

---

Octrooiraad



⑫ A Terinzagelegging ⑪ 8500005

Nederland

⑲ NL

---

⑤4 Stelsel voor het bewerken van informatie en het transporteren daarvan.

⑤1 Int.Cl<sup>4</sup>: G06F 13/12, G06F 13/38, G06F 15/16.

⑦1 Aanvrager: P. Romeijn Electro-techniek B.V. te Tilburg.

⑦4 Gem.: Ir. A.M.E. Siemens  
Roskam 8  
4813 GZ Breda.

---

②1 Aanvraag Nr. 8500005.

②2 Ingediend 3 januari 1985.

③2 --

③3 --

③1 --

⑥2 --

---

④3 Ter inzage gelegd 1 augustus 1986.

De aan dit blad gehechte stukken zijn een afdruk van de oorspronkelijk ingediende beschrijving met conclusie(s) en eventuele tekening(en).

---

N256

Uitvinders: Herman Johan Bennink, Tjeerd Walter Böttcher en Johannes Antonie Gerardus de Goeif allen te Tilburg en Bernard van den Dolder te Enschede.

- 5 Stelsel voor het bewerken van informatie en het transporteren daarvan.

De uitvinding heeft betrekking op een stelsel voor het bewerken van informatie en het transporteren daarvan in twee richtingen tussen verschillende eenheden, opgebouwd uit een aantal aan een gemeenschappelijk transmissiekanaal aangesloten eenheden, welke volgens het baton-pass principe toegang krijgen tot het transmissiekanaal.

- 15 Dergelijke stelsels zijn in velerlei uitvoeringsvormen bekend.

De uitvinding heeft ten doel om een stelsel van de in de aanhef genoemde soort te verschaffen, dat een efficiënte communicatie mogelijk maakt. Het stelsel wordt volgens de uitvinding daardoor gekenmerkt, dat elk van de eenheden een lokaal microprocessor gedeelte bevat voor het verzorgen van de communicatie met de lokale omgeving alsmede een communicatie-microprocessor gedeelte voor het verzorgen van de communicatie met het transmissiekanaal.

- 25 Een verdere voordelige uitvoeringsvorm van het stelsel volgens uitvinding wordt daardoor gekenmerkt, dat zowel het lokale als het communicatie-microprocessor gedeelte elk een eigen microprocessor, een eigen werkgeheugen en een eigen programmeergeheugen bevat.

- 30 Daarbij is volgens de uitvinding van voordeel, dat het werkgeheugen van het lokale microprocessor gedeelte direct toegankelijk is voor de microprocessor van het communicatie-gedeelte.

- 35 Volgens de uitvinding wordt bij voorkeur de volgorde waarin de eenheden berichten kunnen verzenden bepaald aan de hand van een in de eenheden opgeslagen, veranderbare tabel.

Van voordeel is daarbij volgens de uitvinding

85 00005

verder, dat de informatie tussen het lokale en het communicatie-microprocessor gedeelte uitgewisseld wordt met behulp van hand-shake technieken.

Volgens de uitvinding worden daarbij de  
5 systeembus van het lokale microprocessor gedeelte en de systeembus van het communicatie-microprocessor gedeelte aan elkaar gekoppeld met behulp van een buffer.

Vervolgens zal de uitvinding meer in detail worden beschreven aan de hand van de tekeningen, waarin:

10 Fig. 1 schematisch een stelsel volgens de uitvinding toont;

Fig. 2 een blokschema toont van een eenheid te gebruiken in een stelsel volgens de uitvinding;

15 Fig. 3 een blokschema toont volgens fig. 2, waarin informatietransport is aangegeven;

Fig. 4 schematisch de wijze toont, waarop de eenheid beslist of een ontvangen bericht geaccepteerd wordt;

20 Fig. 5 schematisch de wijze van verzenden van een bericht toont; en

Fig. 6 schematisch de aansluiting weergeeft van de hand-shake poorten volgens de figuren 1 en 2.

In het in fig. 1 weergegeven stelsel volgens de uitvinding is een aantal eenheden 16, 17 en 18 weergegeven,  
25 elk met een lokaal microprocessor gedeelte 19 en een communicatie-microprocessor gedeelte 20. Elk van de eenheden 16, 17 en 18 is aangesloten op een seriele verbindingslijn (het transmissiekanaal).

De in fig. 2 weergegeven eenheid volgens de  
30 uitvinding bevat een lokaal microprocessor gedeelte met de onderdelen 1, 2, 3, 7, 8 en 9 en een communicatie-microprocessor gedeelte met de onderdelen 4, 5, 6, 11, 12, 13 en 14. Gemeenschappelijk voor beide gedeelten zijn de onderdelen 10 en 15. Het lokale microprocessor gedeelte  
35 bevat een centrale verwerkingseenheid 1 (CPU), bijv.

8500005

opgebouwd uit een Z80 microprocessor, een programmeergeheugen  
8 (ROM), bijv. bestaande uit 8k EPROM, opgebouwd uit bijv.  
twee stuks 2732 (Intel, NEC of gelijkwaardig), een werk-  
geheugen 9 (RAM), bijv. opgebouwd uit twee stuks uPD449  
5 of uPD446 (NEC of gelijkwaardig), een handshake poort 3  
(Z80-PIO), in/uit lijnen 2 voor communicatie met de lokale  
omgeving (bijv. twee stuks Z80-PIO en een Intel 8279),  
waarmee weergeef-eenheden, toetsenborden en drukknoppen  
kunnen worden aangestuurd en elektrische (logische)  
10 signalen in en uit worden getransporteerd, en een buffer  
7 om rechtstreeks andere apparatuur aan de lokale  
systeembussen te kunnen aansluiten (bijv. SN74LS244 en  
SN74LS245). Voorts is er een voedingseenheid met een  
batterij 15 voor het leveren van de benodigde voedings-  
15 spanning en het onder spanning houden van de werkgeheu-  
gens, wanneer de voedingsspanning niet ingeschakeld is.  
Selectiemechanismen, resetmechanismen en poortschakelin-  
gen voor het opwekken van diverse logische signalen zijn  
terwille van de eenvoud niet weergegeven. De centrale  
20 verwerkingseenheid 1 bestuurt het lokale gedeelte van de  
eenheid. De programmastappen worden daarvoor over de  
systeembus uit het programmeergeheugen 8 gehaald, gedecodeerd  
en uitgevoerd. In het programmeergeheugen 8 wordt het  
programma ingebracht, dat het systeem bestuurt. De inhoud  
25 van het programmeergeheugen 8 is niet vluchtig. Het gaat niet  
verloren als de voedingsspanning uitgeschakeld wordt. In  
het programmeergeheugen 8 kan alleen maar uitgelezen worden.  
In het programmeergeheugen 8 kan geen informatie ingevoerd  
worden. Het werkgeheugen 9 is aangesloten op de  
30 systeembussen. In het werkgeheugen kan geschreven en  
gelezen worden. De inhoud van het werkgeheugen 9 wordt  
onder programmabesturing gewijzigd.

Het communicatie-microprocessor gedeelte bevat  
een Z80 microprocessor 5 als centrale verwerkingseenheid,  
35 een programmeergeheugen 12, bijv. bestaande uit 8k EPROM,  
bijv. gerealiseerd met behulp van twee stuks 2732, Intel

8500005

NEC of gelijkwaardig, het werkgeheugen 11 (RAM), bijv. 12  
stukks uPD449 of uPD446 (NEC of gelijkwaardig), een hand-  
shake poort (Z80-PIO) om commando's met het lokale gedeelte  
te kunnen uitwisselen (het onderdeel 4), een bufferbank 10,  
5 om de systeembus van het communicatie-gedeelte en van het  
lokale gedeelte aan elkaar op gedefinieerde wijze te kunnen  
verbinden, en een protocol controller (Z80-SIO/2) 13 als  
uitvoering van de schakeling voor het lagere niveau commu-  
nicatieprotocol, een zender/ontvanger 14, die de spannings-  
10 niveau's van ontvangen en te verzenden signalen aanpast en  
tevens de codering van de informatie verzorgt.

Selectiemechanismen, de gedetailleerde  
uitvoeringen van klokschakelingen, logische poorten voor  
het opwekken van verschillende logische signalen, de reset-  
15 schakeling, de lijntrafo voor de galvanische scheiding  
tussen de seriële lijn en de eenheid, en andere onderdelen  
zijn terwille van de eenvoud niet in de tekening  
weergegeven. Met het verwijzingscijfer 6 is nog de  
klokschakeling aangegeven. Het communicatie-gedeelte  
20 verzorgt de uitwisseling van gegevens tussen de eenheden  
onderling over de seriële verbindingslijn. Bij het  
verzenden van gegevens vanuit het lokale gedeelte, worden  
deze gegevens door het communicatie-gedeelte uit het lokale  
gedeelte gehaald, voorzien van een aantal controle-woorden,  
25 omgezet in seriële vorm en via de seriële verbindingslijn  
verzonden. Van de over de seriële verbindingslijn ontvangen  
gegevens wordt eerst nagegaan of deze onverminkt ontvangen  
zijn (lagere niveau protocol), en zoja, onderworpen aan een  
aantal controles om te beslissen of dit type boodschap door  
30 de eenheid ontvangen mag worden (hogere niveau protocol).  
Een andere belangrijke taak van het communicatie gedeelte  
is het doorgeven van de baton. Om namelijk te voorkomen dat  
er meer dan een eenheid tegelijkertijd een boodschap op de  
lijn zet, wordt het zogenaamde baton-pass mechanisme  
35 toegepast. De werking van dit mechanisme wordt in het kort  
hierna beschreven.

Het uitwisselen van gegevens door de systeemdelen

85 00005

onderling gebeurt d.m.v. boodschappen. In deze boodschappen staat een bestemmingsadres, een afzender, een commando, en indien noodzakelijk extra informatie. Een eenheid mag alleen een boodschap versturen (op de lijn zetten) als deze 5 eenheid de baton in zijn bezit heeft. Aangezien er maar 1 baton in het systeem aanwezig is, kan er dus maar een boodschap tegelijk op de lijn gezet worden. Na ontvangstbevestiging wordt de baton doorgegeven aan een andere eenheid. Komt er geen ontvangstbevestiging dan wordt de 10 boodschap een aantal keren herhaald, waarna de baton wordt doorgegeven aan een andere eenheid. Het doorgeven aan een eenheid gebeurt volgens een tabel waarin de route van de baton staat. Is deze tabel niet aanwezig in de eenheid, dan wordt de baton volgens een bepaald algoritme doorgegeven. 15 Verder kan het voorkomen dat t.g.v. een storing er geen of meer dan een baton in het systeem voorkomt. Als er geen baton in het systeem voorkomt, zal na het aflopen van een time-out een baton gegenereerd worden en de communicatie komt opnieuw op gang. Als er meer dan een baton in het 20 systeem aanwezig is, worden alle in het systeem aanwezige batons vernietigd. Het systeem is nu in de toestand gekomen dat er geen batons zijn, hoe dit opgelost wordt is reeds hierboven beschreven. Het baton-pass mechanisme moet "spraakverwarring" op de lijn voorkomen. Daarnaast wordt 25 aan alle boodschappen op de lijn een cyclische karakter check (CRC) toegevoegd. Dit is een controle-woord waarmee aan de ontvangende kant gecontroleerd kan worden of de booschap onverminkt ontvangen is.

Zodra van een bericht is vastgesteld dat het 30 bestemmingsadres overeenkomt met het adres van de eenheid, start de PROTOCOL CONTROLLER 13 het binnenhalen van het bericht. Daarbij worden de serie-data omgezet naar parallel (8 bits breed). Zodra een byte beschikbaar is gekomen, wordt dit aan de communicatie-processor 5 gemeld middels 35 een interrupt. De processor 5 haalt vervolgens het byte op

8500005

van de PROTOCOL CONTROLLER 13 en plaatst deze in zijn RAM-geheugen 11 (het bericht wordt door de communicatie-processor tijdelijk in de RECEIVER QUEUE geplaatst). Parallel aan de acties van de processor 5 blijft de PROTOCOL CONTROLLER 13  
5 data van de lijn binnenhalen totdat het einde van het bericht is gedetecteerd. De processor 5 haalt elk beschikbaar gekomen byte direkt weg bij de PROTOCOL CONTROLLER 13. Automatisch wordt door de PROTOCOL CONTROLLER 13 een check uitgevoerd op het onverminkt zijn  
10 van het binnengehaalde bericht. Over het bericht wordt een controle-woord bepaald dat aan het einde van de ontvangst-handeling wordt getest tegen het controle-woord dat in het bericht is meegezonden (laatste 2 bytes van het bericht; deze test staat bekend onder de naam CRC-check). Het  
15 resultaat van deze vergelijking is voor de processor beschikbaar.

Tijdens het verplaatsen van een ontvangen bericht door de processor 5 interpreteert deze het bericht en beslist met gebruikmaking van beschikbare gegevens of het  
20 ontvangen bericht zodanig is dat het door de unit geaccepteerd mag worden. Deze beslissing wordt als volgt verricht (zie ook figuur 4).

De inhoud van het commandoveld van het bericht is een index in de COMMANDO LIJST (COMMANDLIST). De inhoud van  
25 de COMMANDLIST is een verwijzing naar een rij van de 2-dimensionale array RECEIVING COMMUNICATION TABLE (RECCOMTABLE). Tevens wordt via de SOURCE ADDRESS LIST (SALIST) en de inhoud van het "source address"-veld van het bericht (als index in SALIST) een kolom van de array  
30 RECCOMTABLE aangewezen. De zo aangewezen rij en kolom in RECCOMTABLE leveren een event-waarde op welke tot een van de volgende twee acties kan leiden:

De gevonden event-waarde is "nil". Dit betekent dat het ontvangen bericht niet geaccepteerd mag worden.  
35 Zodra het gehele bericht is ontvangen en bij de ontvangst

8500005

geen vermindering is geconstateerd, wordt naar de zender van het bericht (d.i. de eenheid met adres als is aangegeven in het "source address"-veld van het bericht) een negatieve ontvangstbevestiging (NACK-bericht) verzonden. Vervolgens  
5 wordt het ontvangen bericht uit het RAM-geheugen 11, d.i. uit de RECEIVER QUEUE, "verwijderd". Hiermee is de activiteit rond het ontvangen van het bericht voltooid en start het scannen van de lijn opnieuw.

De gevonden event-waarde is ongelijk "nil". Dit  
10 betekent dat de eenheid het bericht accepteert indien het onverminkt is ontvangen en de juiste lengte heeft. De zender van het bericht krijgt in dit geval een positieve ontvangstbevestiging (ACK-bericht) toegezonden. Het bericht blijft in de RECEIVER QUEUE gehandhaafd. Het scannen van de  
15 lijn op nieuwe berichten wordt weer voortgezet. Vervolgens tracht de communicatie-processor 5 berichten vanuit deze queue zo snel mogelijk verder te transporteren. Daartoe wordt de voor het ontvangen bericht reeds bepaalde event-waarde gebruikt als index in de RECEIVING MEMORY  
20 LOCATION TABLE (RECMEMLOCTABLE). Deze tabel wordt bij het opstarten van de unit gevuld vanuit het lokale deel en bevat per event-waarde de volgende gegevens:

Pointer in het geheugen. Deze pointer geeft aan, waar in het geheugen het bericht uiteindelijk moet worden  
25 geplaatst. Uit deze pointer-waarde volgt nu voor de communicatie-processor eenvoudig het gegeven voor welk unit-deel het bericht bestemd is. Immers de pointer wijst naar een locatie in het lokale RAM-geheugen (adressen vanaf 8000H) of in het communicatie-geheugen (adressen tot  
30 8000H).

Berichtlengte. Dit gegeven is medebepalend voor het correct verklaren van een ontvangen bericht (zie boven).

Indien het bericht bestemd is voor het lokale  
35 deel wordt de event-waarde aan het lokale deel aangeboden

8500005

via de hand-shake poorten 3 en 4. Dit is te beschouwen als een REQUEST welke vanuit het lokale deel wordt beantwoord met een toestemming of een verwerping. Bij een verwerping blijft het bericht in de queue staan tot een later  
5 tijdstip. De communicatie-processor 5 trekt zijn REQUEST in, gevolgd door het intrekken van de verwerping door de lokale processor 1. Bij een toestemming start de communicatie-processor 5 het transport van het bericht naar het RAM-geheugen 9 van de lokale processor zoals dat door  
10 de pointer uit de RECMEMLOCTABLE wordt aangegeven. Het transport tussen het lokale- en communicatie deel verloopt via de buffers 10. Zodra het transport is voltooid, trekt de communicatie-processor 5 zijn REQUEST terug, ten teken dat het bericht is overgezet. De lokale processor 1  
15 beantwoordt dit door het intrekken van zijn toestemming. Hiermee is de ontvangst van het bericht vanaf de lijn voltooid en kan de lokale processor 1 het bericht verwerken.

Tot slot is het bij het ontvangen van berichten  
20 mogelijk dat de data vanaf de lijn verminkt zijn, bijvoorbeeld t.g.v. storingen op de lijn. Aangezien in dit geval ook geen zekerheid bestaat over de juistheid van het bestemmings-adres, wordt het bericht uit de RECEIVING QUEUE verwijderd. Er vindt geen ontvangstbevestiging plaats (ACK-  
25 noch NACK-bericht). De zender zal na uit-timing het bericht gaan herhalen. Dit wordt als een normaal bericht afgewerkt, zoals hierboven is beschreven.

Bij het verzenden van een bericht vindt een soortgelijke procedure plaats als bij het ontvangen van  
30 berichten (zie figuur 5). De lokale processor 1 biedt nadat hij het (de) bericht(en) op de juiste plaats(en) heeft klaargezet aan de communicatie-processor 5 een event-waarde aan via de hand-shake poorten 3 en 4. Dit is te beschouwen als een zend-request, nu echter vanuit het lokale deel. Dit  
35 REQUEST wordt vanuit het communicatie-deel beantwoord met

8500005

een bevestiging (het bericht kan nog in een TRANSMIT QUEUE worden opgenomen) of een verwerping (geen ruimte meer beschikbaar in het RAM-geheugen 11 van het communicatie-deel).

5 Bij een verwerping worden het REQUEST en de verwerping weer ingetrokken en neemt de lokale processor 1 de nodige acties. Bij een bevestiging haalt de communicatie-processor 5 het (de) bericht(en) vanuit de lokale RAM 9 via de buffer 10 tussen het lokale en  
10 communicatie-deel naar zijn eigen RAM 11 en plaatst deze in de juiste TRANSMIT QUEUE('s) overeenkomstig de bij dat (die) bericht(en) behorende prioriteitsklasse(n). Elke prioriteitsklasse heeft zijn eigen TRANSMIT QUEUE. Nadat deze transporten zijn verricht, worden bevestiging en REQUEST  
15 weer ingetrokken ten teken dat het (de) bericht(en) is (zijn) weggehaald. Voor de lokale processor 1 is dit identiek aan het gegeven dat het (de) bericht(en) is (zijn) verzonden. De eigenlijke verzending over de lijn is geheel de verantwoordelijkheid van het communicatie-deel.

20 De bepaling van het aantal berichten en hun RAM-locatie(s) wordt als volgt door de communicatie-processor verricht:

De van het lokale deel ontvangen event-waarde is een index in de SEND EVENT TABLE (SENDEVENTTABLE). Deze  
25 tabel bevat per event-waarde een pointer en een "bericht aantal".

De pointer is een index in de TRANSMITTING COMMUNICATION TABLE (TRANSCOMTABLE). Deze tabel bevat per index de volgende 4 elementen:

30 - Bericht Type. Dit is een index in de TRANSMITTING MEMORY LOCATION TABLE (TRANSMEMLOCTABLE). Deze tabel wordt vanuit het lokale deel bij het opstarten van de unit gevuld. De tabel bevat per index:

-- een pointer in het geheugen. Deze pointer  
35 geeft aan waar in het geheugen het bericht (althans het

85 00005

data-veld van het bericht) gereed staat. Vanaf deze locatie wordt het bericht verplaatst naar de juiste TRANSMIT QUEUE.

-- berichtlengte. Dit is het aantal te transporteren bytes.

5 - Prioriteitsklasse. Dit is de prioriteitsklasse van het bericht en bepalend voor de te selecteren TRANSMIT QUEUE.

- Unit List pointer. Dit is een index in de UNIT LIST (UNITLIST), waarvan de inhoud weer verwijzen naar 10 een index in de ADDRESS LIST (ADRLIST). Deze ADRLIST bevat het uiteindelijke unit-adres welke het "destination"-adres voor het bericht vormt.

- Unit Aantal. Deze waarde geeft aan naar hoeveel units het bericht (volgend uit het Bericht Type element) 15 dient te worden verzodne. Het is het aantal opeenvolgende elementen welke uit de UNITLIST genomen moeten worden, beginnend bij de index volgend uit het Unit-List-pointer element.

Het "Bericht Aantal" geeft aan hoeveel bericht- 20 typen verzonden moeten worden t.g.v. de event-waarde. Dit aantal komt overeen met het aantal indexen dat uit de TRANSMCOMTABLE geselecteerd moeten worden, beginnend bij die index als is aangegeven door de pointer uit de SEND-EVENTTABLE.

25 De organisatie van de TRANSMIT QUEUE's is zodanig opgezet dat per prioriteitsklasse een bericht hoogstens een keer in een QUEUE wordt opgenomen. Vanuit deze QUEUE's worden de berichten, voorzien van afzender en bestemmings-adressen, over de lijn verzonden. Het 30 communicatie-deel wacht totdat het van een andere unit de BATON krijgt aangeboden. Is dat het geval en zijn niet alle QUEUE's leeg, dan wordt het oudste bericht uit de hoogste prioriteitsklasse verzonden. Na ontvangst van een bevestigingsbericht of het overschrijden van het maximaal 35 aantal keren dat een bericht herhaald mag worden, wordt de

8500005

de BATON doorgegeven aan een andere unit. Dit gebeurt ook indien alle TRANSMIT QUEUE's leeg zijn. Wordt een verzonden bericht niet door de ontvanger aangenomen of beantwoord met een negatieve ontvangstbevestiging, dan kan het bericht in  
5 een QUEUE blijven staan, afhankelijk van de prioriteits-klasse. Het lokale deel wordt hiervan op de hoogte gebracht.

Tot zover de behandeling van data-uitwisseling via de transmissielijn. De behandeling van de tweede  
10 categorie data-uitwisseling is als volgt:

De data-uitwisseling via de I/O-lijnen 2 geschiedt geheel onder controle van de lokale processor. De betekenis van de I/O-poorten 2 is per unit-type verschillend en kan als input of output worden  
15 gedefinieerd. De input kan daarbij op polling- of op interrupt-basis worden binnengehaald, afhankelijk van de geprogrammeerde mode van de I/O-bouwstenen. De besturingssoftware van de lokale processor 1 bepaalt in feite de acties op de verkregen input, hetgeen outputs op bepaalde  
20 I/O-lijnen 2 tot gevolg kan hebben. Verkregen inputs kunnen aanleiding zijn tot verzenden van bepaalde berichten als hiervoor besproken. Inputs van de I/O-lijnen 2 gaan naar de lokale processor 1 en kunnen van daaruit eventueel naar het lokale RAM-geheugen 9 worden verplaatst. Outputs naar de  
25 I/O-lijnen 2 komen van de lokale processor 1, die ze eventueel weer uit het lokale RAM-geheugen 9 heeft opgehaald.

Vervolgens zal besproken worden hoe het lokale en communicatie-deel elkaar de nodige informatie toespelen.  
30 In het voorgaande zagen we reeds dat dit deels geschiedt via de hand-shake poorten 3 en 4. Figuur 6 geeft schematisch aan hoe deze poorten op elkaar zijn aangesloten. Van elke Z80-PIO is een poort als input en de andere als output gedefinieerd. De input en output poorten

8500005

zijn daarbij kruislings aangesloten. De poorten worden geprogrammeerd in de zogenaamde strobe mode. Dit houdt in dat als er aan de input poort een wijziging optreedt, er een interrupt gegenereerd wordt naar de bijbehorende processor. Deze verwerkt dan de nieuwe input in de geactiveerde interrupt afhandelings-routine.

De beide processoren geven elkaar via deze poorten de informatie welke van een van de volgende typen is:

- 10 - commando. Opdracht tot het uitvoeren van actie(s) m.u.v. het verzenden van berichten.
- status. Het overdragen van toestandsinformatie.
- event. Het melden van een ontvangen bericht (van communicatie naar lokaal) resp. het opdrachtgeven tot 15 verzending van berichten (van lokaal naar communicatie).

Bevestiging van de ontvangen informatie geschiedt expliciet, door het geven van bevestigingsinformatie, of impliciet, door het intrekken van de gegeven informatie.

De informatie welke wordt overgedragen is een 20 byte lang. Voor de meeste toepassingen is dit ruim voldoende. Indien de informatie meer bytes vereist, wordt eerst in het lokale RAM-geheugen 9 alle informatie opgeslagen, waarna via de hand-shake poorten 3 en 4 pointer informatie wordt aangeboden. De uiteindelijke 25 informatie-overdracht vindt in dit geval dus plaats via het lokale RAM-geheugen 9.

Tot slot volgt nog een verklaring van de wijze waarop de communicatie-processor 5 in het RAM-geheugen 9 van het lokale deel kan opereren. De opzet van het systeem 30 is zodanig gekozen dat de communicatie-processor 5 bij elke I/O-poort en bij elke RAM-lokatie terecht kan. Hieruit volgt dus dat alleen de lokale programma ROM's slechts voor de lokale processor 1 bereikbaar zijn. Daarnaast zijn de interrupt-bussen ook volledig gescheiden, zodat interrupts 35 slechts op die processor ingrijpen waar het interrupt

8500005

thuishoort.

Indien nu de communicatie-processor 5 wil  
opereren op I/O-poorten of geheugen van het lokale deel,  
worden de buffers tussen de lokale en communicatie-delen  
5 vrijgegeven, zodat als het ware beide systeem-delen aan  
elkaar gekoppeld worden. Zodra de communicatie-processor 5  
de werkzaamheden in het lokale deel heeft beëindigd, worden  
de buffers 10 tussen lokale en communicatie-delen  
gedisabled. Beide systeem-delen zijn dan weer gescheiden  
10 van elkaar en de lokale processor 1 gaat weer verder met  
zijn eigen programma met die instructie waar hij was  
gebleven.

8500005

Conclusies:

1. Stelsel voor het bewerken van informatie en het transporteren daarvan in twee richtingen tussen  
5 verschillende eenheden, opgebouwd uit een aan een aantal  
aan een gemeenschappelijk transmissiekanaal aangesloten  
eenheden welke volgens het baton-pass principe toegang  
krijgen tot het transmissiekanaal, met het kenmerk, dat elk  
van de eenheden een lokaal microprocessor gedeelte bevat  
10 voor het verzorgen van de communicatie met de lokale  
omgeving alsmede een communicatie-microprocessor gedeelte  
voor de communicatie met het transmissiekanaal.

2. Stelsel volgens conclusie 1, met het kenmerk,  
dat zowel het lokale als het communicatie-microprocessor  
15 gedeelte elk een eigen microprocessor, een eigen  
werkgeheugen en een eigen programma-geheugen bevatten.

3. Stelsel volgens C1 of 2, met het kenmerk, dat  
het werkgeheugen van het lokale microprocessor gedeelte  
direct toegankelijk is voor de microprocessor van het  
20 communicatie-gedeelte.

4. Stelsel volgens C1, 2 of 3, met het kenmerk,  
dat de volgorde waarin de eenheden berichten kunnen  
verzenden wordt bepaald aan de hand van een in de eenheden  
opgeslagen, veranderbare tabel.

25 5. Stelsel volgens een der voorafgaande  
conclusies, met het kenmerk, dat informatie tussen het  
lokale en het communicatie-microprocessor gedeelte  
uitgewisseld wordt met behulp van hand-shake technieken.

6. Stelsel volgens een der voorafgaande  
30 conclusies, met het kenmerk, dat de systeembus van het  
lokale microprocessor gedeelte en de systeembus van het  
communicatie-microprocessor gedeelte aan elkaar gekoppeld  
kunnen worden met behulp van een buffer.

7. Eenheid kennelijk uitgevoerd om te worden  
35 toegepast in een stelsel volgens een der voorafgaande  
conclusies.

Eindhoven, 13 december 1984

8500005

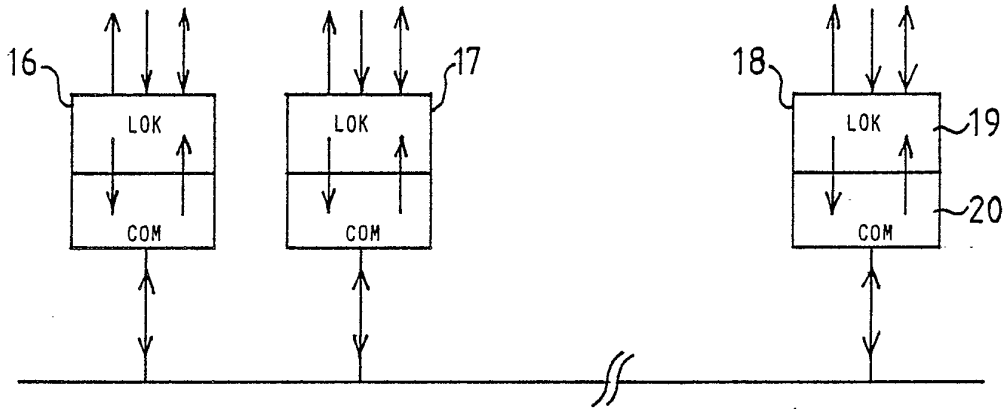


FIG. 1

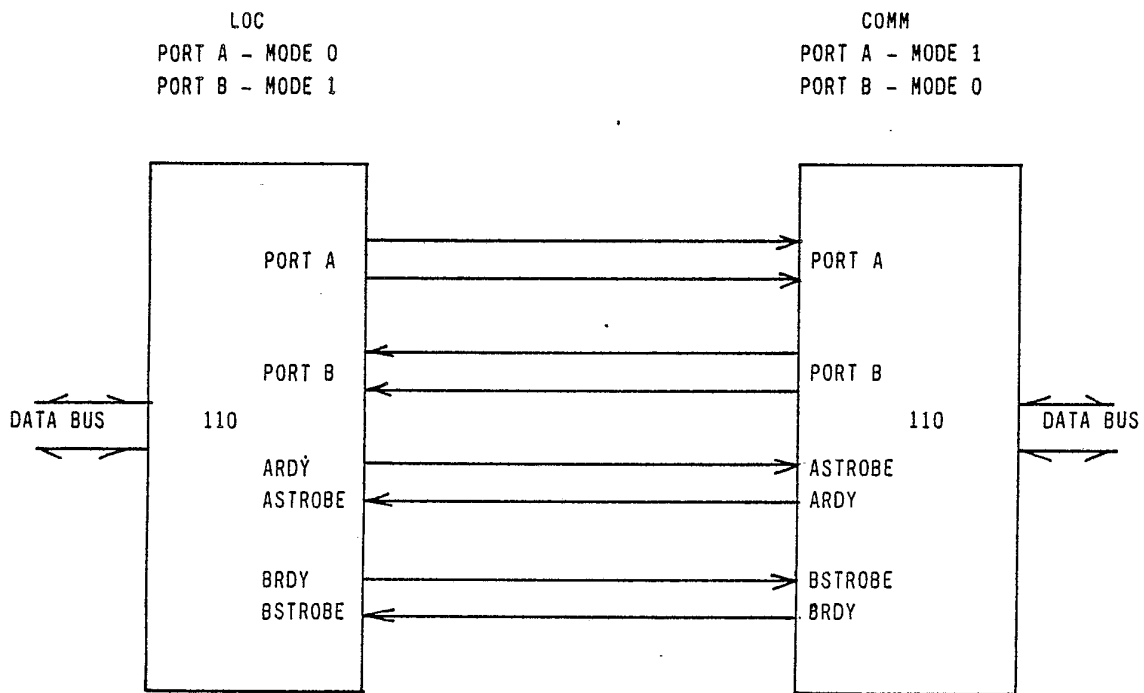


FIG. 6

8500005

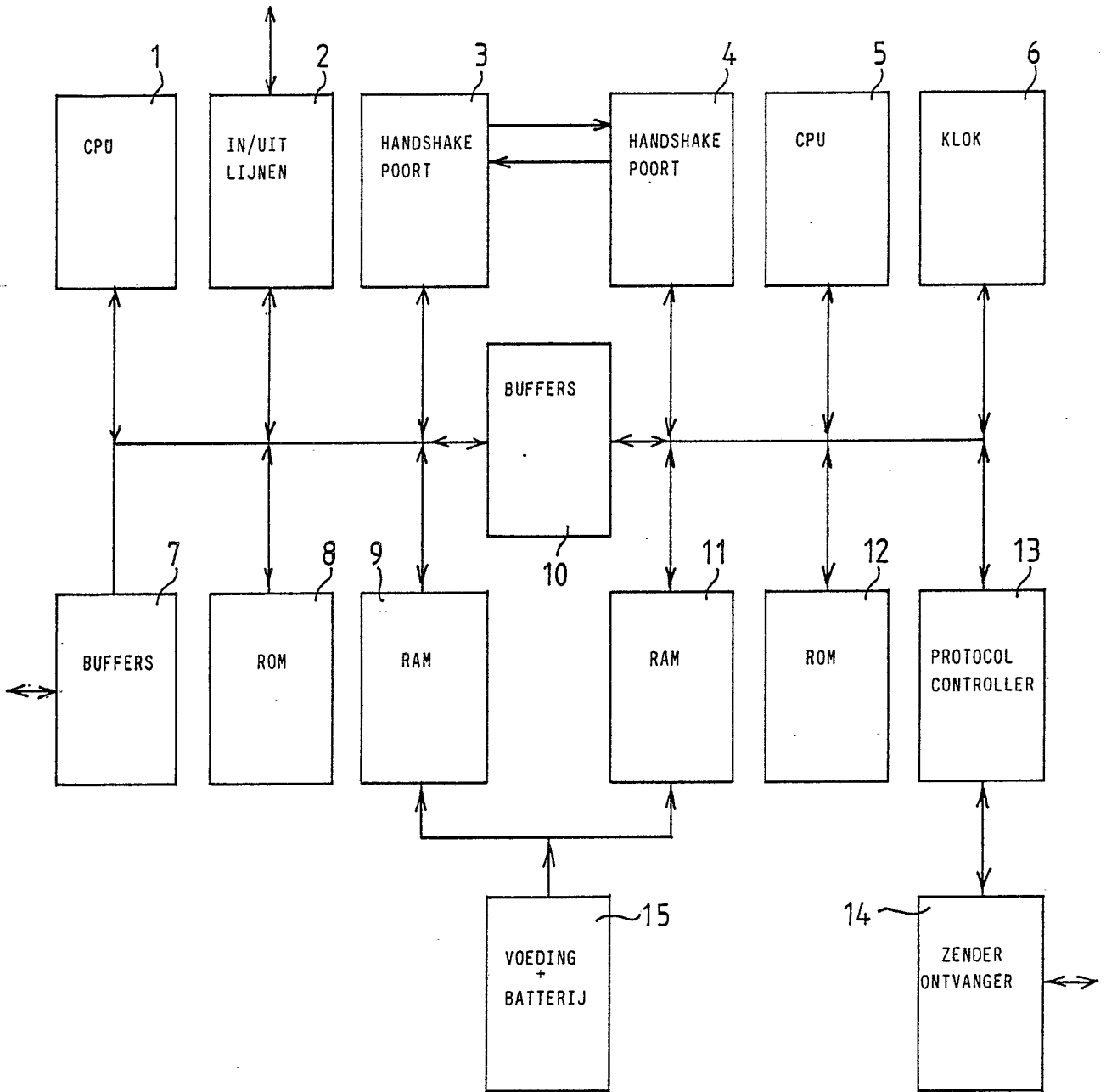


FIG. 2

8500005

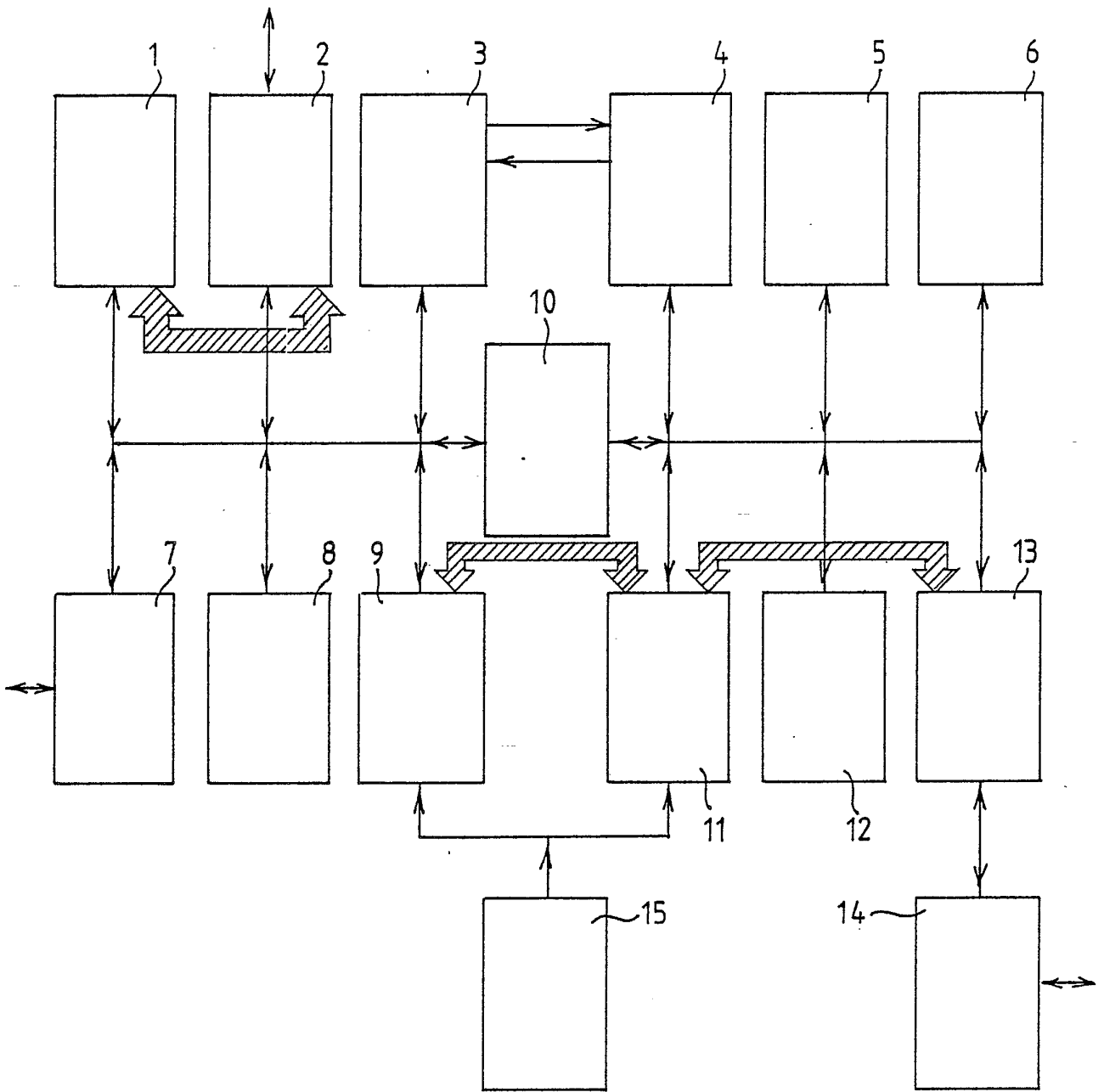


FIG. 3

8500005

BEGIN VLG	BESTEMMINGS ADRES	BRON ADRES	COMMANDO VELD	NR VELD	DATA VELD	FRAME CHECK SEQUENCE	EIND VLG
--------------	----------------------	---------------	------------------	------------	--------------	-------------------------	-------------

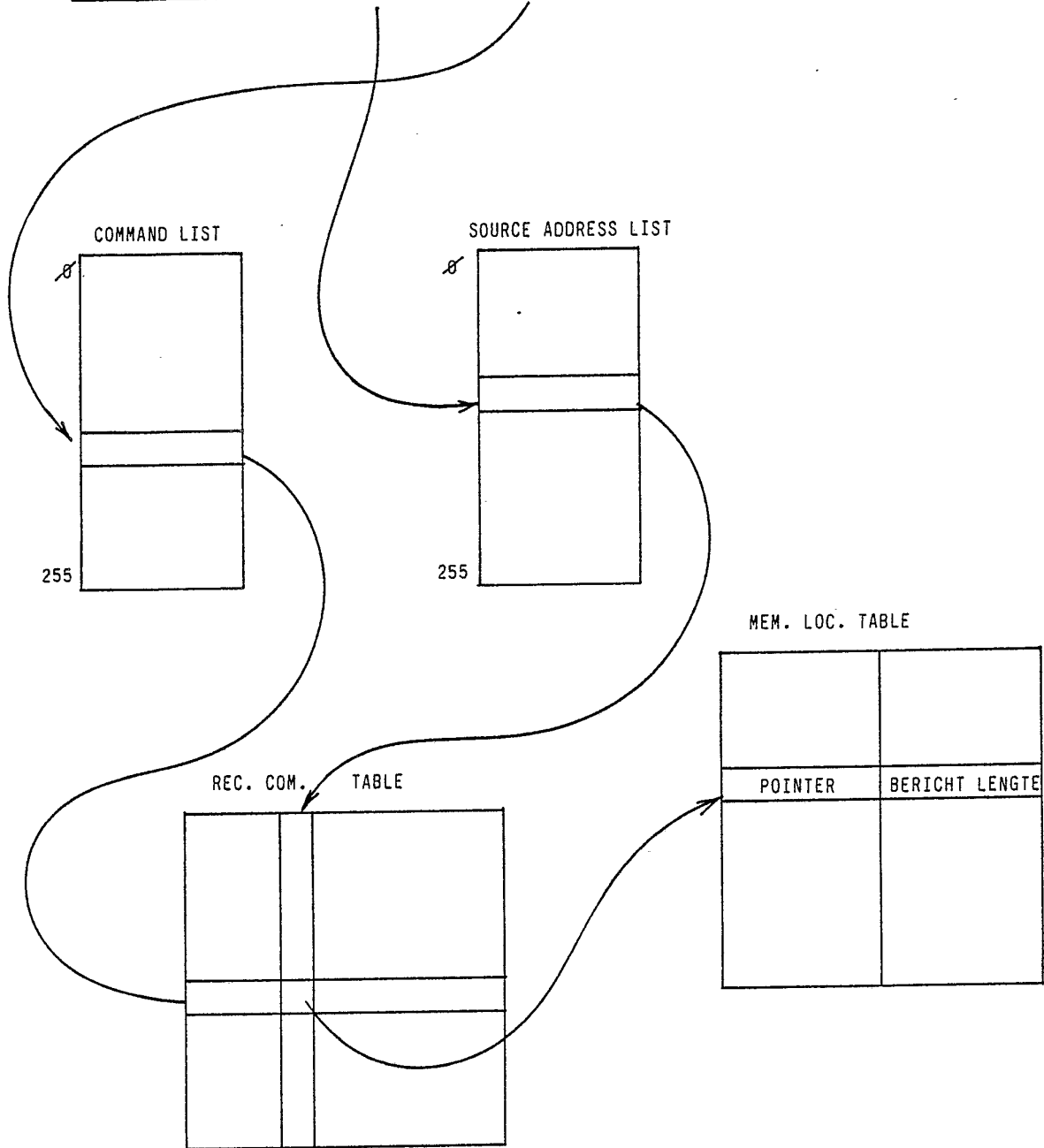
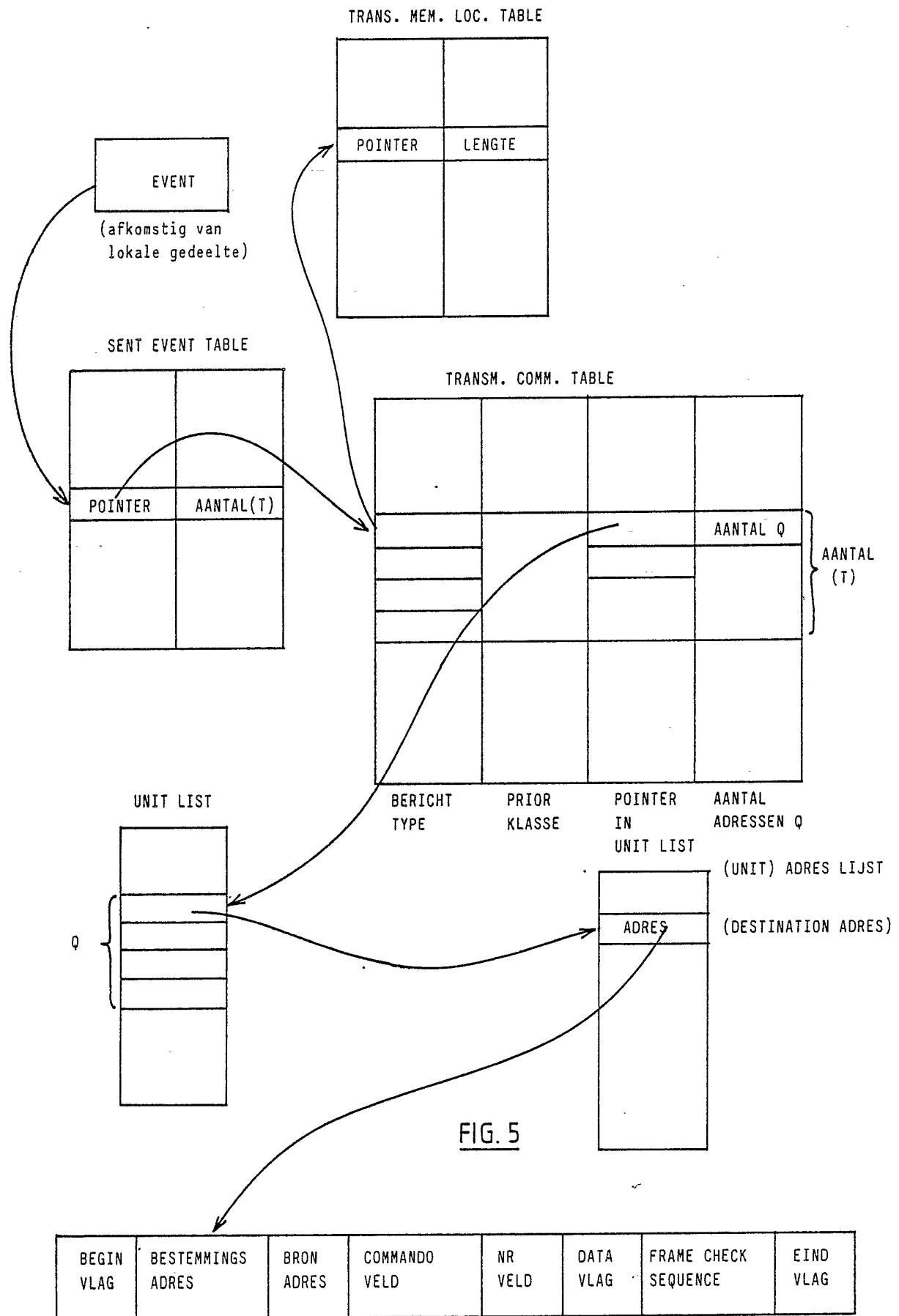


FIG. 4

8500005



8500005