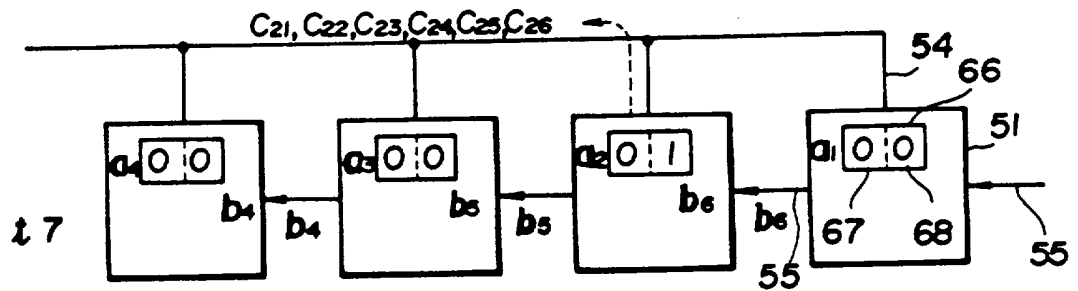


FIG. 16H

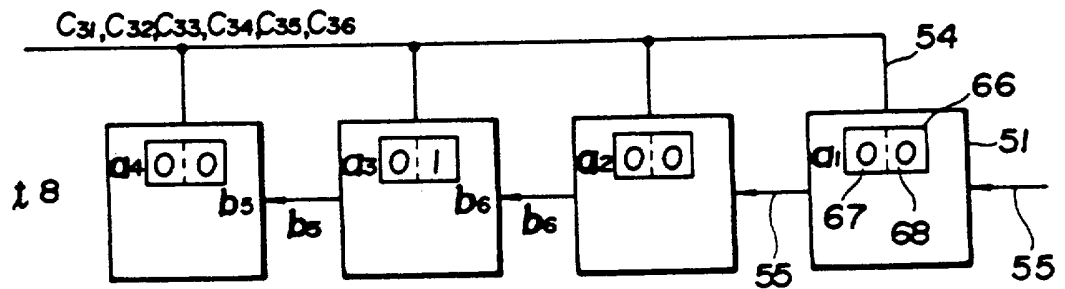


FIG. 16I

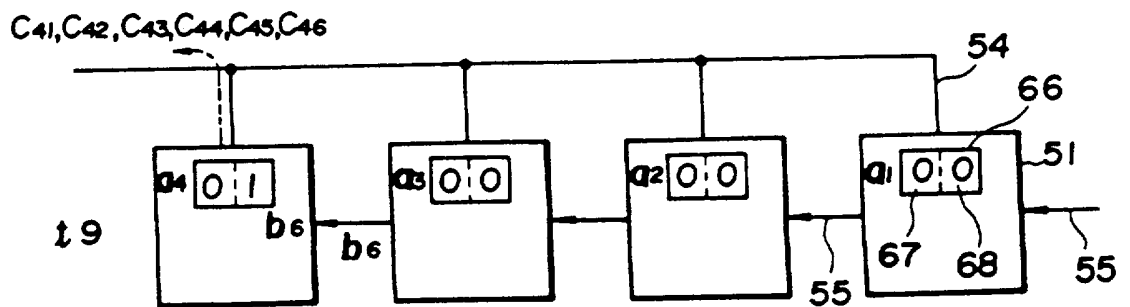
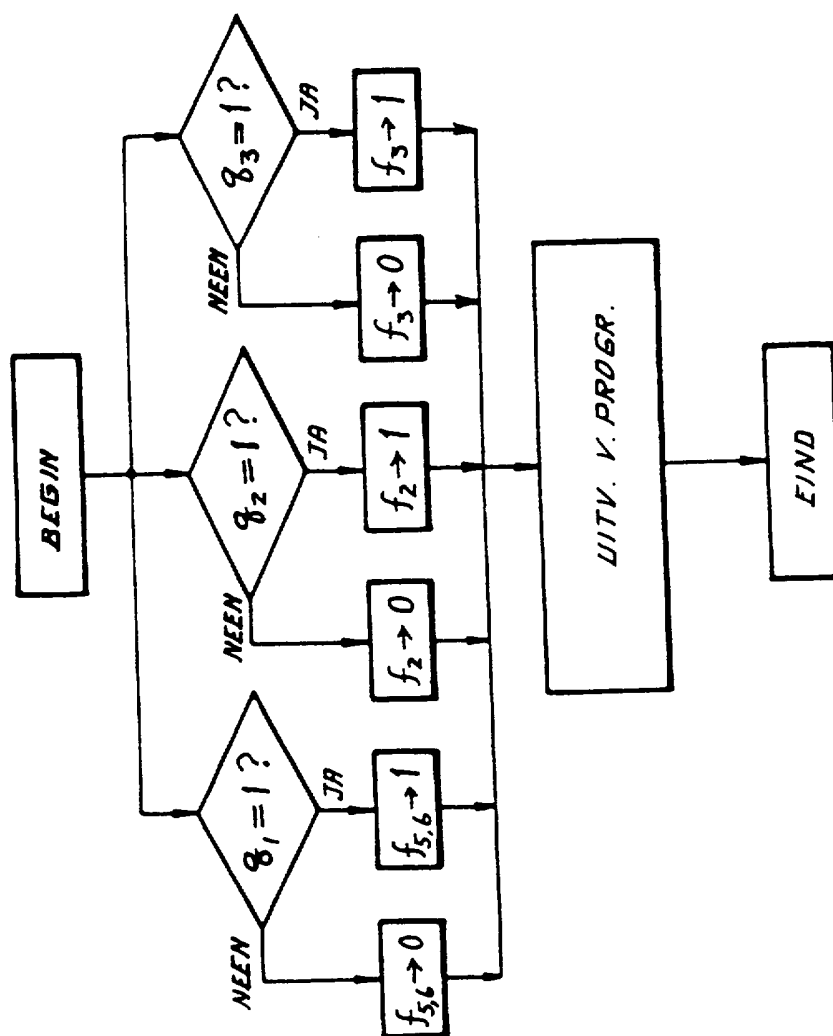


FIG. 17



F I G. 18

