

コンピュータの仕組み (9)

柴山 潔

コンピュータの仕組み

- 1 コンピュータシステム
- 2 ハードウェア
- 3 内部装置
- 4 プロセッサ(6)
- 5 メモリ
- 6 外部装置
- 7 論理回路
- 8 オペレーティングシステム

4 プロセッサ(6)

4.1 プロセッサの動作とその制御

4.2 プロセッサの機能 —演算と制御—

4.3 演算器

4.4 順序制御機構(2)

分岐命令

= ジャンプ(jump)命令

- 別のアドレスに格納されているマシン命令を実行(特別なアドレスをPCに設定)
 - 並び順ではなく, まったく別のアドレスに格納されている命令の実行に移行
 - PCをカウントアップするのではなく, オペランドで指定した「実行したいマシン命令アドレス」をPCに直接設定

分岐と分岐命令

➤ 順序制御の基本は分岐

- **分岐**: 並び順実行を何らかの理由で**変更**
- **分岐命令**: 分岐を指示する**順序制御命令**

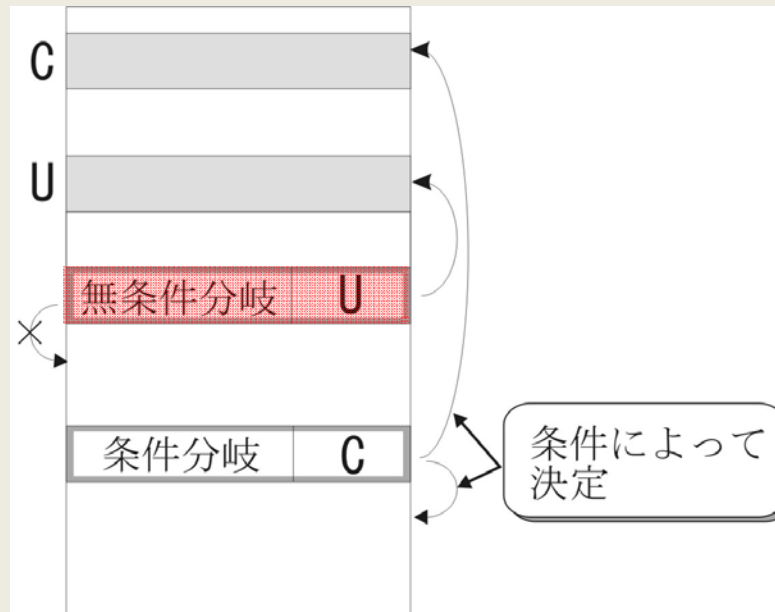
分岐命令の分類

■ 分岐する理由の有無による分岐命令の分類

1. **無条件分岐**: 強制的に, 並び順実行を中止, 別のマシン命令列の実行に移行
2. **2方向条件分岐**: 条件によって, 次のどちらかを選択
 - (1) **分岐不成立**: そのまま並び順で実行を続行
 - (2) **分岐成立**: 並び順実行を中止, 別のマシン命令列の実行に移行(無条件分岐と同じ)

無条件分岐

- オペランドは分岐先アドレス (図では, "U")
- 順序制御機構では, 無条件分岐命令のオペランドとして取り出した分岐先アドレスをPCに設定



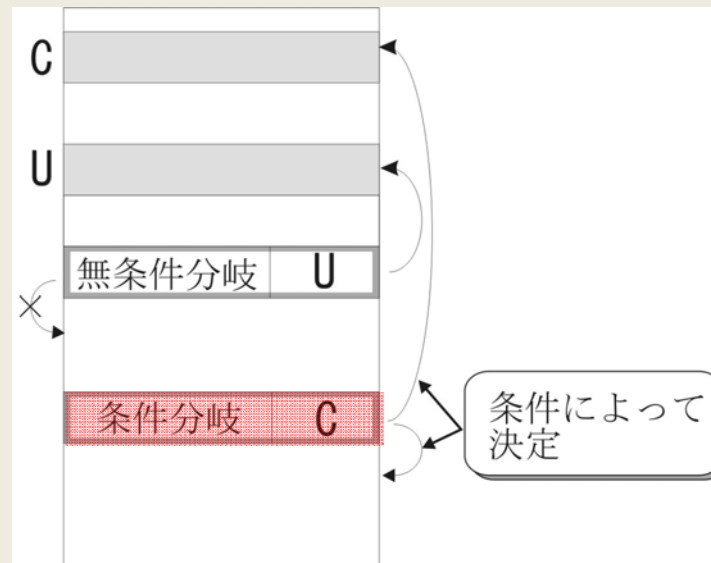
分岐命令の分類

■ 分岐する理由の有無による分岐命令の分類

1. **無条件分岐**: 強制的に, 並び順実行を中止, 別のマシン命令列の実行に移行
2. **2方向条件分岐**: 条件によって, 次のどちらかを選択
 - (1) **分岐不成立**: そのまま並び順で実行を続行
 - (2) **分岐成立**: 並び順実行を中止, 別のマシン命令列の実行に移行(無条件分岐と同じ)

2方向条件分岐(1)

- 順序制御機構としては, (1)分岐**不成立**: PCを単にカウントアップ; (2)分岐**成立**: オペランドで指定した**分岐先アドレス("C")**をPCに設定(=無条件分岐と同じ)



2方向条件分岐(2)

- 2方向条件分岐の成立/不成立かの判定条件 = 論理値 ("1"/"0")
 - (a) 関係演算 (例: ゼロ/正負判定) の結果
 - (b) フラグ(flag) = ユーザが設定/利用可能な論理値(記号)
 - (c) ハードウェア信号 (例: オーバフロー/ゼロ除算検知, 入出力装置の状態信号)

分岐と分岐元プログラム

- 分岐(無条件, 条件) → いったん分岐すると, 順序制御機構(プログラムカウンタ(PC))は分岐元プログラム(マシン命令列)とは縁を切る
- プログラミングにおける具体的な順序制御例の6.「実行中のプログラムの途中で別のプログラムを呼んだり, そこから戻ったりする」順序制御
 - 途中で割り込ませた分岐先プログラムが終了すれば, 割り込まれた分岐元プログラムへ戻る必要

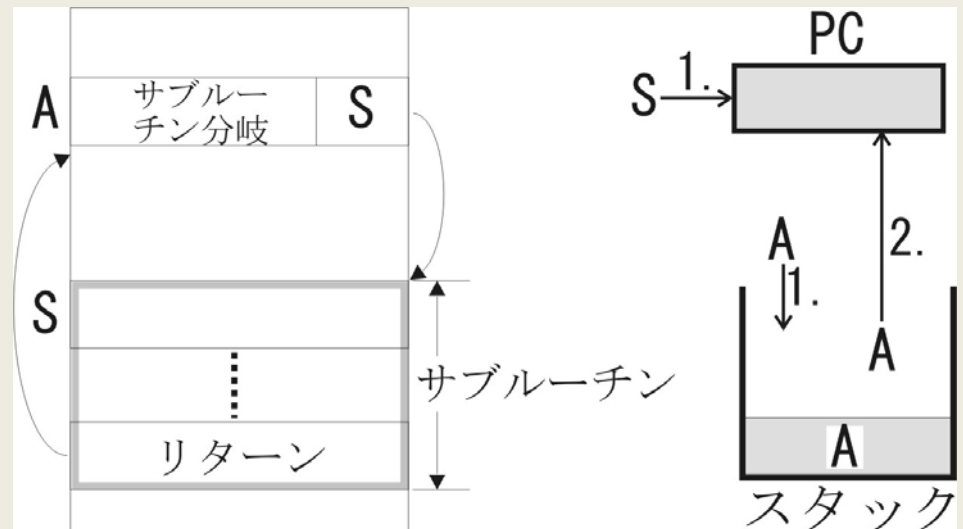
プログラミングにおける具体的な順序制御例(2)

5. 何回も同じプログラムをくり返し実行(くり返し)
6. プログラムの実行途中でいったん別のプログラムを呼ぶ, そこから戻る
7. あるプログラムを複数のコンピュータや人間によって共用
8. あらかじめ予測できないこと(不測の事態)に対処

サブルーチン分岐

- あるプログラムの途中で、いったん**サブルーチン (subroutine)**という別のプログラムの実行を割り込ませ、それが**終了**すれば、割り込まれた元のプログラムの実行に**戻る**ような順序制御

- サブルーチン** = 7.「複数のプログラムで**共用**する部分プログラム」



サブルーチン分岐命令

=サブルーチン呼び出し命令

- サブルーチンへの分岐を実行するマシン命令

サブルーチン分岐命令の種類

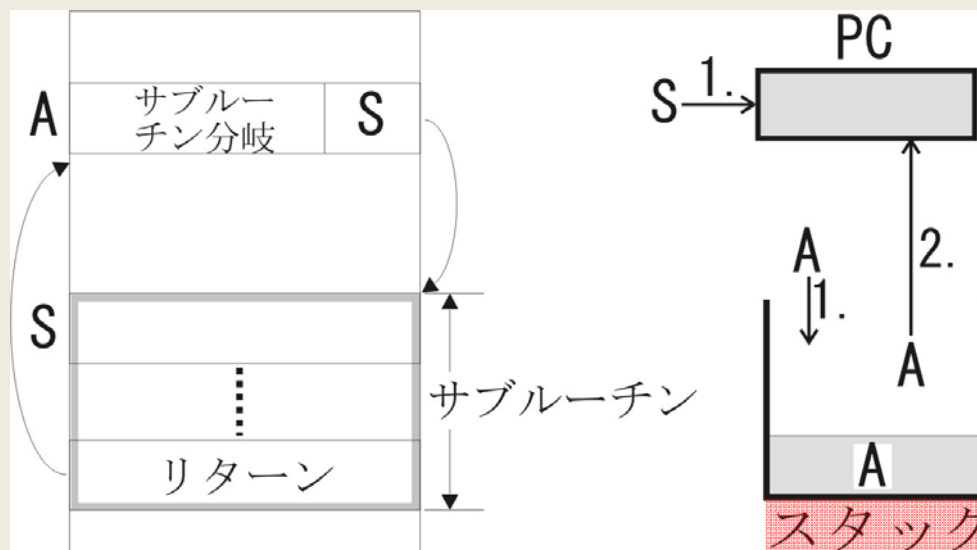
- (1) 無条件サブルーチン分岐: 強制的にサブルーチン分岐
- (2) 条件サブルーチン分岐: 条件によってサブルーチン分岐するか否かを決定

リターン(return; 戻り, 復帰)

- 「分岐元へ戻る」こと
- リターン命令: リターンを行う順序制御命令
 - サブルーチンの最後に実行するマシン命令, 「分岐元へ戻る無条件分岐命令」
 - 分岐元のサブルーチン分岐命令とそれによる分岐先プログラム(サブルーチン)のリターン命令とは対

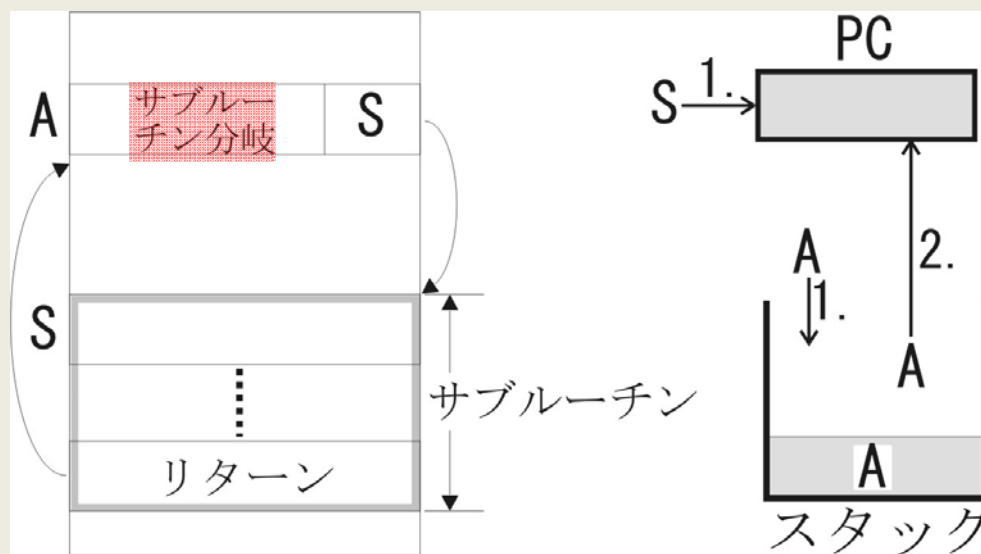
サブルーチン分岐機能の実現

- **スタック(stack)**: サブルーチンからの**戻り(分岐元)アドレス**をサブルーチン実行中に**退避**しておくハードウェア機構
 - 出し入れ共用の口(**出入り口**)を**1つだけ**持つバケツのような入れ物



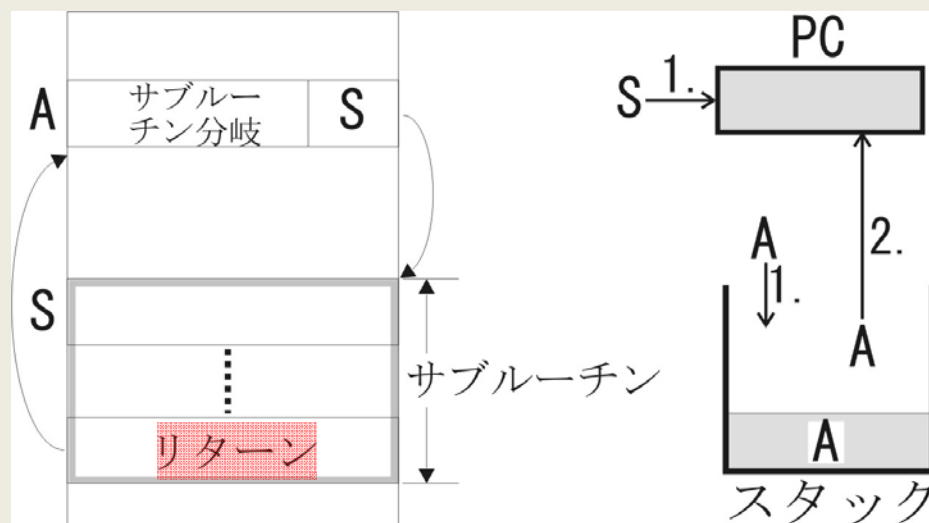
サブルーチン分岐でのスタックの動作

1. サブルーチン分岐命令 (= サブルーチンに分岐): スタックに現在のマシン命令 (分岐元) アドレス (図では“A”) を退避, サブルーチン分岐命令のオペランドであるサブルーチン (分岐先) アドレス (“S”) をPCに設定 (分岐)



リターンでのスタックの動作

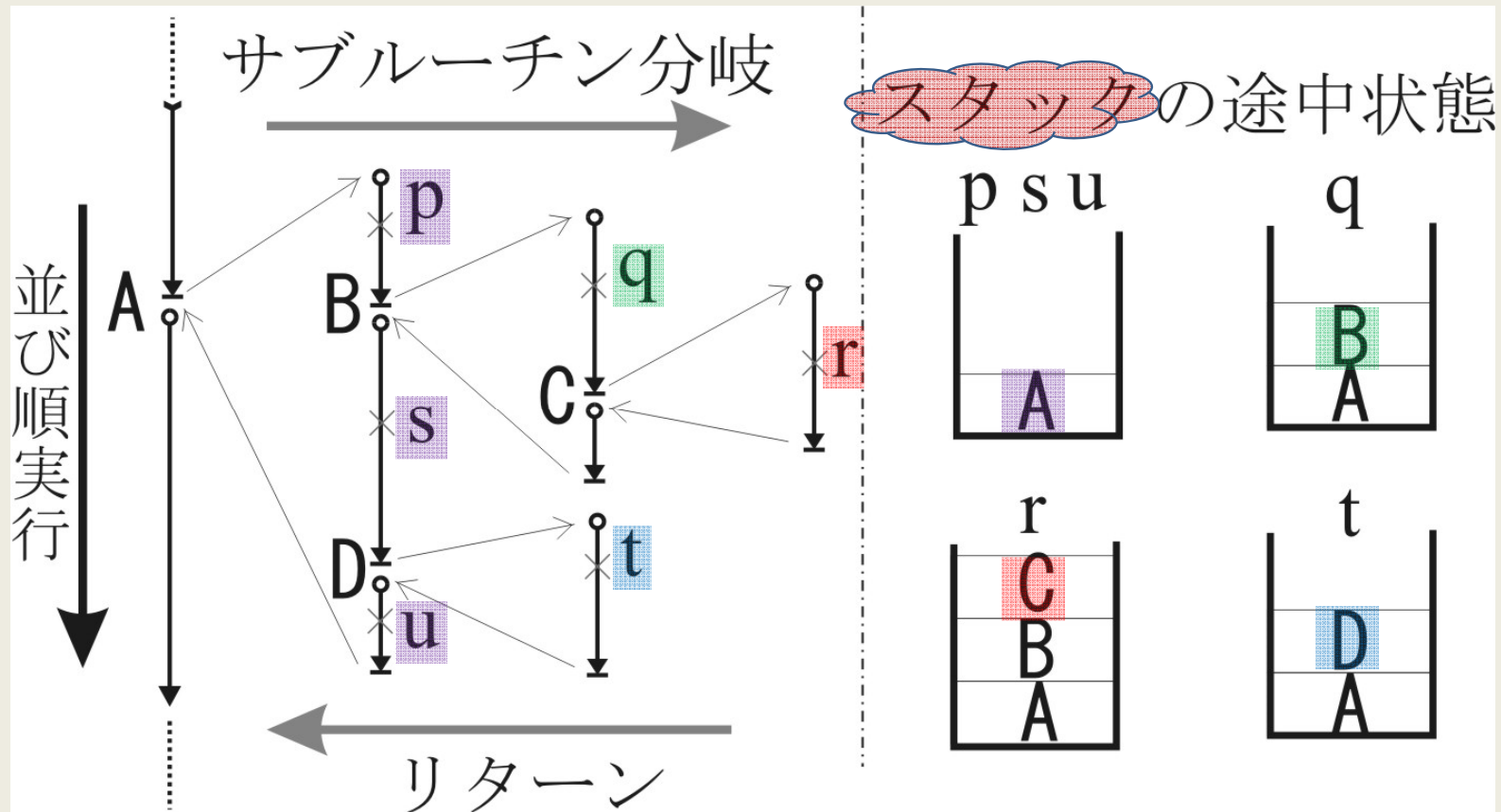
2. **リターン**命令 (= 分岐元に復帰): サブルーチンの実行が完了する際に実行, **スタック**から**分岐元のアドレス** ("A")を読み出してPCに設定 (厳密には, "A+1"に戻る)
- リターン機能では, **スタックの一番上** (出入り口)にあるアドレスを使用 → リターン命令はオペランドなし



多重のサブルーチン分岐

- **スタック**の出入り口は**唯一で共用** → **一番上**(出入り口)には**一番最後**に格納されたアドレス, **一番下**(底)には**一番最初**に格納されたアドレス
 - **一番最後**にスタックに**積まれた**アドレスを**一番最初**に取り出す (LIFO; Last In First Out)
 - サブルーチンの**途中で**さらに**別の**サブルーチンを呼ぶ(= **多重**のサブルーチン分岐)機能を実現

多重のサブルーチン分岐 (図)



不測の事態と順序制御

- プログラムは予測できる事態を想定して記述, 順序制御命令は, 予測した(予測できる)場合に対処するための命令
 - プログラミングにおける具体的な順序制御例の8.「不測の事態の発生」には, プログラムや順序制御命令では対応不可
- プログラムには, 予測した場合すべてに対処するように, 順序制御命令を埋め込む
 - 不測の(=予測していなかった, 予測できなかった)事態が発生 → 順序制御機構はどうする?

(再掲)

プログラミングにおける具体的な順序制御例(2)

5. 何回も同じプログラムをくり返し実行(くり返し)
6. プログラムの実行途中でいったん別のプログラムを呼ぶ, そこから戻る
7. あるプログラムを複数のコンピュータや人間によって共用
8. あらかじめ予測できないこと(不測の事態)に対処

割り込み(1)

- 種々の「不測の事態」のうち、比較的頻繁に起こる事態に対して、(1) 発生を検知するハードウェア機構; (2) それらを処理するための共用のプログラム (OSの一部); をあらかじめ (=コンピュータの製作時に) 用意
- 「不測の事態」の発生 → その事態に応じて対応するOSのプログラムが呼び出される, OSのプログラムに切り替わる仕組み
- 割り込み要因: 割り込みの原因となった不測の事態(事象)

割り込み(2)

- コンピュータの順序制御機構では、「不測の事態」も、厳密にいうと、「予測した事態(割り込み)」

割り込み要因例(1)

= 不測の事態(事象)

1. **ハードウェア異常**: ハードウェアの異常状態, (例) **停電**, メモリから読み出した**データの誤り**など
2. **人間による行為**: **リセット**(=リセットスイッチの押下)など

割り込み要因例(2)

3. 予定したマシン命令機能以外の事態の発生

(例) (1) 定義していない命令コードの命令を実行; (2) 「データ」を「命令」として実行; (3) 命令オペランドで指定した場所にデータなし; (4) 演算結果のオーバーフロー; (5) ゼロで除算; など

= 命令実行例外 ← 「マシン命令としての正規の機能以外の事態が発生」という意味

割り込み要因例(3)

4. OS (システムプログラム) への強制的な分岐: OSを割込ませたいときに、「OSを呼び出す機能の順序制御命令」を実行
= OS呼び出し命令(SVC; SuperVisor Call), **トラップ**(trap; わな) 命令: 「割り込みは『OSを呼び出す機能』をもつ」ことを利用

割り込み要因例(4)

5. 入出力装置などの(内部装置と同期していない)外部装置からの(同期の)要求

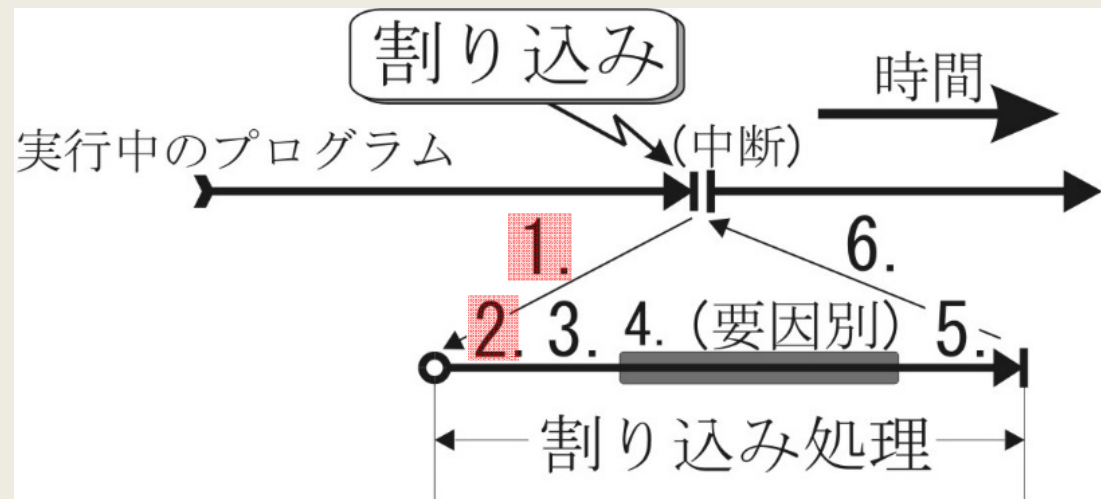
(例) (1) 入力装置からのデータの**入力**; (2) 出力装置の動作の**完了**; (3) 入出力装置の**異常**(例:電源オフ, 紙詰まり, インク切れなど); (4) 通信装置を介する他のコンピュータによる**通信の要求**; など

= 入出力割り込み

割り込み処理(1)

- 割り込みに対処する一連のOSの処理プログラム(1~6)の実行
- 割り込み(要因)の発生に対する順序制御機能(ソフトウェア)と機構(ハードウェア)の動作

1. 割り込みの受け付け
2. 割り込み要因の識別



割り込み処理(2)

3. 割り込まれたプログラムの実行環境(ハードウェアの状態)の**退避**
4. **割り込み要因別**に割り込み要因に**対処する処理の実行**
5. 退避していた割り込まれたプログラムの実行環境の**回復**
6. 割り込まれたプログラムへの**戻り**

