(19) Bureau voor de Industriële Eigendom Nederland

(11) **1004826**

(12) C **OCTROOI**[20]

(21) Aanvrage om octrooi: **1004826**

(22) Ingediend: **18.12.1996**

(51) Int.Cl.[7]
G06F17/50

(30) Voorrang:
22.12.1995 GB 9526457

(41) Ingeschreven:
10.12.1997 I.E. 1998/02

(47) Dagtekening:
06.11.2000

(45) Uitgegeven:
02.01.2001 I.E. 2001/01

(73) Octrooihouder(s):
Nokia Mobile Phones Limited te Salo, Finland (FI).

(72) Uitvinder(s):
Rebecca Gabzdyl te Frimley (GB)
Brian McGovern te Camberley (GB)

(74) Gemachtigde:
Ir. P. Eveleens Maarse c.s. te 2517 GK Den Haag.

(54) **Programmageheugenstelsel voor processoren.**

(57) Werkwijze voor het ontwerpen van een processor. De werkwijze maakt gebruik van de volledige flexibiliteit van een oorspronkelijke instructieverzameling bij het schrijven van programma's voor de werking van de processor, waarbij de deelverzameling van de bij het schrijven van een programma gebruikte instructiewoorden vervolgens worden gebruikt bij het definiëren van de instructiedecodeerschakeling van de processor.

NL C 1004826

De inhoud van dit octrooi wijkt af van de oorspronkelijk ingediende beschrijving met conclusie(s) en eventuele tekening(en). De oorspronkelijk ingediende stukken kunnen bij het Bureau voor de Industriële Eigendom worden ingezien.

PROGRAMMAGEHEUGENSTELSEL VOOR PROCESSOREN

De uitvinding heeft betrekking op instructieverzamelingen voor processoren, en zij heeft meer in het bijzonder betrekking op een werkwijze voor het aan een processor toevoeren van een instructieverzameling.

Processoren voor algemene doeleinden voeren funkties uit in responsie op een verzameling instructies die tezamen een programma vormen. Dit is in een programmageheugen opgeslagen. De geheugencapaciteit, die vereist is voor het opslaan van het programma, is gerelateerd aan de lengte van het instructiewoord en het aantal instructiestappen in het programma.

De instructies in het programmageheugen worden gebruikt voor het opwekken van parallelle besturingssignalen in een decodeereenheid die de door de processor uitgevoerde handelingen bestuurt. De complexiteit van de logica, die vereist is voor het vertalen van de instructiewoorden in besturing, heeft betrekking op de lengte van het in de decodeereenheid ingevoerde instructiewoord.

Voor het optimaliseren van de processorprestaties in snelheid en energieverbruik is het gewenst zoveel mogelijk instructies te verschaffen aan een programmatuurontwerper, waardoor de ontwerper toegang krijgt tot de gehele funktionaliteit van de processor.

Wanneer een volledig flexibele instructiewoordverzameling, die afzonderlijk elke door de processor uit te voeren funktie specificeert, wordt gebruikt, heeft de ontwerper de volledige vrijheid een optimale implementatie van het algoritme te ontwerpen. Een efficiënte instructieverzameling, die een volledige toegang heeft tot de funktionaliteit van een processor die voldoet aan de toekomstige prestatie-eisen van de telecommunicatie-industrie, is veel groter dan de 16 bits die de huidige norm vormen. Voor de doeleinden van dit dokument wordt aangenomen dat een instructiewoord van 90 bits geschikt

is. Zelfs grotere instructieverzamelingen kunnen echter op enige punten gewenst zijn, hetgeen resulteert in zelfs langere instructiewoorden. Met instructiewoorden van 90 bits kan het programmageheugen 90 bits breed zijn, en een lengte hebben van 16 bits om plaats te bieden aan het programma voor het implementeren van een gebruikelijk processoralgoritme voor een radiotelefonietoepassing.

Dit heeft het nadeel voor goedkope toepassingen dat een groot geheugen noodzakelijk is voor het opslaan van het programma, aangezien de instructiewoorden 90 bits lang zullen zijn. De tussenvoegschakeling tussen het programmageheugen en de instructiedecodeerschakeling moet extra onderlinge verbindingen hebben om de instructiewoorden van 90 bits te kunnen laten ontvangen door een decodeereenheid. Dit vergroot opnieuw de afmetingen en het energieverbruik van de processor. Ten slotte neemt de logica, die noodzakelijk is voor het implementeren van de decodeereenheid, grofweg direkt proportioneel toe met het aantal ingangspennen. De decodeereenheid is dan ook meer complex en aldus kostbaarder.

Bij door batterijen gevoede consumentenprodukten, waar de kosten en het energieverbruik sterke overwegingen hebben voor de ontwerper, moet een compromis worden gemaakt tussen de flexibiliteit van de instructieverzameling en de resulterende vermogenseffectiviteit en instructiewoordlengte en toegenomen kosten.

Een conventionele benadering voor het oplossen van dit conflict is het instellen van de instructiewoordlengte door het vooraf bepalen van een beperkt aantal instructies, waaruit de ontwerper kan kiezen. De keuze van de beperkte instructieverzameling is van uiterst groot belang en zal gebruikerlijkerwijs de instructies ontvangen die het meest nuttig worden geacht voor een gebied van funkties dat de processor zou kunnen uitvoeren.

Deze benadering heeft het nadeel dat, wanneer de beperkte instructieverzameling geen specifieke instructie of de samenstellende componenten van deze instructie bevat, de processor niet in staat zal zijn op deze wijze

te funktioneren, alhoewel de processor de apparatuurmogelijkheden heeft tot het funktioneren in overeenstemming met deze instructie. Dit kan leiden tot compromissen in het ontwerp die de effektiviteit van de inrichting, die door de processor wordt aangedreven, kan beperken en die het moeilijk kan maken dat de door de processor bestuurde inrichting de vereiste prestaties behaalt.

Wanneer de instructieverzameling geen specifieke instructie omvat, kan het desondanks mogelijk zijn dezelfde instructie te vormen door het combineren van een aantal andere instructies. Dit zal onvermijdelijk resulteren in het feit dat meer vermogen wordt verbruikt bij het leveren van de gewenste prestatie dan in de situatie, waarin de gewenste instructie zou zijn gevormd als deel van de beperkte instructieverzameling.

In overeenstemming met de onderhavige uitvinding wordt een werkwijze verschaft voor het produceren van een processor die werkt in responsie op besturingssignalen die zijn opgewekt door uit instructiewoorden samengestelde programma's, waarbij de processor in staat is een verzameling funkties uit te voeren en dient voor het uitvoeren van elk van de verzameling funkties in responsie op een desbetreffend besturingswoord, waarbij elk besturingswoord gecorrelleerd is met een desbetreffend instructiewoord van een oorspronkelijke instructieverzameling, waarbij de werkwijze omvat: het schrijven van een programma dat dient voor het besturen van de processor door gebruikmaking van enigerlei instructies die gekozen zijn uit de oorspronkelijke instructieverzameling, het definiëren van een deelverzameling van instructiewoorden uit de oorspronkelijke instructieverzameling omvattende de de in het programma gebruikte instructies, het hernoemen van de desbetreffende instructies van de deelverzameling, en het verschaffen van opwekmiddelen voor het opwekken van de desbetreffende besturingswoorden die gecorrelleerd zijn met de instructiewoorden van de deelverzameling in responsie op de overeenkomstige hernoemde instructiewoorden.

Door het produceren van de beperkte instructieverzameling, die de processor kan gebruiken, met de van tevoren aanwezige kennis van het programma waarmee de processor zal werken, is er geen behoefte een compromis te sluiten tussen de instructiewoordlengte en de prestaties. De prestaties zijn dezelfde als die mogelijk zijn bij gebruik van de oorspronkelijke instructieverzameling bij de keuzen van woorden voor het programma dat uit deze verzameling is samengesteld. De woordlengte kan echter zo kort mogelijk zijn voor het aantal afzonderlijke woorden in het programma waarmee de processor zal werken.

De beperkte instructiewoordlengte maakt het mogelijk dat de instructiedecodeerschakeling eenvoudiger kan zijn, waardoor het aantal componenten, de kosten en het energieverbruik kan worden beperkt.

Het programmageheugen dat noodzakelijk is voor het opslaan van het werkalgoritme wordt tevens gereduceerd in termen van zowel de breedte van het opnieuw genoemde instructiewoord als de lengte daarvan als resultaat van de meer beknopte uitdrukking van de funktionaliteit van de processor.

Wanneer de instructiedecodeerschakeling eenmaal ontworpen is, is er geen mogelijkheid voor het aan de processor toevoeren van extra instructies. Wanneer de een instructie vormende codewoorden of de bouwstenen van een dergelijke instructie niet bestaan, kan een funktie niet worden uitgevoerd. Om de ontwerper enige flexibiliteit te geven wanneer de processor eenmaal is ontwikkeld, kan de onderverzameling van instructiewoorden extra instructiewoorden omvatten die overeenkomen met potentieel nuttige besturingswoorden. Deze extra instructies kunnen instructies omvatten die nuttig zijn bij het corrigeren van gebruikelijke fouten in het programma, of zij kunnen enige basisinstructies omvatten om andere funkties te kunnen ondersteunen.

Wanneer het aantal afzonderlijke instructiewoorden in het programma is vastgesteld, kan de kortste woord-

lengte om deze instructies afzonderlijk aan te duiden, worden ingesteld. Het is waarschijnlijk dat er enige extra woorden zullen zijn, wanneer de onderverzameling van instructiewoorden zijn hernoemd door gebruikmaking van de kortst mogelijke woordlengte. Het aantal woorden dat niet noodzakelijk is voor het eenduidig identificeren van de onderverzameling van instructiewoorden kan voldoende groot zijn, opdat deze verzameling kan worden gebruikt voor het ondersteunen van extra instructies. Deze redundantie zou echter kunnen worden gebruikt voor het vereenvoudigen van de logica van de middelen voor het opwekken van de desbetreffende besturingswoorden. Een instructiewoord met een langere naam dan die minimaal noodzakelijk is voor het eenduidig identificeren van het desbetreffende instructiewoord van de onderverzameling kan worden gebruikt voor het inbouwen van enige redundantie om de complexiteit van de logica te kunnen reduceren.

Werkwijzen volgens deze uitvinding hebben het voordeel dat de programmeur optimale prestaties kan bereiken, aangezien hij gebruik kan maken van een volledig flexibele instructieverzameling. In theorie is er geen grens aan het aantal instructies dat verschaft kan worden in de eerste instructieverzameling buiten de funkties die de processor kan uitvoeren. De beperkende factor voor de breedte van het geheugen van het programmageheugen en de afmetingen van de instructiedecodeerschakeling is het aantal unieke instructies dat gebruikt wordt bij het implementeren van het algoritme.

Zowel de lengte als de breedte van het programmageheugen, als de afmetingen van de instructiedecodeerschakeling worden geminimaliseerd, hetgeen leidt tot voordelen met betrekking tot kosten en vermogen. Het proces kan worden geautomatiseerd, wanneer het programma is vastgesteld, zodat het kan worden geïntegreerd in bepaalde ontwerpprocessen en gemakkelijk kan worden herhaald voor een reeks verschillende programma's, hetgeen zal leiden tot een reeks processoren met geoptimaliseerde instructieverzamelingen, wanneer dit gewenst is.

Het beperkte geheugen en de instructiedecodeerschakeling met beperkte afmetingen maken het mogelijk dat het energieverbruik van de inrichting voor een bepaalde prestatie kan worden gereduceerd.

Voor programma's, die waarschijnlijk een groot aantal gemeenschappelijke instructies zullen gebruiken, zou een enkele processor kunnen worden ontworpen voor het laten werken van een aantal verschillende programma's. De onderverzameling van de instructiewoorden zou dan kunnen worden vastgesteld door gebruikmaking van verschillende programma's die kunnen worden gebruikt voor verscheidene toepassingen. De woordlengte die noodzakelijk is voor het ondersteunen van de desbetreffende funkties zou moeten worden beschouwd, en deze zou moeten worden afgewogen tegen voordelen in gemeenschappelijkheid. De voordelen van de onderhavige uitvinding zouden worden gehandhaafd naarmate de onderverzameling van instructies, die worden gebruikt voor het besturen van de processor, zal worden vastgesteld met volledige voorkennis van de programma's die de processor zou moeten kunnen uitvoeren.

Nu zal bij wijze van voorbeeld een werkwijze in overeenstemming met de uitvinding worden beschreven aan de hand van de bijgaande tekeningen, waarin:

figuur 1 een kernblokschema van een processor is;

figuur 2 een stroomschema van de programmatuurontwikkeling is; en

figuur 3 een stroomschema is van het ROM-compactieproces van een uitvoeringsvorm van de uitvinding.

Een van de gebruiken van de werkwijze volgens de onderhavige uitvinding is bij het ontwerpen van een digitale signaalprocessor (Digital Signal Processor DSP). De uitvinding zal worden beschreven met betrekking tot het ontwerp van een DSP voor de basisbandalgoritmen van een radiotelefoon.

Figuur 1 toont een overzicht van de architectuur van een DSP. Een door DSP-technici geschreven programma is opgeslagen in een programmageheugen PROM 10. Door een oppak- en decodeereenheid (Fetch and Decode (FDU)) 12

wordt op een programma-adresbus (PAB) 11 van 16 bits gebruikelijkerwijs een programma-adres gespecificeerd.

De instructie op die plaats in het programmageheugen wordt via de programmagegevensbus (Program Data Bus (PDB)) 14 geladen tot in de instructiedecodeerschakeling 13 van de FDU 12. Bij een volledig flexibele, dat wil zeggen één die een volledige instructieverzameling ondersteunt op de conventionele wijze ontworpen DSP-schakeling, kan het nodig zijn dat de PDB 14 bijvoorbeeld 90 bits breed is. In de instructiedecodeerschakeling 13 geladen instructies worden gedecodeerd voor het opwekken van besturingswoorden, dat wil zeggen signalen die noodzakelijk zijn voor het besturen van de gegevensbewerkingseenheid (DPU) 15, de gegevensgeheugenadresopwekkingseenheid (Data Memory Address Generation Unit (MAGU)) 16 die de inhoud 17 van het gegevensgeheugen, de FDU of busbenaderingen adresseert om gegevens voor het programma te leveren.

Een DSP volgens de stand van de techniek en een DSP die wordt opgewekt in overeenstemming met de onderhavige uitvinding zullen niet op een verschillende wijze werken. Een DSP, die is opgewekt in overeenstemming met de onderhavige uitvinding, zal echter een kleine programmageheugen 10 hebben en PDB 14 met een beperkte breedte hebben en een kleinere instructiedecodeerschakeling 13 binnen de FDU 12 hebben dan een conventionele DSP, die dezelfde funktionaliteit levert.

Eén werkwijze in overeenstemming met de onderhavige uitvinding die een digitale signaalprocessor verschaft voor een radiotelefoon met de bovenstaande voordelen zal hieronder worden beschreven aan de hand van de figuren 2 en 3. Evenals bij werkwijzen volgens de stand van de techniek moeten de algoritmen van de basisband worden gecodeerd, dat wil zeggen dat het programma of de programma's moeten worden geschreven. Terwijl echter in werkwijzen volgens de stand van de techniek reeds bij deze trap een beslissing moet zijn genomen met betrekking tot de gevolgen tussen de aan de programmeur beschikbare

instructieverzameling en de acceptabele instructiewoordlengte zal de ontwerper in overeenstemming met de onderhavige uitvinding de beschikking hebben over een volledige instructieverzameling.

In figuur 2 is een ontwerpwerkwijze in overeenstemming met de onderhavige uitvinding geïllustreerd. Als aanvangsstap is het bronbestand beschreven. Het bronbestand is een tekstbestand dat door de programmeur tot in de computer wordt gevoerd. Het tekstbestand wordt gevormd door "assembly"-taalinstructies en makro-"instructies" met een hoog niveau die gevormd worden door in "assembly"-taal geschreven, vooraf gedefinieerde funkties. De voorprocessor 21 vertaalt automatisch enige makro-"instructies" tot in de "assembler"-bron 22 die de "assembler"-taalinstructies vormen. De "assembler" 23 vertaalt vervolgens het "assembly"-taalbronbestand dat door de programmeur wordt begrepen tot doelbestanden 24 die door de processor worden beschreven. Bij een grote DSP-toepassing zal het programma waarschijnlijk worden verdeeld in verscheidene bronbestanden. Een verbindingselement 25 neemt alle overeenkomstige doelbestanden en verbindt deze met elkaar voor het vormen van één uitvoerbaar bestand 26.

Wanneer de algoritmen voor het uitvoerbare bestand zijn gecodeerd, zullen zij in een simulator 26 worden gesimuleerd, en hun juistheid worden geverifieerd. De code wordt dan bevroren. In één uitvoeringsvorm van de onderhavige uitvinding worden na dit punt geen veranderingen uitgevoerd met betrekking tot de gecodeerde algoritmen, aangezien de voor gebruik beschikbare instructieverzameling nu gedefinieerd is. Later zal echter een voorbeeld worden besproken, hoe desondanks veranderingen aan het algoritme kunnen worden uitgevoerd na deze trap, alhoewel in het ideale geval geen verdere veranderingen noodzakelijk zouden moeten zijn.

Wanneer het algoritme is geverifieerd, zet de doelformaatomzettingsschakeling 28 het uitvoerbare bestand 26 om in een formaat dat kan worden gebruikt voor het opwek-

ken van het programmageheugen. Volgende stappen van het proces kunnen worden gevolgd door gebruikmaking van een stuk gereedschap dat voor de doeleinden van deze beschrijving wordt aangeduid als een ROM-compactor 29 om slechts die instructies door de DSP te laten ondersteunen die in de algoritmen worden gebruikt.

Het uitvoerbare bestand 26 dat gevormd wordt door het uitgangssignaal van het verbindingselement 25 wordt omgezet naar een formaat dat de ROM-compactor 30 kan gebruiken via de objectformaatomzetter 28.

Voor een programma of een algoritme van gebruikelijke lengte voor een radiotelefonietoepassing zullen ongeveer 25000 van de beschikbare 2^{90} instructieverzamelingen worden gebruikt. Voor het eenduidig identificeren van elk van de 2^{90} instructies is een woordlengte van 90 bits vereist. De werkelijke 25000 afzonderlijke woorden zouden kunnen worden aangeduid door gebruikmaking van een woord van 16 bits. In het algemeen kunnen 2^N woorden op eenduidige wijze worden aangeduid door gebruikmaking van een woord van N bits. Er is geen correlatie tussen het aantal instructies dat de volledig flexibele instructieverzameling levert en het aantal instructies dat wordt gebruikt bij het coderen van het algoritme. Wanneer wij een algoritme voorstellen dat wordt gecodeerd door gebruikmaking van 25 codelijnen, zullen er maximaal 25 afzonderlijke instructies worden gebruikt. Sommige instructies zullen bijna onvermijdelijk meer dan eenmaal worden gebruikt. Elke duplicatie zal resulteren in een reductie in het aantal unieke instructies dat gebruikt wordt voor het schrijven van de code.

Het programma wordt geanalyseerd en elke afzonderlijke instructiewoord wordt in het met 31 aangeduide programma gebruikt. Niet in het programma gebruikte instructies worden genegeerd. Het resultaat is een beperkte instructieverzameling 32 die slechts instructies bevat die werkelijk in het programma worden gebruikt. Wanneer dit noodzakelijk is, kan een onderverzameling van instructies, die nuttig kunnen zijn voor het verwijderen

van fouten uit programma's, worden toegevoegd aan de beperkte instructieverzameling 33, hetgeen leidt tot een beperkte instructieverzameling 34.

Tevens kan het gewenst zijn verdere deelverzamelingen van de oorspronkelijke instructieverzameling toe te voegen die kunnen worden gebruikt, wanneer veranderingen op het programma worden uitgevoerd. Natuurlijk zal telkens, wanneer een gewenste onderverzameling van instructies worden beschouwd, de lijst van instructies in het programma worden gecontroleerd om duplicatie van instructies te vermijden. Tevens moet de ontwerper beslissen of de voordelen de nadelen te boven gaan, wanneer het opnemen van extra instructies vereist dat een extra regel moet worden toegevoegd aan de PDB, en dat als gevolg daarvan een extra pen moet worden toegevoegd aan de instructiedecodeerschakeling of dat enige redundantie wordt verwijderd die nuttig is bij het vereenvoudigen van het ontwerp van de instructiedecodeerschakeling. Door echter uit te gaan van de verzameling van instructies, die reeds worden gebruikt bij het coderen van een algoritme, is de volledige vrijheid bij het ontwerpen van de volledig flexibele verzameling gebruikt, en is er geen afweging tussen prestaties en woordlengte. Extra instructies die zijn toegevoegd aan de mogelijkheden van de processor kunnen worden gebruikt voor enige mate van flexibiliteit, nadat het algoritme is gecodeerd. Het is waarschijnlijk dat enige veranderingen worden beperkt tot correctie van fouten en dat, met dit in het achterhoofd, extra instructies kunnen worden gekozen.

Vervolgens bepaalt de ROM-compactor de bitbreedte N, waarvoor 35 is vereist. Dit wordt vastgesteld door het onderzoeken van het totale aantal instructies in de gereduceerde of beperkte instructieverzameling. De bitbreedte is de breedte van de woorden die zullen worden gebruikt voor het hernoemen van de gereduceerde of beperkte instructieverzamelingen die worden gebruikt voor het coderen van de instructies.

Vervolgens worden de instructies gesorteerd voor het bij elkaar groeperen van soortgelijke instructies. Elk van de instructies wordt gecorrelleerd met een besturingswoord dat de signalen levert voor het besturen van de funktie van de DSP. Wanneer de verzameling instructies, die in het programma wordt gebruikt, is geïdentificeerd, kunnen deze worden voorzien van een desbetreffend voorvoegsel. Dit maakt het mogelijk dat de instructies gemakkelijk worden gesorteerd, zodat soortgelijke instructies bij elkaar worden geplaatst voor het minimaliseren van de resulterende logica in de instructiedecodeerschakeling.

Voor het verder minimaliseren van de logica en de decodeerschakeling wordt enige redundantie, resulterend uit het feit dat het aantal instructies kleiner is dan 2^N, gebruikt bij dit sorteerproces. De gekozen woordbreedte kan enige extra ingebouwde redundantie behoeven om de logica van de instructiedecodeerschakeling 13 eenvoudig te houden. De gekozen bitbreedte kan dan ook langer zijn dat de minimaal noodzakelijke bitbreedte voor het op eenduidige wijze identificeren van de gereduceerde of de beperkte instructieverzameling.

Wanneer zij zijn gesorteerd, worden de instructies gecodeerd door gebruikmaking van opeenvolgende binaire getallen 36. Eenvoudig gesteld, kan het nieuwe instructiewoord worden geplaatst op de regel van de gesorteerde lijst van eenduidige instructies. Dit resulteert in een optimaal gecodeerde instructieverzameling 37. Het uitvoerbare bestand 26 wordt dan automatisch opnieuw geschreven 38 door het vervangen van de oude instructiewoorden door de desbetreffende nieuwe instructiewoorden voor het opwekken van een nieuwe programmacode 39 die kan worden opgeslagen in een programmageheugen 40 met een beperkte breedte. Het programmageheugen kan worden geadresseerd op conventionele wijze door gebruikmaking van een 16 bits adresbus. De programmagegevensbus zal 15 of 16 bits lang zijn in afhankelijkheid van de mate van duplicatie in het programma in plaats van 80 of 90 bits

lang te zijn, hetgeen noodzakelijk zou zijn, wanneer het programma zou zijn geschreven door gebruikmaking van de oorspronkelijke instructieverzameling.

Om het programma te kunnen implementeren is een instructiedecodeerschakeling noodzakelijk, die besturingswoorden levert die gecorrelleerd zijn aan de desbetreffende nieuwe instructiewoorden. De instructiedecodeerschakeling is een logische eenheid, die een uitgangssignaal met het gecorrelleerde besturingswoord afgeeft in responsie op de toevoer van het desbetreffende nieuwe instructiewoord.

De decodeerlogica wordt bij 41 opgewekt door het correlleren van elk nieuw instructiewoord met het besturingswoord dat bij het oude instructiewoord behoort. Aldus worden de ingangs- en uitgangssignalen van een instructiedecodeerschakeling gespecificeerd. Er zijn apparatuurbeschrijvingstalen beschikbaar voor het aanduiden van de ingangsaansluitingen en de uitgangsaansluitingen van een instructiedecodeerschakeling. Deze worden gebruikt in samenhang met een stuk gereedschap, zoals Synopsys, voor het bepalen van een geschikte logische rangschikking voor de instructiedecodeerschakeling. Dit stuk gereedschap wekt een instructiedecodeernetlijst 42 op die een voor de toepassing specifieke geïntegreerde schakelingsontwerpstroom 43 (Application Specific Integrated Circuit (ASIC)) levert voor het opwekken van de logica voor de instructiedecodeerschakeling.

Wanneer de rangschikking voor de instructiedecodeerschakeling is vastgesteld, dan deze worden vervaardigd tezamen met de andere schakelingen van de oppak- en decodeereenheid en het programmageheugen. De diverse blokken van de DSP zullen meestal geïntegreerd zijn op een ASIC.

Volgens een andere uitvoeringsvorm zou, in plaats van het ontwerpen van een processor die specifiek geschikt is voor een specifieke toepassing, een processor kunnen worden gebruikt die geschikt is voor het laten werken van twee of meer programma's met een hoge mate van

gemeenschappelijkheid; dat wil zeggen van programma's die een grote kern van soortgelijke instructies toepassen.

De onderhavige uitvinding omvat elke nieuwe maatregel, een combinatie van maatregelen, of een onderverzameling van maatregelen die hierin expliciet of door generalisatie zijn beschreven, onafhankelijk van het feit of deze al of niet betrekking heeft op de geclaimde uitvinding of deze voortvloeit uit één of alle van de aangeduide problemen.

CONCLUSIES

1. Werkwijze voor het produceren van een processor die werkt in responsie op besturingssignalen die zijn opgewekt door uit instructiewoorden samengestelde programma's, waarbij de processor in staat is een verzameling funkties uit te voeren, en dient voor het uitvoeren van elk van de verzameling funkties in responsie op een desbetreffend besturingswoord, waarbij elk besturingswoord gecorrelleerd is met een desbetreffend instructiewoord van een oorspronkelijke instructieverzameling, waarbij de werkwijze omvat: het schrijven van een programma dat dient voor het besturen van de processor door gebruikmaking van enigerlei instructies die gekozen zijn uit de oorspronkelijke instructieverzameling, het definiëren van een deelverzameling van instructiewoorden uit de oorspronkelijke instructieverzameling, omvattende de de in het programma gebruikte instructies, het hernoemen van de desbetreffende instructies van de deelverzameling, en het verschaffen van opwekmiddelen voor het opwekken van de desbetreffende besturingswoorden die gecorrelleerd zijn met de instructiewoorden van de deelverzameling in responsie op de overeenkomstige hernoemde instructiewoorden.

2. Werkwijze volgens conclusies 1, waarbij het uit de hernoemde instructies gevormde programma is opgeslagen in een programmageheugen.

3. Werkwijze volgens conclusie 1 of 2, waarbij de opwekmiddelen een instructiedecodeerschakeling omvatten.

4. Werkwijze volgens een van de voorafgaande conclusies, waarbij de funktionaliteit van de processor wordt bevestigd door gebruikmaking van een toepassing die gecodeerd is in de oorspronkelijke instructieverzameling.

5. Werkwijze volgens een van de voorafgaande conclusies, waarbij de deelverzameling van instructies instructies omvatten voor het uitvoeren van funkties die gewenst zijn voor het corrigeren van fouten.

6. Werkwijze volgens een van de voorafgaande conclusies, waarbij de hernoemde instructies een minimale lengte hebben, zodat elk instructiewoord van de deelverzameling eenduidig identificeerbaar is.

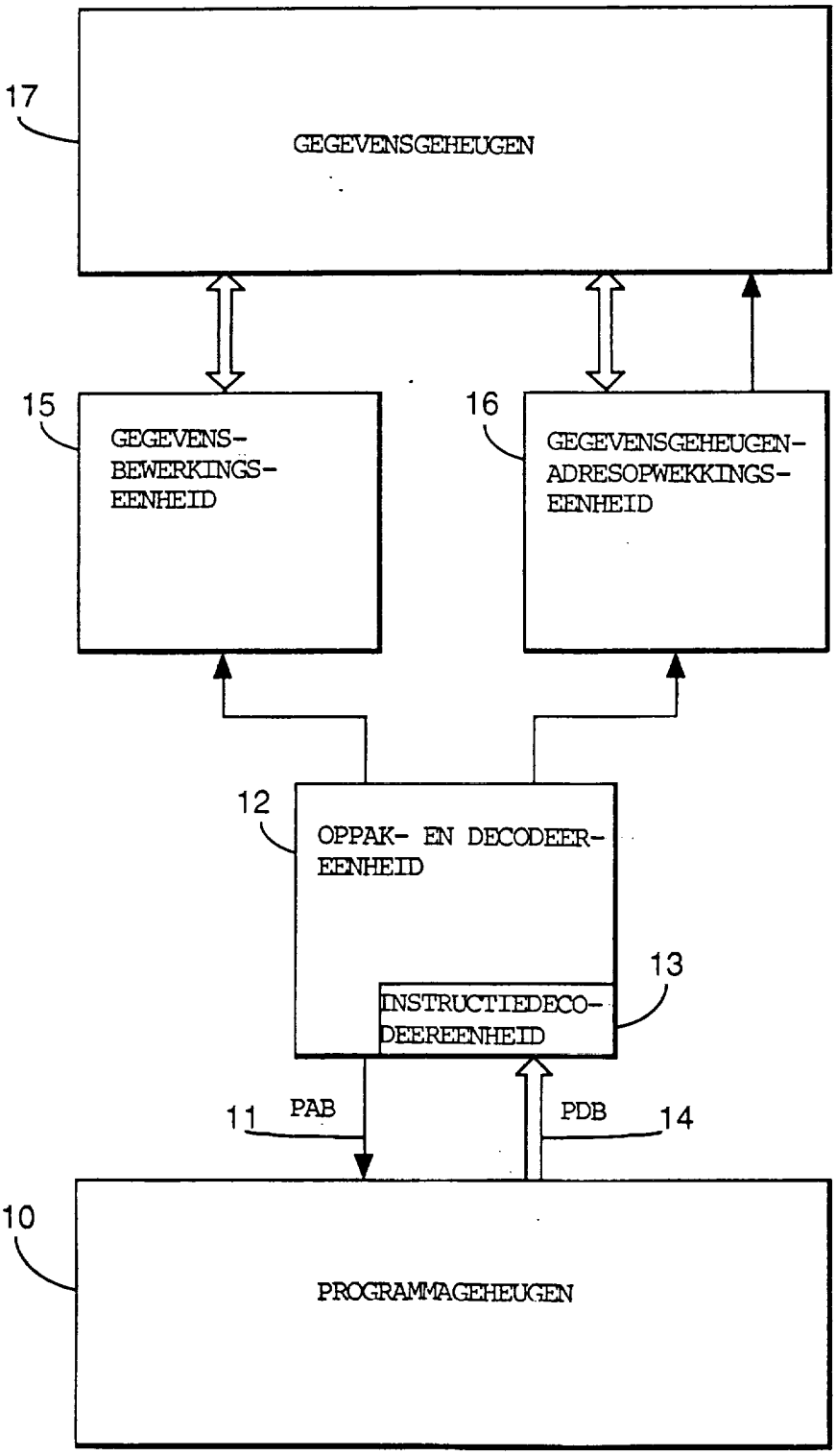
7. Werkwijze volgens een van de conclusies 1-5, waarbij de hernoemde instructies langer zijn dan de minimale lengte voor elk instructiewoord van de deelverzameling om eenduidig identificeerbaar te zijn.

8. Processor, welke geproduceerd is in overeenstemming met conclusie 1.

9. Processor volgens conclusie 8, met het kenmerk, dat de opwekmiddelen een instructie-decodeerschakeling omvatten.

10. Processor volgens conclusie 8 of 9, met het kenmerk, dat het uit de hernoemde instructies gevormde programma is opgeslagen in een programma-geheugen.

Fig.1.



1004826

Fig.2.

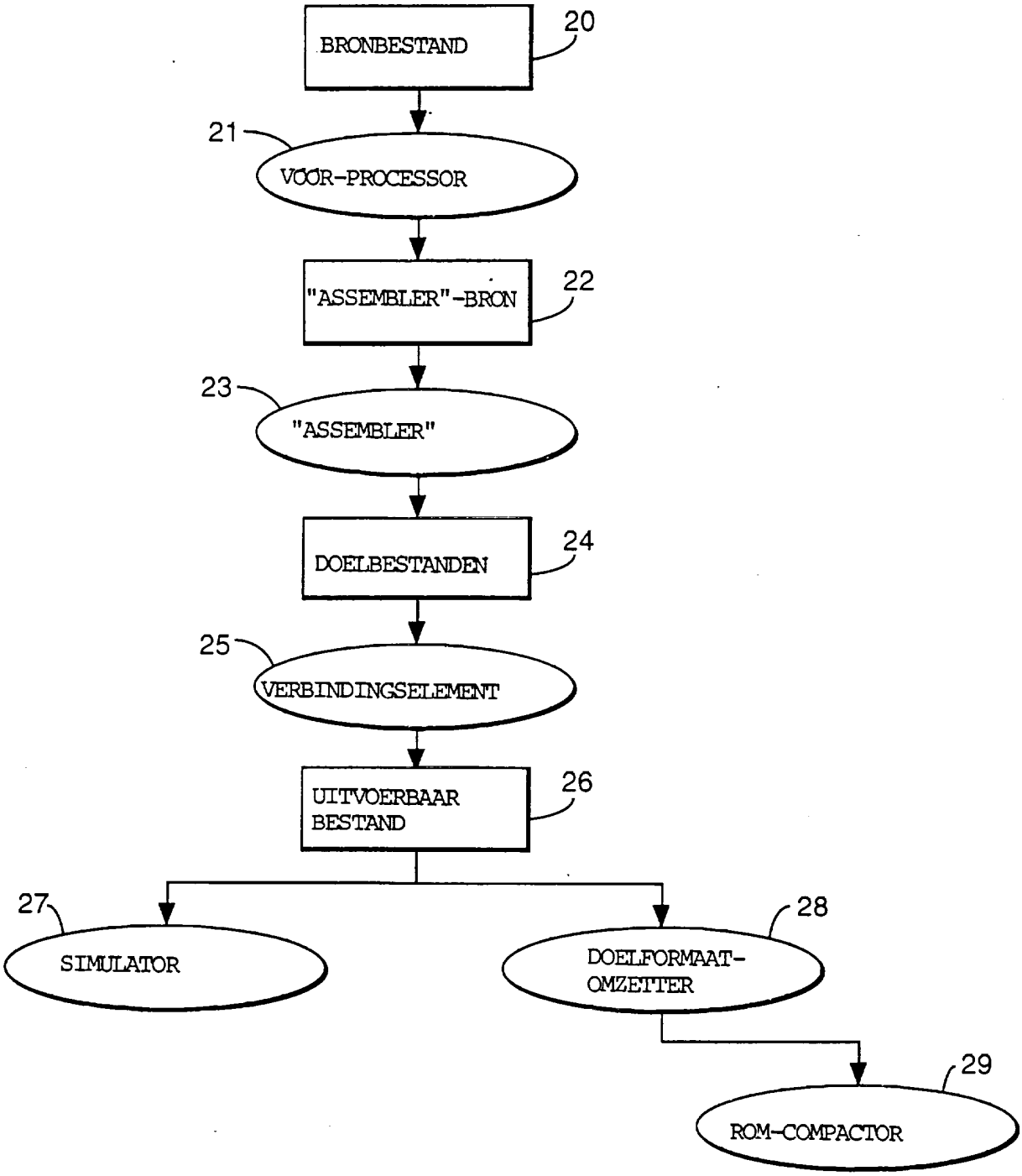
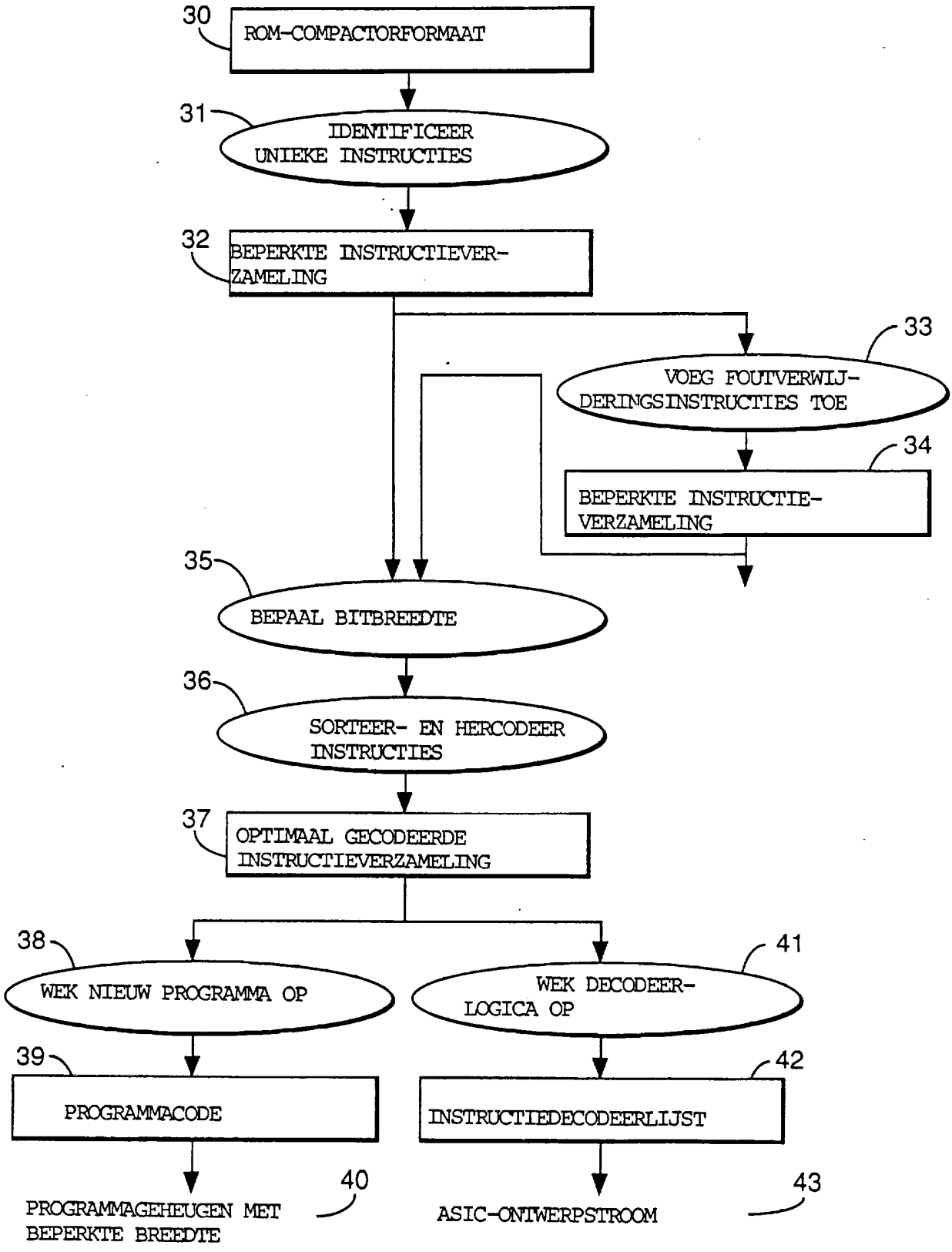

BRONBESTAND
20
21
VOOR-PROCESSOR
"ASSEMBLER"-BRON
22
23
"ASSEMBLER"
DOELBESTANDEN
24
25
VERBINDINGSELEMENT
UITVOERBAAR BESTAND
26
27
SIMULATOR
28
DOELFORMAAT-OMZETTER
29
ROM-COMPACTOR

Fig.3.


30
ROM-COMPACTORFORMAAT
31
IDENTIFICEER UNIEKE INSTRUCTIES
32
BEPERKTE INSTRUCTIEVER-ZAMELING
33
VOEG FOUTVERWIJ-DERINGSINSTRUCTIES TOE
34
BEPERKTE INSTRUCTIE-VERZAMELING
35
BEPAAL BITBREEDTE
36
SORTEER- EN HERCODEER INSTRUCTIES
37
OPTIMAAL GECODEERDE INSTRUCTIEVERZAMELING
38
WEK NIEUW PROGRAMMA OP
41
WEK DECODEER-LOGICA OP
39
PROGRAMMACODE
42
INSTRUCTIEDECODEERLIJST
40
PROGRAMMAGEHEUGEN MET BEPERKTE BREEDTE
43
ASIC-ONTWERPSTROOM

Bureau voor de Industriële Eigendom

RAPPORT BETREFFENDE HET ONDERZOEK NAAR DE STAND VAN DE TECHNIEK

Octrooiaanvrage Nr.:
NO 133724
NL 1004826

VAN BELANG ZIJNDE LITERATUUR

Categorie	Vermelding van literatuur met aanduiding voor zover nodig, van speciaal van belang zijnde passages	Van belang voor conclusie(s)Nr.:	Internationale classificatie
X	ALOMARY A Y ET AL: "AN ASIP INSTRUCTION SET OPTIMIZATION ALGORITHM WITH FUNCTIONAL MODULE SHARING CONTRAINT" IEICE TRANSACTIONS ON FUNDAMENTALS OF ELECTRONICS, COMMUNICATIONS AND COMPUTER SCIENCES,JP,INSTITUTE OF ELECTRONICS INFORMATION AND COMM. ENG. TOKYO, deel 76A, nr. 10, 1 Oktober 1993 (1993-10-01), bladzijden 1713-1719, XP000422017 ISSN: 0916-8508 * bladzijde 1713, linker kolom, regel 22 - bladzijde 1714, rechter kolom, regel 25 * * bladzijde 1716, linker kolom, regel 25 - bladzijde 1717, rechter kolom, regel 8 * ---	1,3-7	G06F17/50
X	JUN SATO ET AL: "AN INTEGRATED DESIGN ENVIRONMENT FOR APPLICATION SPECIFIC INTEGRATED PROCESSOR" PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON COMPUTER DESIGN - VLSI IN COMPUTERS AND PROCESSORS,US,LOS ALAMITOS, IEEE. COMP. SOC. PRESS, deel -, 14 Oktober 1991 (1991-10-14), bladzijden 414-417, XP000245924 ISBN: 0-8186-2270-9 * het gehele document * --- -/--	1,3-7	

Onderzochte gebieden van de techniek

G06F

Indien gewijzigde conclusies zijn ingediend, heeft dit rapport betrekking op de conclusies ingediend op :.....

Plaats van onderzoek	Datum waarop het onderzoek werd voltooid	Vooronderzoeker (EOB)
'S-GRAVENHAGE	4 Augustus 2000	Nicoli, F

CATEGORIE VAN DE VERMELDE LITERATUUR

X : op zichzelf van bijzonder belang
Y : van bijzonder belang in samenhang met andere documenten van dezelfde categorie
A : achtergrond van de stand van de techniek
O : verwijzend naar niet op schrift gestelde van de techniek
P : literatuur gepubliceerd tussen voorrangs- en indieningsdatum
T : niet tijdig gepubliceerde literatuur over theorie of principe ten grondslag liggend aan de uitvinding
E : andere octrooipublicatie maar gepubliceerd op of na indieningsdatum
D : in de aanvrage genoemd
L : om andere redenen vermelde literatuur
.....
& : lid van dezelfde octrooifamilie, corresponderende literatuur document

EOB FORM 02.93 (P0414)

3

Bureau voor de Industriële Eigendom

RAPPORT BETREFFENDE HET ONDERZOEK NAAR DE STAND VAN DE TECHNIEK

Octrooiaanvrage Nr.:
NO 133724
NL 1004826

VAN BELANG ZIJNDE LITERATUUR

Categorie	Vermelding van literatuur met aanduiding voor zover nodig, van speciaal van belang zijnde passages	Van belang voor conclusie(s)Nr.:	Internationale classificatie
X	ALOMARY A ET AL: "PEAS-I: A hardware/software co-design system for ASIPs" PROCEEDINGS EURO-DAC '93. EUROPEAN DESIGN AUTOMATION CONFERENCE WITH EURO-VHDL '93 (CAT. NO.93CH3352-2), PROCEEDINGS OF EURO-DAC 93 AND EURO-VHDL 93- EUROPEAN DESIGN AUTOMATION CONFERENCE, HAMBURG, GERMANY, 20-24 SEPT. 1993, bladzijden 2-7, XP002144259 1993, Los Alamitos, CA, USA, IEEE Comput. Soc. Press, USA ISBN: 0-8186-4350-1 * bladzijde 3, linker kolom, regel 28 - bladzijde 5, linker kolom, regel 29 * ---	1,3-7	
A	PAULIN P G ET AL: "DSP DESIGN TOOL REQUIREMENTS FOR EMBEDDED SYSTEMS: A TELECOMMUNICATIONS INDUSTRIAL PERSPECTIVE" JOURNAL OF VLSI SIGNAL PROCESSING,NL,KLUWER ACADEMIC PUBLISHERS, DORDRECHT, deel 9, nr. 1/02, 1995, bladzijden 23-46, XP000525885 ISSN: 0922-5773 * bladzijde 36, rechter kolom, alinea 4 - bladzijde 44, rechter kolom, alinea 5 * -----	1-7	

Onderzochte gebieden van de techniek

Indien gewijzigde conclusies zijn ingediend, heeft dit rapport betrekking op de conclusies ingediend op :.....

3

Plaats van onderzoek	Datum waarop het onderzoek werd voltooid	Vooronderzoeker (EOB)
'S-GRAVENHAGE	4 Augustus 2000	Nicoli, F

CATEGORIE VAN DE VERMELDE LITERATUUR

X : op zichzelf van bijzonder belang
Y : van bijzonder belang in samenhang met andere documenten van dezelfde categorie
A : achtergrond van de stand van de techniek
O : verwijzend naar niet op schrift gestelde van de techniek
P : literatuur gepubliceerd tussen voorrangs- en indieningsdatum
T : niet tijdig gepubliceerde literatuur over theorie of principe ten grondslag liggend aan de uitvinding
E : andere octrooipublicatie maar gepubliceerd op of na indieningsdatum
D : in de aanvrage genoemd
L : om andere redenen vermelde literatuur
& : lid van dezelfde octrooifamilie, corresponderende literatuur document

EOB FORM 02.83 (P0414)