

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2004-508620

(P2004-508620A)

(43) 公表日 平成16年3月18日(2004.3.18)

(51) Int.Cl. ⁷	F I	テーマコード (参考)
G06F 12/00	G06F 12/00 594	5B035
G06F 9/50	G06K 19/00 N	5B060
G06K 19/07	G06F 9/06 640H	5B076

審査請求 未請求 予備審査請求 有 (全 45 頁)

(21) 出願番号	特願2002-524095 (P2002-524095)	(71) 出願人	596007511
(86) (22) 出願日	平成13年8月21日 (2001.8.21)		ギーゼッケ ウント デフリエント ゲー
(85) 翻訳文提出日	平成15年2月24日 (2003.2.24)		エムペーハー
(86) 国際出願番号	PCT/EP2001/009664		Giesecke & Devrient
(87) 国際公開番号	W02002/019264		GmbH
(87) 国際公開日	平成14年3月7日 (2002.3.7)		ドイツ連邦共和国 D-81677 ミュ
(31) 優先権主張番号	100 40 974.1		ンヘン プリンツレーゲンテンシュトラッ
(32) 優先日	平成12年8月22日 (2000.8.22)		セ 159
(33) 優先権主張国	ドイツ (DE)	(74) 代理人	100073184
			弁理士 柳田 征史
		(74) 代理人	100090468
			弁理士 佐久間 剛
		(72) 発明者	ツィージンガー, ダーニエル
			ドイツ連邦共和国 80637 ミュンヘ
			ン リリュエブラウン-ヴェーク 15
			最終頁に続く

(54) 【発明の名称】 携帯型データ記憶媒体のスタックの仮想拡張方法

(57) 【要約】

実行可能なプログラムコードの再ロードを可能にする、携帯型データ記憶媒体（10）のスタックの仮想拡張のための方法が提案される。物理的に存在するスタックより大きなスタックを必要とするプログラムの実行を可能にするため、スタック（26）は少なくとも2つのスタックセグメント（30）にセグメント化され、セグメント（30）の占有状態が監視される。セグメント（30）の完全な占有が認められると、最も古いスタックセグメント（30）が別の記憶媒体（18）にスワップ書出しされる。別の記憶媒体（18）は、スタック（26）に比較して低速のメモリとすることができる。スタック（26）で再び十分なメモリ空間が利用可能になると、スワップ書出しされたセグメント（30）が戻される。

【特許請求の範囲】

【請求項 1】

スタック（26）として構成された携帯型データ記憶媒体上の揮発性読出—書込メモリ（16）のデータ構造の仮想拡張のための：

b）前記スタック（26）の占有状態を表す占有パラメータが検出されるステップ；及び

c）前記ステップb）で検出された前記占有パラメータをあらかじめ定められた極限值（GW）と比較することにより前記占有パラメータが監視されるステップ；

d）極限值（GW）と前記占有パラメータとの間のあらかじめ定められた関係が達成されると直ちに、前記データ構造の一部が別の記憶媒体（18, 20）にスワップ書出しされるステップ；

を含む方法において：

a）前記スタック（26）が少なくとも2つのスタックセグメント（30）にセグメント化されるステップを含み；

前記別の記憶媒体（18, 20）へのスワップがセグメント毎に実施されることを特徴とする方法。

【請求項 2】

前記別の記憶媒体が読出—書込メモリ、特に不揮発性読出—書込メモリ（18）であることを特徴とする請求項1に記載の方法。

【請求項 3】

前記方法が：

e）前記占有パラメータと第2の極限值との間のあらかじめ定められた関係が達成されると直ちに、特に前記占有パラメータが前記第2の極限值より小さくなると直ちに、前記ステップd）においてスワップ書出しされた前記スタックセグメント（30）を前記スタック（26）にスワップするステップ；

をさらに含むことを特徴とする請求項1または2に記載の方法。

【請求項 4】

前記方法が：

f）前記不揮発性読出—書込メモリ（18）にスワップ書出しされた前記スタックセグメント（30）の宛先アドレスを、好ましくは前記揮発性読出—書込メモリ（16）のレジスタにおいて、管理するステップ；前記宛先アドレスは、スタックセグメント（30）が新しくスワップ書出しされる毎にセグメントサイズの整数倍だけインクリメントされ、スタックセグメント（30）が前記スタック（26）にスワップ書戻しされる毎にセグメントサイズの整数倍だけデクリメントされる；

をさらに含むことを特徴とする請求項1から3のいずれかに記載の方法。

【請求項 5】

サイズが 2^{n-m} のセグメントへの前記セグメント化が実施され、ここで $m = 1, 2, 3, \dots, 2^m =$ セグメント化数、及び $2^n =$ 前記スタック（26）のサイズであり、前記データ記憶媒体のシステムの最適均衡化が得られるように m が選ばれることを特徴とする請求項1から4のいずれかに記載の方法。

【請求項 6】

前記極限值（GW）及び／または前記第2の極限值が、前記セグメントサイズ、すなわち 2^{n-m} に、あるいは前記セグメントサイズの整数倍に一致することを特徴とする請求項5に記載の方法。

【請求項 7】

前記ステップd）における前記極限值（GW）と前記占有パラメータとの間の前記関係が、関係式“占有パラメータ $\leq$ 極限值”で表されることを特徴とする請求項1から6のいずれかに記載の方法。

【請求項 8】

前記スマートカード（10）が実行可能なプログラムコードの再ロードを可能にするオペレーティングシステムを有することを特徴とする請求項1から7のいずれかに記載の方法

10

20

30

40

50

。

【請求項 9】

スタックセグメント (30) のスワップ書出し及び／またはスワップ書込みが既知のブロック毎転送命令を用いて実施されることを特徴とする請求項 1 から 8 のいずれかに記載の方法。

【請求項 10】

前記占有パラメータがスタックポインタ (28) として形成されることを特徴とする請求項 1 から 9 のいずれかに記載の方法。

【請求項 11】

前記ステップ d) において、最も古くから占有されているスタックセグメント (30) がスワップ書出しされることを特徴とする請求項 1 から 10 に記載の方法。 10

【請求項 12】

前記スワップ書出しされたスタックセグメント (30) が、サブプログラムの帰還アドレス、ローカル変数、中間結果及び／またはコールパラメータを含むことを特徴とする請求項 1 から 11 のいずれかに記載の方法。

【請求項 13】

特に前記ステップ a), b), c) 及び d)、並びに、必要に応じて前記ステップ e) 及び f)、における、前記スタック (26) の管理手段がハードウェアに実装されることを特徴とする請求項 1 から 12 のいずれかに記載の方法。

【請求項 14】

少なくとも、プロセッサ (12)、I/O ユニット (14)、スタック (26) をもつ少なくとも 1 つの揮発性読出－書込メモリ (16)、少なくとも 1 つの不揮発性メモリ (18, 20)、オペレーティングシステム (22) 並びに、データ、制御及びアドレスバス (24) からなるマイクロコントローラ (32) を有する携帯型データ記憶媒体 (10) において、前記オペレーティングシステム (22) が、前記スタック (26) をセグメント化するための手段及び、占有状態が定められた状態をこえると、前記揮発性読出－書込メモリ (16) の少なくとも 1 つのセグメント (30) を前記不揮発性メモリ (18, 20) にスワップ書出しするための手段を有することを特徴とする携帯型データ記憶媒体。 20

【請求項 15】

前記不揮発性メモリが不揮発性読出－書込メモリ (18) であることを特徴とする請求項 14 に記載の携帯型データ記憶媒体。 30

【請求項 16】

前記マイクロコントローラ (32) が、セグメント (30) を前記不揮発性メモリ (18, 20) にスワップ書出しするために請求項 1 から 10 のいずれかに記載の方法を実行するために設計されたページコントローラ (34) をさらに有することを特徴とする請求項 14 に記載の携帯型データ記憶媒体。

【発明の詳細な説明】

【0001】

発明の技術分野

本発明は携帯型データ記憶媒体上の高速読出－書込メモリのデータ構造の仮想拡張のための方法に関し、さらに、少なくとも、プロセッサ、I/O ユニット、スタックを有する高速読出－書込メモリ、低速読出－書込メモリ、オペレーティングシステム並びに少なくとも 1 つのデータ、制御及びアドレスバスからなる、マイクロコントローラを備える携帯型データ記憶媒体に関する。 40

【0002】

発明の背景

携帯型データ記憶媒体は、例えばスマートカードの形態で知られている。スマートカードの特徴は、カード本体に組み込まれ、データ転送及び格納のための手段を装備し、これらのデータを処理するためのユニットを備える集積回路を有することである。集積回路チップを破損または捻れによる損傷から保護するため、及びセキュリティ上の理由から、集積 50

回路チップは小面積に抑えられる。可能であれば、正方形で 25 mm^2 の面積を選ぶことが実際上有利であることがわかっている。したがって、モジュールの全て、特にメモリモジュールを備えるスマートカード上に配置される集積回路チップは、面積が強く制限される。

【0003】

そのように制限される面積により、及び、様々なメモリチップに必要な、タイプに依存する様々な大きさの面積－RAMの1ビットメモリセルにはEEPROMの1ビットメモリセルの約4倍もの面積が必要であり、さらにEEPROMの1ビットメモリセルにはROMの1ビットメモリセルの4倍もの面積が必要である－を考慮すれば、面積の大きなメモリチップ、特にRAMチップの使用を避けて、代わりにEEPROMまたはROMのような面積のより小さなチップを選択するようにしなければならない。しかし、面積の大きなチップの面積のより小さい別のチップによる望ましい置換えによるいかなる機能の喪失も絶対に避けるためには、元来用いられていたチップの機能が維持されなければならない。

【0004】

スマートカードには極めて多様な分野の用途及び目的がある。すなわち、いわゆるメモリーカードは主にテレフォンカードまたは健康保険カードとして用いられる。これらの機能は特定の用途に限られる。しかし、マイクロプロセッサカードは、CPUを収める付加ブロックモジュールがあることを特徴とする。これらのカードは通常、CPUとともに、I/Oユニット、データ、制御及びアドレスバス、揮発性読出－書込メモリ（RAM）、不揮発性読出－書込メモリ（EEPROM）及び不揮発性読出専用メモリ（ROM）も有する。オペレーティングシステムが主としてROMに格納される。

【0005】

マイクロプロセッサカードが、オペレーティングシステムに加えて、別のプログラムにアクセスするか否かに依存して、これらのカードは、ROMに格納された1つの特定のアプリケーションだけのために設計されるか、または別のユーザプログラムに指定される別の作業を行うことができる。このことにより、カードの機能性が著しく高められる。しかし、このような機能性の向上により、必要なメモリ面積も大きくなる。例えば、処理されるべきデータのためのより多くのレジスタ及び非常に多くのサブプログラムのためのより多くの帰還アドレスがRAMに備えられなければならない。しかし、現在普及しているカードにおいては、RAMの該当面積、すなわちスタックとして構成されている面積は、より複雑なアプリケーションには不十分である。上述したように、面積の大きなRAMチップのメモリ面積を任意に拡張することはできないから、RAMがスマートカードの機能性の可能な範囲を制限している。

【0006】

最近のスマートカードは実行可能なプログラムコードの再ロードを可能にするオペレーティングシステムを有している。このタイプのカードでは、アプリケーション提供者自身が、オペレーティングシステム製造業者の知らないプログラムコードをスマートカードで実行できる。例えば、いわゆるJava（登録商標）（ジャバ）カードは、JAVA言語でのプログラミング及び、いかなる十分に強力なプロセッサ上にも実装できる「Java」プロセッサの模擬マシンである、いわゆる“「Java」仮想マシン”に基づく。したがって、「Java」カードはプラットフォームにほとんど依存しない。しかし「Java」には、必須の仮想マシンによる、強いスタック要件がある。したがって、RAMの限られた物理的な大きさが「Java」カードの可能性を制限する。

【0007】

独国特許発明第19626337A1号明細書は、スマートカード上で処理されるべき外界からのメッセージの読込み及び処理のための方法及び装置を開示している。I/Oユニットを介して受け取られる、処理されるべきメッセージは、メモリ空間の制限のためRAMに完全には書き込むことができないサイズを有し得る。セキュリティ上の理由から、外界からのメッセージは、処理が可能となる前にCPUによるアクセス認可管理をさらに受けなければならない。メッセージをより小さなサイズのブロックに単純に分割すれば、認

10

20

30

40

50

証が不必要に手の込んだ作業になり、認証に手間取ることになる。R A Mが非常に小さいにもかかわらず、スマートカードとの通信がメッセージのデータサイズにはほとんど依存しないように、外界からのメッセージはE E P R O Mに直接に、またはR A Mへの中間格納に続いてE E P R O Mに間接に格納される。提案された解決策は、そのサイズのためR A Mに完全には格納できないメッセージすなわちデータのための中間格納を可能にするという手法に基づく。仮想メモリ拡張はR A MのI / Oバッファを介して、すなわち必要に応じて書込みができるR A M領域を介して実施される。上記明細書には、スタックとして構成される、上記のR A M領域の拡張についての示唆は含まれていない。

【0008】

欧州特許第720087号明細書は、拡張メモリを設定することによるハードウェアスタックの仮想拡張の提案を開示している。拡張メモリはR A Mの定められた領域に設定されなければならない。ハードウェアスタックの占有状態がハードウェアスタックの最大容量より僅かに少ない極限值をこえると、最も古いスタックエントリーが拡張メモリに移される。極限值に到達すると、最も古いスタックエントリーが拡張メモリに移される。極限值にまだ到達していなければ、次に古いスタックエントリーも外部に移される。この既知の方法により、有効なメモリ利用を可能にするためにハードウェアスタックが常時少なくともほとんど完全に利用可能であることが保証される。このことは、ハードウェアスタックがセクタに分割され、サブルーチンコール毎にセクタローテーションにより新しいセクタが用いられ、全てのセクタが占有されるとセクタがメモリに移されることで、上記の方法を他の既知の概念から区別する。

10

20

【0009】

しかし、欧州特許第720087号明細書に開示される方法は、個々のスタックエントリーの要求時スワップには高速メモリが必要であり、したがって必然的にR A Mに実装されなければならないため、携帯型データ記憶媒体に移植することができない。携帯型データ記憶媒体上の、特にスマートカード上のR A Mにはこのための十分な容量がない。

【0010】

スタックは、R A Mの他の領域とは異なる特殊な構成を有する。スタックは、L I F O（後入れ先出し）データ構造として実装され、ランダムにアクセスされるR A Mの他のメモリ領域より単純な管理及びアクセスアルゴリズムを基本的に可能にする、データ構造を構成する。したがって、スタックのスワップ領域は、R A Mの他の領域に対して適用可能な規準とは異なる規準に合せることができる。

30

【0011】

高水準言語で自由にプログラムすることができるスマートカードのR A Mは、現在一未だに一慣習的な8ビットコントローラに規制され、一般に256バイトの内部R A M（I R A M）及び、アドレス計算によりアドレスが可能であり、したがって若干低速の、1600バイトの拡張R A M（X R A M）を含む。I R A Mは完全にスタックとして通常用いられ、X R A Mは作業メモリ及びI / Oバッファを収める。

【0012】

発明の概要

したがって、本発明の課題は、特にスマートカードの形態の、携帯型データ記憶媒体に対してスタックとして構成されたR A Mの領域の仮想拡張を可能にし、よってデータ記憶媒体上に物理的に設けられたスタックより物理的に大きなスタックを必要とするであろうアプリケーションの実行も、本来のコンピュータ能力を可能な限り完全に保持しアプリケーションソフトウェアのような一現時点での特定のシステム要件並びに一R A M及びE E P R O Mへのアクセス時間、E E P R O Mの寿命等のような一システム均衡化特性を適応的に考慮しながら、可能にする方法を提供することにある。

40

【0013】

別の課題は、拡張スタックを必要とする複雑なアプリケーションの実行、特に「J a v a」アプレットの実行を可能にする携帯型データ記憶媒体を提供することにある。

【0014】

50

上記の課題は以下の：

- a) スタックを少なくとも2つのセグメントにセグメント化するステップ；
 - b) スタックの占有状態を表す占有パラメータを検出するステップ；
 - c) ステップb) で検出された占有パラメータをあらかじめ定められた極限值と比較することにより占有パラメータを監視するステップ；
 - d) 極限值と占有パラメータとの間のあらかじめ定められた関係が達成されると直ちに、少なくとも1つのスタックセグメントを別の記憶媒体にスワップするステップ；
- を有する、特許請求項1の属概念部分にしたがう方法により解決される。

【0015】

さらに、上記課題は、オペレーティングシステムがスタックをセグメント化する手段及び RAMの少なくとも1つのセグメントを別のメモリ、特にEEPROMのような不揮発性読出—書込メモリにスワップする手段を有する、スマートカードにより解決される。

【0016】

スタックは通常、最後のエントリーが最初に処理されるように、LIFOデータ構造として実施される。サブプログラムの帰還アドレスに加えて、スタックはローカル変数を収める。プログラムスタックは、サブプログラムの特定の帰還アドレスがサブプログラムのコール時に格納されるスタックである。プログラムの実行に際して、サブプログラムの特定の帰還アドレス及びローカル変数が順次に格納される。次いで、特定のサブプログラムがマスタープログラムに帰還するときに、特定の帰還アドレス及びローカル変数が逆順で要求される。非常に古いスタック内容または帰還アドレスがスタックメモリ空間を占有するが、その時点では特定のサブプログラムの実行に必要とされない。したがって、スタックセルブロックまたはセグメントは、別の、より低速のメモリ、例えばEEPROMにスワップさせることができる。特定のスタックメモリ内容は、その時点のプログラムが上位プログラムに帰還しなければならない時の直前にのみ、再び利用可能でなければならず、必要に応じて、前記別の記憶媒体から再びスワップされる。この間、上記のスタックセルのメモリ空間は、別のレコード、例えば別の帰還アドレス及びローカル変数を格納するために、本発明にしたがって用いることができる。この方法により、スタックを仮想拡張させることが可能になる。

【0017】

本発明は、RAMの全体として最も有効な負荷軽減を可能にするためには、特にスタックを、便宜的に仮想拡張させなければならないという事実を利用する。スタックの独特の構成により、本発明のスワップ機構は、別の構成の、あるいは何らかの構成を全くもたないRAMのメモリ領域に適用する場合に比較して、より有効に適用できる。例えば、バッファ領域がスワップされるとすれば、RAMがランダムにアクセスされている場合には、スワップされたバッファメモリ内容のどれがいつRAMで再び利用可能でなければならぬかは予知不能であるから、より低速なメモリにあるスワップされた内容へのアクセス時間は長くなると考えなければならないであろう。スワップされたセル内容がスワップのすぐ後のプログラム実行のために正確にRAMで再び利用可能でなくなるといいう事態を排除することはできない。そのような事態では、時間がかかるEEPROMへのアクセスの回数が多くなる。スタック内容がスワップされるときに、“RAMで利用可能になっている”ための時点があらかじめ計算されるという方式を便宜よく提供することができる。これは、LIFOデータ構造としてのスタック構成により可能である。

【0018】

本発明の有利な実施形態は2のべき乗 2^{n-m} でなされるセグメント化にある。ここで $m=1, 2, 3, \dots$ であり、 2^m はセグメント化数、 2^n はメモリセルまたはバイトにしたがうスタックサイズである。分割のための次元数 m は、スマートカードシステムの最適な均衡が得られるように、すなわち、スマートカードのその時点でのアプリケーションに対する、スタックサイズ、セグメントのスワップ書出し及び書込みにかかる時間、セグメント転送にかかる加算オーバーヘッド、スタックメモリ空間に対する要求等のようなパラメータを考慮して、選ばれることが有利である。すなわち、セグメント化はスマートカ

ードシステムのパラメータに依存し、アプリケーションソフトウェア、例えばサイズ、サブプログラム数等に依存して、選ばれることが有利である。好ましいセグメント化においては、セグメントサイズがEEPROMのブロックサイズに正確に一致する。

【0019】

したがって、例えば、スマートカードシステムが全体として最適化された実行を保証できるように、スマートカードの第1のある用途に対しては、2つの区画へのスタックの分割すなわちセグメント化が最も便宜がよい。スマートカードの別の用途に対しては、いくつもの、より小さいセグメントへのセグメント化が、例えば最適なシステム均衡化をより容易に得られるであろうから、より賢明であり得る。

【0020】

本発明の方法は、特に大きなスタック領域を必要とするスマートカードアプリケーション、例えば複雑な暗号アルゴリズムを含むプログラムあるいは「Java」アプレットを含むカードプログラムの実行も可能にする点で有利である。本発明の方法では、ユーザによるユーザ固有のプログラムのカード上へのロードが可能である。このことは、例えば更新プログラムをカードにロードできるから、カードの機能性を高め、寿命を延ばす点で有利である。

【0021】

本発明の別の実施形態において、最下層のスタックセグメントはEEPROMにスワップされず、フラッシュEEPROMまたはFRAM（強誘電体ランダムアクセスメモリ）のような別の不揮発性読出—書込メモリに、あるいは追加の揮発性読出—書込メモリ、例えば面積は最適化されているがこのためアクセス時間が長くなっているRAMにスワップされる。

【0022】

一実施形態においては、特に、スタックセグメントをアクセス時間が非常に短いRAMからアクセス時間がかかり長いEEPROMメモリにスワップする時間に関する難点が、前もってスタックバランスを用意しておくことによって回避される。すなわち、セグメントは、フリーになるスタック領域に関してより長いアクセス時間を考慮し、正の最終バランスが得られる場合にのみスワップされる。スタックバランスを用意するため、アプリケーションソフトウェアをシミュレータ上で試験することができ、あるいはデータフロー解析を行うことができる。

【0023】

2のべき乗 2^{n-m} でのスタックのセグメント化の選択により、占有パラメータを最小のアドレスビット数、すなわちmでコード化することができる。このことは、2つの区画への分割、すなわちm=1の場合に、1つのスタックセグメントから次のスタックセグメントへの変更の標識付け及び認識のためには、アドレスビットは1つで十分であることを意味する。4つの区画へのスタックの分割、すなわちm=2では、必要なアドレスビットは2つだけである。

【0024】

本発明のさらなる利点及び特徴は、以下の、図面の詳細な説明から得られるであろう。

【0025】

発明の詳細な説明

図1は、以降スマートカードの形態にあるされる、携帯型データ記憶媒体10を簡略に示す。スマートカード10は、プロセッサ12、I/Oユニット14、特にRAM（ランダムアクセスメモリ）16の形態にある揮発性読出—書込メモリ16、特にEEPROM（電氣的消去可能なプログラマブル読出専用メモリ）18の形態にある不揮発性読出—書込メモリ18、特にROM（読出専用メモリ）20の形態にある不揮発性読出専用メモリ20、メモリ18及び20に格納されたオペレーティングシステム22、並びに、データ、制御及びアドレスバス24からなるマイクロコントローラ32を搭載する。メモリ16、18、20のそれぞれは、物理的なサイズがメモリのタイプに依存する、多数の1ビットメモリセルを備える。

10

20

30

40

50

【0026】

RAM 16はシステムの作業メモリである。RAM 16の特徴は、アクセス時間が短い、メモリセルサイズに関して面積をとり、供給電圧が短時間であっても絶たれるとすぐにその内容が失われることである。高水準言語での自由なプログラミングが可能な8ビットコントローラを備える普通のスマートカードにおいて、RAMは、例えば、一般に16000バイトの、アドレス計算によるアドレスが可能な拡張RAM (XRAM) に補助される、256バイトの内部RAM (IRAM) からなる。RAM 16は通常、暗号アルゴリズムのような複雑なアルゴリズムのための作業領域に加えて、様々な領域：レジスタ領域、スタック26、変数領域及びI/Oバッファ領域にも分割される。

【0027】

RAM 16の一部としてのスタック26は、LIFOデータ構造として実装される。さらに、スタック26はそれぞれがある数のセルを含むセグメント30に分割される。スタック26の占有状態についてのステートメントを得るため、スタック26の最終格納エントリ、特に最下層エントリを指すスタックポインタ28が備えられる。

【0028】

ROM 20は供給電圧が絶たれているときでもデータを保持する。このメモリチップへの書込アクセスは不可能である。ここには、主としてオペレーティングシステムが格納される。

【0029】

EEPROM 18は最も精巧なメモリチップである。EEPROM 18はそれぞれがある数のセルを含むブロックに分割される。書込走査毎にブロックに書込みがなされる。データは電力供給が絶たれていてもEEPROMに保持される。しかし、EEPROMは、EEPROMブロックに対する書込時間が実際の書込バイト数にはほとんど依存しないから、アクセス時間が非常に長い。書込時間は普通、ブロック当たり3ミリ秒程度である。

【0030】

本発明の好ましい実施形態において、スワップされたスタックセグメント30のアドレス指定は、同様にRAM 16に格納されることが好ましい宛先アドレスレジスタにより解かれる。セグメント30がスタック26からEEPROM 18にスワップされると、EEPROM 18の初期アドレスで初期化された宛先アドレスレジスタがインクリメントされる。ここで想定されている事例においては、アドレス値が例えばセグメントサイズだけ、すなわち $2^n - m$ だけインクリメントされる。別のセグメント30がスワップ書込みされたときには、この宛先アドレスレジスタが再び $2^n - m$ だけインクリメントされ、よって宛先アドレスレジスタは常に最も新しいセグメントを指す。スタック26で再び利用可能となるために、スワップ書込みされたセグメント30がEEPROMからスワップ書出しされると、宛先アドレスレジスタは、例えばここで想定されている場合には $2^n - m$ だけ、デクリメントされる。

【0031】

本発明の別の実施形態において、スマートカードは、上述したコンポーネントだけでなく、さらに、ハードウェア補助メモリ管理、特に公開キーアルゴリズムの計算のための数値演算コプロセッサ、EEPROMの誤り検出及び補正ユニット、乱数発生器等のようなコンポーネントも有する。

【0032】

本発明の特に有効な実施形態において、本発明の方法はハードウェアに実装され、したがってソフトウェア実装より実行が相当に高速である。この目的のため、マイクロコントローラ32はページコントローラをさらに有する。ページコントローラは、スタックセグメント30をスワップ書出しするための手段及び、必要に応じて、本発明の方法にしたがう手段を有する、専用のいわゆるメモリ管理ユニット(MMU)として形成される。

【0033】

本発明の方法がソフトウェアに実装されれば、プロセッサ12がオペレーティングシステム12とともに、スタックセグメント30の転送を担当する。

10

20

30

40

50

【0034】

以下に、本発明の方法が機能するモードを図2を参照して説明する。

【0035】

プログラム実行中の、サブプログラムの、階層的に上位のサブプログラムまたはマスタープログラムへの帰還のための特定のアドレスは、その時点でのローカル変数、計算結果及びコールパラメータとともにスタック26に格納される。プログラムの実行が進むほど、スタック26内の占有領域は大きくなる。それにもかかわらず、実際に、物理的に存在するはずの利用可能領域より大きな利用可能領域をスタック26に必要とするプログラムの実行が可能となるように、スタック26の、既に書込みがなされている、すなわち占有されている個々のセグメント30が、より大きいより低速なメモリ、好ましくはEEPROM18にスワップされる。可能な限り後にならないとプログラム実行に再び必要とされない、最も古くに書込みがなされたセグメント30が選択されることが有利である。

【0036】

プログラム実行がかなり進行すると、セグメント30のいくつかはEEPROM18に置かれている。ここで再びスタック26上のフリーなメモリ空間として十分な容量があり、スワップされたセグメントがアクセスされなければならないときには、スワップされたセグメント30が次々に、個々にまたはまとめてスタック26にスワップ書戻しされる。このスワップ書戻しは、スワップ書込み及び書出しの所要時間、特定のスタック要素へのアクセス頻度等のようなシステムパラメータが最適に考慮されるように実行されることが有利である。

【0037】

スワップされるべきセグメントのスワップ書出し及びスワップ書戻しは、それぞれの場合に、1回の、相関転送操作で、セグメント毎に実施される。スワップ書出し及びスワップ書戻しは、既知のブロック毎転送命令を用いて便宜よくなされる。

【0038】

極限值GW及びセグメント化サイズの適切な決定により、マイクロコントローラが付加MMUを含めて構成されていれば、CPUの、例えば“立ち往生”を予防する、すなわち、その時点で必要なスタック要素がEEPROMからスタック26に戻しロードされるまでまず待たなければならない事態にプロセッサが陥ることを予防することができる。

【0039】

自身のエネルギー源の欠如によるスマートカードのシステム特性のため、処理の進行中に、例えばデータがEEPROMに書き込まれている間に、カードの電力供給が、例えば読取器からの取出しにより、中断されることが可能となる。したがって、時々刻々処理されるデータの状態は不定であり得る。詳しくは、データが失われるか、またはメモリに部分的にしか転送されないでいることがあり得る。特に、RAMに置かれたスタック26、スタックポインタ28及び宛先アドレスレジスタが失われ得る。これは、とりわけ、新しいレコードを一部に収め、同時に古いレコードを一部に収めているファイルにおこり得る。この不整合は、アプリケーションに問題を生じさせるであろう。そのような不整合を確実に回避するためには、ある特に適切な方策が、スマートカード上で、いわゆる原子的シーケンス、すなわち二値原理にしたがって完全に実行できるかまたは全く実行できないかのいずれかでしかないシーケンスとして、構築される。スマートカードのハードウェアは原子的シーケンスをサポートしないから、この原理はソフトウェアに実装されなければならない。例えば特定のレコードを分割せずに全体として処理できるように、本発明に事実上したがって、RAM、特にスタックに一定の領域を確保することがおそらく必要であろう。

【0040】

スマートカードシステム、特にマイクロコントローラ32のソフトウェアは、原子的オペレーション、例えば「Java」アプレットの特有のバイトコードを実行するように設計され、したがって、本発明の有利な別の実施形態においては、スタック要件を前もって決定し、例えば専用に備えられたテーブルに格納することができる。次いで、スワップ書出

し及び書込みのための特定のセグメントの管理が上記テーブルを参照して実行される。

【0041】

物理的限界の下で利用可能なCPUスタックに対し、本発明の方法は理論的にサイズが無制限のスタック26を提供する点で有利である。

【0042】

スマートカードシステムがプログラムの実行中のスワップ書出し及び書込みを制御できるように、RAM16のスタック26の占有状態を表す情報が与えられる。これは、好ましくは、最新のエンタリーが実施されたスタック26のメモリセルを常に指すスタックポインタ28の形態にある、占有パラメータにより実現される。スタック26に新しいエンタリーがある毎に、スタックポインタ28がインクリメントされる。

10

【0043】

必要に応じて、どのスタック26の占有状態でスタックセグメント30がスワップ書出しされるべきかを決定することができる。特定のアプリケーションに依存して、例えば2つの区画へのセグメント化の場合には、比較的大きなセグメント30は極めて稀にしかスワップ書出しされないと便宜がよいことがあり得る。他方で、例えばセグメント化がより細かい場合には、より細かな分割によって、より小さいセグメントがより頻繁にスワップ書出しされると便宜が一層よいことがおこり得る。

【0044】

本発明の方法の上記の拡張された設計性は、極限值GWの定義により可能になる。極限值GWは、スマートカードシステムの総体の最適均衡化が保証されるように、すなわちRAMへのアクセス時間とEEPROM18へのアクセス時間が互いに調和され、CPU計算時間及び管理、特にスワップされたセグメント30のアドレス指定のための付加メモリ要件が考慮に入れられるように、選ばれることが好ましい。極限值GWは、特定のセグメント限界をこえたときに最も古いセグメント30がスワップ書出しされるように、第1の便宜のよい設定におけるセグメントサイズに基づく。セグメント30のサイズがEEPROM18のブロックサイズと正確に一致すると非常に便宜がよい。そうすれば、RAM16とEEPROM18との間のスタックセグメントの転送を、1つのブロック毎転送命令を用い、時間的に最適化された態様で実施できる。別の便宜のよい設定においては、極限值GWがセグメントサイズの倍数に一致するように設定され、よって複数のセグメント30が必ずブロック毎に同時にスワップ書出しされる。次いで、セグメント30は、同じ方式で、セグメント30のブロックとしてスタック26にスワップ書戻しすることができる。スワップ書戻しは同様に1セグメント毎に実施することができる。一般に、あるステップでスワップ書出しされるセグメントの数は、あるステップでスタック26に戻し転送されるセグメントの数に一致する必要はない。全転送セグメントの合計だけを均衡させなければならない。

20

30

【0045】

極限值GWを定めることにより、EEPROM18のメモリチップの寿命を考慮に入れることも可能になる。EEPROMは技術上限られた数の書込－消去サイクルしか行えず、書込－消去される頻度が高いほど記憶保持時間が短くなる。したがって、システム性能がEEPROMの寿命と最適に調和するように、極限值GWが選ばれることが好ましい。

40

【0046】

好ましい実施形態において、極限值GWとスタックポインタ28との間の関係は、関係式“スタックポインタ28 ≤ 極限值GW”で表すことができるように、選ばれる。すなわち、スタック26の占有状態があらかじめ定められたセグメントサイズをこえると直ちに、最も古いスタックセグメント30がスワップ書出しされる。スワップ書出しと同様に、RAM16へのセグメントのスワップ書戻しもある極限值により決定されることと便宜がよい。上記の極限値はGWと同じとすることができる。セグメント30のスワップ書戻しのための極限値は、本発明の方法の可変性を高めるために、極限值GWと異なる値とすることもできる。さらに、本方法を改変すると便宜がよいこともある。例えば短いプログラムの場合には、スワップ書出しがプログラムの開始時にすぐには行われず、スタック26が

50

ほとんど完全に占有されていることをスタックポインタ 28 が指示したときにのみ行われることが、有利であり得る。

【0047】

本発明の主要な実施形態においては通常、マイクロコントローラ 32 はメモリチップ、すなわち RAM 16, ROM 20 及び EEPROM 18 を備える。RAM 16 内のスタック 26 の個々のブロックが、本発明にしたがって EEPROM 18 にスワップ書出しされる。あるいは、マイクロプロセッサ 32 は 2 つの RAM チップ、すなわち RAM 16 及び付加 RAM チップ、並びに ROM 20 を備える。付加 RAM チップは面積が最適化されているが、RAM 16 よりアクセス時間が長い。本発明にしたがえば、RAM 16 に存在するスタック 26 のセグメント 30 が、付加 RAM チップのメモリ領域にスワップされる。

10

【図面の簡単な説明】

【図 1】

部分的に示されるスマートカード上の、本発明のマイクロコントローラの基本モジュールの略図を示す

【図 2】

本発明にしたがうスタックセグメントの略図を示す

【符号の説明】

- 10 携帯型データ記憶媒体
- 12 プロセッサ
- 14 I/O ユニット
- 16 揮発性読出—書込メモリ
- 18 不揮発性読出—書込メモリ
- 20 不揮発性読出専用メモリ
- 22 オペレーティングシステム
- 24 データ、制御及びアドレスバス
- 26 スタック
- 28 スタックポインタ
- 30 スタックセグメント
- 32 マイクロコントローラ
- 34 ページコントローラ
- GW 極限值

20

30

【国際公開パンフレット】

(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES
PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum
Internationales Büro



(43) Internationales Veröffentlichungsdatum
7. März 2002 (07.03.2002)

PCT

(10) Internationale Veröffentlichungsnummer
WO 02/19264 A1

- (51) Internationale Patentklassifikation: **G06K 19/07**, 19/073, G07F 7/10
- (21) Internationales Aktenzeichen: PCT/EP01/09664
- (22) Internationales Anmeldedatum: 21. August 2001 (21.08.2001)
- (25) Einreichungssprache: Deutsch
- (26) Veröffentlichungssprache: Deutsch
- (30) Angaben zur Priorität: 100 40 974.1 22. August 2000 (22.08.2000) DE
- (71) Anmelder (für alle Bestimmungsstaaten mit Ausnahme von US): **GIESECKE & DEVRIENT GMBH** [DE/DE]; Prinzregentenstrasse 159, 81677 München (DE).
- (72) Erfinder; und
(75) Erfinder/Anmelder (nur für US): **CIESINGER, Daniel** [DE/DE]; Lily-Braun-Weg 15, 80637 München (DE).
FREY, Thomas [DE/DE]; Grossvenedigerstrasse 6, 85560 Ebersberg (DE).
MERCK, Martin [DE/DE]; Schweinfurterstrasse 3a, 81243 München (DE).
STOCKER, Thomas [DE/DE]; Schildensteinstrasse 16, 81673 München (DE).
- (74) Anwalt: **KLUNKER, SCHMITT-NILSON, HIRSCH**; Winzerstrasse 106, 80797 München (DE).
- (81) Bestimmungsstaaten (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.
- (84) Bestimmungsstaaten (regional): ARIPO-Patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), eurasisches Patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), europäisches Patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI-Patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).
- Veröffentlicht:**
— mit internationalem Recherchenbericht
— vor Ablauf der für Änderungen der Ansprüche geltenden Frist; Veröffentlichung wird wiederholt, falls Änderungen eintreffen
- Zur Erklärung der Zweibuchstaben-Codes und der anderen Abkürzungen wird auf die Erklärungen ("Guidance Notes on Codes and Abbreviations") am Anfang jeder regulären Ausgabe der PCT-Gazette verwiesen.

(54) Title: METHOD FOR VIRTUALLY ENLARGING THE STACK OF A PORTABLE DATA CARRIER

(54) Bezeichnung: VERFAHREN ZUR VIRTUELLEN VERGRÖßERUNG DES STACKS EINES TRAGBAREN DATENTRÄGERS

(57) Abstract: The invention relates to a method for virtually enlarging the stack of a portable data carrier (10) which enables executable programme codes to be reloaded. In order to enable programmes requiring a larger stack than the physically available stack to be carried out, the stack (26) is split into at least two stack segments (30) and the filling state thereof is monitored. When the complete load of a segment (30) is known, the respectively oldest stack segment (30) is transferred into another storage medium (18). Said other storage medium (18) can be a slower memory in relation to the stack (26). When sufficient memory space is available again in the stack (26), the transferred segment (30) is transferred back.

(57) Zusammenfassung: Vorgeschlagen wird ein Verfahren zur virtuellen Vergrößerung des Stacks eines tragbaren, das Nachladen von ausführbaren Programmcode erlaubenden Datenträgers (10). Um die Ausführung von Programmen zu ermöglichen, die einen größeren als den physisch vorhandenen Stack benötigen, wird der Stack (26) in mindestens zwei Stacksegmente (30) segmentiert und deren Füllungsstatus überwacht. Wird die vollständige Belegung eines Segments (30) erkannt, wird das jeweils älteste Stacksegment (30) in ein weiteres Speichermedium (18) ausgelagert. Das weitere Speichermedium (18) kann ein im Vergleich zum Stack (26) langsamer Speicher sein. Steht im Stack (26) wieder ausreichend Speicherplatz zur Verfügung, wird das ausgelagerte Segment (30) wieder zurückgeführt.

WO 02/19264 A1

WO 02/19264

PCT/EP01/09664

Verfahren zur virtuellen Vergrößerung des Stacks
eines tragbaren Datenträgers

5

Die Erfindung betrifft ein Verfahren zur virtuellen Vergrößerung der Datenstruktur eines schnellen Schreib-/Lese-Speichers auf einem tragbaren Datenträger und bezieht sich weiterhin auf einen tragbaren Datenträger mit einem Mikrocontroller, zumindest bestehend aus einem Prozessor, einer I/O-Einheit, einem schnellen Schreib-/Lese-Speicher mit einem Stack, einem langsamen Schreib-/Lese-Speicher, einem Betriebssystem und mindestens einem Daten-, Steuer- und Adreßbus.

Tragbare Datenträger sind unter anderem in Gestalt von Chipkarten bekannt. Diese zeichnen sich dadurch aus, daß sie eine integrierte Schaltung im Kartenkörper eingebaut haben, die mit Mitteln zur Datenübertragung und -speicherung und mit einer Einheit zur Bearbeitung dieser Daten ausgestattet ist. Um den Chip vor Bruch- oder Torsionsschäden zu schützen sowie aus Sicherheitsgründen wird die Fläche des Chips klein gehalten. Es hat sich in der Praxis als vorteilhaft erwiesen, möglichst eine quadratische Fläche von 25mm² zu wählen. Der auf der Chipkarte befindliche Chip mit all seinen Bauteilen, insbesondere seinen Speicherbauteilen, ist in seiner flächenmäßigen Ausdehnung daher stark eingeschränkt.

Aufgrund dieser Ausdehnungsbeschränkung und unter Berücksichtigung des typabhängig unterschiedlichen Platzbedarfes der verschiedenen Speicherbausteine – eine 1-Bit-Speicherzelle eines RAMs benötigt etwa viermal so viel Platz wie eine 1-Bit-Speicherzelle eines EEPROMs, dieses wiederum viermal so viel wie eine 1-Bit-Zelle eines ROMs – entsteht die Notwendigkeit, möglichst wenig der flächenintensiven Speicherbausteine, d.h. insbesondere der RAM-Bausteine, und stattdessen mehr von weniger flächenintensiven

WO 02/19264

PCT/EP01/09664

- 2 -

Bausteinen, wie ein EEPROM oder ein ROM, zu verwenden. Um jedoch bei einer wünschenswerten Ersetzung eines flächenintensiven Bausteines durch einen weniger flächenintensiven anderen keine Funktionseinbuße hinnehmen zu müssen, muß die Funktionalität des ursprünglich verwendeten Bausteines gewahrt bleiben.

Chipkarten haben unterschiedlichste Anwendungsgebiete und Einsatzzwecke. So werden sogenannte Speicherkarten vornehmlich als Telefonkarten oder Krankenversicherungskarten eingesetzt. Ihre Funktionalität ist auf eine spezifische Anwendung beschränkt. Mikroprozessorkarten hingegen zeichnen sich durch ein zusätzlichen Block von Bauteilen aus, der eine CPU beinhaltet. Üblicherweise verfügen diese Karten neben der CPU ferner eine I/O-Einheit, einen Daten-, Steuer- und Adreßbus, einen flüchtigen Schreib-/Lese-Speicher (RAM), einen nicht-flüchtigen Schreib-/Lese-Speicher (EEPROM) sowie einen nicht-flüchtigen Nur-Lese-Speicher (ROM). Das Betriebssystem ist dabei hauptsächlich im ROM abgelegt.

Davon abhängig, ob sie neben dem Betriebssystem noch auf weitere Programme zugreifen, sind die Karten entweder nur für eine spezielle, im ROM abgelegte, Anwendung ausgelegt, oder können unterschiedliche Aufgaben erfüllen, die in den weiteren Anwendungsprogrammen spezifiziert sind. Damit steigert sich die Funktionalität der Karten deutlich. Mit diesem Funktionalitätsgewinn steigt jedoch auch der Bedarf an Speicherplatz. So müssen beispielsweise im RAM mehr Register für die zu verarbeitenden Daten und mehr Rücksprungsadressen für die Vielzahl der Unterprogramme bereitgestellt werden. Bei den derzeit gebräuchlichen Karten ist jedoch der hierfür relevante Bereich des RAM's, nämlich der als Stack organisierte Bereich für komplexere Anwendungen nicht ausreichend. Da aber, wie oben dargestellt, der Speicherplatz des flächenintensiven RAM-Bauteils nicht beliebig erwei-

WO 02/19264

PCT/EP01/09664

- 3 -

tert werden kann, beschränkt der RAM den möglichen Funktionalitätsumfang der Chipkarten.

- Neuere Chipkarten verfügen über Betriebssysteme, die ein Nachladen von ausführbarem Programmcode ermöglichen. Bei diesem Kartentyp kann ein Anwendungsanbieter selbst einen Programmcode mit der Chipkarte ausführen, den der Betriebssystemhersteller nicht kennt. Sogenannte Java-Cards etwa basieren auf der Programmierung in der Sprache JAVA und auf einer sogenannten „Java Virtual Machine“, die eine Simulation eines Java-Prozessors ist, der auf jedem ausreichend leistungsstarken Prozessor implementiert werden kann. Damit sind Java-Cards weitestgehend plattformunabhängig. Allerdings benötigt Java durch die notwendige „Virtual Machine“ einen hohen Stackbedarf. Die beschränkte physikalische Größe des RAMs beschränkt daher die Möglichkeiten der Java-Card.
- 15 Aus der DE 196 26 337 A1 sind ein Verfahren und eine Vorrichtung bekannt, bei dem/der auf einer Chipkarte zu verarbeitende Nachrichten aus der Außenwelt eingelesen und prozessiert werden. Die über die I/O-Einheit empfangenen und zu verarbeitenden Nachrichten können dabei eine Größe aufweisen, die wegen der Speicherplatzbeschränkung nicht vollständig in das RAM eingeschrieben werden kann. Aus Sicherheitsgründen müssen von außen kommende Nachrichten zudem einer Zugriffsberechtigungs-Kontrolle von der CPU unterzogen werden, bevor sie weiterprozessiert werden dürfen. Eine einfache Zerteilung der Nachrichten in Blöcke von geringerer Größe würde die Authentifikation unnötig aufwendig machen und verzögern.
- 20 Um trotz zu kleinem RAMs eine Kommunikation mit der Chipkarte zu ermöglichen, die weitestgehend unabhängig von der Datengröße der Nachrichten ist, sollen von außen kommende Nachrichten entweder direkt in das EEPROM oder indirekt im Anschluß an eine Zwischenspeicherung im RAM
- 25

WO 02/19264

PCT/EP01/09664

- 4 -

im EEPROM abgelegt werden. Die vorgeschlagene Lösung beruht auf dem Ansatz, für Nachrichten, bzw. Daten, die aufgrund ihrer Größe nicht vollständig im RAM gespeichert werden können, Möglichkeiten zur Zwischenspeicherung zu schaffen. Die scheinbare Speichervergrößerung erfolgt dabei
5 über den I/O-Puffer des RAM, d.h. über den wahlfrei beschreibbaren Teil des RAMs. Hinweise zur Vergrößerung desjenigen Bereiches des RAM's, der als Stack organisiert ist, enthält die Schrift nicht.

Aus der EP 720 087 ist der Vorschlag entnehmbar, einen Hardwarestack durch die Einrichtung eines Erweiterungsspeichers virtuell zu vergrößern.
10 Der Erweiterungsspeicher soll in einem definierten Bereich des RAM's eingerichtet werden. In den Erweiterungsspeicher werden die jeweils ältesten Stackeinträge ausgelagert, wenn der Füllzustand des Hardwarestacks einen Grenzwert überschreitet, der knapp unter der maximalen Aufnahmekapazität des Hardwarestacks liegt. Bei Erreichen des Grenzwertes wird der jeweils
15 älteste Stackeintrag in den Erweiterungsspeicher ausgelagert. Ist der Grenzwert immer noch nicht unterschritten, wird anschließend auch der nächstälteste Stackeintrag ausgelagert. Dieses bekannte Verfahren stellt sicher, daß der Hardwarestack stets zumindest nahezu vollständig nutzbar ist und da-
20 durch eine effektive Speichernutzung möglich wird. Hierdurch unterscheidet es sich von einem anderen bekannten Konzept, nach dem der Hardwarestack in Sektoren gegliedert wird und bei jedem Unterprogrammaufruf durch eine Sektorrotation ein neuer Sektor genutzt wird, wobei im Falle der Belegung aller Sektoren ein Sektor in einen Speicher ausgelagert wird.

25 Das aus der EP 720 087 entnehmbare Verfahren ist gleichwohl nicht auf tragbare Datenträger übertragbar, weil das bedarfsweise Auslagern einzelner Stackeinträge einen schnellen Speicher erfordert und deshalb zwingend im

WO 02/19264

PCT/EP01/09664

- 5 -

RAM realisiert werden muß. Auf tragbaren Datenträgern, speziell auf Chipkarten, steht im RAM hierfür jedoch nicht ausreichend Platz zur Verfügung.

Der Stack ist im Unterschied zu den anderen Bereichen im RAM speziell organisiert. Er bildet eine Datenstruktur, die als LIFO-Datenstruktur (last in first out) implementiert ist und grundsätzlich einfachere Verwaltungs- und Zugriffsalgorithmen ermöglicht als die übrigen Speicherbereiche des RAMs, auf die randomisiert zugegriffen wird. Auslagerung von Teilen des Stacks kann deshalb anderen Kriterien genügen als in Bezug auf den übrigen Teil des RAMs anzuwenden sind.

Der RAM von in Hochsprache frei programmierbaren Chipkarten wird gegenwärtig durch die - noch - üblichen 8-Bit Controller beschränkt und umfaßt typischerweise 256 Byte internen IRAMs sowie 1600 Byte über Adreßrechnungen adressierbaren, deshalb geringfügig langsameren erweiterten XRAM. Davon wird in der Regel der IRAM komplett als Stack genutzt, während im XRAM der Arbeitsspeicher und der I/O-Puffer liegen.

Aufgabe der vorliegenden Erfindung ist es deshalb ein Verfahren zu schaffen, das eine virtuelle Vergrößerung des als Stack organisierten Bereiches des RAM's für einen tragbaren Datenträger, insbesondere in Gestalt einer Chipkarte ermöglicht, so daß auch Anwendungen lauffähig sind, die einen physikalisch größeren Stack benötigen würden, als er auf dem Datenträger physikalisch zur Verfügung gestellt wird - unter möglichst vollständiger Beibehaltung der ursprünglichen Rechenleistung und unter adaptiver Berücksichtigung der jeweils aktuellen Systemanforderungen - wie der Applikations-Software - und Systembalancierungseigenschaften - wie den Zugriffszeiten auf RAM und EEPROM, Lebensdauer des EEPROM, etc.

WO 02/19264

PCT/EP01/09664

- 6 -

Eine weitere Aufgabe liegt in der Bereitstellung eines tragbaren Datenträgers, auf dem komplexe Anwendungen, insbesondere Java-Applets lauffähig sind, die einen erweiterten Stack voraussetzen.

5 Diese Aufgabe wird gelöst durch ein Verfahren nach dem Oberbegriff von Anspruch 1, das folgende Schritte aufweist:

- a) Segmentierung des Stacks in mindestens zwei Stacksegmente,
- 10 b) Erfassen eines Füllungsparameters, der einen Füllungszustand für den Stack beschreibt,
- c) Überwachen des Füllungsparameters, indem der in Schritt b) erfaßte Füllungsparameter mit einem vorbestimmten Grenzwert verglichen
15 wird
- d) Auslagern zumindest eines Stacksegmentes in ein weiteres Speichermedium, sobald ein vorbestimmtes Verhältnis zwischen Grenzwert und Füllungsparameter erreicht ist.

20

Weiterhin wird die Aufgabe durch eine Chipkarte gelöst, bei der das Betriebssystem Mittel aufweist, um den Stack zu segmentieren und um zumindest ein Segment des RAM's in einen weiteren Speicher, insbesondere in einen nicht-flüchtigen Schreib-/Lese-Speicher wie ein EEPROM, auszulagern.

25

Üblicherweise ist der Stack als LIFO-Datenstruktur implementiert, so daß der jeweils jüngste Eintrag auch als erstes wieder bearbeitet wird. Neben Rücksprungradressen von Unterprogrammen beinhaltet er lokale Variablen.

- Der Programmstack ist ein Stack, in den beim Aufruf von Unterprogrammen die jeweiligen Rücksprungadressen der Unterprogramme abgelegt sind. Bei Abarbeitung des Programmes werden nun nacheinander die jeweiligen Rücksprungadressen und die lokalen Variablen des Unterprogrammes abgelegt. In umgekehrter Reihenfolge werden diese dann beim Rücksprung aus dem jeweiligen Unterprogramm in das Hauptprogramm benötigt. Bereits lang zurückliegende Stackinhalte, bzw. Rücksprungadressen belegen zwar den Stackspeicherplatz, werden aber aktuell für die Ausführung des jeweiligen Programmes nicht benötigt. Deshalb können Blöcke bzw. Segmente von Stackzellen in einen anderen, langsameren Speicher, wie beispielsweise den EEPROM ausgelagert werden. Erst kurz bevor das ablaufende Programm wieder zu dem jeweiligen übergeordneten Programm zurückspringen muß, müssen die jeweiligen Stack-Speicherinhalte wieder verfügbar sein und gegebenenfalls aus dem anderen Speichermedium wieder eingelagert werden. In der Zwischenzeit kann der Speicherplatz dieser Stack-Zellen erfindungsgemäß wiederum für andere zu speichernde Datensätze, z.B. für weitere Rücksprungadressen und lokale Variablen genutzt werden. Mit Hilfe dieses Verfahrens erfährt der Stack eine virtuelle Vergrößerung.
- 20 Die Erfindung macht sich dabei zunutze, daß sinnvollerweise speziell der Stack eine virtuelle Vergrößerung erfahren muß, um den RAM als Ganzes am effizientesten entlasten zu können. Aufgrund der spezifischen Organisation des Stacks ist ein erfindungsgemäßer Auslagerungsmechanismus wirkungsvoller anwendbar als bei den anders oder nicht organisierten Speicherbereichen des RAMs. Würden etwa Bereiche des Puffers ausgelagert, müßte mit erhöhten Zugriffszeiten auf die ausgelagerten Inhalte in dem langsameren Speicher gerechnet werden, da nicht vorhersagbar ist, wann welcher ausgelagerte Pufferspeicherinhalt wieder im RAM zur Verfügung stehen muß, wenn dieser einem Random-Access Zugriff unterliegt. Es kann
- 25

nicht ausgeschlossen werden, daß für den Programmablauf gerade die Inhalte der ausgelagerten Zellen kurz nach deren Auslagerung wieder im RAM verfügbar sein müssen. Damit ergibt sich eine erhöhte Anzahl von zeitaufwendigen Zugriffen auf das EEPROM. Sinnvoll kann vorgesehen sein, den
5 Zeitpunkt des „Zu-Verfügung-Stehens-im-RAM“ bei der Auslagerung von Stackinhalten vorauszukalkulieren. Dies ist aufgrund der Organisation des Stacks als LIFO-Datenstruktur möglich.

Eine vorteilhafte Ausführungsform der Erfindung besteht darin, daß die
10 Segmentierung in 2-er Potenzen 2^{n-m} erfolgt, mit $m = 1, 2, 3, \dots$, wobei 2^m die Anzahl der Segmentierungen und wobei 2^n die Größe des Stacks nach Speicherzellen oder nach Bytes betrifft. Die Maßzahl m für die Teilung wird vorteilhafterweise so gewählt, daß eine optimale Balancierung des Chipkartensystems erreicht wird, d.h. daß Parameter wie Größe des Stacks, Zeit für
15 Aus- und Einlagern der Segmente, zusätzlicher Verwaltungsaufwand für den Segmenttransfer, Bedarf an Stack-Speicherplatz etc. für die aktuelle Anwendung der Chipkarte Berücksichtigung finden. Die Segmentierung wird also vorteilhafterweise in Abhängigkeit von Parametern des Chipkartensystems und in Abhängigkeit von der Applikations-Software, wie Größe, Anzahl der Unterprogramme etc. gewählt. In einer bevorzugten Segmentierung
20 entspricht die Segmentgröße genau der Größe der Blöcke des EEPROMs.

Damit ergibt sich beispielsweise, daß für einen ersten bestimmten Einsatz der Chipkarte eine Zweiteilung des Stacks, bzw. eine Segmentierung in zwei
25 Teile am sinnvollsten ist, damit das Chipkartensystem als Ganzes einen optimierten Ablauf gewährleisten kann. Für einen weiteren bestimmten Einsatz der Chipkarte kann eine Segmentierung in mehrere, kleinere Segmente klüger sein, da hiermit beispielsweise leichter eine optimale Systembalancierung erreicht werden kann.

Vorteilhafterweise sind mittels des erfindungsgemäßen Verfahrens auch Chipkartenanwendungen lauffähig, die einen besonders großen Stackbereich benötigen, wie beispielsweise Programme, die komplexe kryptographische Algorithmen beinhalten oder Karten mit Java-Applets. Hier haben die Anwender die Möglichkeit, ihre spezifischen Programme auf die Karte zu laden. Damit erhöht sich vorteilhafterweise die Funktionalität der Karte und deren Lebensdauer, da z.B. Programm-Updates auf die Karte gespielt werden können.

10

In einer alternativen Ausführungsform der Erfindung wird das jeweils unterste Stacksegment nicht in den EEPROM ausgelagert, sondern in einen anderen nicht-flüchtigen Schreib/Lese-Speicher, wie ein Flash-EEPROM oder ein FRAM (ferroelectric random access memory) oder in einen zusätzlichen flüchtigen Schreib/Lese-Speicher, beispielsweise ein RAM, das flächenoptimiert ist, dadurch allerdings langsamere Zugriffszeiten aufweist.

In einer Ausführungsvariante werden insbesondere zeitliche Nachteile beim Auslagern von Stacksegmenten aus dem RAM-Speicher mit sehr schnellen Zugriffszeiten in den EEPROM-Speicher mit wesentlich erhöhten Zugriffszeiten vermieden, indem vorab eine Stackbilanz erstellt wird. Damit wird ein Segment nur dann ausgelagert, wenn sich bei Berücksichtigung der erhöhten Zugriffszeiten in Relation zu dem frei werdenden Stackbereich eine positive Endbilanz ergibt. Zur Erstellung der Stackbilanz kann die Applikations-Software auf einem Simulator getestet werden oder eine Datenflußanalyse vorgenommen werden.

Durch die Wahl der Segmentierung des Stacks in Zweier-Potenzen 2^{n-m} ergibt sich der Vorteil, daß der Füllungsparameter in einer minimalen Anzahl,

WO 02/19264

PCT/EP01/09664

- 10 -

- nämlich (m), von Adress-Bits codiert werden kann. Das bedeutet, daß bei einer Zweiteilung, d.h. $m=1$, ein Adress-Bit ausreicht, um den Wechsel von einem Stacksegment zu dem nächsten zu kennzeichnen und zu erkennen. Bei einer Aufteilung des Stacks in vier Teile, d.h. $m=2$, werden lediglich zwei
- 5 Adress-Bits benötigt.

Weitere Vorteile und Merkmale der Erfindung ergeben sich aus der nachfolgenden detaillierten Figurenbeschreibung, in der:

- 10 Fig. 1 eine schematische Darstellung von Grundbausteinen eines erfindungsgemäßen Mikrocontrollers auf einer teilweise dargestellten Chipkarte und
- Fig. 2 eine schematische Darstellung eines Stacks ist, der erfindungsgemäß
- 15 segmentiert ist.

- Figur 1 deutet schematisch einen tragbaren Datenträger 10 an, für den im folgenden die Gestalt einer Chipkarte angenommen wird. Auf der Chipkarte 10 befindet sich ein Mikrocontroller 32, bestehend aus einem Prozessor 12,
- 20 einer I/O-Einheit 14, einem flüchtigen Schreib-/Lesespeicher 16, insbesondere in Form eines RAMs 16 (Random Access Memory), einem nicht-flüchtigen Schreib-/Lesespeicher 18, insbesondere in Form eines EEPROM 18 (Electrical Erasable Programmable Read Only Memory), einem nicht-flüchtigen Nur-
- 25 lese-Speicher 20, insbesondere in Form eines ROMs 20 (Read Only Memory), einem in den Speichern 18 und 20 abgelegten Betriebssystem 22 sowie einem Daten-, Steuer- und Adreßbus 24. Jeder Speicher 16, 18, 20 umfaßt eine Vielzahl von 1-Bit-Speicherzellen, deren physische Größe vom Typ des Speichers abhängt.

Der RAM 16 ist der Arbeitsspeicher des Systems. Er zeichnet sich durch schnelle Zugriffszeiten aus, ist aber hinsichtlich der Größe der Speicherzellen platzintensiv und verliert seinen Inhalt, sobald die Versorgungsspannung, auch kurzzeitig, nicht mehr vorhanden ist. In gängigen, eine freie Programmierung in Hochsprache erlaubenden Chipkartensystemen mit 8-Bit Controller besteht der RAM z.B. aus einem 256 Byte großen internen IRAM, der um einen erweiterten, über Adreßrechnungen adressierbaren XRAM mit typisch 1600 Byte ergänzt wird. Üblicherweise ist der RAM 16 in verschiedene Bereiche gegliedert: einen Bereich für Register, einen Stack 26, einen Bereich für Variablen und einen Bereich für I/O-Puffer, sowie einen Arbeitsbereich für komplexe Algorithmen, wie beispielsweise kryptographische Algorithmen.

Der Stack 26 - als Teil des RAMs 16 - ist als LIFO-Datenstruktur implementiert. Weiterhin ist er in Segmente 30 gegliedert, die jeweils eine bestimmte Zahl von Zellen umfassen. Um eine Aussage über einen Füllungszustand des Stacks 26 zu erhalten, ist ein Stackpointer 28 vorgesehen, der auf den jeweils zuletzt gespeicherten Eintrag, insbesondere den untersten Eintrag, im Stack 26 zeigt.

Das ROM 20 behält die Daten, auch wenn die Versorgungsspannung nicht anliegt. Ein schreibender Zugriff auf diesen Speicherbaustein ist nicht möglich. Hier ist hauptsächlich das Betriebssystem abgelegt.

Das EEPROM 18 ist der aufwendigste Speicherbaustein. Er ist in Blöcke gegliedert, die jeweils eine bestimmte Zahl von Zellen umfassen. Bei jedem Schreibvorgang wird jeweils ein Block beschrieben. Die Daten bleiben im EEPROM auch ohne Stromzufuhr erhalten. Allerdings weist er sehr lange Zugriffszeiten auf, da die Schreibdauer für EEPROM-Blöcke weitgehend un-

abhängig von der Anzahl tatsächlich geschriebener Bytes ist. Gängige Schreibdauern liegen in der Größenordnung von 3 ms/Block.

5 In einer bevorzugten Ausführungsform der Erfindung ist eine Adressierung von jeweils ausgelagerten Stacksegmenten 30 über ein Ziel-Adress-Register gelöst, das vorzugsweise ebenfalls im RAM 16 abgelegt ist. Wird ein Segment 30 aus dem Stack 26 in das EEPROM 18 ausgelagert, so wird das mit der Anfangsadresse des EEPROMs 18 initialisierte Ziel-Adress-Register inkrementiert. Im hier angesetzten Fall wird etwa der Adresswert um die Größe des Segmentes, also um 2^{n-m} erhöht. Soll nun ein weiteres Segment 30 eingelagert werden, wird dieses Ziel-Adress-Register wiederum, um 2^{n-m} inkrementiert, so daß es immer aktuell auf das neueste Segment zeigt. Wird ein eingelagertes Segment 30 wieder aus dem EEPROM ausgelagert, um wieder im Stack 26 zur Verfügung zu stehen, wird das Ziel-Adress-Register 15 dekrementiert, im hier angesetzten Fall etwa um 2^{n-m} .

In einer alternativen Ausführungsform der Erfindung weist die Chipkarte neben den erwähnten Bauteilen noch zusätzliche Bauteile auf, wie eine hardwareunterstützte Speicherverwaltung, einen mathematischen Coprozessor, insbesondere für die Berechnung von Public-Key-Algorithmen, eine 20 Einheit zur Fehlererkennung und -korrektur im EEPROM, einem Zufallsgenerator etc.

In einer besonders effizienten Ausführungsform der Erfindung ist das Verfahren in Hardware realisiert und somit wesentlich schneller in der Ausführung als eine Software-Realisierung. Hierzu weist der Mikrocontroller 32 25 zusätzlich einen Pagecontroller auf. Der Pagecontroller ist als eine spezifische sogenannte Memory Managing Unit (MMU) ausgebildet, die Mittel

WO 02/19264

PCT/EP01/09664

- 13 -

zum Aus- und gegebenenfalls Einlagern von Stacksegmenten 30 nach dem erfindungsgemäßen Verfahren aufweist.

5 Ist das erfindungsgemäße Verfahren in Software realisiert, so ist der Prozessor 12 zusammen mit dem Betriebssystem 22 für den Transfer von Stacksegmenten 30 zuständig.

Nachfolgend wird unter Bezugnahme auf Figur 2 die Funktionsweise des erfindungsgemäßen Verfahrens beschrieben.

10

Bei Ablauf eines Programmes werden die jeweilige Adressen für den Rücksprung von den Unterprogrammen in die hierarchisch übergeordneten Unterprogramme oder in das Hauptprogramm zusammen mit aktuellen lokalen Variablen den Berechnungsergebnissen sowie den Aufrufparametern in den Stack 26 abgelegt. Je weiter das Programm fortfährt, desto größer wird der belegte Bereich im Stack 26. Damit trotzdem auch Programme ausführbar sind, die einen größeren verfügbaren Bereich im Stack 26 erfordern, als er eigentlich physikalisch vorhanden wäre, werden einzelne bereits beschriebene oder belegte Segmente 30 des Stacks 26 in einen größeren, aber langsameren Speicher, vorzugsweise den EEPROM 18 ausgelagert. Dabei wird vorteilhafterweise das am längsten zurückliegend beschriebene Segment 30 gewählt, das für die Programmausführung erst wieder möglichst spät benötigt wird.

25 Bei fortgeschrittenem Programmablauf befinden sich einige Segmente 30 in dem EEPROM 18. Ist nun wieder ausreichend Kapazität als freier Speicherplatz auf dem Stack 26 vorhanden und soll auf ein ausgelagertes Segment zugegriffen werden, werden die ausgelagerten Segmente 30 nacheinander - einzeln oder zusammen - wieder in den Stack 26 rückgelagert. Diese Rückla-

gerung wird vorteilhafterweise so ausgeführt, daß die Systemparameter - wie Dauer des Aus- und Einlagerns, Häufigkeit des Zugriffs auf das jeweilige Stackelement etc. - optimal berücksichtigt werden.

- 5 Die Auslagerung und ebenso die Rücklagerung auszulagernder Segmente erfolgen jeweils segmentweise in einer einzelnen, zusammenhängenden Transferoperation. Zweckmäßig erfolgen Aus- bzw. Wiedereinlagerung unter Verwendung bekannter Blocktransferanweisungen.
- 10 Durch geeignete Bestimmung eines Grenzwertes GW und der Größe der Segmentierung kann bei Mikrocontrollergestaltung mit zusätzlicher MMU vermieden werden, daß die CPU beispielsweise „gestallt“ wird, d.h. daß der Prozessor erst warten muß, bis das aktuell benötigte Stackelement wieder aus dem EEPROM in den Stack 26 zurück geladen wird.
- 15 Ein durch das Fehlen einer eigenen Energiequelle bedingte Systemeigenschaft von Chipkarten bildet die Möglichkeit, daß während einer laufenden Prozessierung, z.B. einem Schreiben von Daten in das EEPROM die Stromzufuhr der Karte unterbrochen wird, etwa durch Herausziehen aus dem Lesegerät. Der Zustand der augenblicklich prozessierten Daten kann dann unbestimmt sein. Daten können insbesondere verloren oder nur zum Teil auf den Speicher übertragen worden sein. Insbesondere können der im RAM befindliche Stack 26, der Stackpointer 28 und das Zieladressregister verloren gehen. Dadurch können unter anderem Dateien entstehen, die teilweise den
20 neuen Datensatz und gleichzeitig teilweise den alten Datensatz enthalten. Diese Inkonsistenz würde zu Anwendungsproblemen führen. Um solche Inkonsistenzen sicher zu vermeiden, werden bestimmte, besonders relevante Aktionen auf der Chipkarte in sog. atomare Abläufe strukturiert, das heißt in Abläufe, die nur nach dem binären Prinzip ganz oder gar nicht ausführbar
25

sind. Da die Hardware von Chipkarten atomare Abläufe nicht unterstützt, muß dieses Prinzip in Software realisiert werden. Dabei kann es notwendig werden, bestimmte Bereiche im RAM, insbesondere im Stack, erfindungsgemäß virtuell freizuhalten, um beispielsweise spezielle Datensätze als Ganzes und ungeteilt prozessieren zu können.

Ist das Chipkartensystem, insbesondere die Software im Mikrocontroller 32, ausgelegt, um atomare Operationen auszuführen, beispielsweise einen spezifischen Bytecode eines Java-Applets, dann kann in einer vorteilhaften alternativen Ausführungsform der Erfindung der Stackbedarf vorher ermittelt und beispielsweise in einer dafür vorgesehenen Tabelle abgelegt werden. Die Verwaltung der jeweils aus- und einzulagernden Segmente wird dann anhand dieser Tabelle ausgeführt.

Durch das erfindungsgemäße Verfahren wird bei einem physikalisch begrenzt verfügbarem CPU-Stack in vorteilhafter Weise ein Stack 26 mit theoretisch unbegrenzter Größe bereitgestellt.

Damit das Chipkartensystem beim Ablauf eines Programmes die Aus- und Einlagerung steuern kann, werden Informationen bereitgestellt, die den Füllungszustand des Stacks 26 des RAMs 16 angeben. Dies wird über einen Füllungsparameter, vorzugsweise in Form eines Stackpointers 28 realisiert, der immer auf die Speicherzellen des Stacks 26 zeigt, in die der letzte Eintrag erfolgt ist. Bei jedem neuen Eintrag im Stack 26 wird der Stackpointer 28 inkrementiert.

Wahlweise kann nun bestimmt werden, ab welchem Füllungszustand des Stacks 26 ein Stacksegment 30 ausgelagert werden soll. Abhängig von der jeweiligen Anwendung kann es zweckmäßig sein, daß nur sehr selten und

dafür relativ große Segmente 30 ausgelagert werden, z.B. bei einer Segmentierung in zwei Teile. Andererseits kann bisweilen eine kleinere Untergliederung sinnvoller sein, z.B. bei einer höheren Segmentierung, wobei häufiger kleinere Segmente ausgelagert werden.

5 Diese erweiterte Gestaltbarkeit des erfindungsgemäßen Verfahrens wird durch die Festlegung eines Grenzwertes GW ermöglicht. Er wird vorzugsweise so gewählt, daß eine optimale Balancierung des gesamten Chipkartensystems gewährleistet ist, d.h. daß die Zugriffszeiten auf den RAM und auf
10 den EEPROM 18 aufeinander abgestimmt sind und die CPU-Rechenzeit und der zusätzliche Speicherbedarf durch die Verwaltung, insbesondere die Adressierung der ausgelagerten Segmente 30, berücksichtigt ist. Der Grenzwert GW wird in einer ersten zweckmäßigen Einstellung auf die Größe eines Segmentes festgelegt, so daß bei Überschreiten der jeweiligen Segmentgrenze das älteste Segment 30 ausgelagert wird. Sehr zweckmäßig entspricht die
15 Größe eines Segments 30 genau der Größe eines Blockes im EEPROM 18. Die Übertragung der Stack-Segmente zwischen RAM 16 und EEPROM 18 kann dann zeitoptimiert mittels einer einzigen Blocktransferanweisung erfolgen. In einer weiteren zweckmäßigen Einstellung wird der Grenzwert GW so
20 eingestellt, daß er einem Vielfachen der Segmentgröße entspricht, so daß blockweise immer mehrere Segmente 30 gleichzeitig ausgelagert werden. Diese Segmente 30 können dann in gleicher Weise als Block von Segmenten 30 wieder in den Stack 26 eingelagert werden. Die Rücklagerung kann aber ebenso auch segmentweise einzeln erfolgen. Generell muß die Anzahl der in
25 einem Schritt ausgelagerten Segmente nicht der Anzahl von Segmenten entsprechen, die in einem Schritt wieder in den Stack 26 zurück transferiert wird. Lediglich die Summe über alle transferierten Segmente muß ausgeglichen sein.

- Durch Festlegung des Grenzwertes GW ist es auch möglich, die Lebensdauer des EEPROM 18-Speicherbausteins zu berücksichtigen. EEPROMs erlauben nur eine technisch begrenzte Anzahl von Schreib-/Löschzyklen, wobei die Speicherdauer kürzer wird, je öfter sie beschrieben und gelöscht werden.
- 5 Vorzugsweise wird der Grenzwert GW deshalb so gewählt, daß die Performance des Systems und deren Lebensdauer optimal aufeinander abgestimmt sind.

- In der bevorzugten Ausführungsform wird das Verhältnis zwischen Grenzwert GW und Stackpointer 28 so gewählt, daß es sich mit der Relation „Stackpointer 28 $\leq$ Grenzwert GW“ beschreiben läßt. Das heißt, sobald der Füllungszustand des Stacks 26 über die vorbestimmte Segmentgröße steigt, wird das jeweils älteste Stacksegment 30 ausgelagert. Wie die Auslagerung wird zweckmäßig auch die Wiedereinlagerung von Segmenten in den RAM 16 durch einen Grenzwert bestimmt. Dieser kann identisch mit GW sein. Der Grenzwert für die Wiedereinlagerung der Segmente 30 kann von dem Grenzwert GW aber auch verschieden sein, um die Variabilität des erfindungsgemäßen Verfahrens zu erhöhen. Weiter sind auch Abwandlungen dieses Verfahrens sinnvoll. Beispielsweise kann es bei kurzen Programmen vorteilhaft sein, daß zu Beginn des Programmes nicht sofort ausgelagert wird, sondern erst, wenn der Stackpointer 28 die nahezu vollständige Füllung des Stacks 26 bedeutet.
- 10
15
20

- Üblicherweise besteht der Mikrocontroller 32 in der hauptsächlichen Ausführungsform der erfindungsgemäßen Chipkarte 10 aus den Speicherbausteinen RAM 16, ROM 20 und EEPROM 18. Einzelne Blöcke des Stacks 26 innerhalb des RAMs 16 werden erfindungsgemäß in das EEPROM 18 ausgelagert. Alternativ dazu besteht der Mikrocontroller 32 aus zwei RAM-Bausteinen, dem RAM 16 und einem zusätzlichen RAM-Baustein, und einem
- 25

WO 02/19264

PCT/EP01/09664

- 18 -

ROM 20. Dabei ist der zusätzliche RAM-Baustein flächenoptimiert; er weist allerdings höhere Zugriffszeiten auf als das RAM 16. Erfindungsgemäß werden dann die Segmente 30 des im RAM 16 vorhandenen Stacks 26 in Speicherbereiche des zusätzlichen RAM-Bausteins ausgelagert.

Patentansprüche

1. Verfahren zur virtuellen Vergrößerung einer Datenstruktur eines flüchtigen Schreib-/Lesespeichers (16) auf einem tragbaren Datenträger, der als Stack (26) organisiert ist, wobei ein Füllparameter (GW), der einen Füllungs-
zustand für den Stack (26) beschreibt, erfaßt und durch Vergleich mit einem vorbestimmten Grenzwert (GW) überwacht wird, und wobei ein Teil der Datenstruktur in ein weiteres Speichermedium (18, 20) ausgelagert wird,
sobald ein vorbestimmtes Verhältnis zwischen Grenzwert (GW) und Füllungsparameter erreicht ist, dadurch gekennzeichnet, daß der Stack (26) in mindestens zwei Stacksegmente (30) segmentiert wird und die Auslagerung in das weitere Speichermedium (18, 20) segmentweise erfolgt.
2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß das weitere Speichermedium ein Schreib-/Lese-Speicher ist, insbesondere ein nicht-flüchtiger Schreib-/Lese-Speicher (18).
3. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß das Verfahren zusätzlich folgenden Schritt umfaßt:
- Einlagern des in Schritt e) ausgelagerten Stacksegmentes (30) in den Stack (26), sobald ein vorbestimmtes Verhältnis zwischen Füllungsparameter und einem zweiten Grenzwert erreicht ist, insbesondere sobald der Füllungsparameter kleiner ist als der zweite Grenzwert.
4. Verfahren nach einem der vorstehenden Ansprüche, gekennzeichnet durch folgenden zusätzlichen Schritt:
- Verwalten einer Ziel-Adresse des jeweils in den nicht flüchtigen Schreib-/Lese-Speicher (18) ausgelagerten Stacksegmentes (30), vorzugsweise in

- einem Register im flüchtigen Schreib-/Lese-Speicher (16), wobei die Ziel-Adresse bei jedem neu ausgelagerten Stacksegment (30) um ein ganzzahliges Vielfaches der Segmentgröße erhöht und bei jedem wieder in den Stack (26) eingelagerten Stacksegment (30) um ein ganzzahliges Vielfaches der Segmentgröße erniedrigt wird.
- 5
5. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß die Segmentierung in Segmente der Größe 2^{n-m} erfolgt, mit $m = 1, 2, 3, \dots$, wobei $2^m =$ Anzahl der Segmentierungen und $2^n =$ Größe des Stacks (26), und daß m so gewählt wird, daß eine optimale Balancierung des Datenträgersystems erreicht wird.
- 10
6. Verfahren nach Anspruch 5, dadurch gekennzeichnet, daß der Grenzwert (GW) und/oder der zweite Grenzwert mit der Größe des Segmentes (30), nämlich 2^{n-m} , oder mit einem ganzzahligen Vielfachen der Segmentgröße übereinstimmt.
- 15
7. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß sich das Verhältnis zwischen dem Grenzwert (GW) und dem Füllungsparameter in Schritt d) mit folgender Relation beschreiben läßt:
- 20
- „Füllungsparameter $\leq$ Grenzwert“.
8. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß die Chipkarte (10) ein Betriebssystem aufweist, das ein Nachladen von ausführbarem Programmcode ermöglicht.
- 25
9. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß das Aus- und/oder Einlagern eines Stacksegmentes (30) mittels bekannter Blocktransfer-Anweisungen erfolgt.

10. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß der Füllungsparameter als Stackpointer (28) ausgebildet ist.
- 5 11. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß in Schritt d) jeweils dasjenige Stacksegment (30) ausgelagert wird, das am ältesten ist.
12. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß das ausgelagerte Stacksegment (30) Rücksprung-
10 adressen von Unterprogrammen, lokale Variablen, Zwischenergebnisse und/oder Aufrufparameter zum Inhalt hat.
13. Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß eine Verwaltung des Stacks (26), insbesondere die Schritte a),
15 b), c), d) und gegebenenfalls die Schritte e) und f) in Hardware realisiert sind.
14. Tragbarer Datenträger (10) mit einem Microcontroller (32), zumindestens
20 bestehend aus einem Prozessor (12), einer I/O-Einheit (14), zumindest einem flüchtigen Schreib-/Lese-Speicher (16) mit einem Stack (26), zumindest einem nicht-flüchtigen Speicher (18, 20), einem Betriebssystem (22) und einem Daten-, Steuer- und Adreßbus (24), dadurch gekennzeichnet, daß das Betriebssystem (22) Mittel aufweist, um den Stack (26) zu segmentieren und um
25 bei Überschreiten eines definierten Füllungszustandes zumindest ein Segment (30) des flüchtigen Schreib-/Lese-Speichers (16) in den nicht-flüchtigen Speicher (18, 20) auszulagern.

WO 02/19264

PCT/EP01/09664

- 22 -

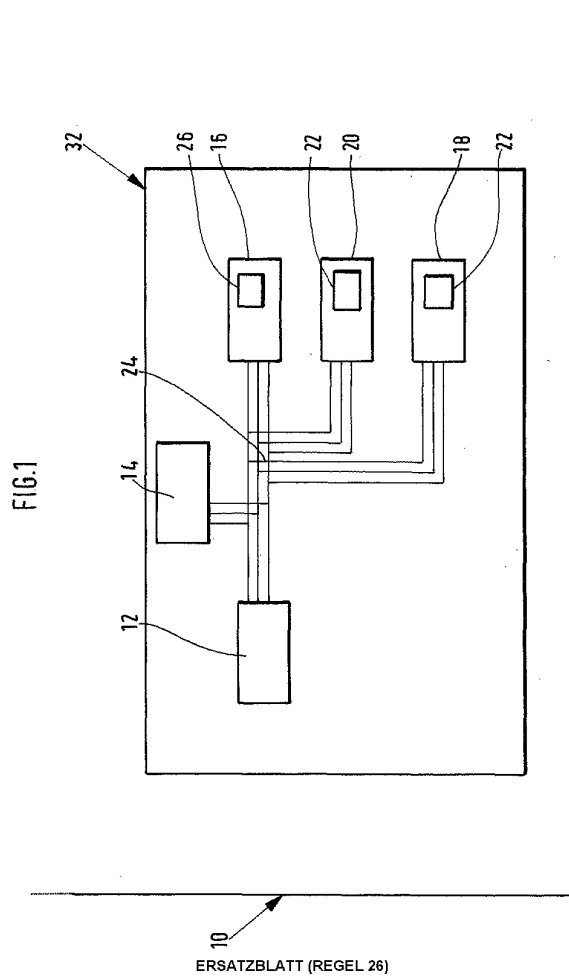
15. Tragbarer Datenträger nach A 14 nach Anspruch 14, dadurch gekennzeichnet, daß der nicht-flüchtige Speicher ein nicht-flüchtiger Schreib-/Lese-Speicher (18) ist.
- 5 16. Tragbarer Datenträger nach Anspruch 14 dadurch gekennzeichnet, daß der Mikrokontroller (32) für die Auslagerung eines Segmentes (30) in den nicht-flüchtigen Speicher (18, 20) zusätzlich einen Pagekontroller (34) aufweist, der zur Durchführung eines Verfahrens nach einem der Ansprüche 1 – 10 ausgelegt ist.

10

WO 02/19264

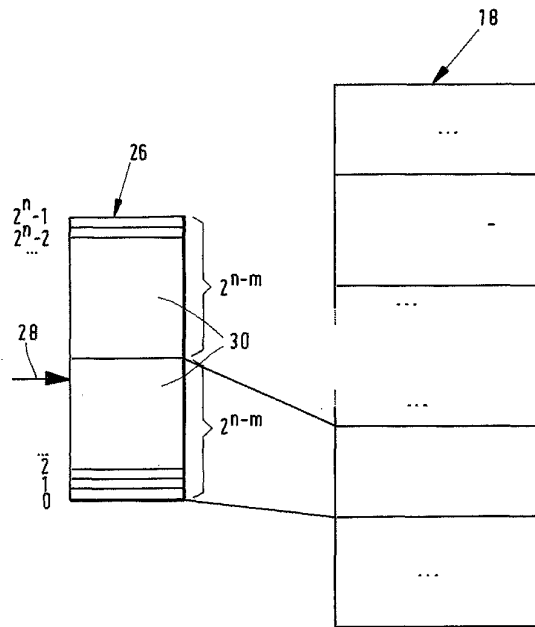
PCT/EP01/09664

1/2



2/2

FIG. 2



ERSATZBLATT (REGEL 26)

【国際調査報告】

INTERNATIONAL SEARCH REPORT		International Application No. PCT/EP 01/09664																					
A. CLASSIFICATION OF SUBJECT MATTER IPC 7 G06K19/07 G06K19/073 G07F7/10																							
According to International Patent Classification (IPC) or to both national classification and IPC																							
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC 7 G06K G07F																							
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched																							
Electronic data base consulted during the international search (name of data base and, where practical, search terms used) EPO-Internal, WPI Data, PAJ																							
C. DOCUMENTS CONSIDERED TO BE RELEVANT <table border="1"> <thead> <tr> <th>Category *</th> <th>Citation of document, with indication, where appropriate, of the relevant passages</th> <th>Relevant to claim No.</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>FR 2 777 673 A (BULL CP8) 22 October 1999 (1999-10-22) page 1, line 1 - line 8 page 14, line 19 - page 15, line 28</td> <td>1,2, 8-10,12</td> </tr> <tr> <td>A</td> <td>-----</td> <td>14</td> </tr> <tr> <td>Y</td> <td>WO 98 52160 A (MONDEX INT LTD) 19 November 1998 (1998-11-19) page 8, line 4 - line 10 page 9, line 8 - page 13, line 7 figures 2,3</td> <td>14,15</td> </tr> <tr> <td>A</td> <td>-----</td> <td>1</td> </tr> <tr> <td>Y</td> <td>DE 196 26 337 A (IBM) 9 January 1997 (1997-01-09) cited in the application column 3, line 28 - line 57</td> <td>14,15</td> </tr> <tr> <td></td> <td>----- -/-</td> <td></td> </tr> </tbody> </table>			Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.	X	FR 2 777 673 A (BULL CP8) 22 October 1999 (1999-10-22) page 1, line 1 - line 8 page 14, line 19 - page 15, line 28	1,2, 8-10,12	A	-----	14	Y	WO 98 52160 A (MONDEX INT LTD) 19 November 1998 (1998-11-19) page 8, line 4 - line 10 page 9, line 8 - page 13, line 7 figures 2,3	14,15	A	-----	1	Y	DE 196 26 337 A (IBM) 9 January 1997 (1997-01-09) cited in the application column 3, line 28 - line 57	14,15		----- -/-	
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.																					
X	FR 2 777 673 A (BULL CP8) 22 October 1999 (1999-10-22) page 1, line 1 - line 8 page 14, line 19 - page 15, line 28	1,2, 8-10,12																					
A	-----	14																					
Y	WO 98 52160 A (MONDEX INT LTD) 19 November 1998 (1998-11-19) page 8, line 4 - line 10 page 9, line 8 - page 13, line 7 figures 2,3	14,15																					
A	-----	1																					
Y	DE 196 26 337 A (IBM) 9 January 1997 (1997-01-09) cited in the application column 3, line 28 - line 57	14,15																					
	----- -/-																						
☒ Further documents are listed in the continuation of box C. ☒ Patent family members are listed in annex.																							
* Special categories of cited documents: *A* document defining the general state of the art which is not considered to be of particular relevance *E* earlier document but published on or after the international filing date *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) *O* document referring to an oral disclosure, use, exhibition or other means *P* document published prior to the international filing date but later than the priority date claimed *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art. *S* document member of the same patent family																							
Date of the actual completion of the international search 8 February 2002		Date of mailing of the international search report 14/02/2002																					
Name and mailing address of the ISA European Patent Office, P.O. 5318 Patentlaan 2 NL - 2200 HV Rijswijk Tel. (+31-70) 340-2000, Tx. 31 651 epo nl, Fax. (+31-70) 340-3016		Authorized officer Bhalodia, A																					

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT		International Application No. PCT/EP 01/09664
C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 354 793 A (HITACHI MAXELL) 14 February 1990 (1990-02-14) the whole document -----	1, 14
A	US 4 833 595 A (IIJIMA YASUO) 23 May 1989 (1989-05-23) column 3, line 42 -column 9, line 13; figures 7, 14 -----	1, 14

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

 International Application No.
 PCT/EP 01/09664

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
FR 2777673	A	22-10-1999	FR 2777673 A1
			22-10-1999
			AU 3154199 A
			01-11-1999
			BR 9906355 A
			19-09-2000
			CN 1272187 T
			01-11-2000
			EP 0990204 A2
WO 9852160	A	19-11-1998	05-04-2000
			WO 9953401 A2
			21-10-1999
			JP 2001502099 T
			13-02-2001
			NO 996177 A
			11-02-2000
			TW 432853 B
			01-05-2001
WO 9852160	A	19-11-1998	AU 736325 B2
			26-07-2001
			AU 6299698 A
			09-09-1998
			AU 7776798 A
			08-12-1998
			AU 7776898 A
			08-12-1998
			AU 7776998 A
			08-12-1998
			AU 7777098 A
			08-12-1998
			AU 7777198 A
			08-12-1998
			AU 7777298 A
			08-12-1998
			AU 7777398 A
			08-12-1998
			AU 7777498 A
			08-12-1998
			EP 0963580 A1
			15-12-1999
			EP 0981807 A2
			01-03-2000
			EP 0985202 A1
			15-03-2000
			EP 0985203 A1
			15-03-2000
			EP 0976114 A2
			02-02-2000
			EP 0985204 A1
			15-03-2000
			EP 0981805 A1
			01-03-2000
			WO 9837526 A1
			27-08-1998
			WO 9852158 A2
			19-11-1998
			WO 9852159 A2
			19-11-1998
			WO 9852160 A2
			19-11-1998
			WO 9852161 A2
			19-11-1998
			WO 9852152 A2
			19-11-1998
			WO 9852162 A2
			19-11-1998
			WO 9852163 A2
			19-11-1998
			WO 9852153 A2
			19-11-1998
			JP 2001513231 T
			28-08-2001
			JP 2001525956 T
			11-12-2001
			JP 2001525957 T
			11-12-2001
			JP 2001525958 T
			11-12-2001
			US 6220510 B1
			24-04-2001
			US 6230267 B1
			08-05-2001
			US 6164549 A
			26-12-2000
			US 6317832 B1
			13-11-2001
			US 6328217 B1
			11-12-2001
DE 19626337	A	09-01-1997	WO 9702543 A1
			23-01-1997
			DE 19626337 A1
			09-01-1997
			DE 59506921 D1
			28-10-1999
EP 0354793	A	14-02-1990	EP 0811204 A1
			10-12-1997
			JP 9510813 T
			28-10-1997
EP 0354793	A	14-02-1990	US 5809241 A
			15-09-1998
			JP 2050786 A
			20-02-1990
			JP 2050787 A
			20-02-1990
EP 0354793	A	14-02-1990	DE 68927361 D1
			28-11-1996
			DE 68927361 T2
			27-03-1997
EP 0354793	A	14-02-1990	EP 0354793 A2
			14-02-1990

Form PCT/ISA(210) (patent family annex) (July 1998)

INTERNATIONAL SEARCH REPORT Information on patent family members				International Application No. PCT/EP 01/09664	
Patent document cited in search report	Publication date	Patent family member(s)	Publication date	Patent family member(s)	Publication date
EP 0354793	A	US	5276903	A	04-01-1994
US 4833595	A	23-05-1989	JP	2537199 B2	25-09-1996
			JP	63000788 A	05-01-1988
			DE	3720394 A1	23-12-1987
			DE	3733794 A1	17-03-1988
			FR	2600446 A1	24-12-1987
			FR	2606909 A1	20-05-1988
			US	4901276 A	13-02-1990

INTERNATIONALER RECHERCHENBERICHT		In
		Internationales Aktenzeichen
A. KLASSIFIZIERUNG DES ANMELDUNGSGEGENSTANDES IPK 7 G06K19/07 G06K19/073 G07F7/10		PCT/EP 01/09664
Nach der Internationalen Patentklassifikation (IPK) oder nach der nationalen Klassifikation und der IPK		
B. RECHERCHIERTE GEBIETE Recherchierte Mindestprüfstoff (Klassifikationssystem und Klassifikationssymbole) IPK 7 G06K G07F		
Recherchierte aber nicht zum Mindestprüfstoff gehörende Veröffentlichungen, soweit diese unter die recherchierten Gebiete fallen		
Während der internationalen Recherche konsultierte elektronische Datenbank (Name der Datenbank und evtl. verwendete Suchbegriffe) EPO-Internal, WPI Data, PAJ		
C. ALS WESENTLICH ANGESEHENE UNTERLAGEN		
Kategorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
X	FR 2 777 673 A (BULL CP8) 22. Oktober 1999 (1999-10-22) Seite 1, Zeile 1 - Zeile 8 Seite 14, Zeile 19 - Seite 15, Zeile 28	1,2, 8-10,12
A	-----	14
Y	WO 98 52160 A (MONDEX INT LTD) 19. November 1998 (1998-11-19) Seite 8, Zeile 4 - Zeile 10 Seite 9, Zeile 8 - Seite 13, Zeile 7 Abbildungen 2,3	14,15
A	-----	1
Y	DE 196 26 337 A (IBM) 9. Januar 1997 (1997-01-09) in der Anmeldung erwähnt Spalte 3, Zeile 28 - Zeile 57 ----- -/-	14,15
☒ Weitere Veröffentlichungen sind der Fortsetzung von Feld C zu entnehmen ☒ Siehe Anhang Patentfamilie		
* Besondere Kategorien von angegebenen Veröffentlichungen : *A* Veröffentlichung, die den allgemeinen Stand der Technik definiert, aber nicht als besondere Vorlesung anzusehen ist *E* älteres Dokument, das jedoch erst am oder nach dem internationalen Anmeldedatum veröffentlicht worden ist *L* Veröffentlichung, die geeignet ist, einen Prioritätsanspruch zweifelsfrei erschließen zu lassen, oder durch die das Veröffentlichungsdatum einer anderen im Recherchenbericht genannten Veröffentlichung belegt werden soll oder die aus einem anderen besonderen Grund angegeben ist (wie ausgeführt) *O* Veröffentlichung, die sich auf eine methodische Offenbarung, eine Benutzung, eine Ausgestaltung oder andere Maßnahmen bezieht *P* Veröffentlichung, die vor dem internationalen Anmeldedatum, aber nach dem beanspruchten Prioritätsdatum veröffentlicht worden ist *T* Spätere Veröffentlichung, die nach dem internationalen Anmeldedatum oder dem Prioritätsdatum veröffentlicht worden ist und mit der Anmeldung nicht kollidiert, sondern nur zum Verständnis des der Erfindung zugrundeliegenden Prinzips oder der ihr zugrundeliegenden Theorie angegeben ist *X* Veröffentlichung von besonderer Bedeutung, die beanspruchte Erfindung kann allein aufgrund dieser Veröffentlichung, nicht als neu oder auf erfindungsfähiger Tätigkeit beruhend betrachtet werden *Y* Veröffentlichung von besonderer Bedeutung, die beanspruchte Erfindung kann nicht als auf erfindungsfähiger Tätigkeit beruhend betrachtet werden, wenn die Veröffentlichung mit einer oder mehreren anderen Veröffentlichungen dieser Kategorie in Verbindung gebracht wird und diese Verbindung für einen Fachmann naheliegend ist *Z* Veröffentlichung, die Mitglied derselben Patentfamilie ist		
Datum des Abschlusses der internationalen Recherche		Abschließdatum des internationalen Recherchenberichts
8. Februar 2002		14/02/2002
Name und Postanschrift der internationalen Recherchenbehörde Europäisches Patentamt, P.B. 5518 Patentlaan 2 NL - 2200 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax (+31-70) 340-3016		Bevollmächtigter Bediensteter Bhalodia, A

Formblatt PCT/ISA/210 (Juli 2) (Juli 1993)

INTERNATIONALER RECHERCHENBERICHT		in: anuales Aktenzeichen PCT/EP 01/09664
C.(Fortsetzung) ALS WESENTLICH ANGESEHENE UNTERLAGEN		
Kategorie	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
A	EP 0 354 793 A (HITACHI MAXELL) 14. Februar 1990 (1990-02-14) das ganze Dokument	1,14
A	US 4 833 595 A (IIJIMA YASUO) 23. Mai 1989 (1989-05-23) Spalte 3, Zeile 42 -Spalte 9, Zeile 13; Abbildungen 7,14	1,14

Formblatt PCT/ISA/210 (Fortsetzung von Blatt 2) (Juli 1992)

INTERNATIONALER RECHERCHENBERICHT				Internationales Aktenzeichen	
Angaben zu Veröffentlichungen, die zur selben Patentfamilie gehören				PCT/EP 01/09664	
Im Recherchenbericht angeführtes Patentdokument	Datum der Veröffentlichung	Mitglied(er) der Patentfamilie	Datum der Veröffentlichung		
FR 2777673	A	22-10-1999	FR 2777673 A1	22-10-1999	
			AU 3154199 A	01-11-1999	
			BR 9906355 A	19-09-2000	
			CN 1272187 T	01-11-2000	
			EP 0990204 A2	05-04-2000	
			WO 9953401 A2	21-10-1999	
			JP 2001502099 T	13-02-2001	
			NO 996177 A	11-02-2000	
			TW 432853 B	01-05-2001	
WO 9852160	A	19-11-1998	AU 736325 B2	26-07-2001	
			AU 6299698 A	09-09-1998	
			AU 7776798 A	08-12-1998	
			AU 7776898 A	08-12-1998	
			AU 7776998 A	08-12-1998	
			AU 7777098 A	08-12-1998	
			AU 7777198 A	08-12-1998	
			AU 7777298 A	08-12-1998	
			AU 7777398 A	08-12-1998	
			AU 7777498 A	08-12-1998	
			EP 0963580 A1	15-12-1999	
			EP 0981807 A2	01-03-2000	
			EP 0985202 A1	15-03-2000	
			EP 0985203 A1	15-03-2000	
			EP 0976114 A2	02-02-2000	
			EP 0985204 A1	15-03-2000	
			EP 0981805 A1	01-03-2000	
			WO 9837526 A1	27-08-1998	
			WO 9852158 A2	19-11-1998	
			WO 9852159 A2	19-11-1998	
			WO 9852160 A2	19-11-1998	
			WO 9852161 A2	19-11-1998	
			WO 9852152 A2	19-11-1998	
			WO 9852162 A2	19-11-1998	
			WO 9852163 A2	19-11-1998	
			WO 9852153 A2	19-11-1998	
			JP 2001513231 T	28-08-2001	
			JP 2001525956 T	11-12-2001	
			JP 2001525957 T	11-12-2001	
			JP 2001525958 T	11-12-2001	
			US 6220510 B1	24-04-2001	
			US 6230267 B1	08-05-2001	
			US 6164549 A	26-12-2000	
			US 6317832 B1	13-11-2001	
			US 6328217 B1	11-12-2001	
DE 19626337	A	09-01-1997	WO 9702543 A1	23-01-1997	
			DE 19626337 A1	09-01-1997	
			DE 59506921 D1	28-10-1999	
			EP 0811204 A1	10-12-1997	
			JP 9510813 T	28-10-1997	
EP 0354793	A	14-02-1990	US 5809241 A	15-09-1998	
			JP 2050786 A	20-02-1990	
			JP 2050787 A	20-02-1990	
			DE 68927361 D1	28-11-1996	
			DE 68927361 T2	27-03-1997	
			EP 0354793 A2	14-02-1990	

Formblatt PCT/ISA/E10 (Anhang Patentfamilie)(Juli 1992)

INTERNATIONALER RECHERCHENBERICHT
 Angaben zu Veröffentlichungen, die zur selben Patentfamilie gehören

 In: Internat. Abkürzungen
 PCT/EP 01/09664

Im Recherchanbericht angeführtes Patentdokument	Datum der Veröffentlichung	Mitglied(er) der Patentfamilie	Datum der Veröffentlichung
EP 0354793 A		US 5276903 A	04-01-1994
US 4833595 A	23-05-1989	JP 2537199 B2	25-09-1996
		JP 63000788 A	05-01-1988
		DE 3720394 A1	23-12-1987
		DE 3733794 A1	17-03-1988
		FR 2600446 A1	24-12-1987
		FR 2606909 A1	20-05-1988
		US 4901276 A	13-02-1990

Formblatt PCT/ISA/Z16 (Anhang Patentfamilie)(Juli 1992)

フロントページの続き

(81)指定国 AP(GH,GM,KE,LS,MW,MZ,SD,SL,SZ,TZ,UG,ZW),EA(AM,AZ,BY,KG,KZ,MD,RU,TJ,TM),EP(AT,BE,CH,CY,DE,DK,ES,FI,FR,GB,GR,IE,IT,LU,MC,NL,PT,SE,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AT,AU,AZ,BA,BB,BG,BR,BY,BZ,CA,CH,CN,CO,CR,CU,CZ,DK,DM,DZ,EC,EE,ES,FI,GB,GD,GE,GH,GM,HR,HU,ID,IL,IN,IS,JP,KE,KG,KP,KR,KZ,LC,LK,LR,LS,LT,LU,LV,MA,MD,MG,MK,MN,MW,MX,MZ,NO,NZ,PH,PL,PT,RO,RU,SD,SE,SG,SI,SK,SL,TJ,TM,TR,TT,TZ,UA,UG,US,UZ,VN,YU,ZA,ZW

(72)発明者 フライ, トーマス

ドイツ連邦共和国 8 5 5 6 0 エーベルスベルク グロースヴェネディッシャーシュトラッセ 6

(72)発明者 メルク, マルティーン

ドイツ連邦共和国 8 1 2 4 3 ミュンヘン シュヴァインフルターシュトラッセ 3アー

(72)発明者 シュトッケル, トーマス

ドイツ連邦共和国 8 1 6 7 3 ミュンヘン シルデンシュタインシュトラッセ 1 6

Fターム(参考) 5B035 BB09 BB11 CA11 CA29

5B060 AC07 BA11

5B076 BA05 BA06 BA10