コンピュータ・サイエンス2

第3回
コンピュータの動作原理

人間科学科コミュニケーション専攻
白銀 純子

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

第3回の内容

- コンピュータの起動
- ソフトウェア開発
- ソフトウェアのインストールと起動
- コンピュータの特徴と実際

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

前回の復習

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

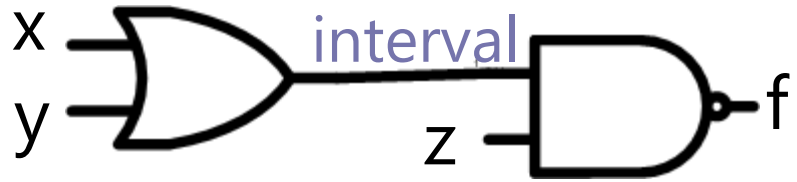
組み合わせ回路(p. 37)

- 組み合わせ回路: 入力された内容によって出力が1つに決定される回路
  - 複数の論理回路の入力と出力をつなぐことで構成
  - 入力が何であるかで出力が決まる回路
  - 入力: 電気が線の中を通っているかいないかの状態

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

組み合わせ回路[例]

- 組み合わせ回路の真理値表



真理値表完成の手順

1. 入力xとyの出力を求める(この出力を「interval」とする)
   - intervalはxとyのOR
2. intervalと入力zの出力fを求める
   - 出力fはintervalとzとのNAND

入力			interval	出力 (f)
x	y	z		
0	0	0	0	1
0	0	1	0	1
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	1	1
1	1	1	1	0

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

加算回路

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

加算器(p. 37)

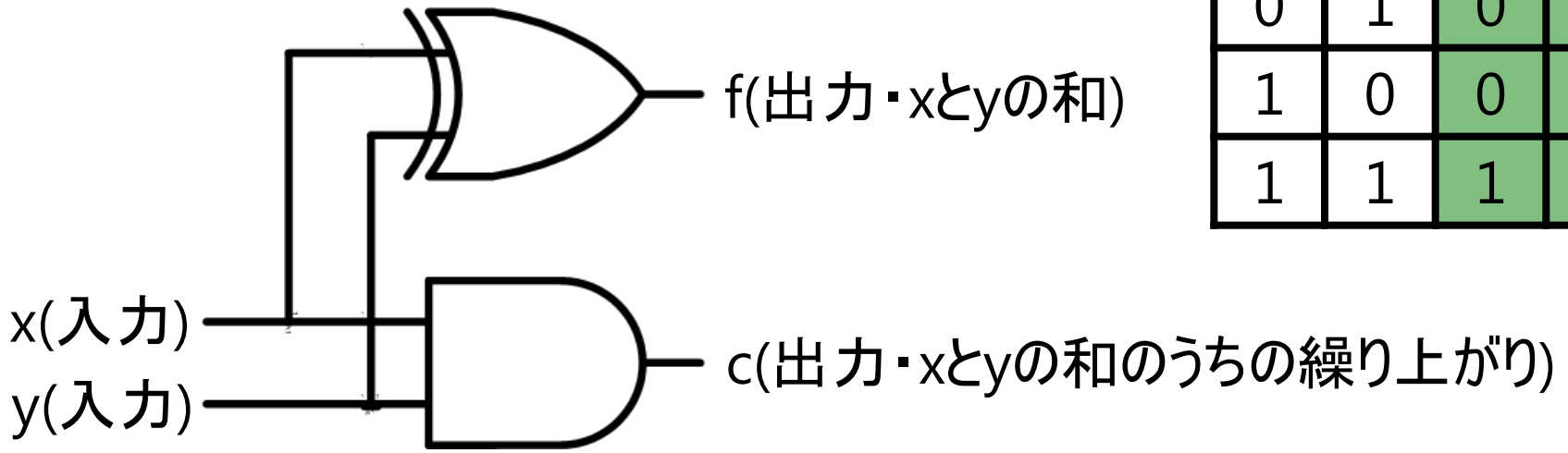
- 加算器: CPUの中で、2進数1桁の足し算をするための回路
  - 半加算器: 繰り上がりの加算をしない回路
    - 入力: 1ビットの数2つ
    - 出力: 2つの数を足した結果と繰り上がり
  - 全加算器: 繰り上がりの加算をする回路
    - 入力: 1ビットの数2つと1つ下の桁からの繰り上がり
    - 出力: 3つの数を足した結果と繰り上がり

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

半加算器

- 2進数1桁の数2つの加算をする回路
- 入力: 2進数1桁の数2つ
- 出力: 2つの数の和と繰り上がり

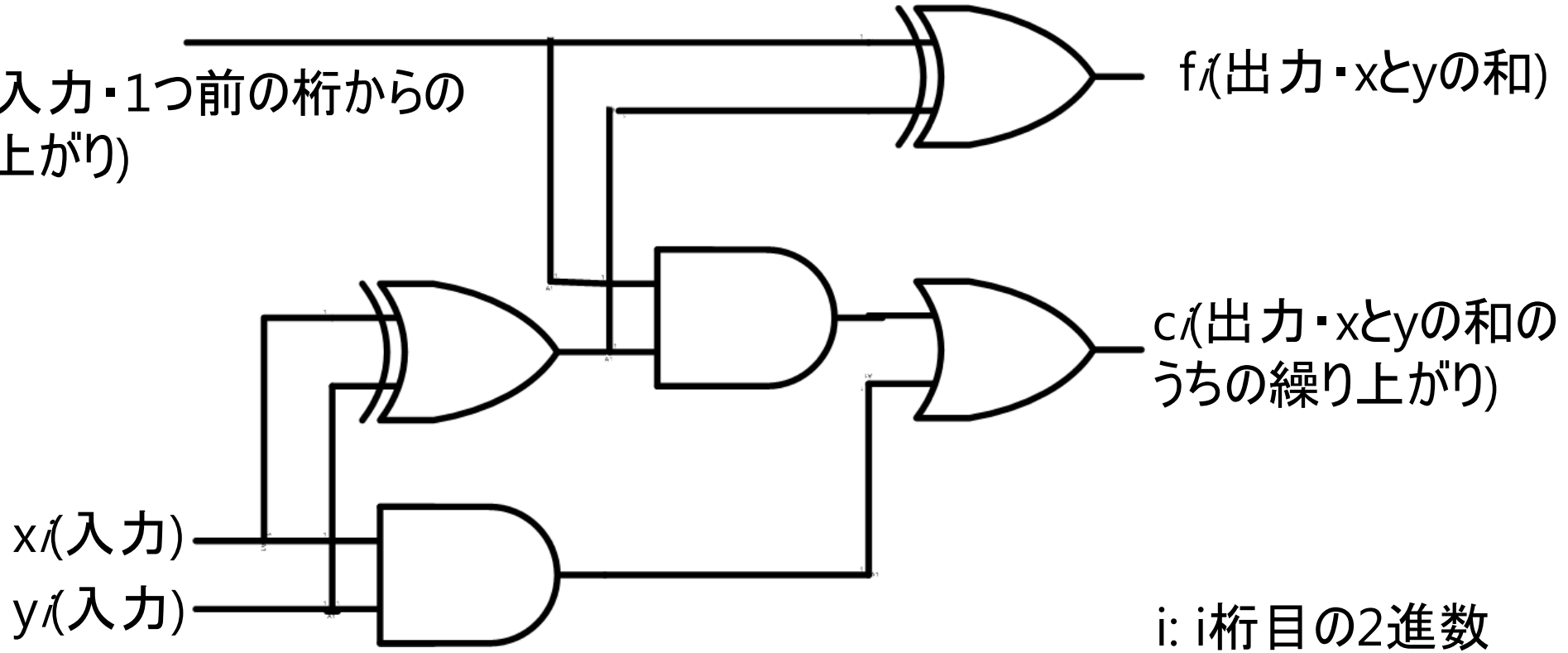
入力		出力	
x	y	c	f
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

全加算器[1](p. 37)

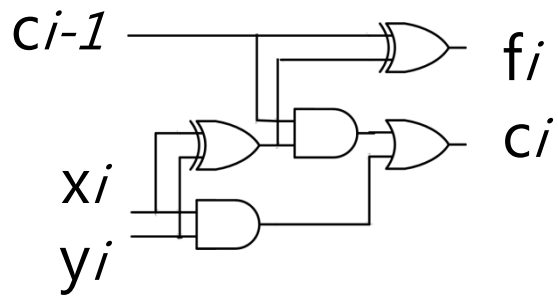
- 2進数1桁の数2つの加算をする回路
- 入力: 2桁の数2つと1つ前の桁からの繰り上がり
- 出力: 2つの数の和と繰り上がり

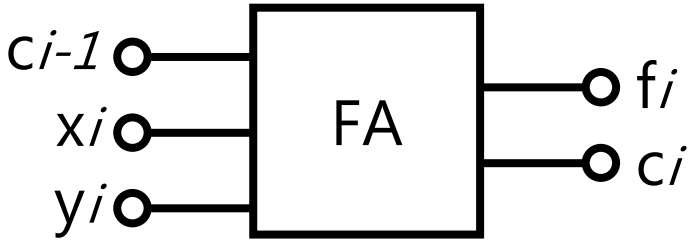


Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

全加算器[2](p. 37)

入力			出力	
x_i	y_i	c_{i-1}	c_i	f_i
0	0	0	0	0
0	1	0	0	1
1	0	0	0	1
1	1	0	1	0
0	0	1	0	1
0	1	1	1	0
1	0	1	1	0
1	1	1	1	1





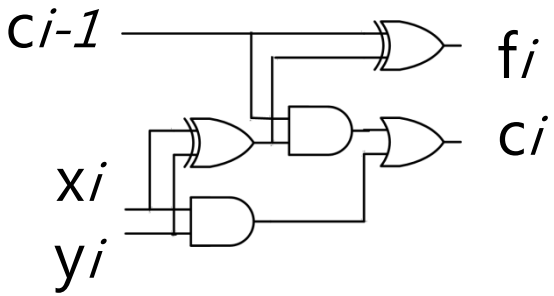
全加算器の略表記
(i: 2進数の中のi桁目)

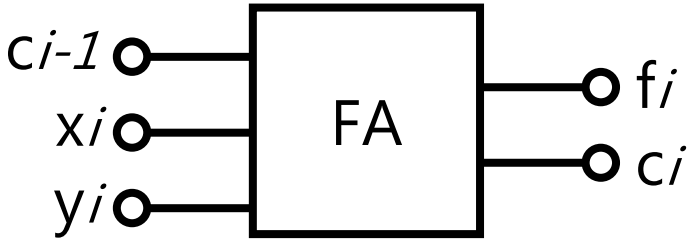
※テストの時に真理値表などが示されるかどうかは、その時々で異なる
(2009年度秋の基本情報技術者試験の問題では、何もなし)

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

全加算器[3](p. 37)

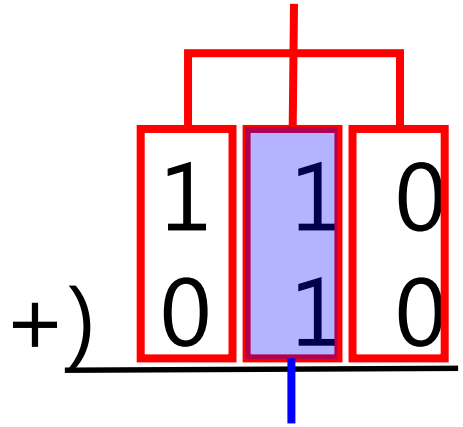
全加算器: 2進数1桁の足し算を行う回路



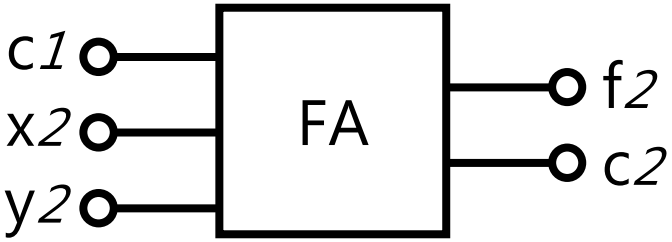


全加算器の略表記
(i: 2進数の中のi桁目)

全加算器が3つ必要
(1つの桁を1つの全加算器で計算)

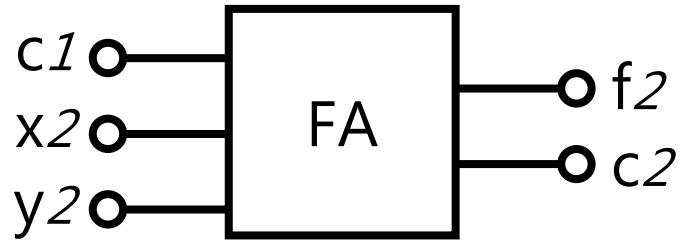


Ex. 2桁目の計算は...



Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

全加算器[4](p. 37)



c1(1桁目からの繰り上がり)

x2

y2

```
  0
1 1 0
+) 0 1 0
```

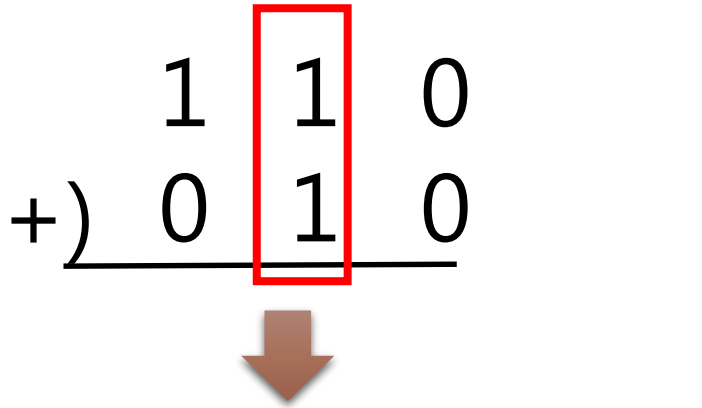
x2 : 1, y2 : 1, c1 : 0

入力			出力	
xi	yi	ci-1	ci	fi
0	0	0	0	0
0	1	0	0	1
1	0	0	0	1
1	1	0	1	0
0	0	1	0	1
0	1		1	0
1	0		1	0
1	1		1	1

2桁目の計算結果: f2 : 0, c2 : 1

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

全加算器[5](p. 37)



加算回路での計算は、必ず真理値表に従う

- 人間がするような計算はしていない
- オーバーフローが起こらない限り、人間が計算した結果と加算回路での結果は同じになる

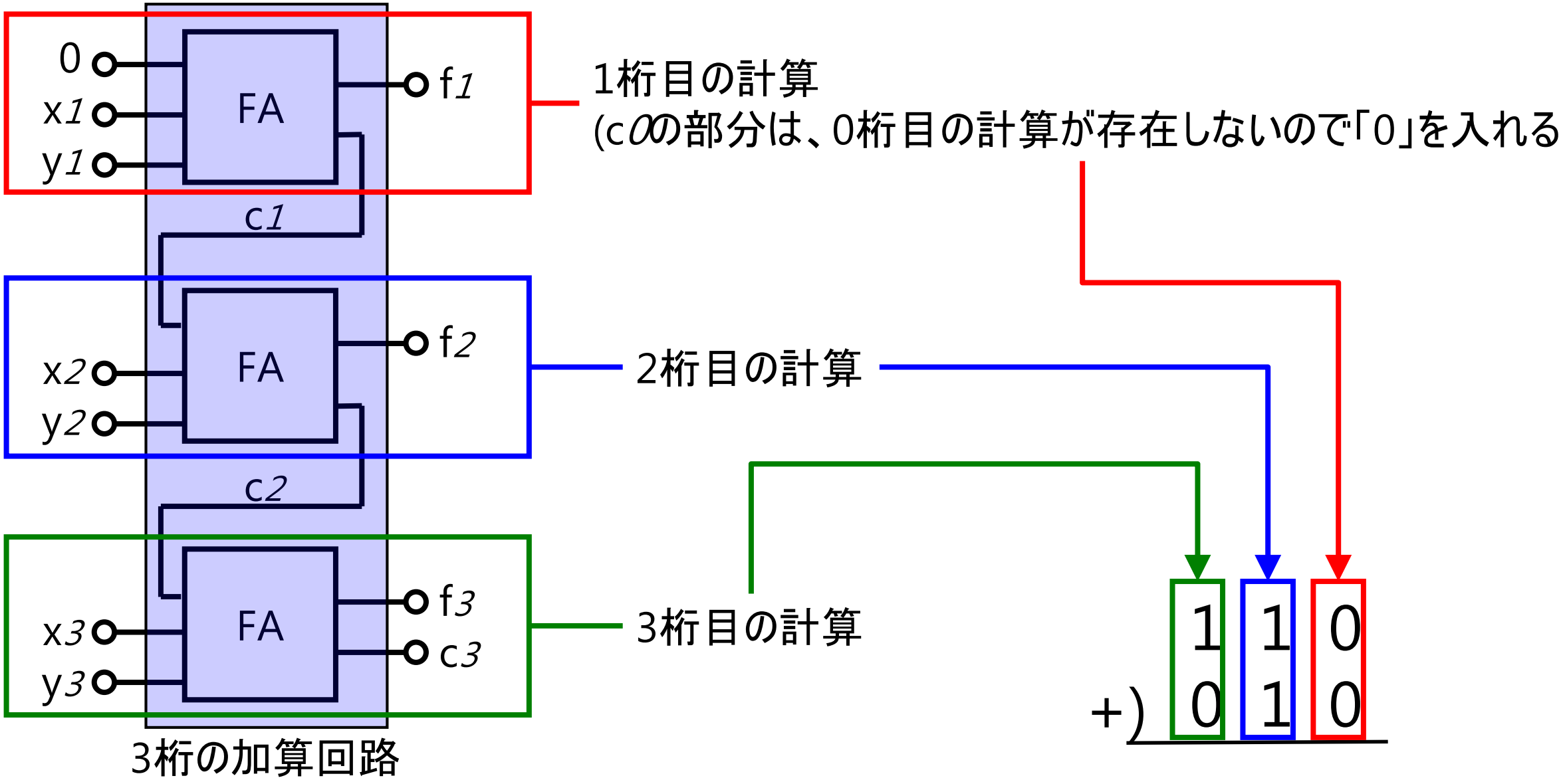
Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

加算回路[1](p. 37)

- 加算回路: ALUの中で足し算をするための回路
  - 全加算器を複数組み合わせることで構成
  - 全加算器をいくつ組み合わせるかで、何桁の2進数の足し算ができるかが決定

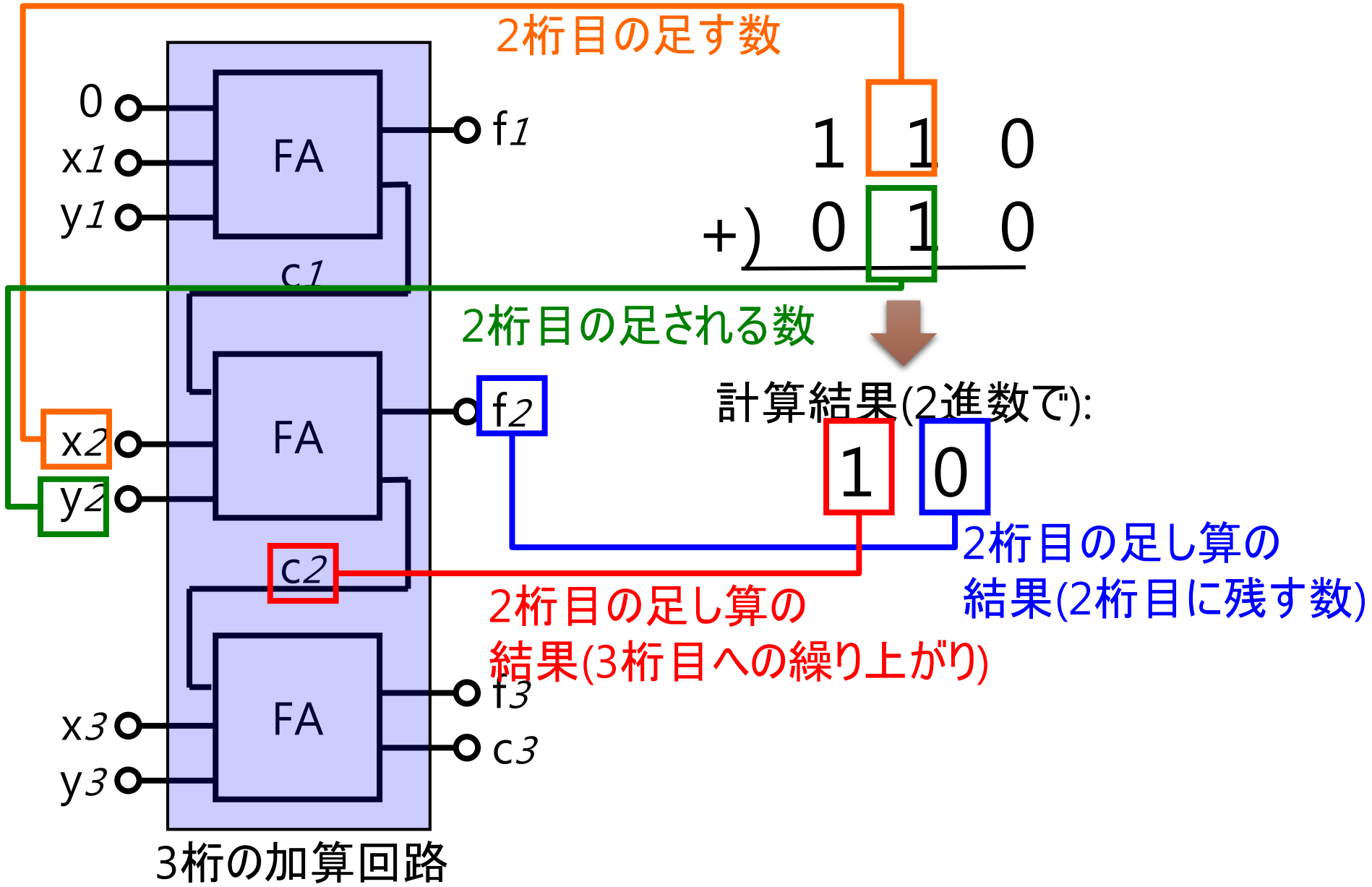
Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

加算回路[2](p. 37)



Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

加算回路[3](p. 37)



Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

コンピュータの起動

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

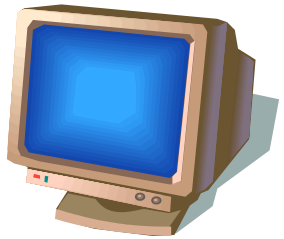
コンピュータの起動(p. 52)

- コンピュータの電源を入れて、デスクトップが画面に表示されるまでの処理

電源ボタンを押す

Windows

OSが起動する



デスクトップ画面が表示される

- BIOSが起動
- ビデオカード上のビデオROMが起動
- ブートディスクを識別

- カーネルが起動
- デバイスドライバを読み込み
- ログオン画面を表示

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

OS起動までの処理[1](p. 52)

1. マザーボードのBIOS-ROMにより制御開始
   - BIOS: 各種装置に記録されているソフトウェアで、それぞれの装置の初期設定や制御などを担当
   - BIOS-ROM: BIOSを記録している部品

単純に言うと、マザーボードが動作開始

2. マザーボードのBIOSが起動し、各種装置を制御
   - CPUのテスト(故障などの不具合がないか)
   - メインメモリのテスト(故障などの不具合がないか)
   - チップセットや各種ポートの設定
     - チップセット: マザーボード上で、CPUやメインメモリ、拡張カードなどの間でデータをやり取りを管理するための回路

単純に言うと、マザーボードに取り付けられているデバイスのチェック

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

OS起動までの処理[2](p. 52)

3. ビデオカード上のビデオROMが起動
   - ビデオROM: グラフィックスボードのBIOS

単純に言うと、ビデオカードが動作開始

4. マザーボードのBIOSによる処理内容をビデオROMが受け取って表示
   - メインメモリに対するテストの結果
   - サウンドボードやLANボードの設定状況
   - ドライブやディスクの設定状況
   - etc.

マザーボードのBIOSの処理結果(ビデオROMの処理ではない)

単純に言うと、マザーボードでの処理結果をディスプレイに表示開始

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

OS起動までの処理[3](p. 52)

5. ブートするディスクを識別
   - ブート: コンピュータを起動すること
   - ブートするディスク: OSを記録しているHDDやSSD
     - コンピュータにHDDやSSDを複数取り付けることが可能
     - OSを記録していないHDDやSSDを使ってブートすることは不可能

単純に言うと、OSを記録しているHDDやSSDを検索

6. OSの起動プログラムを呼び出して制御を受け渡し

単純に言うと、マザーボードでの処理を終了し、OSを呼び出し

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

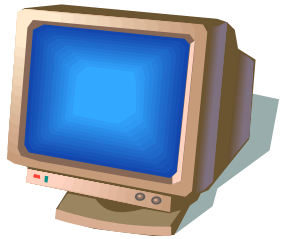
コンピュータの起動(p. 52)

- コンピュータの電源を入れて、デスクトップが画面に表示されるまでの処理

電源ボタンを押す

Windows

OSが起動する



デスクトップ画面が表示される

- BIOSが起動
- ビデオカード上のビデオROMが起動
- ブートディスクを識別

- カーネルが起動
- デバイスドライバを読み込み
- ログオン画面を表示

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

デスクトップの表示まで[1](p. 52)

1. カーネルの起動
   - カーネル: OSの中でも最も中核となる重要な部分
     - プロセスの実行制御やメモリ管理
     - ファイルシステムの管理
     - 主記憶装置へのプログラムの登録
     - etc.

単純に言うと、OSが動作開始

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

デスクトップの表示まで[2](p. 52)

2. デバイスドライバの読み込み
   - デバイスドライバ: 補助記憶装置や入出力装置などの制御を行うソフトウェア
     - HDDやグラフィックスボード、サウンドボード、LANボードなどそれぞれの装置にデバイスドライバは必要

単純に言うと、取り付けられているデバイスを利用可能に準備

3. ログオン画面を表示
   - 利用者がユーザ名とパスワードを入力することでデスクトップが表示

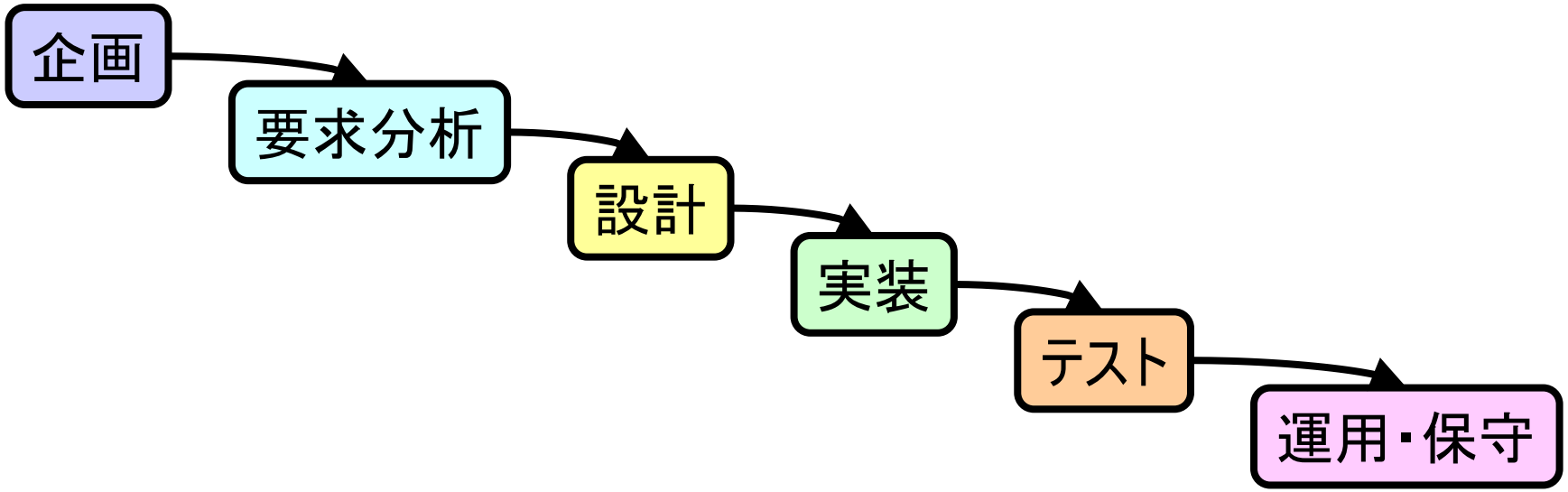
Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

ソフトウェアの開発

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

ソフトウエアの開発の流れ

- ソフトウェアの開発計画から実際の開発、できあがったソフトウェアの利用やメンテナンスの流れ: ソフトウェアのライフサイクル
- ソフトウェアは、利用者側が開発側に開発を依頼
  - 利用者側: 開発が業務としない企業や企業内の部署
  - 開発側: 開発を業務とする企業や企業内の部署



Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

企画

- ソフトウェアの利用者側がソフトウェアの導入を決定する
(業務改善などのため)
  - RFP(提案依頼書)などを作成し、開発側に提示
    - RFP (Request For Proposal): 欲しいソフトウェアの内容や予算、導入スケジュールなどの条件を書いた、利用者側(客側・依頼者側)が作成する文書
    - 提示する開発側は、多くの場合、複数の業者(競争入札で発注する業者を選定するため)
  - 開発側がRFPについて検討し、ソフトウェアの詳細やスケジュール(提案書)、コストの見積もり(見積書)を提案
  - 利用者側が、複数の業者からの提案を検討し、発注する業者を選定、契約

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

要求分析

- 開発側が利用者側から、ソフトウェアに対する要望を聞いて分析
  - どのような機能があれば良いか?
  - 開発の上での条件はあるか?
  - 機能以外で重要なこと(セキュリティや保守性など)はあるか?
  - etc.

    - 利用者側のニーズが実現可能かどうかなどを検討
    - 実現可能と判断したニーズについて、実現するための詳細(コンピュータで実現するために必要な事項)を決定(仕様化)
    - 仕様化したニーズ(要求)を文書化(記述した文書を「要求仕様書」)

※要求のことを「要件」と呼ぶことも

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

設計

- 要求仕様書に書かれている要求を、どのようにコンピュータで実現するかを決定
  - どのような技術を使うか?
  - すでに存在するソフトウェアやハードウェアで、使えるものはあるか?
  - どのような設計図にするか?
  - etc.

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

実装

- 設計の結果をプログラミング言語で記述

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

テスト

- 実装した(プログラミング言語で記述した)ソフトウェアに問題がないかをチェック
  - うまく動かないところはないか?
    - 入力したものに対し、適切な出力が出ているか?
    - おかしな動作をするところがないか?
  - 要求仕様書に記述された要求が、すべて実現されているか?
  - 要求仕様書に記述された要求が、適切に実現されているか?
  - etc.

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

運用・保守

- 運用: ソフトウェアを導入し、利用
  - ソフトウェアをインストール
  - インストールしたソフトウェアを動作させ、利用
- 保守: ソフトウェアのメンテナンス
  - 不具合の修正
  - 機能の追加
  - 機能の内容の変更
  - etc.

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

ソフトウェアのインストール

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

ソフトウェア(p. 52)

- OS: HDDやSSDに記録されていなければコンピュータを起動できない
  - HDDに記録されていない場合は、記録する作業を行う
  - 市販されているコンピュータでは、OSはあらかじめHDDやSSDに記録されている

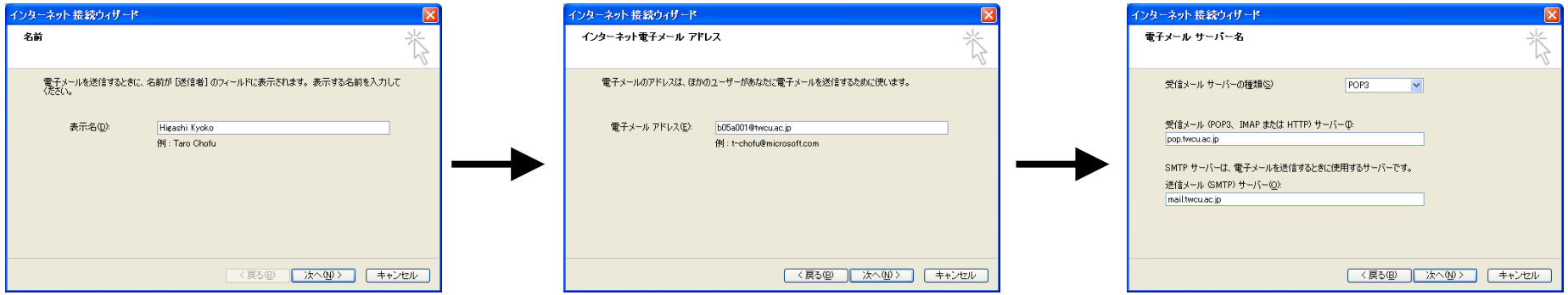
OSを含めてソフトウェアをHDDなどに記録し、利用可能な状態にすること
= インストール

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

インストール[1](p. 52)

- 多くの場合、インストーラを使ってソフトウェアをインストール
  - インストーラ: ソフトウェアをインストールするためのソフトウェア
    - ウィザード(質問に答えていくことで、ソフトウェアをインストールしたり設定するソフトウェア)を使ってインストールするものが多い

例: Outlook Expressのメールアカウントの設定ウィザード



名前の設定　　メールアドレスの設定　　メールサーバの設定

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

インストール[2](p. 52)

- インストーラの機能
  - ウィザードを使ってソフトウェアの初期設定
  - インストール場所に自動的にフォルダを作って、本体のプログラムなどの必要なファイルを保存
  - 起動用アイコン(ダブルクリックするとソフトウェアが起動するアイコン)の作成

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

ソフトウェアの配布形式

- CD-ROMまたはDVD-ROM
  - コンピュータにCD-ROMやDVD-ROMを入れると、自動的にウィザードが起動することが多い
- Webサイトからダウンロード
  - ソフトウェアを利用するために必要なファイルを全てまとめて圧縮し、1つのファイルにしていることが多い
  - ダウンロードしたファイルを解凍し、「setup.exe」ファイルをダブルクリックすると、インストーラが起動することが多い
    - 解凍: 圧縮されているファイルからもとの圧縮前のファイルを取り出すこと(複数のファイルをまとめて圧縮して1つのファイルにしている場合には、その複数のファイルを全て取り出す)

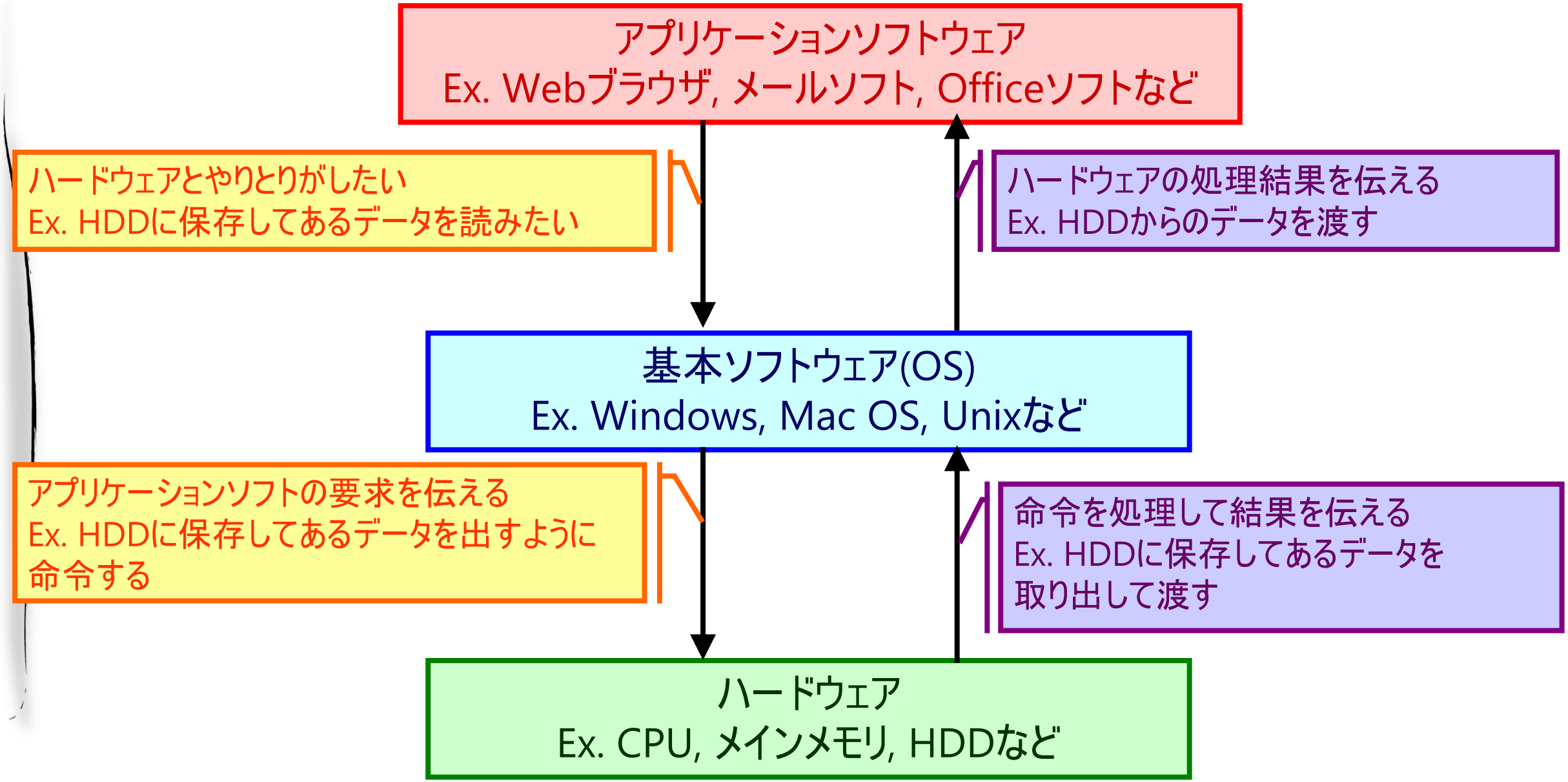
Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

OSとソフトウェアの連携[1]

- ソフトウェア: 基本ソフトウェアとアプリケーションソフトウェアに分類
  - 基本ソフトウェア(OS): ハードウェアを直接管理するためのソフトウェア
    - ハードウェアに対して、様々な命令をする
  - アプリケーションソフトウェア: ハードウェアには直接的には関与しないソフトウェア
    - 通常、マウスやキーボードで人間が操作するソフトウェア
    - ハードウェアとのやりとりが必要な場合は、OSを仲介
    - Webブラウザ, メールソフト, Officeソフト, etc.

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

OSとソフトウェアの連携[2]



Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

プログラムの起動と実行

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

ソフトウェアの起動[1](p. 53)

- ソフトウェアの起動アイコンをダブルクリックor 実行可能プログラムを指定
  - 起動アイコンは、ソフトウェアの実行プログラムそのもの、または実行プログラムのありかを示したもの
    →ダブルクリックすることは、実行可能プログラムの指定と同じ
- 実行可能プログラムがHDDやSSDからメインメモリに転送
  - 「ローディング」と呼ぶ

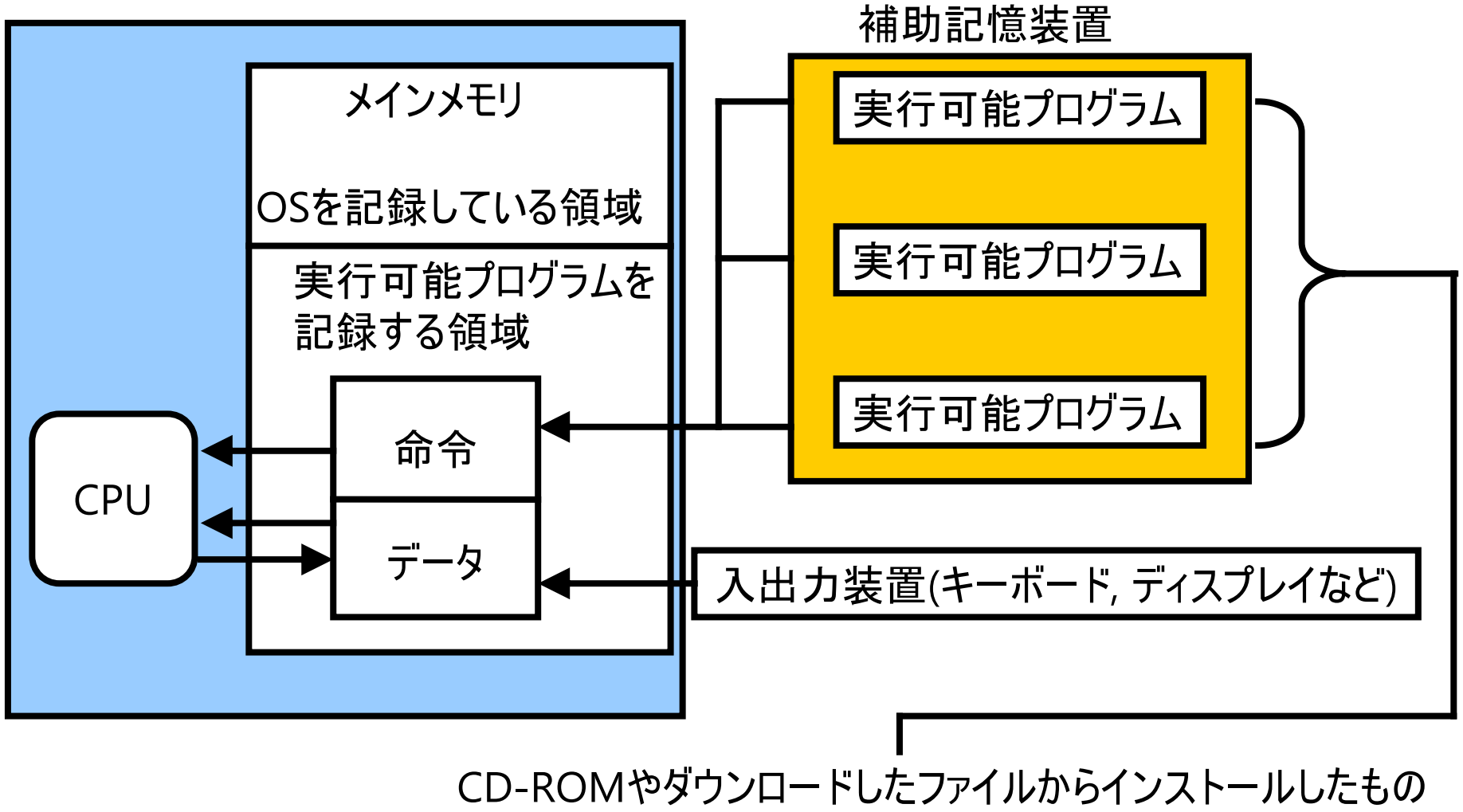
実行可能プログラム: コンピュータが直接命令を読んで実行できるプログラム

- 人間が作成したプログラムは、そのままではコンピュータが直接読んだりできないので、実行可能プログラムに翻訳する(詳細は後日)

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

ソフトウェアの起動[2](p. 53)

- ローディングされたプログラムの命令をCPUが取り出して解読



Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

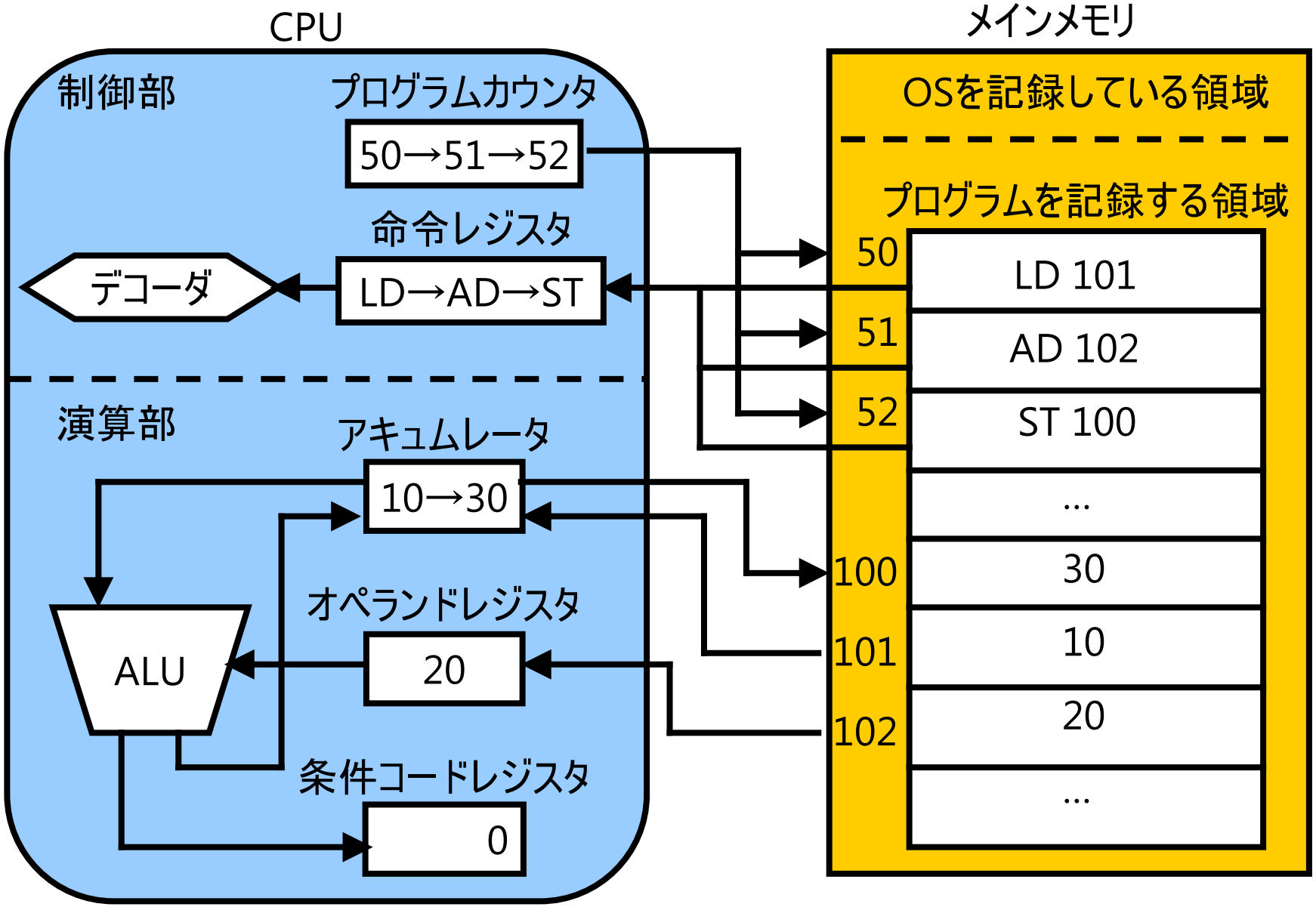
CPUの命令解読例(p. 54)

- 「10+20」の計算をするプログラム(C言語)
  - x, y, zという名前の3つの箱を用意する
    - プログラムでは、データは箱に入れて扱う
  - yとzという箱にそれぞれ10と20を入れる
  - xという箱に、「y+z」の結果を入れる
  - 計算結果を画面に表示する

```
#include<stdio.h>
void main() {
    int x;
    int y = 10;
    int z = 20;
    x = y + z;
    printf("x = %d¥n", x);
}
```

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

CPUの命令解読例[手順図](p. 54)



Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

CPUの命令解読例[手順1](p. 54)

1. プログラムカウンタによって、メインメモリの50番地が設定される
   - これにより、50番地の「LD 101」命令が取り出され、命令レジスタに転送される
2. プログラムカウンタの数値を1増やす
3. デコーダが「LD 101」を解釈して実行する
   - これにより、101番地のデータをアキュムレータに転送する(「LD」はCPUへのデータの転送命令)

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

CPUの命令解読例[手順2](p. 54)

4. プログラムカウンタの数値が51なので51番地の命令を取り出す
   - これにより、51番地の「AD 102」命令が取り出され、命令レジスタに転送される
5. プログラムカウンタの数値を1増やす
6. デコーダが「AD 102」を解釈して実行する
   - これにより、102番地のデータをオペランドレジスタに転送する
   - ALUにより加算処理が行われる(「AD」は加算命令)
     - 「10 + 20 = 30」が実行される
   - 計算結果「30」がアキュムレータに一時的に格納される

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

CPUの命令解読例[手順3](p. 54)

7. プログラムカウンタの数値が52なので52番地の命令を取り出す
   - これにより、52番地の「ST 100」命令が取り出され、命令レジスタに転送される
8. プログラムカウンタの数値を1増やす
9. デコーダが「ST 100」を解釈して実行する
   - これにより、アキュムレータにあるデータ「30」を100番地に格納する(「ST」はデータのメインメモリへの格納命令)
10. 次の命令へ進む

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

CPUの命令解読例[概要](p. 54)

- C言語のプログラムの「x=y+z;」の計算をするためにStep1～Step9の手順
  - Step1～Step4: 「y」という箱からデータを取り出す処理
  - Step5～Step6: 「z」という箱からデータを取り出し、「y+z」の計算をする処理
  - Step7～Step9: 「x」という箱に「y+z」の結果を入れる処理

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

コンピュータの特徴

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

コンピュータの特徴(p. 56)

- コンピュータでのプログラムの処理方式
  - デジタル方式
  - プログラム内蔵方式
  - 逐次制御方式

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

デジタル方式(p. 56)

- 全てのものを2進数に変換して扱う方式
  - プログラム
  - 入力・出力データ
    - 数字, 文字, 記号
    - 図表, 静止画, 動画, 音声, 音楽
    - etc.
- 論理素子・記憶素子: CPUやメモリの構成要素
  - 電圧の高低で動作を制御
    - 電圧の高低を2進数の0と1に対応

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

プログラム内蔵方式[1](p. 56)

- 当初のコンピュータ: 現在プログラムが行っている処理をスイッチと配線の組み換えでその都度操作
  - スイッチと配線の組み換えで様々な命令を表現
  - 非常に手間がかかり、間違いが多発
    - たくさんのスイッチのONとOFFを逐一切り替える必要
    - 同じ処理をするために毎回プログラムを作成する必要

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

プログラム内蔵方式[2](p. 56)

- プログラム内蔵方式: 作成したプログラムを記憶装置に記憶させて利用する方式
  - プログラムを実行するときに命令を記憶装置から取り出して実行
    - 同じ処理をするためにプログラムは1度だけ作成
  - 用途に応じてプログラムを入れ替え、様々な処理を実行

「ノイマン」という人によって提案された方式

- ノイマン:「コンピュータの父」と呼ばれるアメリカの数学者
- プログラム内蔵方式で動くコンピュータを「ノイマン型コンピュータ」
  - ✓ 現在のコンピュータのほとんどはノイマン型コンピュータ

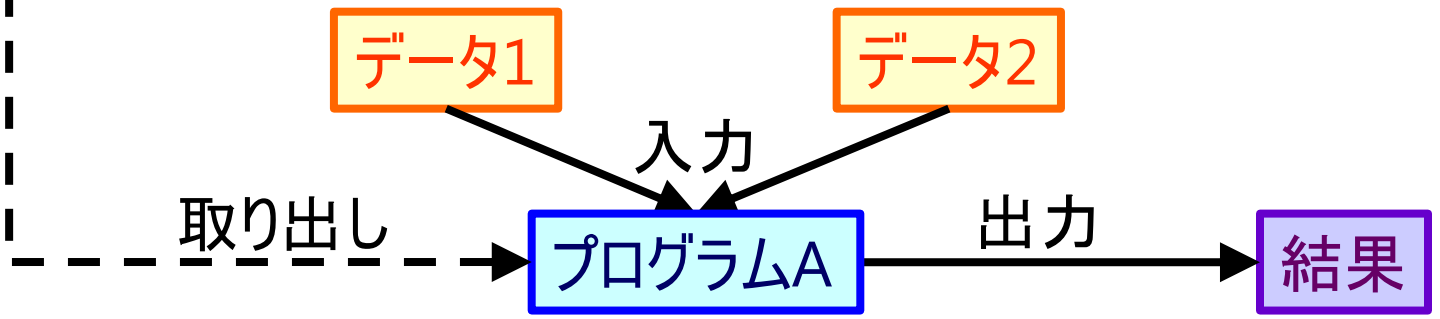
Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

プログラム内蔵方式[3](p. 56)

普段: プログラムをHDDやSSDに記録

プログラム実行時: HDDやSSDから取り出して利用

- 処理に利用するデータを入力



Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

逐次制御方式(p. 56)

- メインメモリから命令が1つずつ取り出されて実行される方式
  - プログラムカウンタによって、命令が格納されている番地を指定
  - その番地から命令を取り出して実行
  - プログラムカウンタの数値を1増やして次の命令の番地を指定

CPUとメインメモリの間の通信経路の転送速度のためにコンピュータ全体の処理の速度が遅くなる問題も存在

フォン・ノイマン・ボトルネック

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

コンピュータの実際

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

コンピュータの分類

- 筐体の大きさから分類すると…
  - スーパーコンピュータ
  - 汎用コンピュータ
  - サーバコンピュータ
  - パーソナルコンピュータ
  - モバイルコンピュータ

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

スーパーコンピュータ

- 一般的な汎用コンピュータよりも、演算速度が大幅に高速
- 利用されている分野
  - 気象学
  - 天文学
  - 流体力学
  - 金融工学, etc.

  } 大規模な数値解析によるシミュレーション用の科学技術計算
- 最近: グリッドコンピューティング
  - インターネットなどの広域ネットワークを利用し、各地のスーパーコンピュータを連結させて動作させる仕組み

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

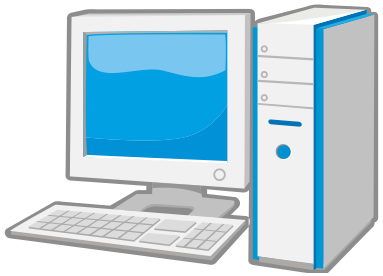
サーバコンピュータ

- クライアントサーバシステムを実現するためのサービスを提供するコンピュータ
  - クライアント: サーバに処理を要求するコンピュータ
  - サーバ: クライアントから要求された処理を行い、結果をクライアントに返すコンピュータ
  - インターネットで広く利用
- サービスを提供するソフトウェアの種類により分類
  - Webサーバ
  - メールサーバ, etc.

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

パーソナルコンピュータ(PC)

- 個人の利用を前提としたコンピュータ
  - 大きさ、価格、性能などが個人向け
  - 以前は個人の趣味で利用、現在はビジネスでも利用
  - クライアントサーバシステムでのクライアントの役割
- 筐体の大きさや形状で分類
  - デスクトップPC
  - ノートPC



デスクトップPC
(タワー型)

デスクトップPC
(一体型)

ノートPC

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.

モバイルコンピュータ

- 移動しながら使用できるコンピュータ
  - 携帯電話(スマートフォンを含む)
  - タブレット端末, etc.

Copyright (C) Junko Shirogane, Tokyo Woman's Christian University 2015, All rights reserved.