



(10) **AT 15189 U1 2017-02-15**

(12) **Gebrauchsmusterschrift**

(21) Anmeldenummer: GM 8014/2016 (51) Int. Cl.: **G06F 11/30** (2006.01)
 (22) Anmeldetag: 07.07.2015 **G06F 11/08** (2006.01)
 (24) Beginn der Schutzdauer: 15.12.2016
 (45) Veröffentlicht am: 15.02.2017

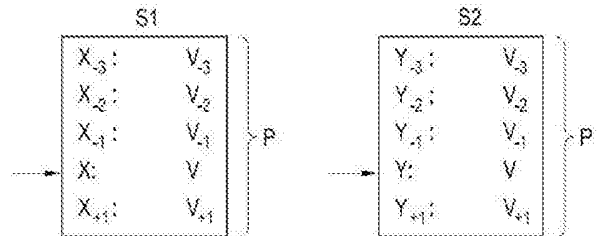
(67) Umwandlung von A 50595/2015	(73) Gebrauchsmusterinhaber: AVL List GmbH 8020 Graz (AT)
(56) Entgegenhaltungen: DE 102013202961 A1 DE 102009010004 A1 US 2004168078 A1 WO 2013160724 A1	(72) Erfinder: Brandl Harald 8020 Graz (AT) Bulatovic Boris 8053 Graz (AT)
	(74) Vertreter: PATENTANWÄLTE PINTER & Weiss OG WIEN (AT)

(54) **Überwachung von Stack-Variablen**

(57) Um Bitfehler in zumindest einem Stack-Speicher (S, S1, S2) eines Getriebe-Steuergeräts sicher zu detektieren, um im Falle der Detektion ein Fehlersignal ausgeben zu können, wird eine Programmlogik zum Ausgeben eines Fehlersignals bei Detektion eines Bitfehlers in zumindest einem Stack-Speicher (S, S1, S2) eines Getriebe-Steuergeräts offenbart, gekennzeichnet durch die folgenden Schritte:

- Erstes Aufrufen der Ablaufvorschrift (P), wobei die zumindest eine Variable (V) in zumindest einen Stack-Speicher (S, S1, S2) an einer ersten Adresse (X) angelegt wird;
- Zweites Aufrufen der Ablaufvorschrift (P), wobei die zumindest eine Variable (V) an einer zweiten Adresse (Y) angelegt wird, die sich von der ersten Adresse (X) unterscheidet;
- Vergleich des Inhalts der ersten Adresse (X) mit dem Inhalt der zweiten Adresse (Y);
- Ausgabe eines Fehlersignals, wenn der Vergleich gemäß Schritt c) einen Unterschied ergibt.

Zudem wird ein Getriebe-Steuergerät mit einer Programmlogik nach offenbart.



Beschreibung

ÜBERWACHUNG VON STACK-VARIABLEN

[0001] Die vorliegende Erfindung betrifft eine Programmlogik zum Ausgeben eines Fehlersignals bei Detektion eines Bitfehlers in zumindest einem Stack-Speicher eines Getriebe-Steuergeräts, wobei bei Abarbeiten einer Ablaufvorschrift durch die Ablaufvorschrift auf zumindest eine Variable in zumindest einem Stack-Speicher zugegriffen wird. In Speichern von Getriebe-Steuergeräten können diverse Bitfehler auftreten. Transiente Bitfehler sind zeitabhängig und damit nicht reproduzierbar. Ursache hierfür können beispielsweise elektrische oder elektromagnetische Störungen darstellen. Permanente Bitfehler hingegen bedeuten, dass ein Bit des Speichers auf einem bestimmten Wert fest hängt. Gerade in sicherheitskritischen Anwendungen, die auf dem Getriebe-Steuergerät abgearbeitet werden, ist es notwendig, sowohl transiente, als auch permanente Bitfehler sicher zu detektieren. Insbesondere permanente Bitfehler können aber schwer detektiert werden und können somit mehrere Ablaufvorschriften oder Funktionen, die auf den fehlerhaften Speicherbereich zugreifen beeinflussen.

[0002] Beim Kompilieren einer Ablaufvorschrift, welches auf einem Getriebe-Steuergerät abgearbeitet werden soll, wird festgelegt, welche von der Ablaufvorschrift zu beeinflussenden Variablen, in auf Getriebe-Steuergeräten integrierten Registern oder im Stack-Speicher abgelegt werden. Der Stack-Speicher ist bekanntermaßen ein Teil des RAM (Random Access Memory) eines Mikroprozessors. Für größere Datenstrukturen, wie z.B. Arrays, wird normalerweise immer ein Stack Speicher verwendet. Lokale, im Stack-Speicher gespeicherte, Variablen werden typischerweise verwendet, um Zwischenergebnisse der Ablaufvorschrift zu berechnen oder aufzunehmen. Die Lebensdauer von Stack-Variablen ist auf die Dauer der Laufzeit der Ablaufvorschrift (globale Variablen) oder einer in der Ablaufvorschrift aufgerufenen Funktion (lokale Variablen) beschränkt. Wenn das Programm oder die Funktion abgearbeitet wurde, wird der den Stack-Variablen zugehörige Stack-Speicher freigegeben und kann von anderen Ablaufvorschriften verwendet werden.

[0003] Da der Stack-Speicher sowohl bei Funktionsaufruf, als auch von der Ablaufvorschrift selbst verwendet wird, müssen permanente Bitfehler in diesem Speicherbereich erkannt werden. Andernfalls würden sich Bitfehler im Stack-Speicher durch die Ablaufvorschrift und in weitere Funktionen fortpflanzen, was zu einem unvorhersehbaren Verhalten des Programms führen kann. In den meisten Standardanwendungen, die auf üblichen Personal Computern laufen, wie beispielsweise E-Mail oder Office-Anwendungen werden üblicherweise, einzelne Bitfehler toleriert, da diese kein sicherheitsrelevantes Risiko darstellen. In sicherheitskritischen Anwendungen können Bitfehler im Stack-Speicher jedoch schwerwiegende Folgen haben, weswegen permanente Bitfehler im Stack-Speicher sicher detektiert werden müssen.

[0004] Eine gängige Lösung, permanente Bit-Fehler in einem Speicherbereich zu detektieren wäre das Durchführen eines sogenannten March-Tests. Bei dieser Art von Tests wird idealerweise jedes Bit des zu testenden Speichers je einmal mit einem Null-Bit und je einmal mit einem Eins-Bit nach einem vorbestimmten Muster überschrieben und ausgelesen. Somit können „hängengebliebene“ Null- oder Eins-Bits detektiert werden. Solche Tests verursachen jedoch Laufzeitverluste und sind zerstörend, d.h. der ursprüngliche Speicherinhalt ist nach dem Testen verloren. Ein Stack-Speicher kann selbstverständlich nicht einfach im laufenden Betrieb gelöscht werden und müsste also zwischengespeichert werden, weshalb ein March-Test insbesondere bei der Abarbeitung von zeitkritischen Anwendungen nachteilig ist.

[0005] Es ist daher die Aufgabe der vorliegenden Erfindung, Bitfehler in einem Stack-Speicher eines Getriebe-Steuergeräts sicher und einfach zu detektieren, um im Falle der Detektion ein Fehlersignal ausgeben zu können.

[0006] Diese Aufgabe wird erfindungsgemäß durch folgende Schritte gelöst:

[0007] a) Erstes Aufrufen der Ablaufvorschrift, wobei die zumindest eine Variable in zumindest einen Stack-Speicher an einer ersten Adresse angelegt wird;

[0008] b) Zweites Aufrufen der Ablaufvorschrift, wobei die zumindest eine Variable in zumindest einen Stack-Speicher an einer zweiten Adresse angelegt wird, die sich von der ersten Adresse unterscheidet;

[0009] c) Vergleich des Inhalts der ersten Adresse des zumindest einen Stack-Speichers nach dem Anlegen der zumindest einen Variable an der ersten Adresse mit dem Inhalt der zweiten Adresse des zumindest einen Stack-Speichers nach dem zweiten Anlegen der zumindest einen Variable an der zweiten Adresse;

[0010] d) Ausgabe eines Fehlersignals, wenn der Vergleich gemäß Schritt c) einen Unterschied ergibt.

[0011] Die Aufgabe wird zudem durch ein Getriebe-Steuergerät mit einer Programmlogik nach den oben beschriebenen Schritten gelöst.

[0012] Durch dieses Verfahren wird es ermöglicht selbst permanente Bitfehler in zumindest einem Stack-Speicher des Getriebe-Steuergeräts zu detektieren, indem die Ablaufvorschrift zwei Mal aufgerufen wird. Beim ersten Aufruf der Ablaufvorschrift wird zumindest eine Variable an einer ersten Adresse angelegt, beim zweiten Aufruf der Ablaufvorschrift wird die zumindest eine Variable an einer zweiten Adresse angelegt. Da die Speicherinhalte des zumindest einen Stack-Speichers nach Beendigung der jeweiligen Programminstanzen freigegeben werden, müssen selbstverständlich die Inhalte der ersten und der zweiten Adresse nach Anlegen der jeweiligen Variablen und vor Beenden der jeweiligen Programminstanzen verglichen werden.

[0013] Eine Ablaufvorschrift wird ein erstes Mal aufgerufen, wobei die zumindest eine Variable im zumindest einen Stack-Speicher an einer ersten Adresse angelegt wird, die Ablaufvorschrift ein zweites Mal aufgerufen wird, wobei die zumindest eine Variable im zumindest einen Stack-Speicher an einer zweiten Adresse, die sich von der ersten Adresse unterscheidet, angelegt wird. Ein Bitfehler wird dann detektiert, wenn sich der Inhalt der ersten Adresse des zumindest einen Stack-Speichers nach dem Anlegen der zumindest einen Variable an der ersten Adresse vom Inhalt der zweiten Adresse des zumindest einen Stack-Speichers nach dem zweiten Anlegen der zumindest einen Variable an der zweiten Adresse unterscheidet.

[0014] Der erste Aufruf der Ablaufvorschrift und der zweite Aufruf der Ablaufvorschrift kann sowohl parallel, als auch sequentiell erfolgen.

[0015] Falls die Aufrufe der Ablaufvorschrift sequentiell erfolgen, kann nach dem ersten Aufruf der Ablaufvorschrift eine Hilfsprozedur aufgerufen werden, aus welcher der zweite Aufruf der Ablaufvorschrift erfolgt. Damit kann sichergestellt werden, dass sich die Adresse der zumindest einen Variable nach dem ersten Aufruf der Ablaufvorschrift von der Adresse der zumindest einen Variable nach dem zweiten Aufruf der Ablaufvorschrift unterscheidet. Dies wird in dieser Verfahrensvariation dadurch erreicht, dass für die Hilfsprozedur selbst der Stack-Speicher an der nächstfreien Adresse verwendet wird. Falls somit die nächstfreie Adresse der durch das Beenden des ersten Aufrufs der Ablaufvorschrift soeben freigegebenen Adresse entspricht, wird diese Adresse von der Hilfsprozedur belegt und eine andere Adresse nach dem zweiten Aufruf der Ablaufvorschrift für das Anlegen der zumindest einen Variablen verwendet.

[0016] Vorteilhafterweise werden die Inhalte der ersten Adresse und der zweiten Adresse der zumindest einen Variablen nach deren Anlegen erfasst und zwischengespeichert.

[0017] Dies ermöglicht ein Vergleichen der Inhalte der ersten Adresse und der zweiten Adresse vor allem bei sequentiellen Programmaufrufen, da die Inhalte der unterschiedlichen Adressen zu unterschiedlichen Zeitpunkten erfasst werden müssen. Wie oben beschrieben wird der verwendete Stack-Speicher nach Beenden eines Programms oder einer Funktion freigegeben. Bei sequentiellen Aufrufen der Ablaufvorschrift ist es daher von Vorteil vor allem den Inhalt der Adresse der zumindest einen Variablen vor Beenden des ersten Ablaufvorschriftaufrufes zu erfassen und zwischenzuspeichern.

[0018] Besonders vorteilhaft ist es nach Anlegen der zumindest einen Variable an der ersten Adresse eine Checksumme über einen die erste Adresse beinhaltenden Teil des zumindest

einen Stack-Speichers gebildet wird und nach dem Anlegen der zumindest einen Variable an der zweiten Adresse eine Checksumme über einen die zweite Adresse beinhaltenden Teil des zumindest einen Stack-Speichers gebildet wird woraufhin ein Vergleich der Inhalte der Adressen über den Vergleich der Checksummen erfolgt.

[0019] Durch Bildung der Checksummen kann Speicherplatz gespart werden, da im Falle eines direkten Speichern der jeweiligen Inhalte des zumindest einen Stack-Speichers die gesamten betreffenden Speicherbereiche gespiegelt werden müssten. Auch in dieser Variation ist es selbstverständlich, dass die Inhalte der jeweiligen Stack-Speicherbereiche vor dem jeweiligen Beenden der Programmaufrufe erfasst werden müssen.

[0020] Die Aufrufe der Ablaufvorschrift können durch ein Überwachungsprogramm erfolgen, was es ermöglicht bestimmte Variablen gezielt zu überwachen. Das Überwachungsprogramm ruft beispielsweise die Ablaufvorschrift auf und steuert die Abarbeitungen, Speicherzugriffe, Checksummenbildung usw. Sofern eine Hilfsprozedur verwendet wird, kann auch der Aufruf dieser Hilfsprozedur durch das Überwachungsprogramm erfolgen.

[0021] Falls ein Bitfehler detektiert wird, kann eine dadurch getriggerte Aktion durchgeführt werden.

[0022] Als Aktion kann ein Fehlersignal ausgegeben werden und/oder ein Interrupt oder ein Reset- ausgelöst werden, da das Weiterarbeiten mit ungültigem Speicher zu unvorhersehbaren Aktionen führen kann.

[0023] Die gegenständliche Erfindung wird nachfolgend unter Bezugnahme auf die Fig. 1 und 2 näher erläutert, die beispielhaft, schematisch und nicht einschränkend vorteilhafte Ausgestaltungen der Erfindung zeigen. Dabei zeigt

[0024] Fig. 1 Eine erste Ausgestaltung mit parallelen Aufrufen der Ablaufvorschrift P

[0025] Fig. 2 Eine zweite Ausgestaltung mit sequentiellen Aufrufen der Ablaufvorschrift P

[0026] Wie in den Fig. 1 und 2 ersichtlich ist, wird eine Ablaufvorschrift P ein erstes Mal aufgerufen und dabei die zumindest eine Variable V im zumindest einen Stack-Speicher S, S1, S2 an einer ersten Adresse X angelegt und die Ablaufvorschrift P ein zweites Mal aufgerufen, wobei die zumindest eine Variable V im zumindest einen Stack-Speicher S, S1, S2 an einer zweiten Adresse Y, die sich von der ersten Adresse X unterscheidet, angelegt. Ein Bitfehler wird detektiert, wenn sich der Inhalt der ersten Adresse X des zumindest einen Stack-Speichers S, S1, S2 nach dem Anlegen der zumindest einen Variable V an der ersten Adresse X vom Inhalt der zweiten Adresse Y des zumindest einen Stack-Speichers S, S1, S2 nach dem zweiten Anlegen der zumindest einen Variable V an der zweiten Adresse Y unterscheidet.

[0027] Der erste Aufruf der Ablaufvorschrift P und der zweite Aufruf der Ablaufvorschrift P können sowohl parallel als auch sequentiell erfolgen. In Folge soll das Verfahren sowohl in Form einer Ausgestaltung mit parallelen, als auch einer Ausgestaltung mit sequentiellen Aufrufen beschrieben werden.

[0028] Eine Ausgestaltung mit parallelen Aufrufen der Ablaufvorschrift P ist in Fig. 1 dargestellt. Bei Verwendung von zwei Getriebe-Steuergeräten ist in der Regel jedem Getriebe-Steuergerät ein Stack-Speicher S1, S2 zugeordnet. Obwohl in der Fig. 2 zwei Stack-Speicher S1, S2 dargestellt sind, funktioniert das Verfahren auch bei Verwendung von zwei Getriebe-Steuergeräten mit einem geteilten Stack-Speicher S. Es wird erfindungsgemäß nach dem ersten Aufruf der Ablaufvorschrift P zumindest eine Variable V im ersten Stack-Speicher S1 an der Adresse X angelegt. Weitere mögliche Variablen V_{-3} bis V_{+1} sind in der Fig. 1 an den Adressen X_{-3} bis X_{+1} im ersten Stack-Speicher S1 dargestellt. Nach dem parallel erfolgten zweiten Aufruf der Ablaufvorschrift P wird die zumindest eine Variable V im zweiten Stack-Speicher S2 an der Adresse Y angelegt. Weitere mögliche Variablen V_{-3} bis V_{+1} befinden sich nun auch an den Adressen Y_{-3} bis Y_{+1} im zweiten Stack-Speicher S2. Die Adresse X befindet sich laut Fig. 1 somit in einem ersten Stack-Speicher S1, die Adresse Y befindet sich normalerweise in einem zweiten Stack-Speicher S2. Wie erwähnt ist auch die Verwendung eines geteilten Stack-Speichers S möglich.

Entscheidend ist, dass sich die Adressen X und Y unterscheiden, was bei einem parallelen Aufruf der Ablaufvorschrift P immer gewährleistet sein muss. Wenn sich der Inhalt der ersten Adresse X nach dem ersten Anlegen der zumindest einen Variablen V vom Inhalt der zweiten Adresse Y nach dem zweiten Anlegen der zumindest einen Variable V voneinander unterscheiden, so wird ein Bitfehler detektiert. Dabei müssen die Inhalte der Adressen X, Y selbstverständlich immer vor dem Beenden der Instanz der jeweiligen Ablaufvorschrift P betrachtet werden, da nach dem Beenden der Instanzen der Ablaufvorschrift P der zumindest eine Stack-Speicher S, S1, S2 an den Adressen X und Adressen X_{-3} bis X_{+1} bzw. Y und Y_{-3} bis Y_{+1} wieder freigegeben wird und damit die Inhalte verloren gehen.

[0029] Die zumindest eine Variable V wird in den Fig. 1 und 2 beispielhaft betrachtet. Selbstverständlich kann jede andere Variable V_{-3} bis V_{+1} in Fig. 1 analog an den zugehörigen Adressen X_{-3} bis X_{+1} bzw. Y_{-3} bis Y_{+1} , oder in Fig. 2 an den Adressen U bis Y, bzw. V bis Z, auf Bitfehler untersucht werden.

[0030] In der Fig. 2 ist eine Ausgestaltung mit einer sequentiellen Abarbeitung der Aufrufe der Ablaufvorschrift P dargestellt. Hierbei wird analog zur Ausgestaltung der parallelen Abarbeitungen der Ablaufvorschrift P die zumindest eine Variable V am Stack-Speicher S, S1, S2 an der ersten Adresse X angelegt. Weitere mögliche Variablen V_{-3} bis V_{+1} sind in der Fig. 1 an den Adressen U bis Y im zumindest einen Stack-Speicher S, S1, S2 dargestellt. Der zweite Aufruf der Ablaufvorschrift P erfolgt jedoch erst nach dem Beenden der ersten Instanz der Ablaufvorschrift, also direkt im Anschluss oder zu einem späteren Zeitpunkt. Dabei wird bei diesem zweiten Aufruf der Ablaufvorschrift P die zumindest eine Variable V am zumindest einen Stack-Speicher S, S1, S2 an der zweiten Adresse Y angelegt. Dies kann erreicht werden, indem ein Offset erzeugt wird, wie z.B. durch einen indirekten zweiten Aufruf der Ablaufvorschrift. Wenn sich der Inhalt an der Adresse X nach dem ersten Anlegen der Variable (und vor Beenden der ersten Instanz der Ablaufvorschrift P) vom Inhalt an der Adresse Y nach dem zweiten Anlegen der zumindest einen Variable V (ebenso vor Beenden der ersten Instanz der Ablaufvorschrift P) unterscheiden, wird ein Bitfehler detektiert. Auch in dieser Ausführung ist entscheidend, dass sich die Adressen X und Y unterscheiden.

[0031] Um sicher zu stellen, dass sich die Adressen X und Y unterscheiden, erfolgt der zweite Aufruf der Ablaufvorschrift P erfindungsgemäß nach einer Verschiebung der als nächstes zu belegenden Adressen des zumindest einen Stack-Speichers S, S1, S2 von den Adressen U, V, W, X, Y auf die Adressen V, W, X, Y, Z. Dies kann vorteilhafterweise erreicht werden, indem der zweite Aufruf der Ablaufvorschrift aus einer Hilfsprozedur H erfolgt, wie in Fig. 1 dargestellt. Nach Beenden des ersten Aufrufs der Ablaufvorschrift P wird somit die Hilfsprozedur H aufgerufen und belegt den zumindest einen Stack-Speicher S, S1, S2 an der Adresse U. Dies ist der Fall, da durch den Aufruf der Hilfsprozedur H am Stack bekanntermaßen ebenfalls Einträge gemacht werden, wie beispielsweise das Zwischenspeichern von Registern zum Wiederherstellen nach dem Rücksprung aus der Hilfsprozedur H. Aus der Hilfsprozedur H wird nun die Ablaufvorschrift P ein zweites Mal aufgerufen, wobei sich durch das Belegen des zumindest einen Stack-Speichers S durch die Hilfsprozedur H an der Adresse U die weiteren Variablen V_{-3} bis V_{+1} und damit auch die zumindest eine Variable V verschieben. Somit ist sichergestellt, dass sich die Speicherorte der zumindest einen Variablen, also die Adressen X und Y unterscheiden. Nach dem Beenden des ersten Aufrufs der Ablaufvorschrift P wird die Adresse X des Stack-Speichers wieder freigegeben. Sofern weitere Voraussetzungen erfüllt sind, wie beispielsweise, dass zwischen den Aufrufen des Programms P keine Interrupts ausgelöst wurden, könnte ohne Verwendung der Hilfsprozedur H und die damit einhergehende Belegung der Adresse U gegebenenfalls die identen Adressen U bis Y am Stack-Speicher S, S1, S2 für die weiteren Variablen V_{-3} bis V_{+1} und vor Allem die Adresse X für die zumindest eine Variable V verwendet werden. Durch Verwendung der Hilfsprozedur H verschiebt sich die Adresse der zumindest einen Variablen V am Stack auf die Adresse Y und es ist sichergestellt, dass sich die Adressen X und Y unterscheiden. Die Adressen U, V, W, Y der weiteren Variablen V_{-3} bis V_{+1} verschieben sich durch das Belegen der Adresse U durch die Hilfsprozedur H ebenso auf die Adressen V, W, X, Z.

[0032] Um die Inhalte der Adressen X und Y nach Anlegen der zumindest einen Variablen V und vor dem Beenden der einzelnen Instanzen der Ablaufvorschrift P vergleichen zu können, können diese vorteilhafterweise global zwischengespeichert werden.

[0033] Nach Anlegen der zumindest einen Variable V an der ersten Adresse X kann eine Checksumme CSX über einen die erste Adresse X beinhaltenden Teil des zumindest einen Stack-Speichers S, S1, S2 gebildet werden. Analog kann nach der Änderung der zumindest einen Variable V an der zweiten Adresse Y eine Checksumme CSY über einen die zweite Adresse Y beinhaltenden Teil des zumindest einen Stack-Speichers S, S1, S2 gebildet werden. Ein Vergleich der Inhalte der Adressen X, Y kann dann über den Vergleich der Checksummen CSX, CSY erfolgen. Hierbei ist für das Speichern der Checksummen CSX, CSY weniger Speicherplatz notwendig, als wenn die Inhalte der Adressen selbst gespeichert werden müssen.

[0034] Vorteilhafterweise werden die Aufrufe der Ablaufvorschrift P durch ein Überwachungsprogramm gehandhabt. Dies ermöglicht es die zumindest eine Variable V des zumindest einen Stack-Speichers S, S1, S2 gezielt zu überwachen.

[0035] Das Überwachungsprogramm kann gegebenenfalls auch die Aufrufe der Hilfsprozedur H verwalten, sofern diese im Rahmen des Verfahrens verwendet wird.

[0036] Bei Detektion eines Bitfehlers kann durch das Überwachungsprogramm vorteilhafterweise eine Aktion ausgelöst werden.

[0037] Diese Aktion kann die Ausgabe eines Fehlersignals, beispielsweise in Form eines Flags, oder eines optischen oder akustischen Signals darstellen.

[0038] Auch das Auslösen eines Interrupts oder eines Resets als Aktion ist möglich, woraufhin beispielsweise ein laufender Prozess abgebrochen wird, um zu verhindern, dass unvorhersehbare Programmzustände eintreten können und dadurch ein Sicherheitsrisiko entsteht.

Ansprüche

1. Programmlogik zum Ausgeben eines Fehlersignals bei Detektion eines Bitfehlers in zumindest einem Stack-Speicher (S, S1, S2) eines Getriebe-Steuergeräts, wobei bei Abarbeiten einer Ablaufvorschrift (P) durch die Ablaufvorschrift (P) auf zumindest eine Variable (V) in zumindest einem Stack-Speicher (S, S1, S2) zugegriffen wird, **gekennzeichnet durch** die folgenden Schritte:
 - a) Erstes Aufrufen der Ablaufvorschrift (P), wobei die zumindest eine Variable (V) in zumindest einen Stack-Speicher (S, S1, S2) an einer ersten Adresse (X) angelegt wird;
 - b) Zweites Aufrufen der Ablaufvorschrift (P), wobei die zumindest eine Variable (V) in zumindest einen Stack-Speicher (S, S1, S2) an einer zweiten Adresse (Y) angelegt wird, die sich von der ersten Adresse (X) unterscheidet;
 - c) Vergleich des Inhalts der ersten Adresse (X) des zumindest einen Stack-Speichers (S, S1, S2) nach dem Anlegen der zumindest einen Variable (V) an der ersten Adresse (X) mit dem Inhalt der zweiten Adresse (Y) des zumindest einen Stack-Speichers (S, S1, S2) nach dem zweiten Anlegen der zumindest einen Variable (V) an der zweiten Adresse (Y);
 - d) Ausgabe eines Fehlersignals, wenn der Vergleich gemäß Schritt c) einen Unterschied ergibt.
2. Getriebe-Steuergerät mit einer Programmlogik nach Anspruch 1.

Hierzu 1 Blatt Zeichnungen

1/1

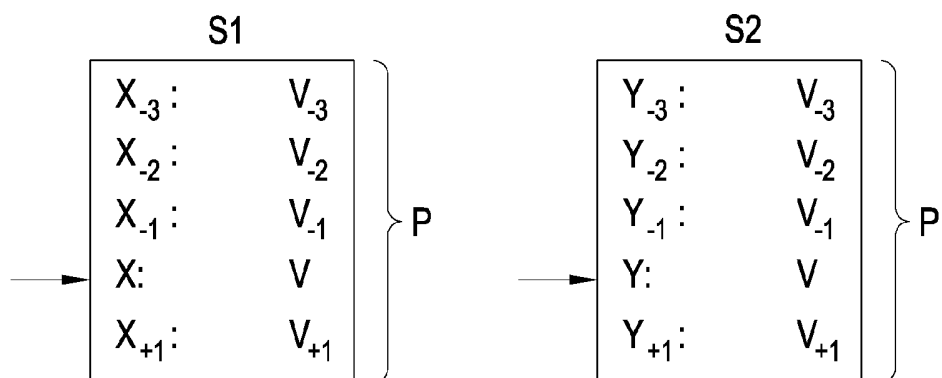


Fig. 1

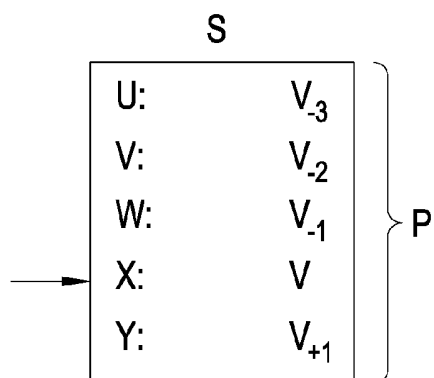
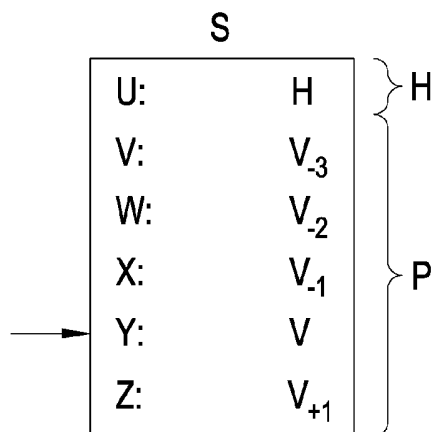


Fig. 2



Klassifikation des Anmeldungsgegenstands gemäß IPC:
G06F 11/30 (2006.01); **G06F 11/08** (2006.01)

Klassifikation des Anmeldungsgegenstands gemäß CPC:
G06F 11/3037 (2013.01); **G06F 11/08** (2013.01)

Recherchierte Prüfzettel (Klassifikation):
G06F, H04L

Konsultierte Online-Datenbank:
EPODOC, WPI, TXTDE, TXTEN, INSPEC, NPL

Dieser Recherchenbericht wurde zu den am **13.09.2016** eingereichten Ansprüchen **1-2** erstellt.

Kategorie ¹⁾	Bezeichnung der Veröffentlichung: Ländercode, Veröffentlichungsnummer, Dokumentart (Anmelder), Veröffentlichungsdatum, Textstelle oder Figur soweit erforderlich	Betreffend Anspruch
X	DE 102013202961 A1 (BOSCH GMBH ROBERT) 28. August 2014 (28.08.2014) Zusammenfassung; Figur 1; Beschreibung der Figur; Ansprüche 1-10;	1-2
X	DE 102009010004 A1 (CONTINENTAL AUTOMOTIVE GMBH) 26. August 2010 (26.08.2010) Zusammenfassung; Figuren 1-3; Beschreibung der Figuren; Ansprüche 1-9;	1-2
X	US 2004168078 A1 (BRODLEY ET AL.) 26. August 2004 (26.08.2004) Zusammenfassung; Figuren 4, 6A-6H; Beschreibung der Figuren; Ansprüche 1-20;	1-2
X	WO 2013160724 A1 (FREESCALE SEMICONDUCTOR INC) 31. Oktober 2013 (31.10.2013) Zusammenfassung; Figuren 1, 3, 6; Beschreibung der Figuren; Ansprüche 1-19;	1-2

Datum der Beendigung der Recherche:
12.10.2016

Seite 1 von 1

Prüfer(in):

STOLL Judith

¹⁾ **Kategorien** der angeführten Dokumente:

- X** Veröffentlichung **von besonderer Bedeutung**: der Anmel-
dungsgegenstand kann allein aufgrund dieser Druckschrift nicht als neu bzw. auf
erfinderischer Tätigkeit beruhend betrachtet werden.
- Y** Veröffentlichung **von Bedeutung**: der Anmeldungsgegenstand kann nicht
als auf erfinderischer Tätigkeit beruhend betrachtet werden, wenn die
Veröffentlichung mit einer oder mehreren weiteren Veröffentlichungen
dieser Kategorie in Verbindung gebracht wird und diese **Verbindung für
einen Fachmann naheliegend** ist.

- A** Veröffentlichung, die den allgemeinen **Stand der Technik** definiert.
- P** Dokument, das von **Bedeutung** ist (Kategorien **X** oder **Y**), jedoch **nach
dem Prioritätstag** der Anmeldung veröffentlicht wurde.
- E** Dokument, das **von besonderer Bedeutung** ist (Kategorie **X**), aus dem
ein „**älteres Recht**“ hervorgehen könnte (früheres Anmeldedatum, jedoch
nachveröffentlicht, Schutz ist in Österreich möglich, würde Neuheit in Frage
stellen).
- &** Veröffentlichung, die Mitglied der selben **Patentfamilie** ist.