

# PATENTOVÝ SPIS

(11) Číslo dokumentu:

## 293 613

(13) Druh dokumentu:

**B6**

(51) Int. Cl. : <sup>7</sup>

G 21 D 3/04

G 21 D 3/06

G 21 C 9/00

G 21 C 17/00

(19)  
ČESKÁ  
REPUBLIKA



ÚŘAD  
PRŮMYSLVÉHO  
VLASTNICTVÍ

(21) Číslo přihlášky: **1992-3773**

(22) Přihlášeno: **18.12.1992**

(30) Právo přednosti: **17.01.1992 US 1992/821734**

(40) Zveřejněno: **17.11.1993**

(Věstník č: 11/1993)

(47) Uděleno: **20.04.04**

(24) Oznámení o udělení ve Věstníku:

(Věstník č: 6/2004)

**16.06.2004**

(73) Majitel patentu:

WESTINGHOUSE ELECTRIC CORPORATION,  
Pittsburgh, PA, US

(72) Původce:

Lipner Melvin Herbert, Monroeville, PA, US

Pitcairn Francine S., Pittsburgh, PA, US

Mundy Roger Andrew, North Huntingdon, PA, US

(74) Zástupce:

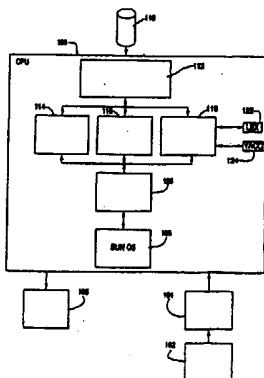
Čermák Karel JUDr. advokát, Národní 32, Praha 1,  
11000

(54) Název vynálezu:

**Způsob monitorování chodu zařízení pomocí CPU**

(57) Anotace:

Způsob monitorování chodu zařízení pomocí CPU (100), jehož podstata spočívá v tom, že zahrnuje ukládání parametrů, vztahujících se k provozním podmínkám zařízení, do relační databáze (112), generování rovnic v CPU (100), definujících způsob vyhodnocení údajů získaných z provozních senzorů zařízení na základě parametrů uložených v relační databázi (112) a monitorování a vyhodnocování uvedených rovnic při chodu zařízení s cílem stanovit, zda v zařízení došlo k poruše. Výhodně je zařízením, jehož chod se monitoruje uvedeným způsobem, jaderná elektrárna.



CZ 293613 B6

## Způsob monitorování chodu zařízení pomocí CPU

### Oblast techniky

5

Vynález se týká způsobu monitorování chodu zařízení pomocí CPU. Zkratku CPU je třeba v rámci vynálezu chápat tak, že zahrnuje počítač nebo pracovní stanici, která je schopna spouštět softwarové moduly. Takové počítače nebo pracovní stanice jsou odborníkům v daném oboru známé a jsou jimi výhodně pracovní stanice systému Sun Micro system Sun-3 a Sun-4. Specificky vyjádřeno, týká se vynález způsobu generování a provádění komplexních automatizovaných procesů (COMPRO, Complex COMputerized PROCedures), které monitorují chod zařízení jaderné elektrárny. Vynález se zejména zabývá problematikou generování rovnic reprezentujících podmínky nebo stavy zařízení a jejich vyhodnocování, čímž se zjišťuje, zda zařízení pracuje v mezích stanovených podmínkami, nebo zda došlo k havarijnímu stavu.

15

Systém COMPRO je automatizovaný a řídicí informační systém, který má operátorům zařízení pomáhat zejména při monitorování a řízení jednotlivých procesů chodu elektrárny, mezi které patří například havarijní stav, normální chod, abnormální chod a odezvy na poplach. Systém COMPRO zajišťuje, aby operátoři zařízení při řešení jednotlivých situací postupovali jednoduchým a logickým způsobem. Současně systém COMPRO poskytuje operátorům zařízení paralelní informace a ti pak mají přehled o celém procesu nebo o celkovém stavu zařízení. Systém COMPRO lze užívat pro všechny pracovní postupy v elektrárně. Většinu pracovních postupů v elektrárně lze s výhodou automatizovat, protože počítač může snadno monitorovat velké množství parametrů současně. Tím, že COMPRO poskytuje informace operátorům elektrárny paralelně, napomáhá tím k lepší celkové orientaci v systému.

Základní operační postupy pro práci v normálních nebo nouzových podmínkách jaderné elektrárny předem vyvinuli zkušení odborníci. Tyto postupy nemají pouze sloužit k vedení operátora zařízení správným směrem v jednotlivých akcích, ale mají také zjišťovat anomální stavy. Při zjištění anomálie mají navrhnout změny v normálním postupu a tím řešit tyto anomálie.

30

### Dosavadní stav techniky

35

COMPRO neboli interaktivní počítačem řízené monitorování provádění komplexních operačních procedur se již běžně používá a je popisováno v patentových přihláškách, například patenty US 4 803 039, 4 815 014 a 5 068 080. Tím byl vytvořen základní systém pro provádění komplexních operačních procedur. Další patenty se však pouze zabývaly zlepšováním a rozšiřováním základního systému a netýkaly se jeho elementární softwarové struktury.

40

Základní systém, tak, jak byl původně vytvořen, měl v sobě systémové parametry (ventily, pumpy, generátory) a podmínky pevně zakódované. Například ventily pro přívod napájecí vody smějí pracovat nad určitou průtokovou rychlostí, jako je 100 stop za vteřinu (asi 30 metrů za vteřinu). Tato podmínka byla pevně zakódována do systémového software, například jako "If valve-input > 100 then valve-condition = TRUE - Když průtoková rychlost > 100, tak stanovená podmínka průtoku ventilem = PRAVDA". Program byl potom zkompilován do spustitelného kódu a tak v něm byly jednoznačně určeny rozhodující podmínky. Když ale bylo třeba změnit jediný parametr programu, musely se provést změny ve zdrojovém kódu systému a potom systém znovu zkompilovat. Rekompilování bylo nákladné, jelikož bylo nutné systém COMPRO editovat, testovat, ověřovat a kontrolovat. Navíc systémový návrhář museli psát a udržovat tisíce řádek počítačového kódu.

50

Cílem vynálezu je takto systém umožňující za chodu vytvářet a vyhodnocovat rovnice pro monitorování chodu zařízení, a modifikovat podmínky bez nutnosti rekompilovat zdrojový kód systému.

5

### Podstata vynálezu

Výše uvedené cíle lze dosáhnout způsobem monitorování chodu zařízení pomocí CPU, jehož podstata spočívá v tom, že zahrnuje ukládání parametrů, vztahujících se k provozním podmínkám zařízení, do relační databáze, generování rovnic v CPU, definujících způsob vyhodnocení údajů získaných z provozních senzorů zařízení na základě parametrů uložených v relační databázi a monitorování a vyhodnocování uvedených rovnic při chodu zařízení s cílem stanovit, zda v zařízení došlo k poruše.

Výhodně generování rovnic v CPU zahrnuje překlad každé rovnice do spustitelného kódu a vkompileování spustitelného kódu přímo do programového zásobníku.

Výhodně generování rovnic zahrnuje výběr parametrů pro rovnice z relační databáze, identifikaci vzájemných logických vztahů z uvedených parametrů, zřetězení parametrů do tvaru singulárních rovnic, zřetězení všech singulárních rovnic s využitím identifikace vzájemných logických vztahů a generování složené rovnice.

Výhodně se při využití parametrů zahrnujících množiny descriptorů, stavů a nastavených hodnot singulární rovnice vytvoří zřetězením jednoho z deskriptorů, jednoho z logických stavů a odpovídající nastavené hodnoty.

Výhodně způsob podle vynálezu dále zahrnuje neustále vyhodnocování rovnic vstupních podmínek, přičemž v případě poruchy zařízení se z množiny procedur vybere odpovídající procedura schopná provést odstranění poruchy.

Výhodně za předpokladu, že každá procedura obsahuje množinu dílčích kroků, které sestávají z množiny rovnic, způsob podle vynálezu dále zahrnuje vyhodnocování všech rovnic souvisejících s procedurou a výběr jednoho kroku prováděné procedury, přičemž se monitorují rovnice související s uvedeným krokem s cílem zjistit, zda procedura pokračuje.

Výhodně se dále provede vyhodnocování rovnic vstupních podmínek s cílem zjistit, zda nastala porucha, přičemž se monitorují rovnice vstupních podmínek s cílem vybrat odpovídající proceduru schopnou eliminovat poruchu a monitorování a vyhodnocování rovnic se provádí až do eliminace poruchy.

Výhodně se za předpokladu, že každá procedura obsahuje množinu dílčích kroků, které sestávají z množiny rovnic, dále provede výběr kroku prováděné procedury, provádí se vyhodnocení všech rovnic všech kroků procedury, monitoruje se výsledek rovnic uvedeného kroku procedury a zjišťuje se, zda došlo k poruše uvedeného kroku, což indikuje, že tento krok nebyl proveden.

Způsob podle vynálezu výhodně dále zahrnuje vyhodnocení rovnic vstupních podmínek s cílem zjistit počátek poruchy, vyhodnocení rovnic procedury s cílem zjištění poruchy procedury/kroku a vyhodnocení rovnice paralelní informace souběžně s vyhodnocením rovnic procedury s cílem zjistit poruchu související s paralelní informací.

Výhodně se výše uvedeným způsobem provádí způsob monitorování chodu jaderné elektrárny, jehož podstata spočívá v tom, že se

- a) z parametrů uložených v relační databázi generují rovnice vstupních podmínek, rovnice kroku procedury a rovnice paralelní informace,
- b) vyhodnotí se rovnice vstupních podmínek s cílem zjistit počátek poruchy,
- c) monitorují se rovnice vstupních podmínek s cílem vybrat z množiny procedur odpovídající proceduru schopnou eliminovat poruchu,
- d) vyhodnocení a monitorování vstupních podmínek provádí až do eliminování poruchy,
- e) vybere se ten z kroků prováděné procedury, jehož provedení vede k eliminování poruchy,
- f) provede se vyhodnocení všech rovnic všech kroků procedury,
- g) monitoruje se výsledek rovnic kroku procedury odpovídajícího kroku,
- h) zjistí se, zda nastala porucha kroku procedury,
- i) jestliže se porucha kroku procedury vyskytla, je indikována zprávou,
- j) současně s prováděním stupňů e) až i) se vyhodnocují všechny rovnice paralelní informace,
- k) monitorují se paralelní informační rovnice související s uvedeným krokem,
- l) v závislosti na výsledku monitorování ve stupni k) se zjistí, zda nastala porucha související s paralelní informací, a
- m) jestliže nastala porucha související s paralelní informací, je indikována zprávou o poruše.

Jak je z výše uvedeného zřejmé, jedná se o metodu využívající relační databázi a řídicí systém pro uchování informací o postupech. Mezi tyto informace spadají deskriptory, logické stavy a nastavené hodnoty. Deskriptory, stavy a nastavené hodnoty jsou vzájemně sřetězeny tak, že vytvářejí singulární a sloužené rovnice, které popisují, jak se budou vyhodnocovat data ze senzorů. Jakmile se rovnice vygenerují, jsou jednotlivě analyzovány a zkompileovány do spustitelné formy. Zkompileované rovnice a podmínky se analyzují a monitorují a tak lze určit, zdali se vyskytla porucha zařízení. Pokud se porucha skutečně vyskytla, systém COMPRO doporučí operátorovi zařízení určitý předem definovaný postup akcí. Systém nepřetržitě monitoruje a vyhodnocuje rovnice a postoupí do dalšího kroku pouze tehdy, když obdrží potvrzení od operátora zařízení. Komponenty jednotlivých rovnic tedy slouží jako vstupy do systému a nemusí být uloženy a zakódovány přímo v programu. Proto není nutné měnit kód systému, ale stačí změny provést pouze v rovnicích. Jelikož jsou izolovány všechny procedurální informace včetně komponentů rovnic, tedy ty části, které je nutno často v kódu programu měnit, eliminuje se potřeba modifikace zdrojového kódu při změnách procedurálních informací nebo při změnách nastavených hodnot.

Výhody tohoto systému jsou jasné. Vynález snižuje celkovou složitost systému tím, že uchovává procedurální údaje a nastavené hodnoty v relační databázi. To usnadňuje údržbu systému, protože je mnohem jednodušší modifikovat procedurální informace uložené v relační databázi, nežli modifikovat zdrojový kód. Tento vynález také umožňuje vytvořit uživatelsky příjemnější prostředí a dále lze převést aplikaci do cizích jazyků, jelikož je snadné změnit znakové řetězce uložené v databázi do jakéhokoliv jazyka.

#### Přehled obrázků na výkresech

Obrázek 1 je blokové schéma systému podle tohoto vynálezu.

Obrázek 2A znázorňuje jeden z mnoha procedurálních kroků implementovaných v systému COMPRO.

Obrázek 2B znázorňuje procedurální krok uložený jako série tabulek v relační databázi.

Obrázek 3 je blokové schéma znázorňující generování a vypočtení rovnic.

Obrázek 4 je vývojový diagram popisující operaci vytváření rovnice.

Obrázek 5 znázorňuje příklady znakových řetězců - jednotlivých singulárních rovnic a složenou rovnici.

5 Obrázek 6 znázorňuje programový zásobník (program stack).

Obrázek 7 je blokové schéma generování a provádění komplexních operačních procedur.

Obrázek 8 znázorňuje, jak se informace zobrazují na obrazovce.

## 10 Příklady provedení vynálezu

### Příklad

15 Přehled hardwarových a softwarových modulů systému je znázorněn na obrázku 1. Pro spouštění programových modulů se používají běžně dostupné pracovní stanice Sun Micro System (pracovní stanice Sun-3 nebo Sun-4), na obrázku 1 znázorněné jako CPU 100. Vstup uživatele 102 do CPU 100 se provádí pomocí vstupního zařízení 104 jako je myš, klávesnice nebo dotykový displej. Dotykový displej je zařízení, které dokáže reagovat na lidský dotyk. Výstupní 20 informace z CPU 100 se zobrazují na výstupním zařízení 106, na tiskárně nebo na displeji. CPU 100 využívá ve výhodném provedení operační systém SUNOS® 108 (verze 3.0.3). CPU 100 uchovává procedurální informace v relační databázi na diskové jednotce 110. K řízení se používá relační databázový řídicí systém 112.

25 Relační databáze je soubor vzájemně souvisejících informací. Relační databáze podle tohoto vynálezu obsahuje série tabulek které spolu vzájemně logicky souvisejí. Na obrázku 2A je příklad jedné z množství procedur, která se skládá z mnoha kroků. Každý krok zahrnuje množství rovnic, které jsou uloženy v relační databázi. Obrázek 2B ukazuje krok 1 procedury EO z obrázku 2A uložený jako série tabulek v relační databázi.

30 Krok 1 podle obrázku 2A zjišťuje, zdali došlo k prasknutí trubky parního generátoru. Dílčí krok a kroku 1 procedury EO nejprve prověří indikátor běhu pumpy A a zjistí, jestli pampa A pracuje. Dílčí krok b kroku 1 procedury EO dále zkontroluje tlakový senzor, který indikuje úroveň tlaku parního generátoru, a určí, je-li tlak vyšší než 50 %. Nakonec se přečte hodnota tlakového 35 senzoru generátoru a zjistí se, zda je tlak nižší než 7 MPa (1000 psia). Krok 1 procedury EO tedy zjistí, že došlo k prasknutí trubky parního generátoru, jestliže

- a) pampa A pracuje, a
- b) tlak je vyšší než 50 %, a
- 40 c) tlak parního generátoru je nižší než 7 MPa.

To znamená, že k prasknutí trubky došlo pouze tehdy, když jsou všechny 3 podmínky splněny (TRUE). COMPRO potom navrhne operátorovi zařízení jednotlivé kroky, které se zobrazí na výstupním zařízení 106 a operátor může přikročit k nápravě, v tomto případě zastaví chod pumpy A.

50 Obrázek 2B ukazuje neúplnou tabulku kroků, ve které je uloženo číslo procedury (jako například EO), čísla kroků, čísla dílčích kroků, (DSS) čísla (klíče) objektů a OR FLAGS (příznaky). Procedura se skládá z kroků a kroky se skládají z dílčích kroků. Dílčí kroky mohou obsahovat ještě další dílčí kroky, v tomto příkladě ale žádné neobsahují.

DSS čísla (klíče) objektů tabulky kroků podle obrázku 2B ukazují na vstupy v DSS tabulce objektů. DSS tabulka objektů obsahuje deskriptor, stav pravda a nepravda a nastavenou hodnotu. Číslo deskriptoru ukazuje na vstupy v tabulce deskriptorů.

5 Tabulka deskriptorů obsahuje jméno deskriptoru, symbol proměnné pro deskriptor a čísla jednotek, která ukazují na tabulku jednotek. Čísla stavů pravda a nepravda DSS tabulky objektů ukazují na odpovídající vstupy tabulky stavů pravda a tabulky stavů nepravda. Nastavené hodnoty DSS tabulky objektů obsahují vlastní čísla (hodnoty).

10 Tabulka stavů pravda obsahuje jednak čísla stavů pravda, vlastní stav pravda a odpovídající rovnici stavu. Číslo stavu pravda ukazuje na číslo stavu pravda v DSS tabulce objektů a označuje číslo řádky tabulky stavů pravda. Stav pravda popisuje pozitivní stav logického stavu, například "větší než" nebo "menší než", a používá se pouze pro účely zobrazování. Rovnice stavu definují tvar rovnice, která má být řešena. Je definována každá jednotlivá část rovnice, jako například

15 [proměnná] < [nastavená hodnota].

Tabulka stavů nepravda obsahuje čísla stavů nepravda a vlastní stav nepravda. Číslo stavu nepravda ukazuje na číslo stavu nepravda v DSS tabulce objektů. Stav nepravda popisuje negativní stav logického stavu, například "ne větší než" nebo "ne menší než", a používá se pouze

20 pro účely zobrazování.

Tabulka jednotek zahrnuje čísla jednotek a názvy jednotek. Číslo jednotky ukazuje na číslo jednotky v tabulce deskriptorů. Název jednotky je například MPa a používá se pouze pro účely

25 zobrazování.

Jak je vidět z tabulky kroků, procedura EO krok 1 má pouze tři dílčí kroky a, b a c. Každý dílčí krok je rovnice vytvořená podle obsahu ostatních tabulek. Každá vytvořená rovnice vyjadřuje podmínku procedury EO z obrázku 2A, krok 1, dílčí kroky a, b a c. DSS čísla (klíče) objektů tabulky kroků ukazují na vstupy v DSS tabulce objektů. DSS tabulka objektů vyjadřuje vztahy

30 mezi jednotlivými tabulkami na obrázku 28 při vytváření rovnic.

Relační databázový řídicí systém 112, jako INGRES<sup>TM</sup> nebo ORACLE, což jsou obchodně běžně dostupné softwarové systémy, pomáhají při správě informací uložených v tabulkách (např. na obrázku 28). INGRES<sup>TM</sup> automaticky zajišťuje čtyři funkce:

35

- 1) příjem dat a instrukcí od uživatele
  - 2) výběr informací z databáze
  - 3) aktualizaci databáze
  - 4) formátuje a předává data uživatelům.
- 40

Tyto funkce značně usnadňují správu databáze a manipulaci s ní. Obsluha s běžnými znalostmi systémů INGRES<sup>TM</sup>, UNIX a jazyka C může vytvořit příslušný software využívající tyto funkce, například pro výběr, zavádění a údržbu databáze podle obrázku 28.

45 INGRES<sup>TM</sup> umožňuje ukládat a vybírat všechna data potřebná ke generování procedur, jako například vstupní podmínky, kroky, dílčí kroky, deskriptory, stavy a nastavené hodnoty. Specialisté obsluhovaného zařízení používají funkce systému INGRES<sup>TM</sup> pro ukládání procedurálních informací do výše popsaných tabulek a pro specifikaci propojení a vzájemných stavů mezi tabulkami, jak je ukázáno na obrázku 2A. INGRES<sup>TM</sup> zajišťuje další funkce, jako prohlížení

50 současného obsahu databáze, vyhledání určitých dat, provádění modifikací dat, mazání starých dat a přidávání nových dat. Funkce pro generování sestav umožňují generovat tiskové sestavy obsahu databáze.

Navíc kromě popsaných funkcí pro výběr, přidávání, modifikování a mazání dat INGRES<sup>TM</sup> zajišťuje integritu dat. Během aktualizace je automaticky chráněna konzistence dat a nepřetržitě se vytváří provozní deník, takže v případě softwarové nebo hardwarové chyby je možno databázi plně obnovit. INGRES<sup>TM</sup> provádí kontroly platnosti, čímž ochraňuje integritu databáze. Každou aktualizaci lze konfrontovat s omezeními, která definovali systémoví specialisté při návrhu systému, aby se zajistilo zadávání správných vstupů.

Tento vynález, jak je vidět podle obrázku 1, obsahuje tři hlavní softwarové moduly:

- 1) modul pro údržbu relační databáze 114
- 2) modul pro sestavování a údržbu procedur 116
- 3) modul pro zpracování procedur 118.

Tyto tři softwarové moduly pomáhají operátorovi zařízení při generování procedur a při jejich provádění tím, že konvertují data uložená v databázi do spustitelného tvaru. Každý z těchto tří softwarových modulů je s výhodou napsán v programovacím jazyce C s vloženým SOL (standardní dotazovací jazyk).

Tyto tři softwarové moduly jsou vzájemně nezávislé spustitelné programy. Moduly pro údržbu relační databáze 114 a pro sestavování a údržbu procedur 116 pracují v režimu off-line, což znamená, že tyto moduly jsou spouštěny před skutečným spuštěním systémového software. Modul pro zpracování procedur 118 využívá běžně dostupné obslužné programy (utility) systému UNIX<sup>TM</sup>, jako jsou lex 122 a yacc 124 pro sestavování a kompilaci rovnic z dat uložených v relační databázi a pro jejich provádění za běhu programu. Lex 122 a yacc 124 jsou nástroje pro kompilování rovnic za běhu systému, kompilování rovnic se tedy neprovádí při kompilování systémového zdrojového kódu. Lex 122 je generátor lexikálních analyzátorů, yacc 124 je kompilovací zapisovací systém. Odborník znalý dané problematiky porozumí, jak používat tyto obslužné programy pro rozbor a kompilaci řetězců.

Softwarové moduly také využívají systému zobrazování v okně 126 (Windowing system), jako jsou běžně dostupné Windows 4GL pro off-line vstup a výstup dat. Dále jsou to Open Windows pro provádění zpracování procedur 118. Windows 4GL pomáhají specialistům zařízení při sestavování a údržbě procedur 116 a s komunikací s relační databází INGRES<sup>TM</sup>. Windows 4GL komunikují se SunOS 108 a umožňují specialistům zařízení vytvářet uživatelské formuláře pro vkládání a výběr dat z relační databáze. Windows 4GL usnadňují programátorům ovládání oken tím, že poskytují vizuální editory umožňující návrh a testování formulářů interaktivním způsobem, generování libovolného počtu oken v aplikaci, interaktivní umístění polí, tlačítek, bitových map a dalších elementů v oknech a jiné funkce. Modul pro údržbu relační databáze 114 používá konvence UNIX<sup>TM</sup> pro aktualizaci databáze.

Modul pro údržbu relační databáze 114 je obslužný program, který pomáhá správci databáze při provádění rutinních operací spojených s údržbou databáze a diskové jednotky 110, jako je vytváření provozního deníku, a při automatickém zápisu kontrolních bodů (checkpointing). Tím se zajišťuje integrita databáze uložené na diskové jednotce 110 a kontroluje se, zda je databáze skutečně aktualizována. Modul pro údržbu relační databáze 114 neověřuje vstupy relační databáze, ale zaručuje, že databáze je platná a že nebyla změněna osobou s neautorizovaným přístupem. Pokud by databáze nebyla řádně udržována, nebyly by také správně prováděny komplexní operační postupy. Proto je správnost databáze hlavním předpokladem bezchybného chodu systému. Odborník znalý této problematiky může využít výše zmíněné funkce pro správu databáze přístupné pomocí konvencí UNIX používajících příkazy INGRES k provádění údržby relační databáze 114.

Modul pro sestavování a údržbu procedur 116 slouží pro zavádění procedurální informace do relační databáze, takže ji lze zpracovat v modulu pro zpracování procedur 118. Jak je ukázáno na

obrázku 28, vstupní procedurální informace zahrnuje například čísla kroků, čísla dílčích kroků, jména deskriptorů, proměnné, nastavené hodnoty, jednotky atd. Specialisté zařízení používají série formulářů. Formuláře jsou zobrazovány systémem zobrazování v okně 126 (Windowing system). Procedurální informace jsou na formulářích uspořádány tak, že podobné (souhlasné) informace budou součástí stejného formuláře. Odborník znalý této problematiky může zavést procedurální informace do databáze tím, že použije výše zmíněné běžně dostupné funkce, které jsou spustitelné nebo přístupné pomocí systému INGRES.

Modul pro zpracování procedur 118 systému COMPRO zajišťuje přímé zpracování procedur. Modul pro zpracování procedur 118 provádí generování a výpočet komplexních operačních procedurálních rovnic pro monitorování zařízení jaderné elektrárny. Diagram na obrázku 3 znázorňuje jednotlivé softwarové moduly modulu pro zpracování procedur 118, který slouží ke generování spustitelných softwarových rovnic. Generátor datové struktury 200 generuje procedurální informaci 202 z relační databáze uložené na diskové jednotce 110 pomocí relačního databázového řídicího systému 112.

Generátor datové struktury 200 v podstatě kopíruje obsah databáze z diskové jednotky 110 do místní paměti. Kopírování obsahu databáze do místní paměti se provádí z důvodu snížení doby přístupu k datům v databázi na diskové jednotce 110. V případě, že by relační databáze INGRES měla rychlejší přímé přístupové schopnosti, nebylo by nutné používat generátor datové struktury 200. Datové struktury jazyka C lze tedy snadno přenášet a systém tím nebude zpomalován. Data z relační databáze se kopírují do místní paměti a jsou uložena v datových strukturách jazyka C, jako jsou "struct" a ukazatel nebo například "char \*pointer", což vytváří strukturu typu tabulky a definují se vzájemné vztahy mezi tabulkami. To tedy znamená, že uspořádání tabulky na obrázku 28 použité pro uložení v relační databázi je udržováno v místní paměti.

Databázové tabulky a asociativní informace, jako například informace o vzájemných vztazích mezi tabulkami, se čtou z diskové jednotky 110 pomocí příkazů systémů SOL a INGRES, například příkazem SELECT. Tabulky a informace o vzájemných vztazích mezi tabulkami uložené v relační databázi se předem nadefinují pro každou aplikaci. Tyto tabulky se načtou do místní paměti s využitím datových struktur jazyka C. Specialisté zařízení například mohou vytvořit a definovat tabulku, jako je tabulka kroků na obrázku 28 tím, že vytvoří a označí každý sloupec tabulky a vloží vhodná data do řádek a sloupců. Odborník znalý této problematiky může prozkoumat tabulku kroků a vytvořit odpovídající datové struktury jazyka C pro uchování této informace.

Stejný proces se bude opakovat pro každou tabulku uloženou v relační databázi. Obsah tabulky uložené na diskové jednotce 110 se přečte například pomocí SOL příkazu SELECT a uloží se v odpovídající datové struktuře jazyka C. Odborník znalý této problematiky může definovat datové struktury jazyka C a C/UNIX software a pomocí INGRES a SOL snadno kopírovat data z databáze uložené na diskové jednotce 110 do místní paměti. Do každého místa v tabulce, kde není žádný vstupní údaj, bude při zpracování modulem pro zpracování procedur 118 dosazena nula nebo v jazyce C "null".

Poté, co generátor datové struktury 200 uloží obsah databáze do lokální paměti, generátor rovnic 204 vygeneruje všechny rovnice a matematické výrazy tak, jak je znázorněno na obrázku 4. Generátor rovnic 204 podle obrázku 4 vygeneruje každou rovnici, která byla dříve pevně zakódována do systémového software.

Jsou zde dva typy rovnic: rovnice složené a rovnice singulární. Složené rovnice se skládají ze singulárních rovnic. Pro každou z podmínek (dílečí kroky a, b a c) podle obrázku 2A pro proceduru EO, krok 1 je vytvořena odpovídající singulární rovnice zřetěžením deskriptoru, logického stavu a v případě potřeby jedné nebo více nastavených hodnot.



Deskriptor představuje jeden z mnoha ventilů, pump, generátorů, přerušovačů atd. zařízení jaderné elektrárny a je uložen v relační databázi. Například deskriptor pro podmínku na obrázku 2A, procedura EO, krok 1, dílčí krok a je "pumpa A". Podobně deskriptory pro podmínky na obrázku 2A, procedura EO, krok 1, dílčí kroky b a c jsou "úroveň tlaku" a "tlak parního generátoru".

5 Deskriptory jsou uloženy v datové struktuře C tabulky deskriptorů pod jménem deskriptoru. Datová struktura C tabulky deskriptorů ukazuje na číslo deskriptoru uložené v datové struktuře C DSS tabulky objektů. DSS tabulka objektů dále ukazuje na DSS čísla (klíče) objektů datové struktury C tabulky kroků. Jméno proměnné uložené v datové struktuře C tabulky deskriptorů představuje jednoznačný identifikátor pro hodnotu, kterou obdržíme při snímání senzory zařízení

10 elektrárny.

Logický stav pro každou rovnici vyjadřuje vztah deskriptoru vzhledem k nastaveným hodnotám. Logický stav může být například  $<$ ,  $>$ ,  $=$ , AND a NOR atd. Logický stav pro obrázek 2A, proceduru EO, krok 1, dílčí krok b je "větší než". Logický stav pro proceduru EO, krok 1, dílčí

15 krok a je buď "PRAVDA", nebo "NEPRAVDA", protože pumpa buď pracuje, nebo nepracuje, což se zjistí pomocí senzoru spínače pumpy A.

Datová struktura C tabulky stavů pravda podle obrázku 28 uchovává jednotlivé logické stavy rovnic. Logický stav rovnice definuje tvar rovnice tím, že se vyberou komponenty, které se mají

20 porovnávat, a logický operátor, který vykonává vlastní porovnání komponent, například " $<$ ". Jak je vidět na obrázku 28, logický stav procedury EO, kroku 1, dílčího kroku b je " $>$ " oproti logickému stavu procedury EO, kroku 1, dílčího kroku c, kde logický stav je " $<$ ".

Nastavená hodnota DSS tabulky objektů je taková hodnota, ke které bude odpovídající deskriptor buď ve stavu "PRAVDA", nebo ve stavu "NEPRAVDA". Jak je vidět na obrázku 2A, nastavené hodnoty pro proceduru EO, krok 1, dílčí kroky b a c jsou uloženy v datové struktuře C

25 DSS tabulky objektů jako 50 a 7 (7 MPa=1000 psia) pod "nastavená hodnota" (Setpoint).

Generátor rovnic 204 začne z datové struktury C číst proceduru a krok 300 pro každý výskyt rovnice spojené s krokem a dílčím krokem. Například ukazatel ukazuje na datovou strukturu C

30 "struct" tabulky kroků na obrázku 28. Datová struktura C pro číslo procedury a číslo kroku se přečte a uloží jako proměnná.

Generátor rovnic 204 provede generování označení složené rovnice 302. Například z datové struktury C tabulky kroků se nejprve přečte číslo procedury. Číslo procedury se uloží do znakového řetězce složené rovnice, číslo je následováno znakem ":". Z datové struktury tabulky kroků se přečte číslo kroku, zkonvertuje se na znakový řetězec a uloží za "EO:" do znakového řetězce

35 složené rovnice. Dále se vloží za číslo kroku do znakového řetězce složené rovnice ":", tím dostaneme "EO:1=", kde EO:1= je označení, které bude uchovávat výsledek rovnice.

40

Z datové struktury C se přečte další číslo procedury a kroku a provede se porovnání 306 s proměnným číslem procedury a číslem kroku. Odlišné číslo procedury nebo číslo kroku indikuje konec složené rovnice. Složená rovnice se pak uloží 308 do souboru (na obrázku není vyznačen). Tento soubor obsahující singulární a složené rovnice se pošle do syntaktického analyzátoru

45 a kompilátoru. Složený znakový řetězec se nastaví na "null" - provede se reinitializace 310. Jestliže číslo procedury a kroku ukazuje na konec datové struktury tabulky kroků, provede se opuštění generátoru rovnic 314. V opačném případě generátor rovnic generuje označení další složené rovnice 302.

Když se porovnáním 306 zjistí, že číslo další procedury a číslo kroku jsou stejné, přečte se logický operátor pro složenou rovnici 316 a uloží se do složeného znakového řetězce. Například se přečte OR FLAG uložený v datové struktuře C tabulky kroků pro proceduru EO, krok 1, dílčí

50 krok a. Pokud je výsledek "null", potom se nic nepřipojí na konec složeného znakového řetězce. Je-li výsledek pravdivý, vloží se na konec složeného znakového řetězce rovnice symbol "I"

(logický operátor OR). Pokud není výsledek pravdivý, vloží se na konec složeného znakového řetězce rovnice symbol "&" (logický operátor AND). V tomto příkladě je tedy OR\_FLAG null, tedy se nic nepřipojí na konec složeného znakového řetězce.

- 5 Dále se generuje označení pro singulární rovnici 318. Například ukazatel bude ukazovat na vstupy v datové struktuře C tabulky kroků, která indikuje rovnici reprezentující proceduru EO, krok 1, dílčí krok a. Generování označení pro singulární rovnici začne přečtením řetězce s číslem procedury "EO" z datové struktury C a jeho uložením do proměnného znakového řetězce. "EO" se také připojí na konec znakového řetězce složené rovnice. Dále se na nejbližší přístupné místo
- 10 proměnného znakového řetězce a znakového řetězce složené rovnice vloží ":". Z datové struktury C tabulky kroků se přečte číslo kroku, zkonvertuje se na znaky a uloží se na další pozici v proměnném znakovém řetězci a znakovém řetězci složené rovnice. Za číslo kroku se dále do proměnného znakového řetězce uloží "=". Každá rovnice bude začínat označením, ve kterém bude číslo procedury a kroku, a pokud rovnice není složená rovnice, tak ještě číslo dílčího kroku.
- 15 Označení rovnice využívá syntaktický analyzátor a kompilátor pro uložení výsledku rovnice. Oddělovačem jednotlivých čísel reprezentovaných jako znaky je ":".

- Zřetěžením deskriptoru, logického operátoru a jedné nebo více nastavených hodnot se vytvoří a uloží singulární rovnice 320. Například se z datové struktury C DSS tabulky objektů přečte
- 20 číslo stavu pravda. Číslo stavu pravda v datové struktuře C DSS tabulky objektů je ukazatel na číslo stavu pravda v tabulce stavů pravda. Stav se nyní přečte do znakového řetězce. Jestliže znakový řetězec rovnice stavu obsahuje [odkaz na název proměnné], proměnná je přečtena z datové struktury C tabulky deskriptorů a nahrazuje [odkaz na název proměnné]. Pokud stavová rovnice obsahuje [odkaz na nastavené hodnoty], nastavená hodnota je přečtena z datové
- 25 struktury C DSS tabulky objektů a nahrazuje [odkaz na nastavené hodnoty]. Znakový řetězec stavové rovnice se pak uloží do proměnného znakového řetězce. Tak je vytvořen znakový řetězec rovnice, který obsahuje deskriptor, logický operátor a nastavenou hodnotu.

- V tomto příkladě se proměnná čte čtením DSS čísla objektu, které je ukazatelem na DSS číslo objektu v DSS tabulce objektů. Tento ukazatel bude ukazovat na první část vstupů v DSS tabulce objektů. Podle čísla ukazatele na DSS tabulku objektů se přečte číslo deskriptoru uložené v datové struktuře C DSS tabulky objektů. Číslo deskriptoru v datové struktuře C DSS tabulky objektů je ukazatel na číslo deskriptoru uložené v datové struktuře C tabulky deskriptorů. Podle čísla ukazatele na deskriptor se přečte proměnná uložená v datové struktuře C tabulky deskriptorů
- 35 a dosadí se do znakového řetězce rovnice stavu.

- V tomto příkladě se nastavená hodnota čte podle hodnoty ukazatele DSS čísla objektu z datové struktury C DSS tabulky objektů. Nastavené hodnoty pro proceduru EO, krok 1, dílčí kroky a a b jsou 50 a 7 (1000 psia). Tyto hodnoty se zkonvertují na znakové řetězce a dosadí do znakového
- 40 řetězce rovnice stavu.

- Jakmile se znakový řetězec rovnice stavu uloží do proměnného znakového řetězce, výsledkem je rovnice pro proceduru EO, krok 1, dílčí krok a. Vytvořená rovnice 320 v proměnném znakovém řetězci se pak uloží do souboru (soubor na obrázku není vyznačen). Ten se pošle do kompilátoru
- 45 a syntaktického analyzátoru a všechny rovnice se najednou vygenerují. Funkce kompilátoru a syntaktického analyzátoru, yacc a lex, budou popsány později.

- Poté, co je proměnný znakový řetězec uložen do souboru 320, přečte se číslo další procedury a kroku 304 a provede se porovnání s předchozím číslem procedury a kroku. Je-li výsledek porovnání pravda, jsou vygenerovány všechny singulární rovnice pro proceduru/krok. Složené rovnice se potom uloží 308 do souboru a provede se reinicializace složeného řetězce 310 (nastavení na "null").
- 50

Rovnice reprezentující proceduru EO, krok 1, dílčí krok b a c se generují stejným způsobem, jak bylo popsáno výše pro dílčí krok a. To znamená, že všechny singulární a složené rovnice pro každou vstupní podmínku, proceduru/krok a paralelní informaci se generují tou samou metodou, která je popsána v předchozím textu. Výsledné znakové řetězce singulárních a složených rovnic pro proceduru EO, krok 1, jsou zobrazeny na obrázku 5.

Výsledkem každé singulární rovnice bude buď pravda, nebo nepravda. Tento výsledek pomůže zjistit, zda zařízení jaderné elektrárny funguje správně. Po vyhodnocení složené rovnice na obrázku 5 se pro proceduru na obrázku 2A zjistí, zda-li došlo k prasknutí parního generátoru. Výsledek vyhodnocení pravda může být popudem ke zprávě, kde se navrhne způsob řešení situace. V případě prasknutí parního generátoru se zastaví pumpy nebo se sníží tlak.

Generátor rovnic 204 nejprve vygeneruje každou rovnici. Potom je soubor obsahující znakové řetězce pro všechny singulární a složené rovnice načítán kompilátorem 206. Syntaktický analyzátor 208 spolu s výkonnými UNIX<sup>TM</sup> utilitami (obslužnými programy) lex 122 a yacc 124 kompiluje zásobníky 210 a tabulky 214 a 216. Syntaktický analyzátor 208 analyzuje rovnice vygenerované ve formě řetězců generátorem rovnic 204. Syntaktický analyzátor 208 může zpracovávat jak jednoduché, tak i složité výrazy. Odborník znalý této problematiky může využívat tyto běžně dostupné utility UNIX lex a yacc pro analýzu a kompilaci řetězců.

Utilita (obslužný program) UNIX<sup>TM</sup> lex 122 se používá pro definování rozlišitelných jednotek ("tokens"), které je syntaktický analyzátor 208 schopen rozpoznat. Rozlišitelné jednotky zahrnují různé typy přípustných proměnných, operátorů (například =, +, -, \*, /, <, >, atd.), funkcí a předdeklarovaných konstant. Další utilita UNIX<sup>TM</sup> yacc 124 poskytuje obecný nástroj pro definování pravidel a struktury pro vstup uživatelsky definovaných rovnic. Lex 122 je rutina vytvořená ve strojovém kódu, kterou využívá yacc 124 pro výběr rozlišitelných jednotek z uživatelsky definovaných rovnic 202 a jejich zorganizování podle vstupních pravidel a struktury. Když yacc 124 rozpozná řád v systémově definovaných rovnicích, kompilátor 206 zavolá funkce lex 122 a yacc 124 pro generování programového zásobníku 210, vstupní tabulky ukazatelů 214 a tabulky symbolů 216.

Kompilátor používá syntaktický analyzátor 208, lex 122 a yacc 124 pro každou rovnici. Je nutno pamatovat, že znakové řetězce rovnic generované generátorem rovnic 204 nemají předem určený obsah, ale jsou determinovány systémovým návrhářem jaderné elektrárny. Kompilátor provádí kroky podle předdefinované logiky, což je důvodem flexibility systému.

Kompilátor používá obvyklé funkce a vytváří tabulku symbolů 216, programový zásobník 210 a vstupní tabulku ukazatelů 214. Programový zásobník 210 pro singulární a složené rovnice z obrázku 5 je na obrázku 6. Jak je vidět, první vstup pro rovnici je označení oblasti, kde se nachází výsledek. Dále následuje logický operátor a zbývající dva vstupy jsou operandy. Jak je ukázáno na obrázku 6, podobné uspořádání platí pro složené rovnice, kde jsou nejprve všechny logické operátory a za nimi následují operandy.

Jak bylo uvedeno výše, programový zásobník 210 obsahuje spustitelné instrukce, zde označené jako pseudoinstrukce, a názvy operandů. Tyto pseudoinstrukce zahrnují:

1. operátory nebo dotazy pro provádění sady základních příkazů skládajících se z operátorů a knihovních funkcí
2. dotazy pro provádění dalších uživatelsky definovaných programových segmentů
3. dotazy pro posuv adres nebo hodnot do kontextového zásobníku 212 z tabulky symbolů 216.

Na konci kompilace obsahuje vstupní tabulka ukazatelů 214 ukazatele na programový zásobník. Vstupní ukazatele ukazují na rovnice paralelní informace, rovnice vstupní podmínky a rovnice

procedury/kroku. Tabulka symbolů 216 obsahuje ukazatele na dočasné proměnné, konstanty a předdeklarované funkce. Modul pro zpracování procedur zobrazený na obrázcích 1 a 7 provádí a vyhodnocuje rovnice uložené v programovém zásobníku 210 spolu s kontextovým zásobníkem 212, vstupní tabulkou ukazatelů 214 a tabulkou symbolů 216.

5

Obrázek 7 je vývojový diagram znázorňující funkci modulu pro zpracování procedur 118. Generátor datové struktury 200 pomocí dotazů INGRES 400 zpřístupňuje a kopíruje relační databázi uloženou na diskové jednotce 110 do místní paměti 402. Generátor rovnic 204 generuje všechny logické rovnice 404 metodou zobrazenou na obrázcích 3 a 4.

10

Nejprve se provede vyhodnocení rovnic vstupních podmínek a zjistí se, zda-li je každá singulární a složená rovnice pravda nebo nepravda. Vyhodnocení rovnic je provedeno použitím vstupní tabulky ukazatelů 214, programového zásobníku 210, kontextového (datového) zásobníku 212 a tabulky symbolů 216. Celý proces začíná vstupní tabulkou ukazatelů 214, která ukazuje na počáteční pozici pro vstup rovnic vstupních podmínek. Pro výpočet a vyhodnocování rovnic se využívá například obvyklé metody postfixového zápisu (reverzní polské notace). Programový zásobník, jak je vidět na obrázku 6, slouží k vypočtení singulárních rovnic EO:1:a, EO:1:b, a EO:1:c a složených rovnic z obrázku 5. Během tohoto výpočtu se tabulka symbolů 216, která je křížovou referencí z názvu proměnné do místa uložení proměnné, využívá pro získání skutečných hodnot proměnných a k uložení výsledků. Kontextový (datový) zásobník 212 se používá pro ukládání vyhledaných proměnných a k ukládání dočasných výsledků.

20

Zatím se provádělo vyhodnocování rovnic vstupních podmínek, aby se zjistilo, zda-li nedošlo k poruše 410 zařízení. Jestliže porucha nenastala, pokračuje se ve vyhodnocování rovnic vstupních podmínek 406 a v monitorování výsledků vyhodnocování 408, dokud se neobjeví podmínka signalizující poruchu zařízení jaderné elektrárny. Specialisté systému určí, jaké rovnice vstupních podmínek signalizují poruchu. Tyto rovnice předem vyvinuli systémoví návrháři a záleží na tom, jaké zařízení se monitoruje, jestli je to jaderná elektrárna, uhelná elektrárna nebo zařízení určené pro chemické zpracování.

30

Jestliže se při testování poruchy 410 vyskytne porucha pro jednu z procedur, je podle hlavičky procedury stanoveno číslo procedury a číslo kroku. Procedura se potom zobrazí operátorovi zařízení. Například když jsou všechny rovnice a, b a c podle obrázku 8 pravda, potom bude provedena procedura EO, krok 1, protože rovnice a, b a c následují v programovém zásobníku 210 hlavičku procedury EO, krok 1. Například procedura EO, krok 1 může způsobit vypnutí pumpy A, poté poslat zprávu na displej operátora zařízení a sdělit mu, že je pumpa A vypnuta. Podobně bude vykonána procedura E1, když jsou všechny rovnice paralelní informace následující proceduru E1 vyhodnoceny jako pravda.

35

Jakmile je stanoveno číslo procedury a kroku 412, vyřeší se všechny aktuální asociované rovnice 414. Procedura vyhodnocování je stejná jako v předchozí části. Dále se provede určení čísla procedury a kroku u právě probíhající procedury. Číslo procedury a kroku se určí z lokální paměti.

40

Ačkoliv byly vyřešeny všechny aktuální asociované rovnice 414 pro krok procedury, je prováděno monitorování 418 pouze těch rovnic, které jsou spojené s tímto krokem. Monitorování zahrnuje nepřetržité kontrolování rovnic a vytváření související zprávy až do té doby, dokud nezmizí podmínky, které zprávu vyvolaly, nebo dokud operátor zprávu neanuluje. Žádné další výsledky kroků nejsou monitorovány, ačkoliv rovnice související s těmito kroky byly vyhodnoceny.

45

50

Podle obrázku 8 se jednotlivé kroky procedury zobrazují na displeji 500 v rámečku pro zobrazování aktuálního kroku 502 (obrázek 7 - procedura monitorování, zobrazování zpráv a promptů (náznaků) 418). Text v rámečku pro zobrazování právě prováděného kroku 502 je návrh očekávané akce nebo odpověď operátora zařízení. Každý krok obsahuje příkaz vysoké úrovně, který

popisuje prováděnou úlohu. Když úloha (krok) vysoké úrovně vyžaduje vícenásobné činnosti, jsou specifikovány dílčí úlohy (dílčí kroky). Prováděním jednotlivých úloh a dílčích úloh se dosáhne požadovaného výsledku. Jednotlivé kroky jsou zapsané tak, že operátor zařízení postupuje směrem dolů v seznamu kroků a provádí je. Zobrazují se také všechny očekávané podmínky a operátor rozhodne o akci, která se provede a potvrdí, že je možno operaci provést.

Procedura monitorování, zobrazování zpráv a promptů (prompt=náznak) 418 zobrazí kroky doporučené akce, ze kterých si operátor zařízení vybírá. Jak je vidět na obrázku 8, zobrazení se provádí v oblasti uživatelských promptů (náznaků) 504. Prompt se bude zobrazovat až do té doby, dokud operátor zařízení neodpoví (zpracování odpovědi uživatele 424).

Souběžně se zobrazováním kroků doporučené akce se zkoumá, zda bylo dosaženo požadované odezvy. Jestliže tomu tak není, tak porucha nadále pokračuje a výstupní zařízení 106 zobrazí instrukce náhradního řešení v rámečku pro zobrazení aktuálního kroku 502. Téměř všechny kroky zobrazené v rámečku pro zobrazení právě aktuálního kroku 502 obsahují nějaké náhradní kroky. Pokud tyto náhradní kroky neexistují, systém pokračuje na další krok nebo dílčí krok, který je zobrazen v části pro zobrazení příštího kroku 506 a provede se znovu vyřešení všech aktuálních asociovaných rovnic 414 (podle obrázku 8).

Než systém přejde na další krok nebo proceduru, vyžádá si potvrzení od uživatele. Prompt se zobrazí v oblasti uživatelských promptů 504 na obrázku 8 (procedura monitorování, zobrazování zpráv a promptů 418 nebo 422) a doporučí operátorovi zařízení patřičné kroky. Jakmile operátor odpoví (zpracování odpovědi uživatele 424), provede se kontrola, zda-li operátor bude pokračovat podle navrhovaných kroků. Jestliže se rozhodne pokračovat a přejít na další krok nebo proceduru, tak, jak je zobrazeno pomocí procedury monitorování, zobrazování zpráv a promptů 418 nebo 422, provede se stanovení čísla kroku a procedury 412. Když se operátor naopak rozhodne ukončit monitorování aktuální procedury 428, tak se program vrací na vyhodnocování rovnic vstupních podmínek 406.

Zároveň s řešením aktuálních asociovaných rovnic 414 probíhá vyhodnocování všech paralelních informačních rovnic 430. Paralelní informace zahrnují například poznámky, výstrahy a stavové změny parametrů a komponent, které byly započaty, ale nebyly dokončeny při opuštění aktivního kroku procedury. Dále zahrnují ty parametry, které je nutno monitorovat vícekrát za jeden krok procedury, například udržování úrovně tlaku parního generátoru. Paralelní informační rovnice jsou uloženy v relační databázi a jsou generovány podobným způsobem, jako rovnice procedury/kroku zobrazené na obrázcích 2A a 2B.

Určení čísla procedury a kroku 432 se provede podobným způsobem, jaký byl popsán výše, což zajistí, že prováděná paralelní informační rovnice je asociována s právě probíhajícím krokem. Některá paralelní informace je globální a bude se vyhodnocovat vždy. Jestliže krok procedury (nebo dílčí krok), poznámka nebo výstraha zahrnují parametr, který by měl být neustále sledován, bude systém monitorovat tento parametr (monitorování paralelní informace 434) tak dlouho, dokud nebude některá z asociovaných rovnic vyhodnocena jako nepravda.

Výsledek vyhodnocování paralelních informačních rovnic je monitorován (monitorování paralelní informace 434), aby se zjistilo, zdali došlo k poruše (vyhodnocení poruchy 436). Jestliže jsou všechny asociované rovnice pro proceduru v paralelní informaci pravda, tak pro tuto proceduru nastala porucha. Když je vyhodnocení poruchy 436 pravda, objeví se v rámečku pro zobrazení paralelní informace 508 (viz obrázek 8) poplašná zpráva a zobrazí se prompt 438. Prompt se zobrazí v oblasti uživatelských promptů 504 a naznačí operátorovi zařízení, jakým směrem by se měl při řešení poruchy ubírat. Vyhodnocování paralelních rovnic 430 bude ovšem i nadále pokračovat, zároveň se ale bude zobrazovat poplašná zpráva a prompt 438.

Operátor zařízení tak tedy může plně řídit systém COMPRO. Systém nikdy neprovede další krok nebo proceduru bez souhlasu operátora zařízení. Cykly pro proceduru/krok a pro paralelní informaci probíhají zároveň. Systém zobrazuje prompty (náznaky) (418, 422 a 440), které operátorovi zařízení napoví, jakou proceduru a krok je nutno provést. Systém čeká na odpověď operátora, zároveň ale pokračují cykly vyhodnocování pro proceduru/krok a pro paralelní informaci. Systém přejde na další proceduru nebo krok teprve po souhlasu operátora. Jestliže operátor proceduru ukončí, provede se ukončení monitorování aktuální procedury a paralelní informace 428 a systém dále pokračuje ve vyhodnocování rovnic vstupních podmínek 406.

## PATENTOVÉ NÁROKY

1. Způsob monitorování chodu zařízení pomocí CPU, **vyznačený tím**, že zahrnuje ukládání parametrů, vztahujících se k provozním podmínkám zařízení, do relační databáze, generování rovnic v CPU, definujících způsob vyhodnocení údajů získaných z provozních senzorů zařízení na základě parametrů uložených v relační databázi a monitorování a provádění uvedených rovnic při chodu zařízení s cílem stanovit, zda v zařízení došlo k poruše.

2. Způsob podle nároku 1, **vyznačený tím**, že generování rovnic v CPU zahrnuje překlad každé rovnice do spustitelného kódu a vkompilování spustitelného kódu přímo do programového zásobníku.

3. Způsob podle nároku 1, **vyznačený tím**, že generování rovnic zahrnuje výběr parametrů pro rovnice z relační databáze, identifikaci vzájemných logických vztahů z uvedených parametrů, zřetězení parametrů do tvaru singulárních rovnic, zřetězení všech singulárních rovnic s využitím identifikace vzájemných logických vztahů a generování složené rovnice.

4. Způsob podle nároku 3 s využitím parametrů zahrnujících množiny deskriptorů, stavů a nastavených hodnot, **vyznačený tím**, že se singulární rovnice vytvoří zřetězením jednoho z deskriptorů, jednoho z logických stavů a odpovídající nastavené hodnoty.

5. Způsob podle nároku 1, **vyznačený tím**, že dále zahrnuje neustále vyhodnocování rovnic vstupních podmínek, přičemž v případě poruchy zařízení se z množiny procedur vybere odpovídající procedura schopná provést odstranění poruchy.

6. Způsob podle nároku 5, **vyznačený tím**, že dále zahrnuje vyhodnocování všech rovnic souvisejících s procedurou a výběr jednoho kroku prováděné procedury, přičemž se monitorují rovnice související s uvedeným krokem s cílem zjistit, zda procedura pokračuje.

7. Způsob podle nároku 1, **vyznačený tím**, že se dále provede vyhodnocování rovnic vstupních podmínek s cílem zjistit, zda nastala porucha, přičemž se monitorují rovnice vstupních podmínek s cílem vybrat odpovídající proceduru schopnou eliminovat poruchu a monitorování a vyhodnocování rovnic se provádí až do eliminace poruchy.

8. Způsob podle nároku 7, **vyznačený tím**, že se dále provede výběr kroku prováděné procedury, provádí se vyhodnocení všech rovnic všech kroků procedury, monitoruje se výsledek rovnic uvedeného kroku procedury a zjišťuje se, zda došlo k poruše uvedeného kroku, což indikuje, že tento krok nebyl proveden.

9. Způsob podle nároku 1, **v y z n a č e n ý t í m**, že dále zahrnuje vyhodnocení rovnic vstupních podmínek s cílem zjistit počátek poruchy, vyhodnocení rovnic procedury s cílem zjištění poruchy procedury/kroku a vyhodnocení rovnice paralelní informace souběžně s vyhodnocením rovnic procedury s cílem zjistit poruchu související s paralelní informací.

5

10. Způsob podle nároku 1, **v y z n a č e n ý t í m**, že zařízením je jaderná elektrárna, přičemž se

a) z parametrů uložených v relační databázi generují rovnice vstupních podmínek, rovnice kroku procedury a rovnice paralelní informace,

10 b) vyhodnotí se rovnice vstupních podmínek s cílem zjistit počátek poruchy,

c) monitorují se rovnice vstupních podmínek s cílem vybrat z množiny procedur odpovídající proceduru schopnou eliminovat poruchu,

d) vyhodnocení a monitorování vstupních podmínek provádí až do eliminování poruchy,

e) vybere se ten z kroků prováděné procedury, jehož provedení vede k eliminování poruchy,

15 f) provede se vyhodnocení všech rovnic všech kroků procedury,

g) monitoruje se výsledek rovnic kroku procedury odpovídajícího kroku,

h) zjistí se, zda nastala porucha kroku procedury,

i) jestliže se porucha kroku procedury vyskytla, je indikována zprávou,

j) současně s prováděním stupňů e) až i) se vyhodnocují všechny rovnice paralelní informace,

20 k) monitorují se paralelní informační rovnice související s uvedeným krokem,

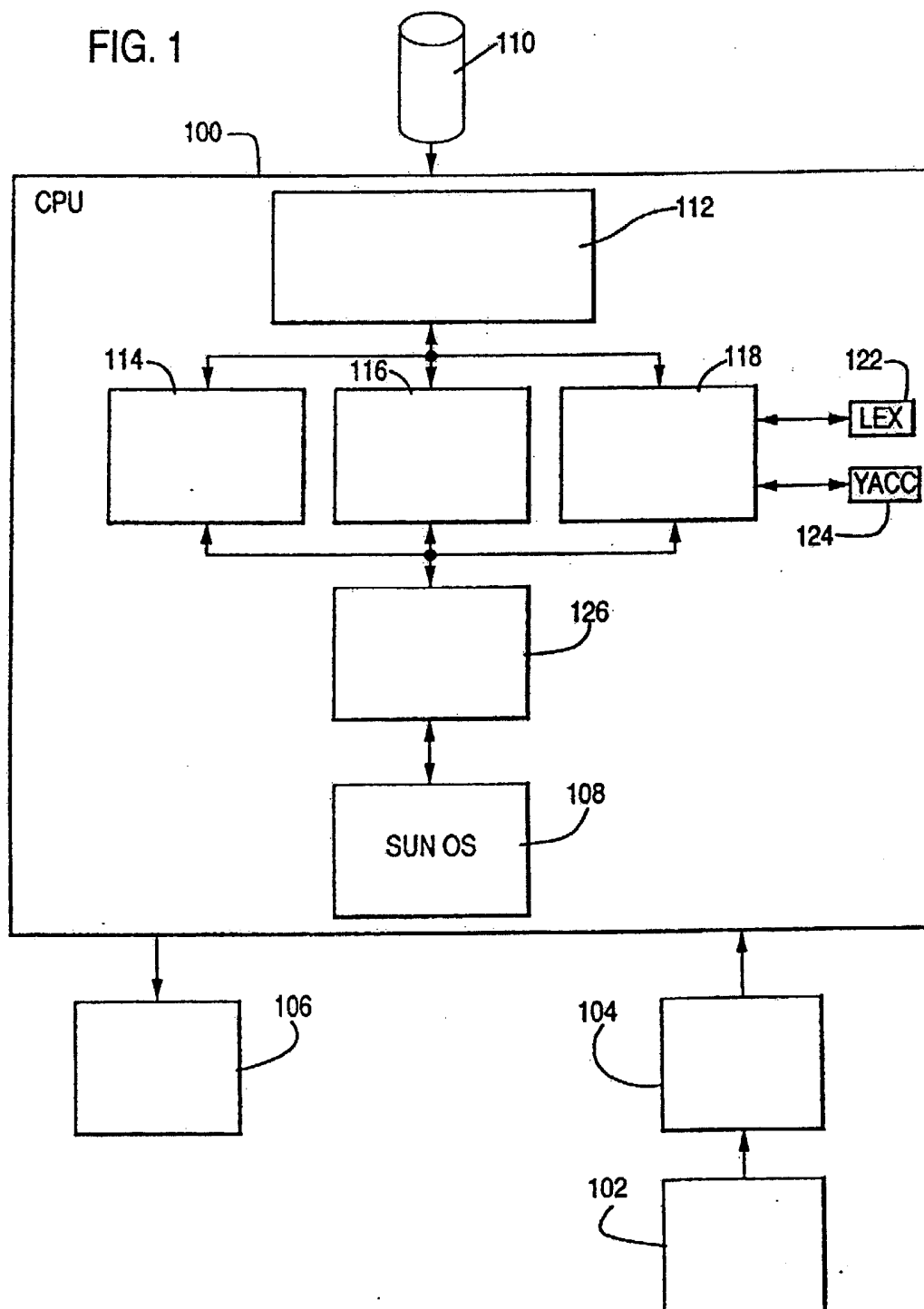
l) v závislosti na výsledku monitorování ve stupni k) se zjistí, zda nastala porucha související s paralelní informací, a

m) jestliže nastala porucha související s paralelní informací, je indikována zprávou o poruše.

25

7 výkresů

FIG. 1





## OBR. 2A

## PROCEDURA EO

1. Zjištění, zda došlo k prasknutí trubky parního generátoru

- |                            |                    |
|----------------------------|--------------------|
| a. pumpa A                 | pracuje            |
| b. úroveň tlaku            | je větší než 50%   |
| c. tlak parního generátoru | je menší než 7 MPa |

## OBR. 6

## PROGRAMOVÝ ZÁSObNÍK 210

## OBR. 5

## VÝSLEDNÉ ROVNICE

EO:1:a:=ABC123=1  
 EO:1:b:=DEF456>50  
 EO:1:c:=GHI789<7

EO:1:=EO:1:a & EO:1:b & EO:1:c

EO:1:a

=

1

ABC123

EO:1:b

&gt;

50

DEF456

EO:1:c

&lt;

7

GHI789

EO:1

&amp;

&amp;

EO:1:c

EO:1:b

EO:1:a

FIG. 2B

|    |    |        |    |      |
|----|----|--------|----|------|
| EO | 1  |        |    |      |
| EO | 1  | a      |    | 1    |
| EO | 1  | b      |    | 2    |
| EO | 1  | c      |    | 3    |
| 1  | 8  | 12     | 16 |      |
| 2  | 10 | 13     | 18 | 50   |
| 3  | 12 | 14     | 19 | 1000 |
| 8  |    | ABC123 |    |      |
| 10 |    | DEF456 |    | 3    |
| 12 |    | GHI789 |    | 6    |
| 12 |    |        |    |      |
| 13 |    |        |    |      |
| 14 |    |        |    |      |
| 16 |    |        |    |      |
| 18 |    |        |    |      |
| 19 |    |        |    |      |
| 3  |    | %      |    |      |
| 6  |    | psia   |    |      |

FIG. 3

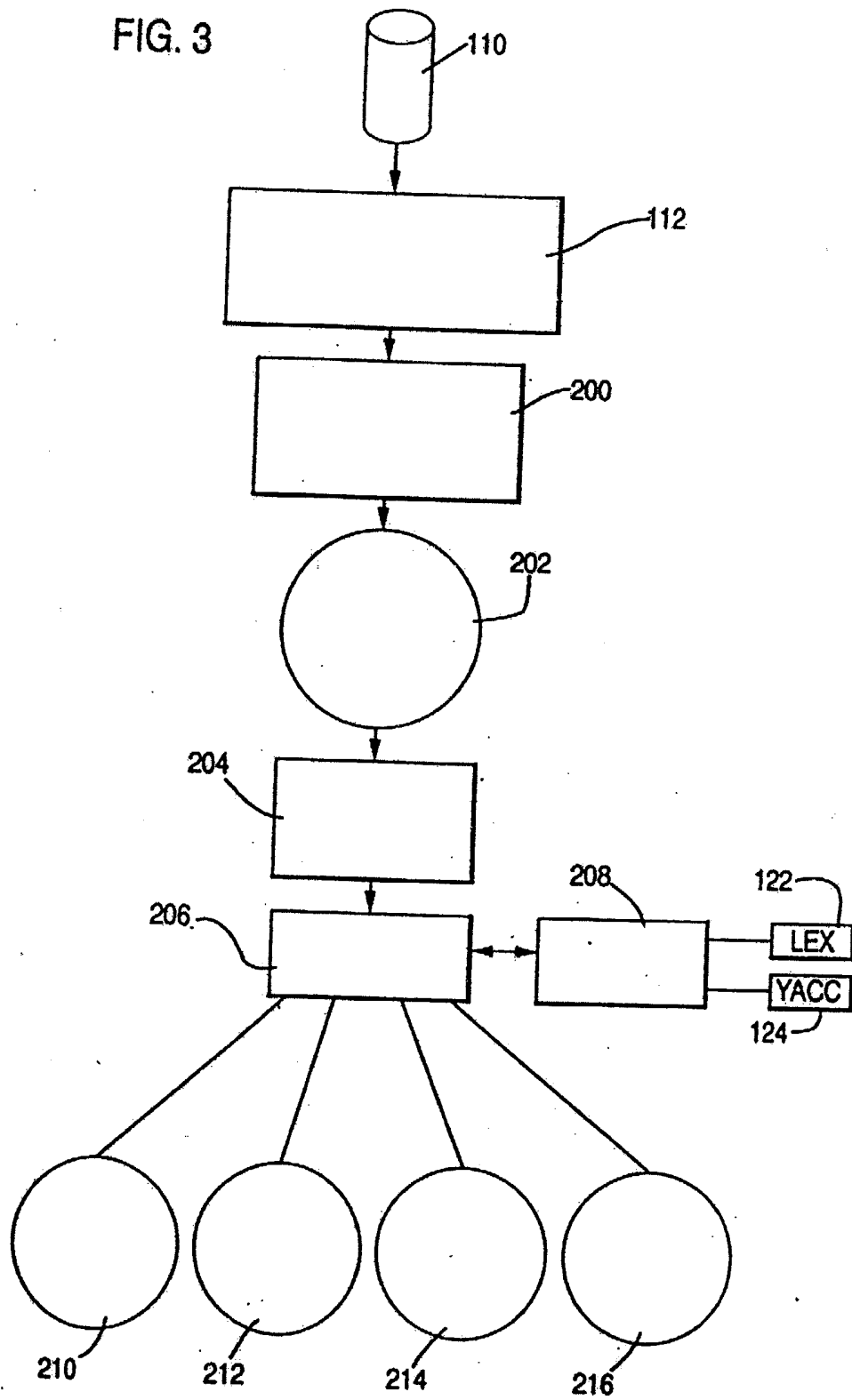


FIG. 4

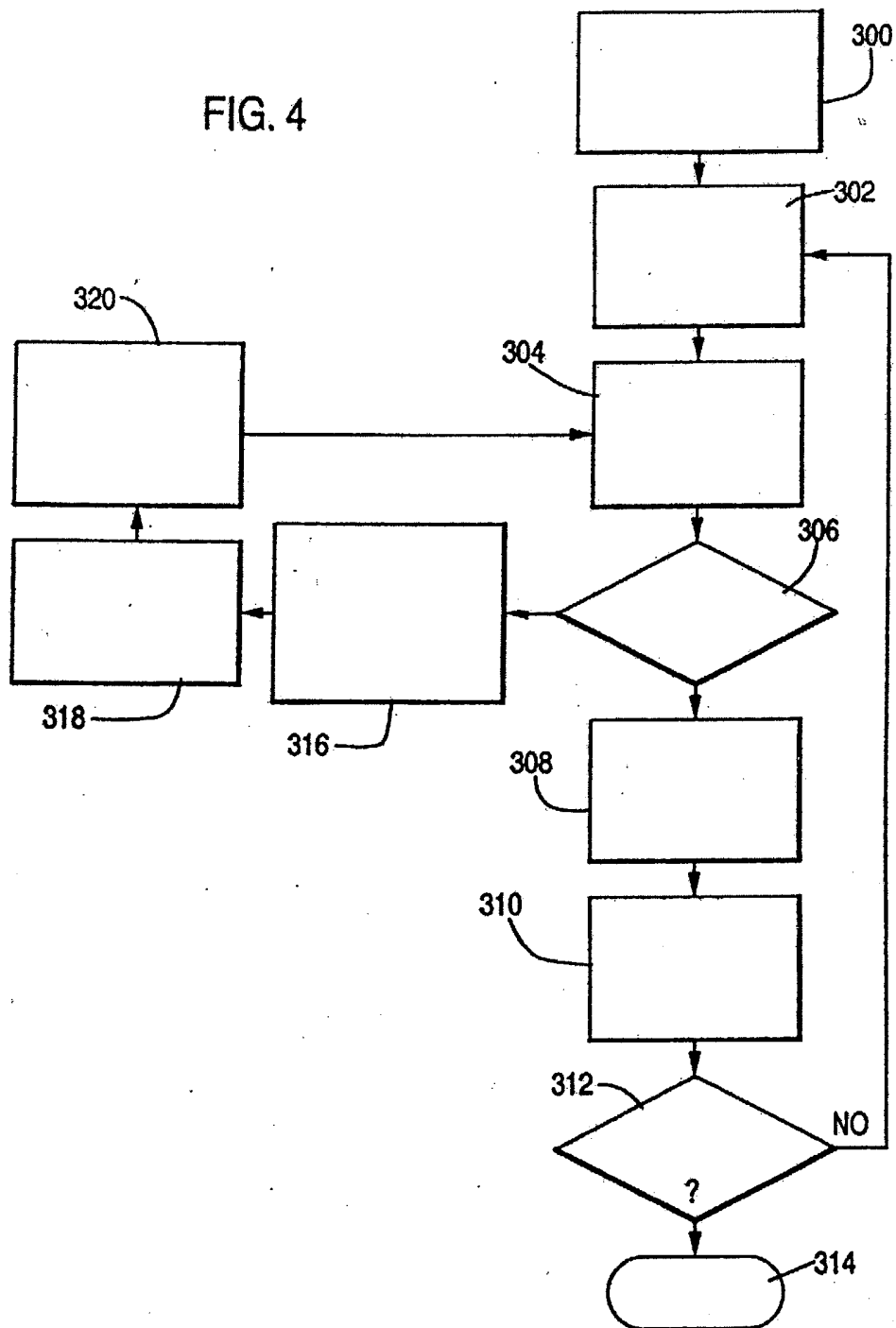


FIG. 7

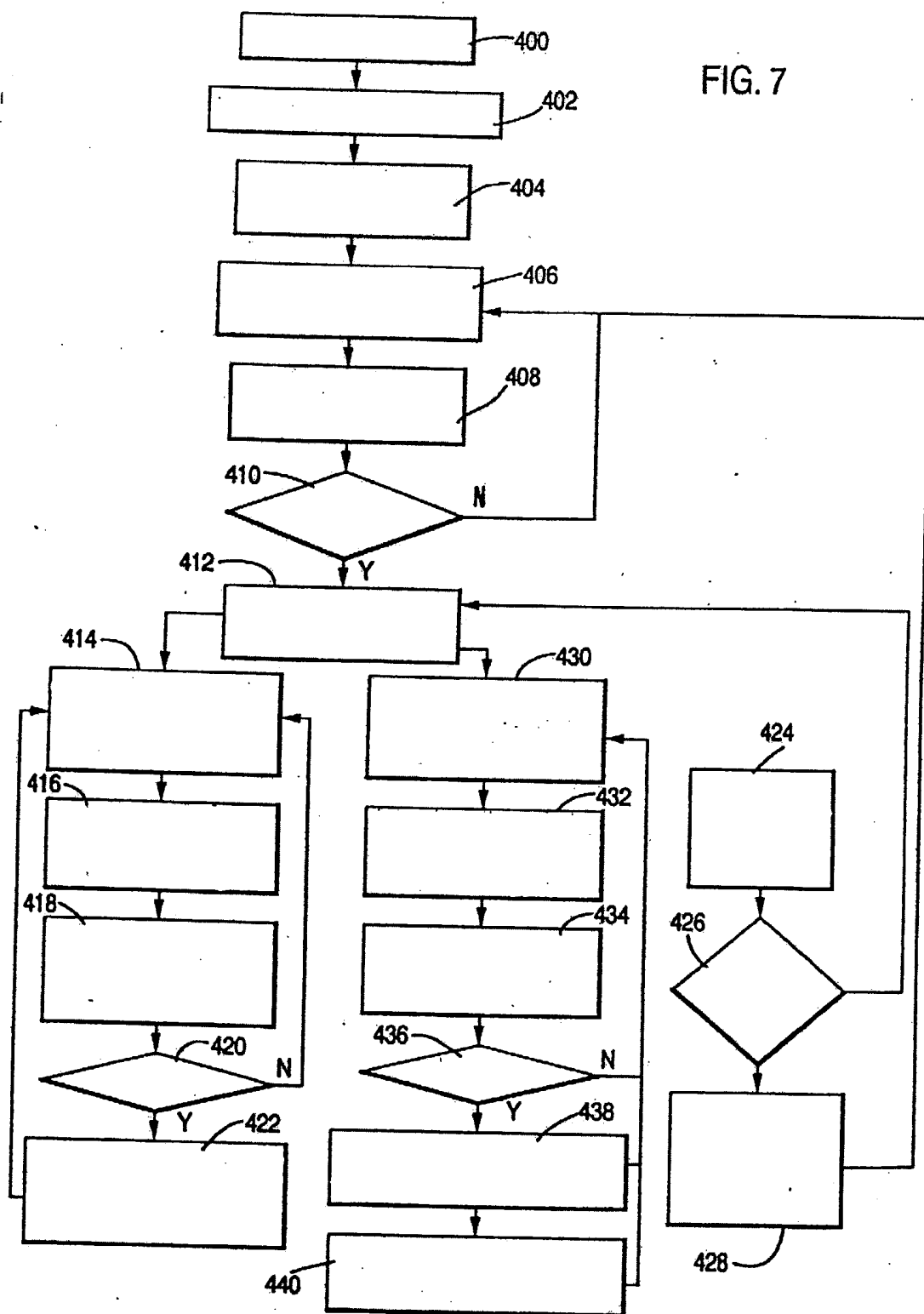
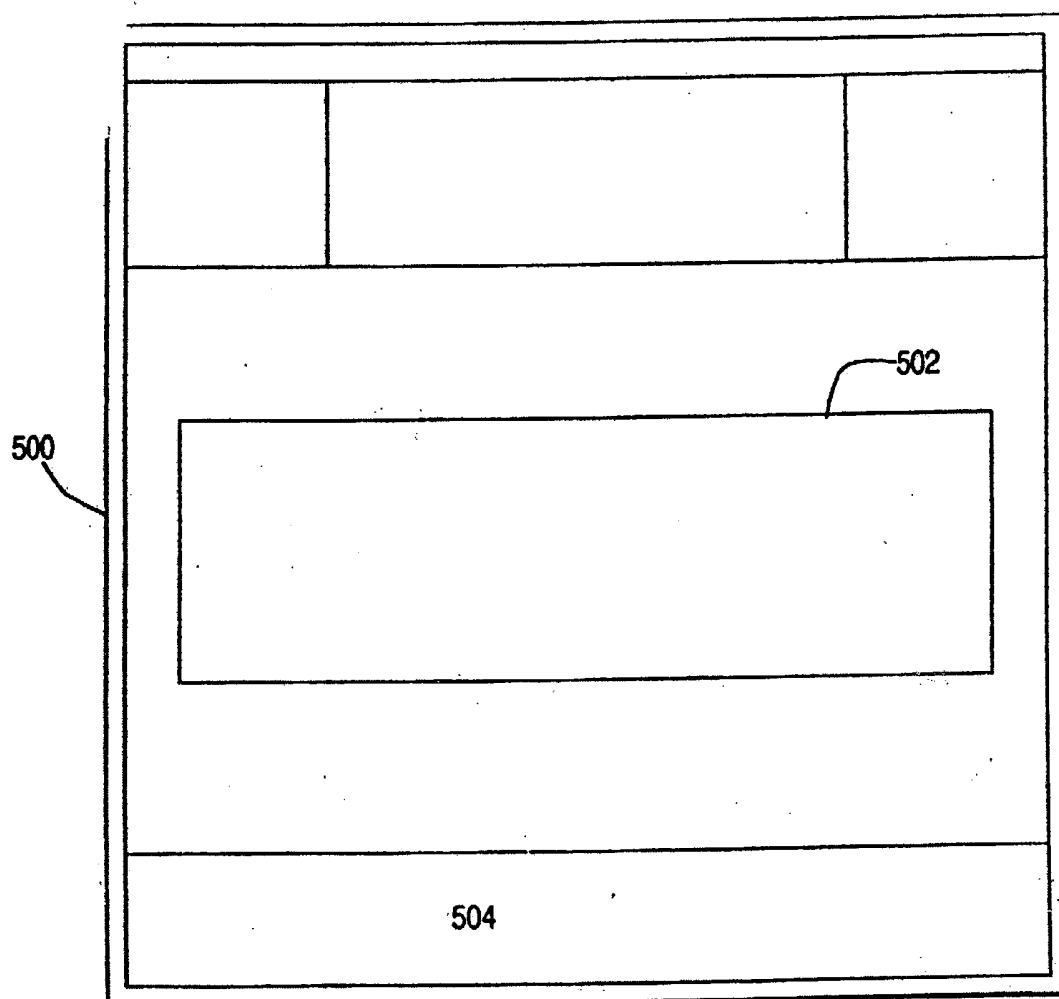


FIG. 8



Konec dokumentu