

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2020年12月3日(03.12.2020)



(10) 国際公開番号

WO 2020/241570 A1

- (51) 国際特許分類:
G06F 8/30 (2018.01) *G06F 8/75* (2018.01)
G06F 8/41 (2018.01) *G06F 9/44* (2018.01)
- (21) 国際出願番号: PCT/JP2020/020541
- (22) 国際出願日: 2020年5月25日(25.05.2020)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
特願 2019-103138 2019年5月31日(31.05.2019) JP
- (71) 出願人: コネクトフリー株式会社 (**CONNECTFREE CORPORATION**) [JP/JP];
〒6008009 京都府京都市下京区四条烏丸西入ル函谷鉾町83番地 Kyoto (JP).
- (72) 発明者: 帝都 久利寿 (**TATE, Kristopher Andrew**); 〒6008009 京都府京都市下京区四

条烏丸西入ル函谷鉾町83番地 コネクトフリー株式会社内 Kyoto (JP).

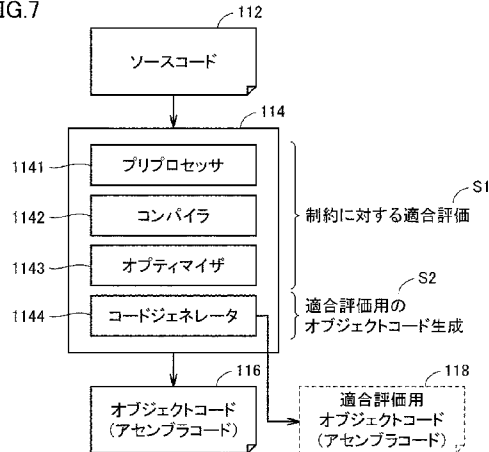
(74) 代理人: 特許業務法人深見特許事務所(**FUKAMI PATENT OFFICE, P.C.**); 〒5300005 大阪府大阪市北区中之島三丁目2番4号 中之島フェスティバルタワー・ウエスト Osaka (JP).

(81) 指定国(表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(54) **Title:** SOFTWARE DEVELOPMENT DEVICE AND SOFTWARE DEVELOPMENT PROGRAM

(54) 発明の名称: ソフトウェア開発装置およびソフトウェア開発プログラム

FIG.7



- 112 Source code
116 Object code (assembler code)
118 Conformity assessment object code (assembler code)
1141 Pre-processor
1142 Compiler
1143 Optimizer
1144 Code generator
S1 Conformity assessment for constraint
S2 Generation of conformity assessment object code

(57) **Abstract:** Provided is an environment making it possible to freely set a variety of constraints for a program to be run on an edge device or the like. This software development device for generating object code from source code comprises: an assessing means for extracting a constraint that is set for the source code, and assessing whether the source code conforms to that constraint within a scope of application of the extracted constraint; and a generating means for generating object code so as to conform to the constraint.

(57) 要約: エッジデバイスなどで実行されるプログラムに対して、様々な制約を自在に設定できる環境を提供する。ソースコードからオブジェクトコードを生成するソフトウェア開発装置は、ソースコードに設定される制約を抽出し、当該抽出した制約の適用範囲において、当該ソースコードが当該制約に適合しているか否かを評価する評価手段と、制約に適合するようにオブジェクトコードを生成する生成手段とを含む。

(84) 指定国(表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

添付公開書類:

一 国際調査報告 (条約第21条(3))

明 細 書

発明の名称：

ソフトウェア開発装置およびソフトウェア開発プログラム

技術分野

[0001] 本開示は、ソフトウェア開発装置およびソフトウェア開発プログラムに関する。

背景技術

[0002] 近年の情報通信技術（Information and Communication Technology：ICT）の進歩は目覚ましく、インターネットなどのネットワークに接続されるデバイスは、従来のパーソナルコンピュータやスマートフォンといった情報処理装置に限らず、様々なモノ（things）に広がっている。このような技術トレンドは、「IoT（Internet of Things；モノのインターネット）」と称され、様々な技術およびサービスが提案および実用化されつつある。将来的には、地球上の数十億人と数百億または数兆のデバイスとが同時につながる世界が想定されている。このようなネットワーク化された世界を実現するためには、よりシンプル、より安全、より自由につながることをできるソリューションを提供する必要がある。

[0003] IoTで利用されるデバイス（「エッジデバイス」とも称される。）のインテリジェント化に伴って、様々な種類のプログラムの作成が必要となっている。一方で、対象となるデバイスで利用可能なリソースは、パーソナルコンピュータなどに比較して制限されたものとなることが多い。

[0004] 利用可能なリソースを考慮してプログラムを作成する方法の一例として、特開2004-038956号公報は、様々なコンピューティングデバイスで利用可能なコンピューティングリソースを発見し示すための、またこうしたリソースをソフトウェアアプリケーションによってアドレス指定可能なサービスとして露出するためのシステムを開示する。

先行技術文献

特許文献

[0005] 特許文献1：特開2004-038956号公報

発明の概要

発明が解決しようとする課題

[0006] エッジデバイスで実行されるプログラムを作成するにあたっては、利用可能なリソースおよびセキュリティの観点から様々な点を考慮する必要がある。しかしながら、上記特許文献1は、対象のコンピューティングデバイスで利用可能なコンピューティングリソースを考慮してプログラムを作成することに着目するにすぎず、プログラムの作成にあたって、様々な点を考慮しなければならないといった課題に対するソリューションを提供するものではない。

課題を解決するための手段

[0007] 本開示のある形態に従えば、ソースコードからオブジェクトコードを生成するソフトウェア開発装置が提供される。ソフトウェア開発装置は、ソースコードに設定される制約を抽出し、当該抽出した制約の適用範囲において、当該ソースコードが当該制約に適合しているか否かを評価する評価手段と、制約に適合するようにオブジェクトコードを生成する生成手段とを含む。

[0008] 生成手段は、制約の適用範囲において、ソースコードが制約に適合していないと評価されると、オブジェクトコードの生成を中止するようにしてもよい。

[0009] 生成手段は、制約の適用範囲において、ソースコードが制約に適合しているか否かを評価できない場合に、ソースコードに対応するオブジェクトコードに加えて、当該オブジェクトコードの実行中に、当該制約に適合しているか否かを評価するための別のオブジェクトコードを生成するようにしてもよい。

[0010] 制約は、オブジェクトコードが実行時に利用するリソースに対する制限あるいは規則、オブジェクトコードの実行状態に対する制限あるいは規則、オ

ブジェクトコードの実行手順に対する制限あるいは規則、ならびに、ソースコードに含まれる命令に対する制限あるいは規則のうち、いずれか1つを含んでいてもよい。

[0011] 評価手段は、制約の適用範囲に呼び出し命令が含まれる場合に、当該呼び出し命令により呼び出される命令についても、制約に適合しているか否かを評価するようにしてもよい。

[0012] 本開示の別の形態に従えば、ソースコードからオブジェクトコードを生成するソフトウェア開発プログラムが提供される。ソフトウェア開発プログラムはコンピュータに、ソースコードに設定される制約を抽出し、当該抽出した制約の適用範囲において、当該ソースコードが当該制約に適合しているか否かを評価するステップと、制約に適合するようにオブジェクトコードを生成するステップとを実行させる。

発明の効果

[0013] 本開示によれば、エッジデバイスなどで実行されるプログラムに対して、様々な制約を自在に設定できる環境を提供できる。

図面の簡単な説明

[0014] [図1]本実施の形態に従うIoTシステムの全体構成の一例を示す模式図である。

[図2]本実施の形態に従うソフトウェア開発装置のハードウェア構成例を示す模式図である。

[図3]本実施の形態に従うコントローラのハードウェア構成例を示す模式図である。

[図4]本実施の形態に従うソフトウェア開発装置において設定可能な制約について説明するための図である。

[図5]本実施の形態に従うソフトウェア開発装置において設定可能な制約のスコープについて説明するための図である。

[図6]本実施の形態に従うソフトウェア開発装置において設定可能な制約の適合を実行時に判断しなければならない場合について説明するための図である。

。

[図7]本実施の形態に従うソフトウェア開発装置が提供する機能構成を示すブロック図である。

[図8]本実施の形態に従うソフトウェア開発装置における呼び出しに対する制約への適合を実現する方法を説明するための図である。

[図9]本実施の形態に従うソフトウェア開発装置におけるソースコードからオブジェクトコードを生成する処理手順を示すフローチャートである。

[図10]本実施の形態に従うソフトウェア開発装置が提供するユーザインターフェイス画面の一例を示す図である。

発明を実施するための形態

[0015] 本開示に係る実施の形態について、図面を参照しながら詳細に説明する。なお、図中の同一または相当部分については、同一符号を付してその説明は繰り返さない。

[0016] 以下の説明においては、典型例として、本実施の形態に従うソフトウェア開発装置100をIoTシステムに適用した場合について説明するが、本開示はIoTシステムに限らず任意のシステムおよびコントローラに適用可能である。

[0017] < A. IoTシステム1 >

まず、本実施の形態に従うソフトウェア開発装置100およびエッジデバイス2を含むIoTシステム1の全体構成について説明する。

[0018] 図1は、本実施の形態に従うIoTシステム1の全体構成の一例を示す模式図である。図1を参照して、IoTシステム1においては、典型的には、ソフトウェア開発装置100においてエッジデバイス2で実行されるプログラム（オブジェクトコード）が生成される。生成されたプログラムは、ソフトウェア開発装置100からエッジデバイス2に含まれるコントローラ200へ転送される。

[0019] ソフトウェア開発装置100には、統合開発環境（IDE：Integrated Development Environment）が提供されており、ユーザは統合開発環境上で任

意のプログラムを作成できる。すなわち、ソフトウェア開発装置１００は、ユーザが任意に作成するソースコードからオブジェクトコードを生成する。

[0020] エッジデバイス２としては、どのようなデバイスであってもよいが、典型的には、工場設備、家庭内の各種装置、社会インフラ設備、車両などの移動体、任意の携帯デバイスなどが想定される。後述するように、コントローラ２００は、プロセッサを有しており、ソフトウェア開発装置１００からのプログラムを実行可能になっている。

[0021] IoTシステム１における処理手順の一例について説明する。まず、ユーザがソフトウェア開発装置１００を用いてソースコードを作成する（（１）ソースコード作成）。そして、作成されたソースコードは、ソフトウェア開発装置１００においてコンパイルされ、オブジェクトコードが生成される（（２）オブジェクトコード生成）。生成されたオブジェクトコードは、エッジデバイス２のコントローラ２００へ転送される（（３）オブジェクトコード転送）。転送されたオブジェクトコードは、コントローラ２００で実行される（（４）オブジェクトコード実行）。

[0022] このような手順によって、ソフトウェア開発装置１００において開発された任意のプログラムをコントローラ２００において実行させることができる。

[0023] 後述するように、本実施の形態に従うソフトウェア開発装置１００は、コントローラ２００で実行されるプログラムに対して、様々な制約を自在に設定できる環境を提供する。典型的には、プログラムが実行されるコントローラ２００が有しているリソース、エッジデバイス２の種類および用途、実行されるプログラムの重要性および確保すべきセキュリティなどに応じて、プログラムに対して様々な制約を任意に設定できる。このような制約の設定によって、限られたリソースを利用したアプリケーションの実現や、アプリケーションの意図しない挙動の防止などを実現できる。

[0024] ＜Ｂ．ハードウェア構成例＞

次に、本実施の形態に従うIoTシステム１に含まれるデバイスのハード

ウェア構成例について説明する。

[0025] (b 1 : ソフトウェア開発装置 100)

ソフトウェア開発装置 100 は、典型的には汎用コンピュータで実現される。

[0026] 図 2 は、本実施の形態に従うソフトウェア開発装置 100 のハードウェア構成例を示す模式図である。図 2 を参照して、ソフトウェア開発装置 100 は、主たるコンポーネントとして、プロセッサ 102 と、メインメモリ 104 と、入力部 106 と、ディスプレイ 108 と、ハードディスク 110 と、通信インターフェイス 122 とを含む。これらのコンポーネントは、内部バス 120 を介して接続されている。

[0027] プロセッサ 102 は、例えば、CPU (Central Processing Unit) や GPU (Graphics Processing Unit) などで構成される。複数のプロセッサ 102 が配置されてもよいし、複数のコアを有するプロセッサ 102 を採用してもよい。

[0028] メインメモリ 104 は、DRAM (Dynamic Random Access Memory) や SRAM (Static Random Access Memory) などの揮発性記憶装置で構成される。ハードディスク 110 は、プロセッサ 102 で実行される各種プログラムや各種データを保持する。なお、ハードディスク 110 に代えて、SSD (Solid State Drive) やフラッシュメモリなどの不揮発性記憶装置を採用してもよい。ハードディスク 110 に格納されたプログラムのうち、指定されたプログラムがメインメモリ 104 上に展開され、プロセッサ 102 は、メインメモリ 104 上に展開されたプログラムに含まれるコンピュータ可読命令 (computer-readable instructions) を順次実行することで、後述するような各種機能を実現する。

[0029] 典型的には、ハードディスク 110 には、ユーザが任意に作成するソースコード 112 と、統合開発環境を実現するためのソフトウェア開発プログラム 114 と、ソースコード 112 から生成されるオブジェクトコード 116 とが格納される。ソフトウェア開発プログラム 114 は、ユーザが任意に作

成するソースコード 112 からオブジェクトコード 116 を生成するものであり、プログラムの開発環境を提供するモジュールを含む。

[0030] 入力部 106 は、ソフトウェア開発装置 100 を操作するユーザの入力操作を受け付ける。入力部 106 は、例えば、キーボード、マウス、表示デバイス上に配置されたタッチパネル、ソフトウェア開発装置 100 の筐体に配置された操作ボタンなどであってもよい。

[0031] ディスプレイ 108 は、プロセッサ 102 での処理結果などを表示する。ディスプレイ 108 は、例えば、LCD (Liquid Crystal Display) や有機 EL (Electro-Luminescence) ディスプレイなどであってもよい。

[0032] 通信インターフェイス 122 は、コントローラ 200 とのデータ交換を担当する。通信インターフェイス 122 は、例えば、USB (Universal Serial Bus) ポート、IEEE 1394 などのシリアルポート、レガシーなパラレルポートといった有線接続端子を含む。あるいは、通信インターフェイス 122 は、イーサネット (登録商標) ポートを含んでいてもよい。

[0033] なお、ソフトウェア開発装置 100 の全部または一部は、コンピュータ可読命令に相当する回路が組み込まれた ASIC (Application Specific Integrated Circuit) などのハードワイヤード回路を用いて実現してもよい。さらにあるいは、FPGA (field-programmable gate array) 上にコンピュータ可読命令に相当する回路を用いて実現してもよい。また、プロセッサ 102 およびメインメモリ、ASIC、FPGAなどを適宜組み合わせて実現してもよい。

[0034] ソフトウェア開発装置 100 は、コンピュータ可読命令を含むソフトウェア開発プログラム 114 を格納する非一過性 (non-transitory) のメディアから、当該格納しているプログラムなどを読み出すためのコンポーネントをさらに有していてもよい。メディアは、例えば、DVD (Digital Versatile Disc) などの光学メディア、USBメモリなどの半導体メディアなどであってもよい。

[0035] なお、ソフトウェア開発プログラム 114 は、メディアを介してソフトウ

エア開発装置１００にインストールされるだけでなく、ネットワーク上の配信サーバから提供されるようにしてもよい。

[0036] (b 2 : コントローラ 200)

コントローラ 200 は、汎用コンピュータを用いて実現してもよいし、処理を実現するために必要なコンポーネントを含む半導体基板を用いて実現してもよい。

[0037] 図 3 は、本実施の形態に従うコントローラ 200 のハードウェア構成例を示す模式図である。図 3 を参照して、コントローラ 200 は、主たるコンポーネントとして、演算処理部 210 と、無線通信モジュール 212 と、USB コントローラ 214 と、通信コントローラ 216 と、1 または複数のパッド 220 と電氣的に接続された I/O ドライバ 218 とを含む。

[0038] 演算処理部 210 は、プログラムを実行する演算部であり、主たるコンポーネントとして、プロセッサ 202 と、メインメモリ 204 と、フラッシュメモリ 206 とを含む。プロセッサ 202 は、例えば、CPU や GPU など構成される。複数のプロセッサ 202 が配置されてもよいし、複数のコアを有するプロセッサ 202 を採用してもよい。メインメモリ 204 は、DRAM や SRAM などの揮発性記憶装置で構成される。フラッシュメモリ 206 は、プロセッサ 202 で実行されるプログラムや必要なデータを保持する不揮発性記憶装置である。フラッシュメモリ 206 に格納されたプログラムのうち、指定されたプログラムがメインメモリ 204 上に展開されて、プロセッサ 202 により実行されることで、各種機能が実現される。

[0039] 無線通信モジュール 212 は、他の任意のデバイスとの間の無線によるデータ交換を担当する。無線通信モジュール 212 は、デバイス、ルータ、移動体基地局などと無線通信するための処理回路およびアンテナなどを含んでもよい。無線通信モジュール 212 が対応する無線通信は、例えば、Wi-Fi (登録商標)、Bluetooth (登録商標)、ZigBee (登録商標)、LPWA (Low Power Wide Area)、GSM (登録商標)、W-CDMA、CDMA 200、LTE (Long Term Evolution)、第 5 世代移

動通信システム（5G）のいずれであってもよい。

[0040] USBコントローラ214は、ソフトウェア開発装置100とのデータ交換を担当する。通信コントローラ216は、他の任意のデバイスとの間の有線によるデータ交換を担当する。通信コントローラ216は、例えば、シリアル通信、パラレル通信、GPIO（General-purpose input/output）などの公知のデータ交換方式に対応するようにしてもよい。

[0041] I/Oドライバ218は、パッド220を介して電氣的に接続された任意のデバイスとの間の電気信号の遣り取りを担当する。I/Oドライバ218は、演算処理部210からの指令に従って電気信号を出力する。また、I/Oドライバ218は、パッド220を介して与えられる電気信号を検知し、その検知結果を演算処理部210へ出力する。より具体的には、I/Oドライバ218は、信号生成回路、信号検知回路、バッファ回路などで構成される。

[0042] コントローラ200は、図示しないバッテリーからの電力により駆動されてもよい。

<C. 制約>

次に、本実施の形態に従うソフトウェア開発装置100において設定可能な制約について説明する。

[0043] 本明細書において、「制約」は、ソースコード112から生成されるオブジェクトコード116（アセンブラコード）の実行において遵守すべき規則を包含する。「制約」としては、オブジェクトコード116が実行時に利用するリソースに対する制限あるいは規則、オブジェクトコード116の実行状態に対する制限あるいは規則、オブジェクトコード116の実行手順に対する制限あるいは規則、ならびに、ソースコード112に含まれる命令に対する制限あるいは規則などを含み得る。

[0044] 図4は、本実施の形態に従うソフトウェア開発装置100において設定可能な制約について説明するための図である。図4（A）に示すソースコード112は、予め定義されたファンクション $f_{n1}()$ により決定される出力値を指定されたアドレスに書き込むためのコード例である。

- [0045] より具体的には、ソースコード112は、出力値変数の定義1121と、出力値変数の定義1121とを含む。ファンクション`fn1()`の戻り値が出力値`Out Value`にセットされる(命令1123)。そして、出力値`Out Value`の値がアドレス「0x1000」に書き込まれる(命令1124)。
- [0046] 図4(A)に示すソースコード112に対して、制約コード1125を追加することで、制約を設定できる。図4(B)に示すソースコード112においては、アクセス可能なメモリ範囲を指定するための制約コード1125が追加されている。例えば、「\$allowedAddressRange=0x0000...0xFFFF」は、「0x0000」～「0xFFFF」のアドレスに対してのみアクセスが可能であることを意味する。
- [0047] このような制約が設定された場合において、出力値`Out Value`の値をアドレス「0x1000」に書き込むための命令1124は、制約に適合しないことになる。すなわち、命令1124は実行不可となる。
- [0048] 図4(B)に示すような制約コード1125は、例えば、コントローラ200において、「0x0000」～「0xFFFF」のメモリ範囲はノンセキユア領域として設定されており、「0x1000」～「0x1FFF」のメモリ範囲はセキユア領域として設定されているような場合に有効である。
- [0049] 制約に適合しない命令の実行を禁止する典型的な手法として、(1)ソースコード112からオブジェクトコード116を生成する過程において判断する方法、および、(2)オブジェクトコード116の生成時に判断する方法が想定される。(1)の方法については、プリプロセッサ、コンパイラ、オプティマイザなどのオブジェクトコード116を生成するための機能が制約への適合を評価する。一方、(2)の方法については、ソースコード112から生成されるオブジェクトコード116に加えて、制約への適合を評価するためのオブジェクトコード(以下、「適合評価用オブジェクトコード」とも称す。)を生成するようにしてもよい。適合評価用オブジェクトコード118は、オブジェクトコード116の一部に組み込まれてもよいし、オブ

ジェクトコード 1 1 6 とは独立して存在してもよい。このような実装例の詳細については後述する。

[0050] 次に、本実施の形態に従う制約の適用範囲（以下、「スコープ」とも称す。）について説明する。

[0051] 図 4（B）に示すように、基本的には、制約コード 1 1 2 5 の記述に引き続く部分に対して、制約コード 1 1 2 5 に規定された制約が適用される。すなわち、制約コード 1 1 2 5 の記述に引き続く部分が制約のスコープとなる。この制約のスコープの終了位置は、任意に設定できるが、基本的には、制約コード 1 1 2 5 を含むカッコの範囲内を制約のスコープとすることができる。

[0052] また、プロシージャやファンクションが呼び出された場合には、それらの呼び出されたプロシージャやファンクションについても制約のスコープとしてもよい。

[0053] 図 5 は、本実施の形態に従うソフトウェア開発装置 1 0 0 において設定可能な制約のスコープについて説明するための図である。図 5 に示すように、ファンクション `f n 1 ()` を呼び出して、ファンクション `f n 1 ()` の戻り値を出力値 `Out Value` にセットするという命令 1 1 2 3 については、呼び出されるファンクション `f n 1 ()` の部分ソースコード 1 1 2 6 についても制約のスコープに含まれるようにしてもよい。

[0054] このように、ソフトウェア開発装置 1 0 0 は、制約のスコープに呼び出し命令が含まれる場合に、当該呼び出し命令により呼び出される命令（プロシージャやファンクション）についても、制約に適合しているか否かを評価する。このような制約のスコープを順次継承することで、呼び出されるプロシージャやファンクションを規定するソースコードについても制約への適合を評価することで、制約を確実に遵守できる。

[0055] 図 6 は、本実施の形態に従うソフトウェア開発装置 1 0 0 において設定可能な制約の適合を実行時に判断しなければならない場合について説明するための図である。図 6（A）には、図 4（B）に示すソースコード 1 1 2 と同

様のソースコード112を示す。図6（A）に示すソースコード112においては、出力値O u t V a l u eの値を書き込むアドレスが「0 x 1 0 0 0」に固定されており（命令1124）、ソースコード112を語句解析および構文解析することで、制約に適合していないと評価できる。

[0056] これに対して、図6（B）に示すソースコード112は、出力値O u t V a l u eの値を書き込むアドレスは出力アドレス初期値I n i P t rを用いて決定されるので、書き込むアドレスをソースコード112の解析だけでは一意に決定できない。

[0057] より具体的には、図6（B）に示すソースコード112においては、出力アドレス初期値変数の定義1127に加えて、出力アドレス初期値I n i P t rを決定する処理1128が規定されている。そして、出力アドレスO u t A d d r sは、出力アドレス初期値I n i P t rに「0 F 0 0」を加算して決定することを規定する出力アドレスを決定する命令1129がソースコード112に規定されている。そして、出力値O u t V a l u eの値が出力アドレスO u t A d d r sにより示されるアドレスに書き込まれる（命令1130）。

[0058] 図6（B）に示されるソースコード112においては、出力アドレスO u t A d d r sの値は、出力アドレス初期値I n i P t rの値に依存することになり、対応するオブジェクトコード116が実行される時点で、動的に決定されることになる。

[0059] 本実施の形態に従うソフトウェア開発装置100は、対応するオブジェクトコード116の実行中においても、制約への適合を評価できる仕組みを提供する（詳細については後述する）。

[0060] < D. 制約の種類 >

本実施の形態に従う制約としては、以下のような種類のものを採用してもよい。

[0061]

[表1]

No	制約コード	設定値	説明
1	\$allowedAddressRange	0x0000 ... 0xFFFF	アクセス可能なメモリ範囲の指定
2	\$allowedScopeDepth	usize (default: none)	スコープが有効なデプス指定 \$allowSelfRecursion は False に設定される
3	\$allowedStackDepth	usize (default: none)	スコープが有効なスタックのデプス指定 \$allowSelfRecursion は False に設定される
4	\$enableRuntimeSafety	True False (default: True)	ランタイムのセーフティ実行の有効/無効
5	\$enforceStrictSwitches	True False (default: True)	()ステートメント内でのすべてのオプション指定を強制/不強制
6	\$allowPanics	True False (default: False)	パニック発生の許容/不許容
7	\$allowUndefinedVariables	True False (default: False)	未定義変数の使用の可否
8	\$allowUnreachable	True False (default: False)	指定された宛先に到達できない状態の許容/不許容
9	\$allowNoReturns	True False (default: False)	戻り値が得られない状態の許容/不許容
10	\$clearMemoryUponEnd	true false (default: false)	スコープが終了するとメモリを完全クリアするか否か
11	\$allowInputToOutput	True False (default: True)	指定されたメモリのスコープからリターンの許容/不許容
12	\$allowConditionalBranches	True False (default: True)	条件分岐の使用の可否
13	\$allowSelfRecursion	True False (default: True)	再帰呼び出しの使用の可否
14	\$allowPtrToZero	True False (default: False)	0x0 を示すポインタの使用の可否
15	\$allowRawPointers	True False (default: False)	物理値を示すポインタの使用の可否
16	\$inlineFunctionCalls	True False (default: False)	呼び出す function のすべてをインライン展開することの強制/不強制
17	\$floatMode = @import("builtin").FloatMode	Strict/ Optimized (default: Strict)	Strict: IEEE コンプライアンスに沿った浮動小数点演算の実行 Optimized: 高速な数学処理への最適化
18	\$evalBranchQuota	Usize (default: 1000)	バックワード可能な最大数
19	\$allowSubWithOverflow	True False (default: True)	オーバーフローを伴う減算処理の許容/不許容
20	\$allowMulWithOverflow	True False (default: True)	オーバーフローを伴う掛算処理の許容/不許容
21	\$allowRem	True False (default: True)	剰余処理の許容/不許容
22	\$allowPtrCast	True False (default: True)	型変換処理の許容/不許容
23	\$allowMemset	True False (default: True)	メモリブロック設定の許容/不許容

[0062] [表2]

No	制約コード	設定値	説明
24	\$allowIntToPtr	True False (default: True)	ポインタへの整数設定の許容/不許容
25	\$allowBreakpoint	True False (default: True)	ブレークポイント使用の許容/不許容
26	\$allowBitCas	True False (default: True)	ビットキャスト演算子の許容/不許容
27	\$allowExternalCalls	True False (default: True)	外部関クションのコールの許容/不許容
28	\$allowExternalVariables	True False (default: True)	外部変数の許容/不許容
29	\$allowVarArgs	True False (default: True)	変数を引数する使用の許容/不許容
30	\$allowErrorTermination	True False (default: True)	スコープから生じるエラーストリームの許容/ 不許容
31	\$allowGlobalVariables	True False (default: True)	グローバル変数使用の許容/不許容
32	\$allowGlobalConstants	True False (default: True)	グローバル定数使用の許容/不許容
33	\$allowIntermediateVariables	True False (default: True)	中間変数/中間定数の使用の許容/不許容 *中間変数/中間定数は、直前のスコープお よび後続のグローバルスコープで使用される 構造体および列挙型の集合を意味する
34	\$allowIntermediateConstants	True False (default: True)	
35	\$allowSuperVariables	True False (default: True)	超変数/超定数の使用の許容/不許容 *超変数/超定数は、現在のスコープでのみ 使用される構造体および列挙型の集合を意 味する
36	\$allowSuperConstants	True False (default: True)	
37	\$allowLocalVariables	True False (default: True)	ローカル変数使用の許容/不許容
38	\$allowLocalConstants	True False (default: True)	ローカル定数使用の許容/不許容
39	\$allowThreadLocalVariables	True False (default: True)	ローカル変数のスレッド使用の許容/不許容
40	\$allowInlineAssembly	True False (default: True)	インライン展開されたアセンブリのスコープへ の適用可否
41	\$coldScope	True False (default: False)	オプティマイザにスコープの実行頻度が低い ことの通知有無
42	\$secureScope	True False	True に設定されることは以下を意味する \$allowInputToOutput を False に設定 \$clearMemoryOnExit を True に設定 \$allowConditionalBranches を False に設定 \$enableRuntimeSafety を True に設定 \$allowUnreachablePaths を False に設定 \$allowUndefinedVariables を False に設定 \$allowPanics を False に設定 \$allowFunctionRecursion を False に設定 \$floatMode を Strict に設定 \$allowErrorTermination を False に設定

[0063] 上述したような制約は、典型的には、オブジェクトコード１１６が実行時に利用するリソースに対する制限あるいは規則、オブジェクトコード１１６の実行状態に対する制限あるいは規則、オブジェクトコード１１６の実行手順に対する制限あるいは規則、ならびに、ソースコード１１２に含まれる命令に対する制限あるいは規則などを含み得る。

[0064] なお、上表に示す制約コードのすべてを実装する必要はなく、要求される仕様に応じてその一部のみを実装するようにしてもよい。また、上表に示す制約コード以外の制約コードを採用してもよい。

[0065] <Ｅ．制約への適合を評価する仕組み>

次に、本実施の形態に従うソフトウェア開発装置１００が制約への適合を評価する仕組みの一例について説明する。

[0066] 図７は、本実施の形態に従うソフトウェア開発装置１００が提供する機能構成を示すブロック図である。図７に示す各機能は、典型的には、ソフトウェア開発装置１００のプロセッサ１０２がソフトウェア開発プログラム１１４を実行することで実現される。

[0067] 図７を参照して、ソフトウェア開発プログラム１１４は、ソースコード１１２の入力を受けて、オブジェクトコード１１６（アセンブラコード）を生成する。より具体的には、ソフトウェア開発プログラム１１４は、プリプロセッサ１１４１と、コンパイラ１１４２と、オプティマイザ１１４３と、コードジェネレータ１１４４とを含む。

[0068] プリプロセッサ１１４１は、ソースコード１１２に対する語句解析および構文解析を実行するとともに、コンパイラ１１４２、オプティマイザ１１４３およびコードジェネレータ１１４４の挙動を制御する。

[0069] コンパイラ１１４２は、ソースコード１１２に対する語句解析および構文解析の結果に基づいて、オブジェクトコードに生成する。オプティマイザ１１４３は、生成されたオブジェクトコードを最適化する。コードジェネレータ１１４４は、オプティマイザ１１４３による最適化の結果に基づいて、最終的なオブジェクトコード１１６を出力する。

- [0070] 上述した制約への適合を評価するにあたって、プリプロセッサ 1141、コンパイラ 1142 およびオプティマイザ 1143 は、ソースコード 112 に規定された制約を抽出するとともに、抽出した制約への適合を評価する（ステップ S1）。なお、設定された制約の内容に応じて、オプティマイザ 1143 がオブジェクトコードを修正するようにしてもよい。このように、ソフトウェア開発装置 100 は、ソースコード 112 に設定される制約を抽出し、当該抽出した制約のスコープにおいて、当該ソースコード 112 が当該制約に適合しているか否かを評価する。
- [0071] コードジェネレータ 1144 は、抽出した制約への適合をオブジェクトコード 116 の実行時でなければ評価できない場合に、制約への適合を評価するためのアセンブラコードである適合評価用オブジェクトコード 118 を生成する（ステップ S2）。
- [0072] このように、ソフトウェア開発装置 100 は、制約に適合するようにオブジェクトコード 116 を生成する。
- [0073] 図 8 は、本実施の形態に従うソフトウェア開発装置 100 における呼び出しに対する制約への適合を実現する方法を説明するための図である。図 8 に示す処理は、ソフトウェア開発プログラム 114 により提供されてもよいし、オブジェクトコード 116 の実行時に適合評価用オブジェクトコード 118 により提供されてもよい。
- [0074] 図 8 を参照して、呼び出し元のプロシージャに設定された制約スコープ（以下、「親スコープ」とも称す。）を示すデータセット 150 が生成される。親スコープのデータセット 150 は、変数、制約、プロシージャなどの情報を含む。親スコープのデータセット 150 に対応付けて、管理オブジェクト 152 も生成される。
- [0075] 呼び出し先のプロシージャに継承されるべき制約スコープ（以下、「子スコープ」とも称す。）を示すデータセット 154 は、親スコープのデータセット 150 に関連付けられる。子スコープのデータセット 154 に対応付けて、管理オブジェクト 156 も生成される。

[0076] 例えば、プロシージャやファンクションが呼び出されると、親スコープのデータセット150に関連付けられる管理オブジェクト152が参照される（ステップS11）。そして、親スコープのデータセット150に基づいて子スコープのデータセット154が生成される（ステップS12）。子スコープのデータセット154から管理オブジェクト152への参照（ステップS13）に応答して、管理オブジェクト152から管理オブジェクト156が生成される（ステップS14）。生成された管理オブジェクト156は、子スコープのデータセット154と関連付けられる（ステップS15）。

[0077] このような一連の処理が繰り返されることで、プロシージャやファンクションが呼び出しに応じて、設定された制約のスコープが継承される。

[0078] < F. 処理手順 >

次に、本実施の形態に従うソフトウェア開発装置100におけるソースコードからオブジェクトコードを生成する処理手順について説明する。

[0079] 図9は、本実施の形態に従うソフトウェア開発装置100におけるソースコードからオブジェクトコードを生成する処理手順を示すフローチャートである。図9に示す各ステップは、典型的には、プロセッサ102がソフトウェア開発プログラム114を実行することで実現される。

[0080] 図9を参照して、ソフトウェア開発装置100は、入力されたソースコード112に対して語句解析および構文解析を実行する（ステップS100）。ソフトウェア開発装置100は、解析結果に基づいて、制約が設定されているか否かを判断する（ステップS102）。すなわち、ソフトウェア開発装置100は、ソースコード112に設定される制約を抽出し、当該抽出した制約のスコープにおいて、ソースコード112が当該制約に適合しているか否かを評価する。ここで、制約が設定されていないならば（ステップS102においてNO）、途中のステップがスキップされて、ステップS118以下の処理が実行される。

[0081] 制約が設定されていれば（ステップS102においてYES）、ソフトウェア開発装置100は、設定されている制約のうち1つを選択し（ステップ

S 1 0 4)、選択された制約のスコープに含まれるソースコードが当該制約に適合するか否かを判断する(ステップS 1 0 6)。

[0082] 選択された制約のスコープに含まれるソースコードのうち当該制約に適合しない部分があれば(ステップS 1 0 6においてN O)、ソフトウェア開発装置1 0 0は、制約に適合しない旨のメッセージを出力し(ステップS 1 0 8)、オブジェクトコード1 1 6を生成する処理を中断する(ステップS 1 1 0)。そして、処理は終了する。このように、ソフトウェア開発装置1 0 0は、制約のスコープにおいてソースコード1 1 2が当該制約に適合していないと評価されると、オブジェクトコード1 1 6の生成を中止する。

[0083] 図1 0は、本実施の形態に従うソフトウェア開発装置1 0 0が提供するユーザインターフェイス画面の一例を示す図である。図1 0を参照して、ユーザインターフェイス画面3 0 0は、ソースコードを作成するための編集領域3 0 2と、編集領域3 0 2において作成されたソースコードのコンパイルを開始するためのコンパイルボタン3 0 4と、エラーメッセージを表示するためのメッセージ表示領域3 0 6とを含む。制約に適合しない旨のメッセージは、メッセージ表示領域3 0 6に表示されてもよい。

[0084] 再度図9を参照して、選択された制約のスコープに含まれるソースコードが当該制約に適合するか否かを判断できなければ(ステップS 1 0 6において「不明」)、ソフトウェア開発装置1 0 0は、ソースコードが当該制約に適合するか否かを判断できない部分をマークする(ステップS 1 1 2)。そして、処理はステップS 1 1 4へ進む。

[0085] 選択された制約のスコープに含まれるすべてのソースコードが当該制約に適合していれば(ステップS 1 0 6においてY E S)、ソフトウェア開発装置1 0 0は、設定されているすべての制約についての評価が完了したか否かを判断する(ステップS 1 1 4)。設定されている制約のうち評価が完了していないものが残っていれば(ステップS 1 1 4においてN O)、ソフトウェア開発装置1 0 0は、未評価の制約のうち1つを選択し(ステップS 1 1 6)、ステップS 1 0 6以下の処理を実行する。

[0086] 設定されているすべての制約についての評価が完了していれば（ステップ S 1 1 4 において Y E S）、ソフトウェア開発装置 1 0 0 は、オブジェクトコード 1 1 6 を生成する（ステップ S 1 1 8）。

[0087] 続いて、ソフトウェア開発装置 1 0 0 は、ソースコードが当該制約に適合するか否かを判断できない部分がマークされているか否かを判断する（ステップ S 1 2 0）。すなわち、上述のステップ S 1 1 2 においていずれかの部分がマークされているか否かが判断される。

[0088] ソースコードが当該制約に適合するか否かを判断できない部分がマークされていなければ（ステップ S 1 2 0 において N O）、ステップ S 1 2 2 の処理はスキップされる。これに対して、ソースコードが当該制約に適合するか否かを判断できない部分がマークされていれば（ステップ S 1 2 0 において Y E S）、ソフトウェア開発装置 1 0 0 は、マークされた部分が実行時ににおいて設定された制約に適合するか否かを判断するための適合評価用オブジェクトコード 1 1 8 を生成する（ステップ S 1 2 2）。このように、ソフトウェア開発装置 1 0 0 は、制約のスコープにおいて、ソースコード 1 1 2 が制約に適合しているか否かを評価できない場合に、ソースコード 1 1 2 に対応するオブジェクトコード 1 1 6 に加えて、オブジェクトコード 1 1 6 の実行中に、当該制約に適合しているか否かを評価するための別のオブジェクトコード（適合評価用オブジェクトコード 1 1 8）を生成する。

[0089] 最終的に、ソフトウェア開発装置 1 0 0 は、生成したオブジェクトコードを出力する（ステップ S 1 2 4）。すなわち、ソフトウェア開発装置 1 0 0 は、ソースコード 1 1 2 に含まれる制約に適合するようにオブジェクトコードを生成する。そして、処理は終了する。

[0090] < G. 変形例 >

上述の説明においては、説明の便宜上、1 つの制約を設定する場合について例示したが、これに限らず複数の制約を重ねて設定することもできる。また、複数の制約をそれぞれのスコープを互いに一部重複させて設定することもできる。

[0091] また、上述の説明においては、ソースコード１１２に制約コードを埋め込む構成例について例示したが、これに限らず、ソースコード１１２とは別に制約を規定する定義ファイルを用意にしてもよい。この場合には、定義ファイルには、制約のスコープとなるプロシージャ名やファンクション名を特定する情報と、適用される制約の内容とを関連付けて含めるようにしてもよい。

[0092] このように、制約の設定方法については、任意の方法を採用できる。

＜H. 利点＞

本実施の形態に従うソフトウェア開発装置１００によれば、エッジデバイスなどで実行されるプログラムに対して、様々な制約を自在に設定できる環境を提供できる。これによって、利用可能なリソースおよびセキュリティの観点から様々な点を考慮した上で、エッジデバイスで実行されるプログラムを作成できる。

[0093] 今回開示された実施の形態はすべての点で例示であって制限的なものではないと考えられるべきである。本発明の範囲は上記した説明ではなくて請求の範囲によって示され、請求の範囲と均等の意味および範囲内でのすべての変更が含まれることが意図される。

符号の説明

[0094] １ １０Ｔシステム、２ エッジデバイス、１００ ソフトウェア開発装置、１０２、２０２ プロセッサ、１０４、２０４ メインメモリ、１０６ 入力部、１０８ ディスプレイ、１１０ ハードディスク、１１２ ソースコード、１１４ ソフトウェア開発プログラム、１１６ オブジェクトコード、１１８ 適合評価用オブジェクトコード、１２０ 内部バス、１２２ 通信インターフェイス、１５０、１５４ データセット、１５２、１５６ 管理オブジェクト、２００ コントローラ、２０６ フラッシュメモリ、２１０ 演算処理部、２１２ 無線通信モジュール、２１４ ＵＳＢコントローラ、２１６ 通信コントローラ、２１８ ドライバ、２２０ パッド、３００ ユーザインターフェイス画面、３０２ 編集領域、３０４ コンパ

イルボタン、306 メッセージ表示領域、1121, 1127 定義、1123, 1124, 1129, 1130 命令、1125 制約コード、1126 部分ソースコード、1128 処理、1141 プリプロセッサ、1142 コンパイラ、1143 オプティマイザ、1144 コードジェネレータ。

請求の範囲

- [請求項1] ソースコードからオブジェクトコードを生成するソフトウェア開発装置であって、
- ソースコードに設定される制約を抽出し、当該抽出した制約の適用範囲において、当該ソースコードが当該制約に適合しているか否かを評価する評価手段と、
- 前記制約に適合するようにオブジェクトコードを生成する生成手段とを備える、ソフトウェア開発装置。
- [請求項2] 前記生成手段は、前記制約の適用範囲において、前記ソースコードが前記制約に適合していないと評価されると、前記オブジェクトコードの生成を中止する、請求項1に記載のソフトウェア開発装置。
- [請求項3] 前記生成手段は、前記制約の適用範囲において、前記ソースコードが前記制約に適合しているか否かを評価できない場合に、前記ソースコードに対応するオブジェクトコードに加えて、当該オブジェクトコードの実行中に、当該制約に適合しているか否かを評価するための別のオブジェクトコードを生成する、請求項1または2に記載のソフトウェア開発装置。
- [請求項4] 前記制約は、前記オブジェクトコードが実行時に利用するリソースに対する制限あるいは規則、前記オブジェクトコードの実行状態に対する制限あるいは規則、前記オブジェクトコードの実行手順に対する制限あるいは規則、ならびに、前記ソースコードに含まれる命令に対する制限あるいは規則のうち、いずれか1つを含む、請求項1～3のいずれか1項に記載のソフトウェア開発装置。
- [請求項5] 前記評価手段は、前記制約の適用範囲に呼び出し命令が含まれる場合に、当該呼び出し命令により呼び出される命令についても、前記制約に適合しているか否かを評価する、請求項1～4のいずれか1項に記載のソフトウェア開発装置。
- [請求項6] ソースコードからオブジェクトコードを生成するソフトウェア開発

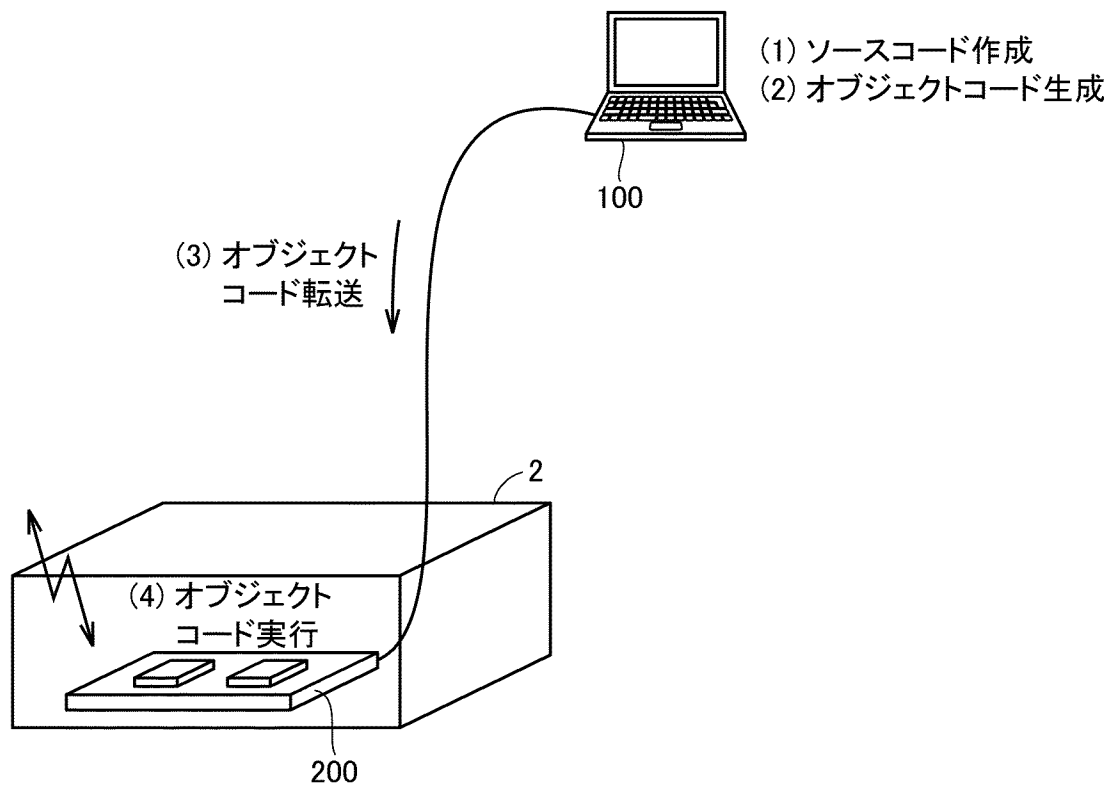
プログラムであって、前記ソフトウェア開発プログラムはコンピュータに

ソースコードに設定される制約を抽出し、当該抽出した制約の適用範囲において、当該ソースコードが当該制約に適合しているか否かを評価するステップと、

前記制約に適合するようにオブジェクトコードを生成するステップとを実行させる、ソフトウェア開発プログラム。

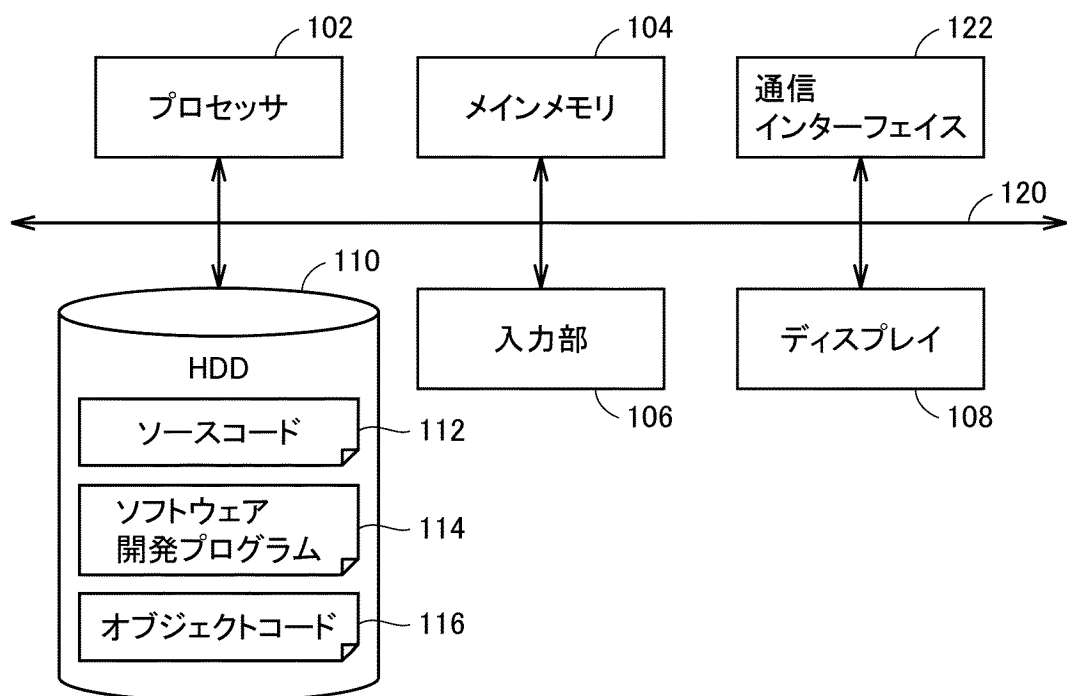
[図1]

FIG.1

1

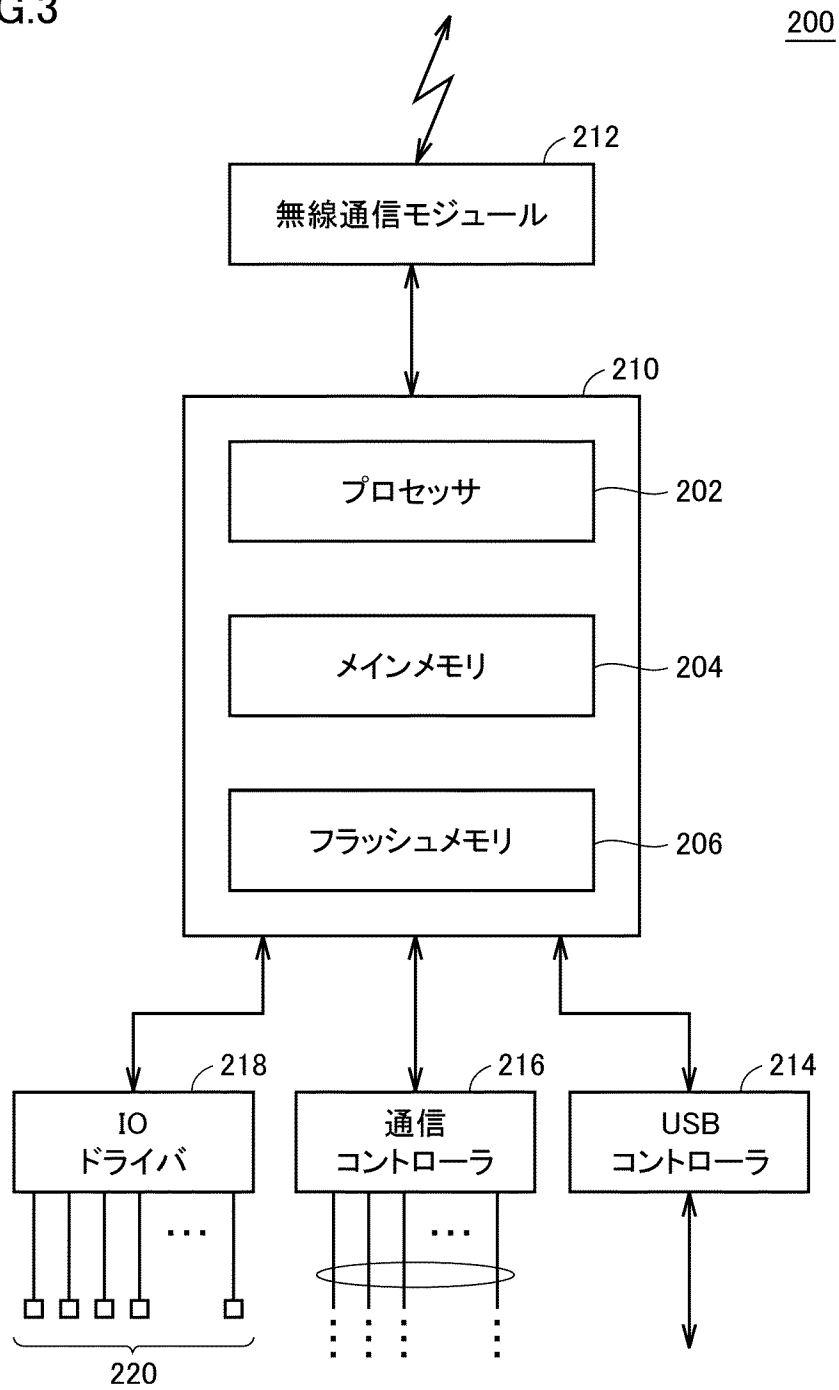
[図2]

FIG.2

100

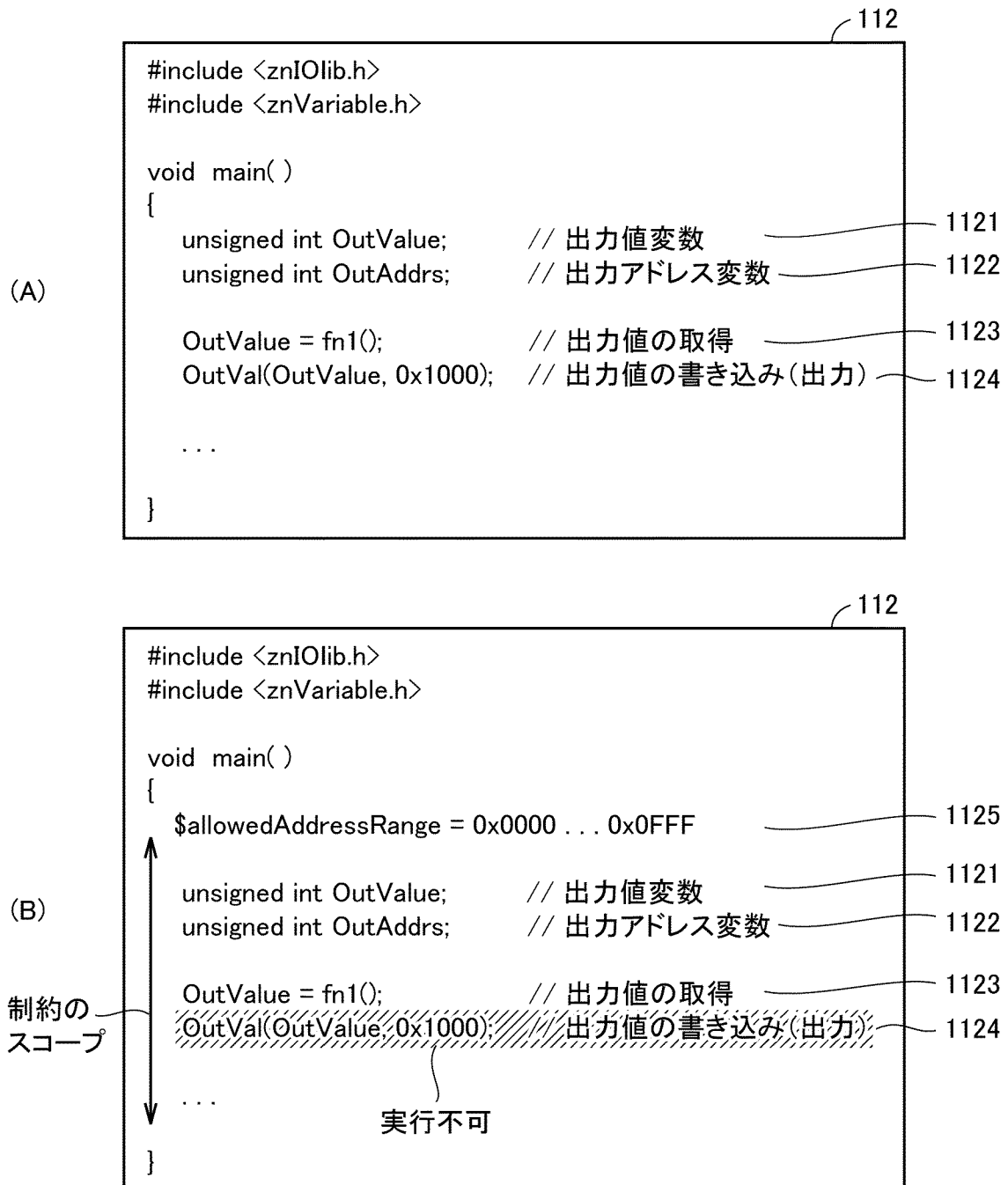
[図3]

FIG.3



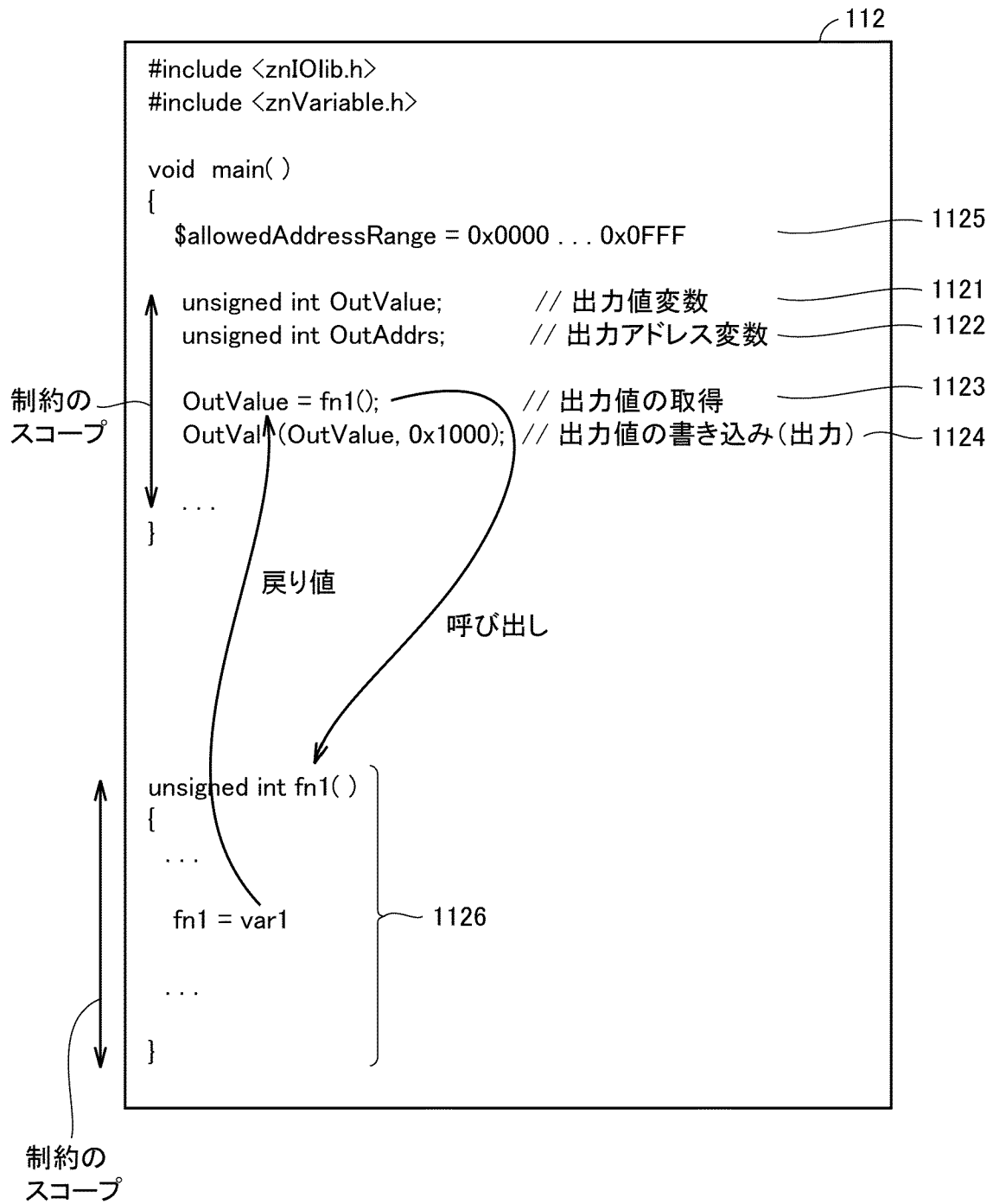
[図4]

FIG.4



[図5]

FIG.5



[図6]

FIG.6

(A)

```

#include <znIOlib.h>
#include <znVariable.h>

void main( )
{
    $allowedAddressRange = 0x0000 ... 0x0FFF 1125

    unsigned int OutValue;    // 出力値変数 1121
    unsigned int OutAddr;    // 出力アドレス変数 1122

    OutValue = fn1();        // 出力値の取得 1123
    OutVal(OutValue, 0x0100); // 出力値の書き込み(出力) 1124

    ...

}

```

(B)

```

#include <znIOlib.h>
#include <znVariable.h>

void main( )
{
    $allowedAddressRange = 0x0000 ... 0x0FFF 1125

    unsigned int OutValue;    // 出力値変数 1121
    unsigned int OutAddr;    // 出力アドレス変数 1122
    unsigned int IniPtr;      // 出力アドレス初期値変数 1127

    OutValue = fn1();        // 出力値の取得 1123

    ... //出力アドレス初期値IniPtrを決定する処理 1128

    OutAddr = IniPtr + 0F00;  // 出力アドレスの決定 1129

    OutVal(OutValue, OutAddr); // 出力値の書き込み(出力) 1130

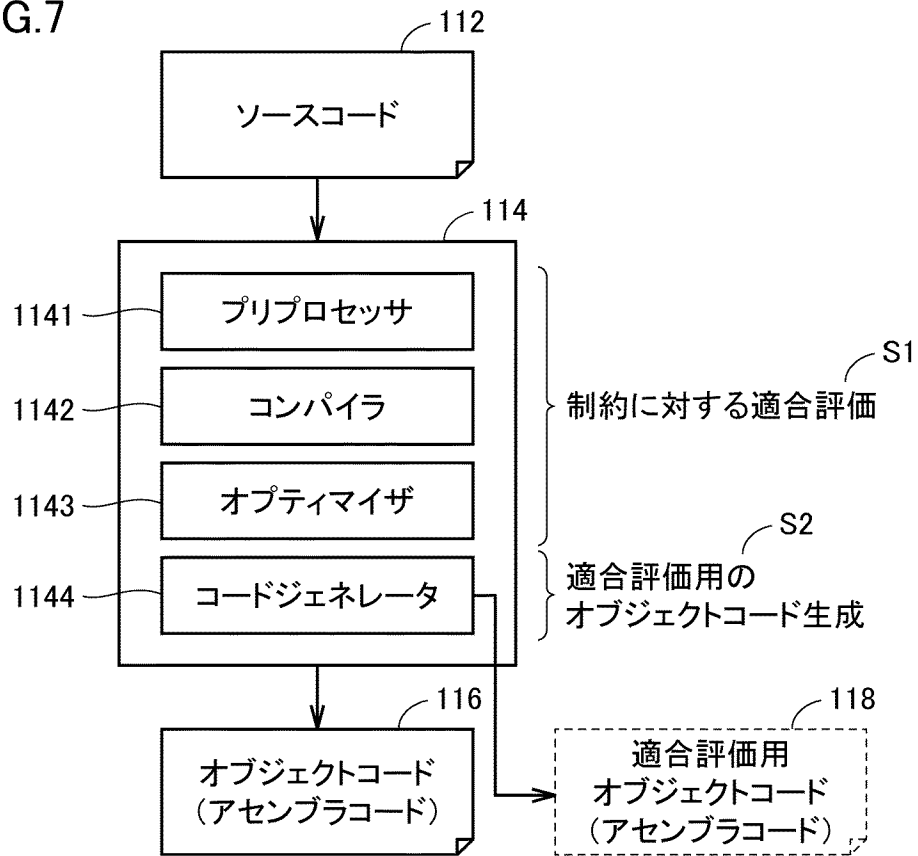
    ...

}

```

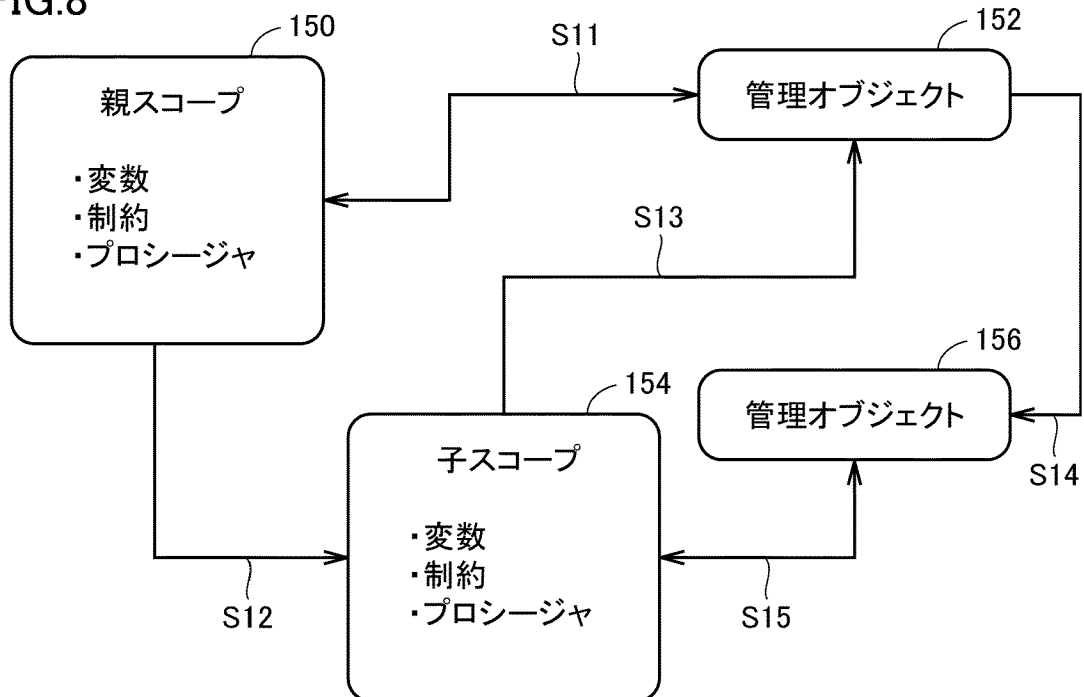
[図7]

FIG.7



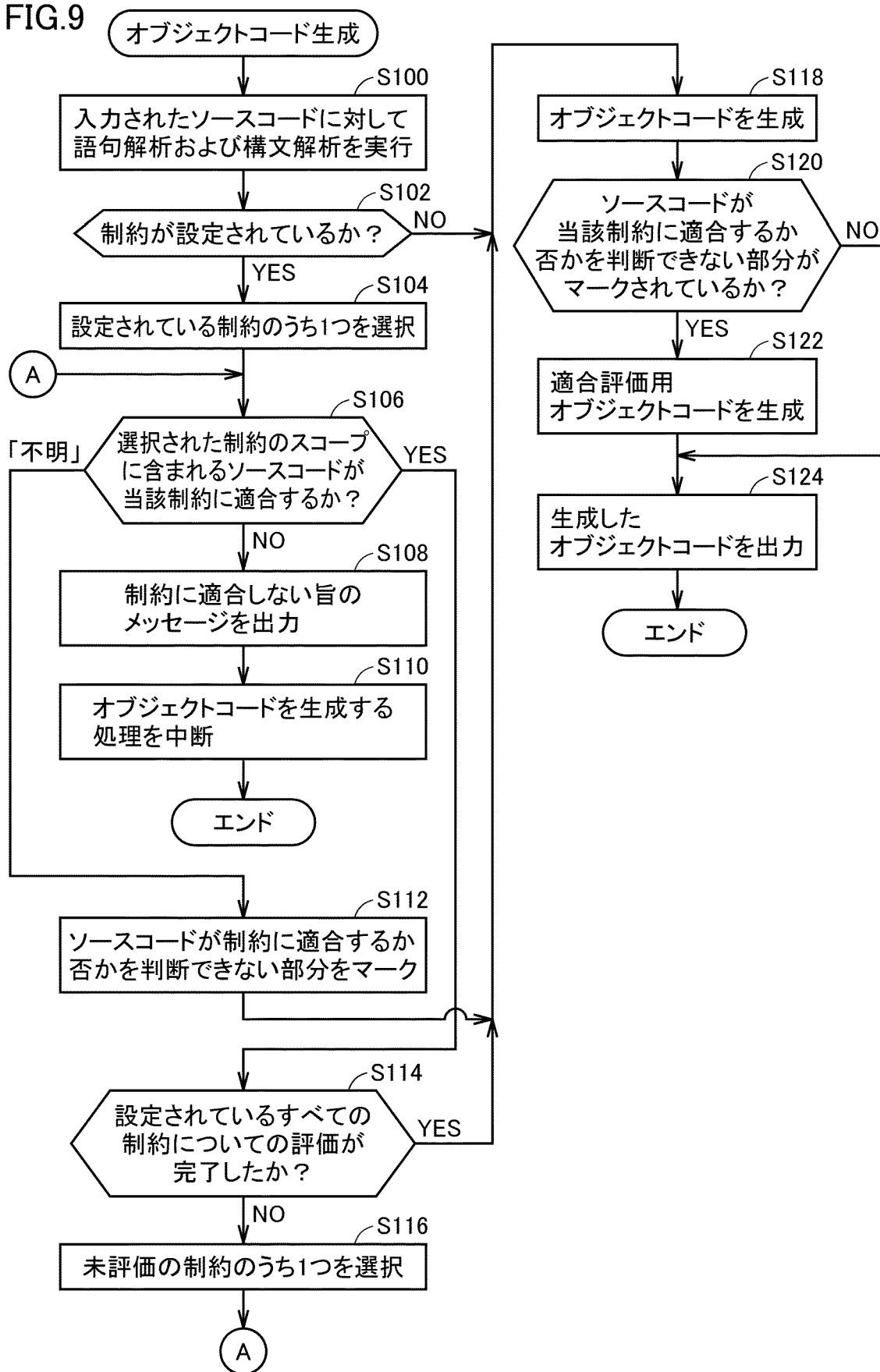
[図8]

FIG.8



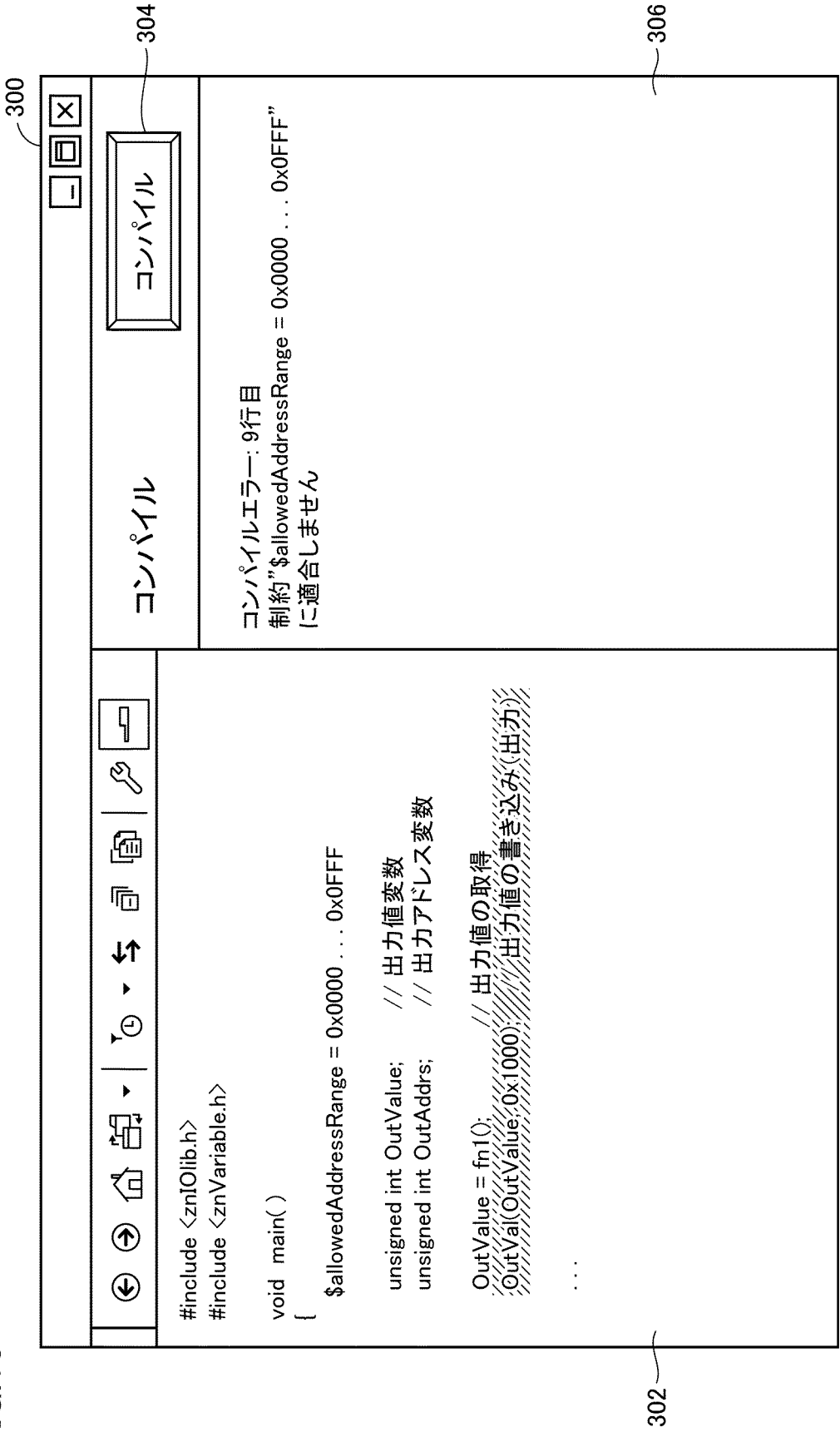
[図9]

FIG.9



[図10]

FIG.10



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2020/020541

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl. G06F8/30(2018.01) i, G06F8/41(2018.01) i, G06F8/75(2018.01) i,
G06F9/44(2018.01) i

FI: G06F9/44ZIT, G06F8/41170, G06F8/30, G06F8/75

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)

Int.Cl. G06F8/30, G06F8/41, G06F8/75, G06F9/44

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Published examined utility model applications of Japan 1922-1996

Published unexamined utility model applications of Japan 1971-2020

Registered utility model specifications of Japan 1996-2020

Published registered utility model applications of Japan 1994-2020

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	JP 2013-140513 A (FUJI ELECTRIC CO., LTD.)	1-2, 4-6
Y	18.07.2013 (2013-07-18), paragraphs [0010],	1-2, 4-6
A	[0036]-[0038], [0046], [0087], [0088], fig. 1, 2	3
Y	JP 2006-285582 A (CANON INC.) 19.10.2006 (2006-10-	1-2, 4-6
A	19), paragraphs [0023]-[0025], [0053]-[0062], fig. 3	3
A	JP 2007-122631 A (INTERNATIONAL BUSINESS MACHINES CORPORATION) 17.05.2007 (2007-05-17), paragraphs [0001]-[0007]	1-6

☐	Further documents are listed in the continuation of Box C.	☒	See patent family annex.
--------------------------	--	-------------------------------------	--------------------------

* Special categories of cited documents:	"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
"A" document defining the general state of the art which is not considered to be of particular relevance	"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
"E" earlier application or patent but published on or after the international filing date	"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
"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)	"&" document member of the same patent family
"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	

Date of the actual completion of the international search 15.07.2020	Date of mailing of the international search report 28.07.2020
Name and mailing address of the ISA/ Japan Patent Office 3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan	Authorized officer Telephone No.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/JP2020/020541

JP 2013-140513 A 18.07.2013 (Family: none)

JP 2006-285582 A 19.10.2006 (Family: none)

JP 2007-122631 A 17.05.2007 US 2017/0169020 A1
paragraphs [0002]-[0007]

A. 発明の属する分野の分類（国際特許分類（IPC）） G06F 8/30(2018.01)i; G06F 8/41(2018.01)i; G06F 8/75(2018.01)i; G06F 9/44(2018.01)i FI: G06F9/44 ZIT; G06F8/41 170; G06F8/30; G06F8/75										
B. 調査を行った分野										
調査を行った最小限資料（国際特許分類（IPC）） G06F8/30; G06F8/41; G06F8/75; G06F9/44										
最小限資料以外の資料で調査を行った分野に含まれるもの <table border="0"> <tr> <td>日本国実用新案公報</td> <td>1922-1996年</td> </tr> <tr> <td>日本国公開実用新案公報</td> <td>1971-2020年</td> </tr> <tr> <td>日本国実用新案登録公報</td> <td>1996-2020年</td> </tr> <tr> <td>日本国登録実用新案公報</td> <td>1994-2020年</td> </tr> </table>			日本国実用新案公報	1922-1996年	日本国公開実用新案公報	1971-2020年	日本国実用新案登録公報	1996-2020年	日本国登録実用新案公報	1994-2020年
日本国実用新案公報	1922-1996年									
日本国公開実用新案公報	1971-2020年									
日本国実用新案登録公報	1996-2020年									
日本国登録実用新案公報	1994-2020年									
国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）										
C. 関連すると認められる文献										
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号								
X	JP 2013-140513 A（富士電機株式会社）18.07.2013（2013-07-18） [0010], [0036]-[0038], [0046], [0087]-[0088], 図1-2	1-2, 4-6								
Y		1-2, 4-6								
A		3								
Y	JP 2006-285582 A（キヤノン株式会社）19.10.2006（2006-10-19） [0023]-[0025], [0053]-[0062], 図3	1-2, 4-6								
A		3								
A	JP 2007-122631 A（インターナショナル・ビジネス・マシーンズ・コーポレーション）17.05.2007（2007-05-17） [0001]-[0007]	1-6								
☐ C欄の続きにも文献が列挙されている。 ☒ パテントファミリーに関する別紙を参照。										
* 引用文献のカテゴリー “A” 特に関連のある文献ではなく、一般的な技術水準を示すもの “E” 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの “L” 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す） “O” 口頭による開示、使用、展示等に言及する文献 “P” 国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献 “T” 国際出願日又は優先日後に公表された文献であって出願と抵触するものではなく、発明の原理又は理論の理解のために引用するもの “X” 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの “Y” 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの “&” 同一パテントファミリー文献										
国際調査を完了した日 15.07.2020		国際調査報告の発送日 28.07.2020								
名称及びあて先 日本国特許庁(ISA/JP) 〒100-8915 日本国 東京都千代田区霞が関三丁目4番3号		権限のある職員（特許庁審査官） 渡部 博樹 5B 3868 電話番号 03-3581-1101 内線 3545								

国際調査報告
パテントファミリーに関する情報

国際出願番号

PCT/JP2020/020541

引用文献			公表日	パテントファミリー文献	公表日
JP	2013-140513	A	18.07.2013	(ファミリーなし)	
JP	2006-285582	A	19.10.2006	(ファミリーなし)	
JP	2007-122631	A	17.05.2007	US 2017/0169020 A1	
				[0002]-[0007]	