

# 基礎情報処理

Information Processing Basics  
論理回路からコンピュータまで1・2

2004年10月28日

高等教育研究開発推進センター  
小山田耕二

## 3.3 順序回路とハードウェア

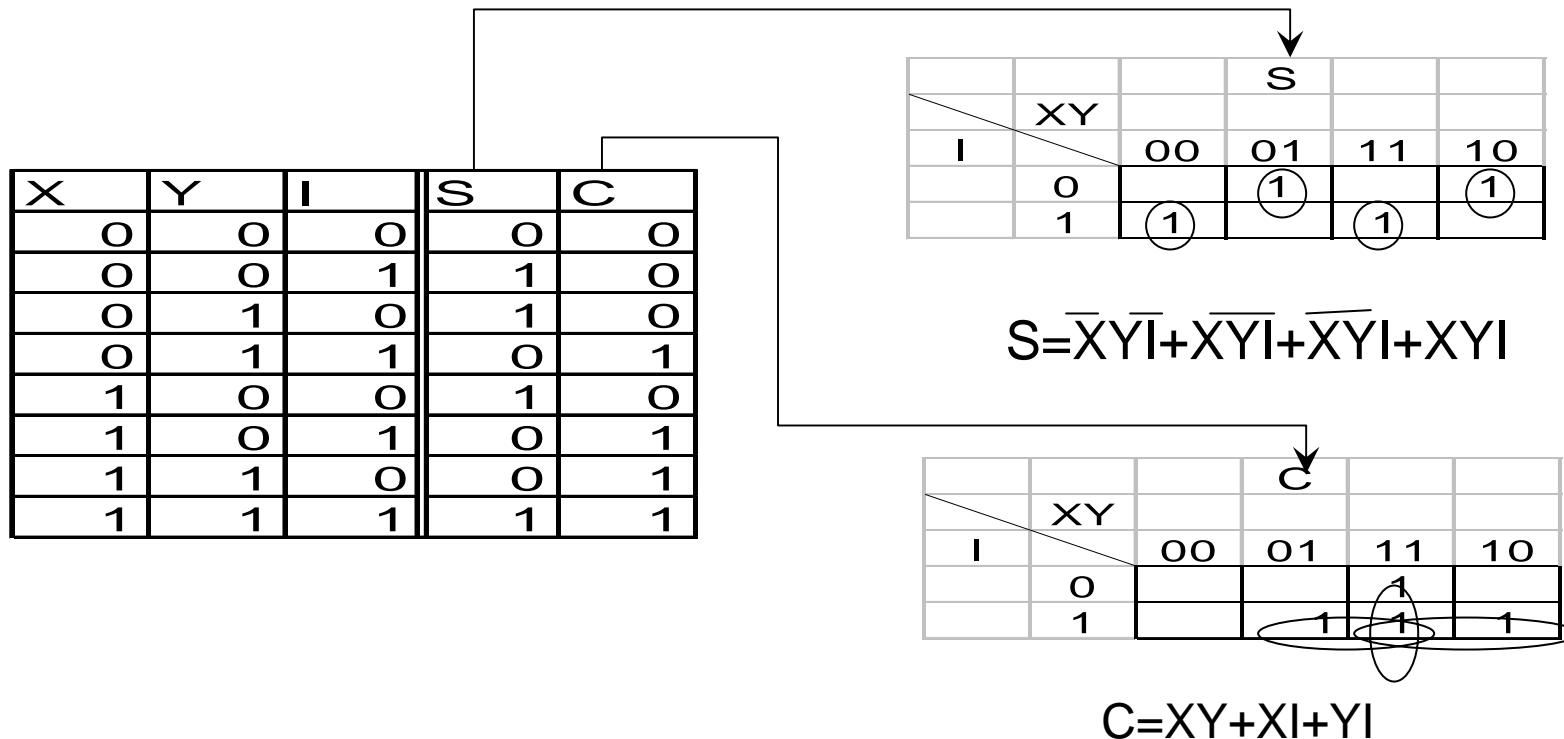
3.3.1 フリップフロップ

3.3.2 順序回路

3.3.3 コンピュータの状態モデル

# 復習: カルノー図

変数値の組によって求めた関数値を2次元(平面)格子図に書き入れて論理関数を表す方法



# 3.3.1 フリップフロップ

flip-flop

論理値の“0”か“1”のいずれかを安定状態として持つ1ビットのメモリ

## •SRフリップフロップ

- S,Rともに“0”のときは、現状態 $Q$ は不変
- Sが“0”Rが“1”のときは現状が何であっても次状態 $Q^+$ は“0”
- Sが“1”Rが“0”のときは現状が何であっても次状態 $Q^+$ は“1”
- S,Rともに“1”のときは、次状態 $Q^+$ は不定

SRフリップフロップの特性表

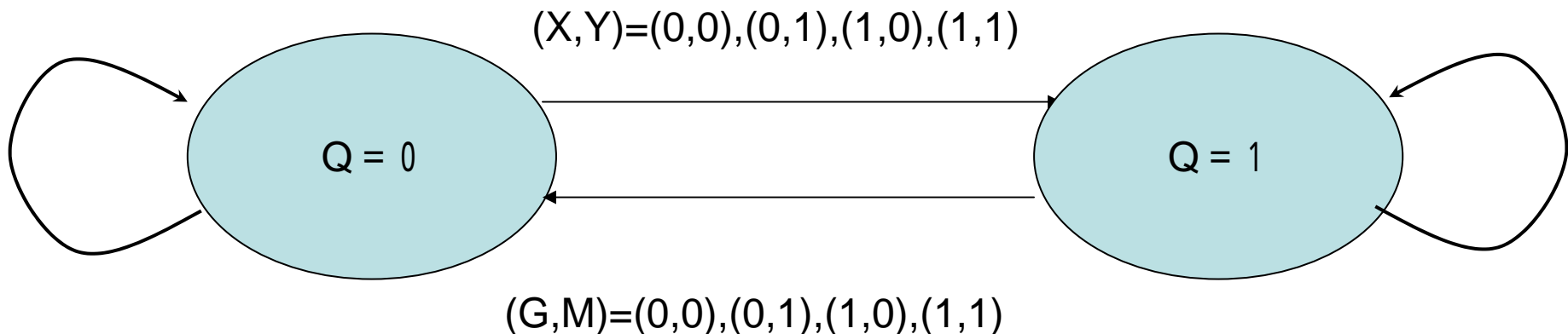
$S$	$R$	$Q^+$
0	0	$Q$
0	1	0
1	0	1
1	1	-

# 自動販売機的设计

入力として、50円硬貨Xまたは100円硬貨Yのいずれか一方のみを受け付ける硬貨投入口、出力として、100円分に相当する品物Gと釣り銭(50円硬貨)Mの2種類、のそれぞれを備えた自動販売機を2入力2出力順序回路として設計することを考える。

- 50円硬貨を受け付けた場合、 $X=1$ 、そうでない場合、 $X=0$
- 100円硬貨を受け付けた場合、 $Y=1$ 、そうでない場合、 $Y=0$
- 品物が出てきた場合、 $G=1$ 、そうでない場合、 $G=0$
- 釣り銭が出てきた場合、 $M=1$ 、そうでない場合、 $M=0$

自動販売機に0円が入力された状態 ( $Q=0$ ) と50円が入力された状態 ( $Q=1$ ) の2状態を設定し、状態遷移を考えよ。

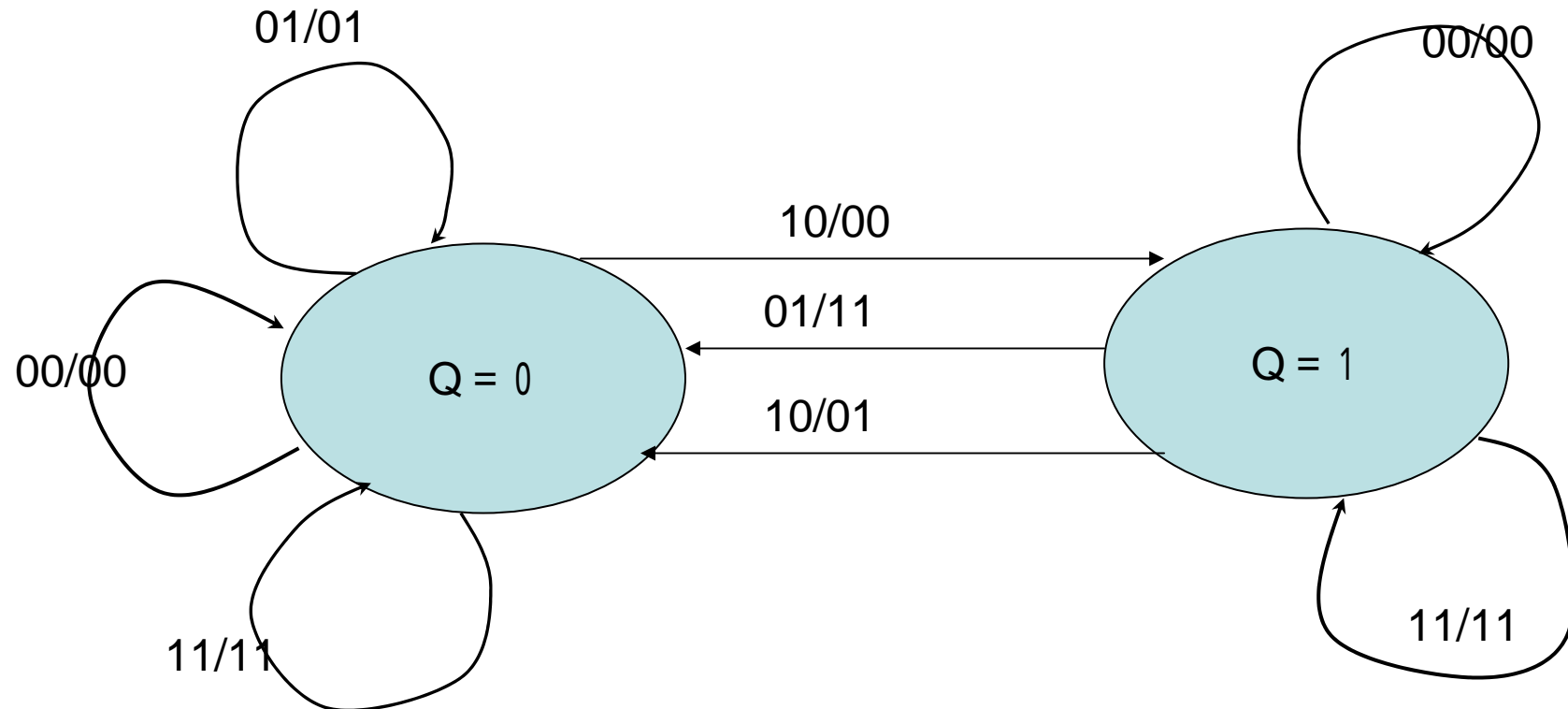


## 3.3.2 順序回路

状態遷移の枝ラベルをXY/MG とすると

10/00:50円投入して、何も出ない

01/01:100円投入して、モノだけ出る



# SRフリップフロップ

flip-flop

SRフリップフロップの特性表

$S$	$R$	$Q^+$
0	0	$Q$
0	1	0
1	0	1
1	1	-

		SR			
		00	01	11	10
$Q$	0			-	1
	1	1		-	1

SRフリップフロップの展開特性表

$S$	$R$	$Q$	$Q^+$	$\bar{Q}^+$
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	-	-
1	1	1	-	-

$$Q^+ = \bar{S}\bar{R}Q + S\bar{R}\bar{Q} + S\bar{R}Q$$

$$Q^+ = \bar{S}\bar{R}Q + S\bar{R}\bar{Q} + S\bar{R}Q + S\bar{R}\bar{Q} + S\bar{R}Q$$

$$X = ABC + A\bar{B}\bar{C}$$

$$X = AB(C + \bar{C}) = AB$$

$$Q^+ = S + RQ$$

# SRフリップフロップ

flip-flop

SRフリップフロップの特性表

$S$	$R$	$Q^+$
0	0	$Q$
0	1	0
1	0	1
1	1	-

		SR			
		00	01	11	10
$Q$	0			-	1
	1	1		-	1

$$Q^+ = S + R\overline{Q}$$

SRフリップフロップの展開特性表

$S$	$R$	$Q$	$Q^+$	$\overline{Q}^+$
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	-	-
1	1	1	-	-

		SR			
		00	01	11	10
$Q$	0	1	1	-	
	1		1	-	

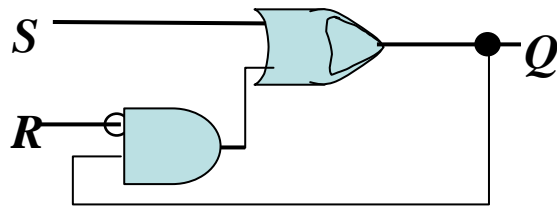
$$\overline{Q}^+ = R + \overline{S}Q$$



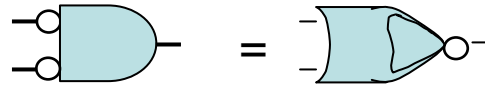
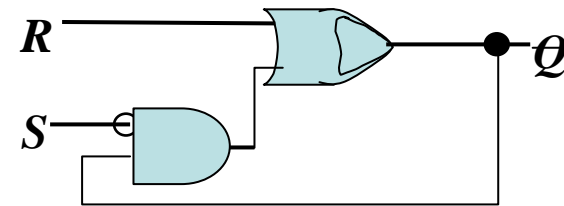
# SRフリップフロップ

flip-flop

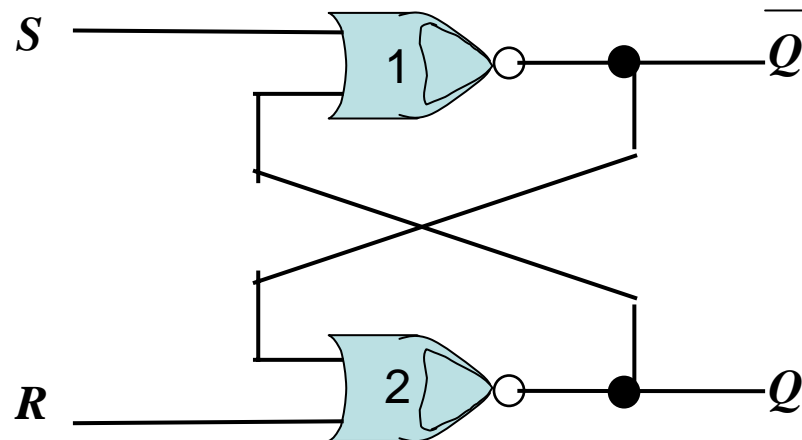
$$Q^+ = S + \bar{R}Q$$



$$\bar{Q}^+ = R + \bar{S}\bar{Q}$$



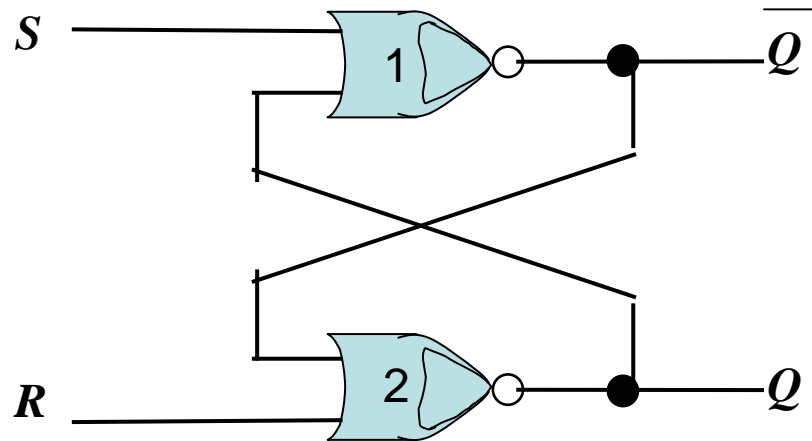
$$A \cdot B = \overline{A + B}$$



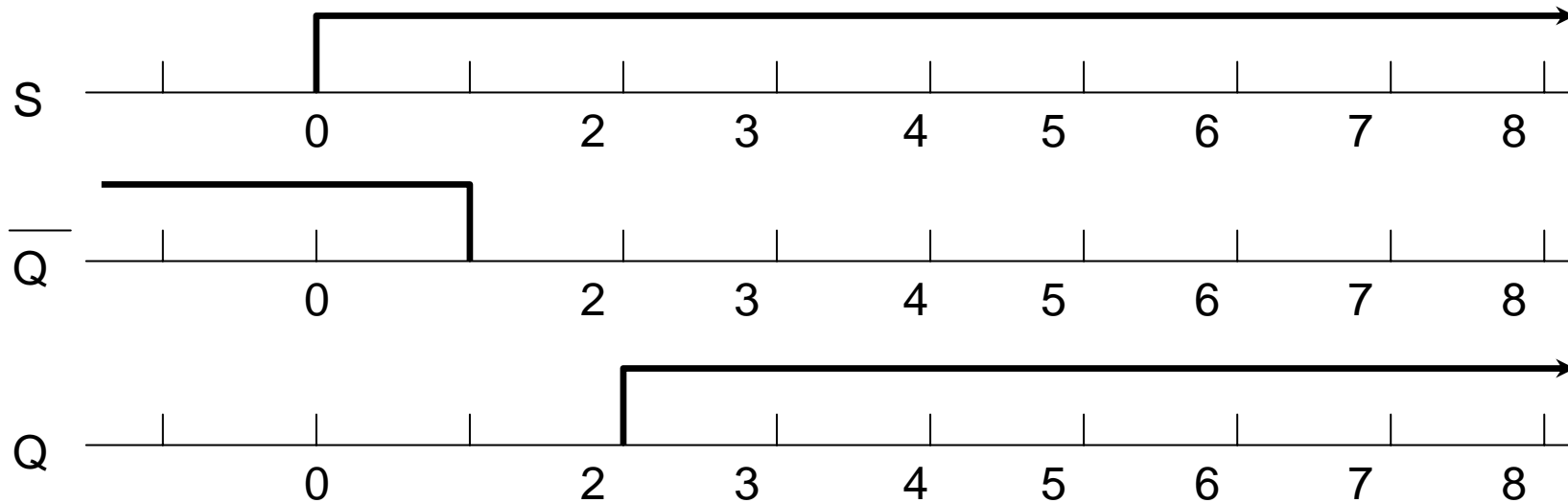
SRフリップフロップの基本回路

# SRフリップフロップ

flip-flop

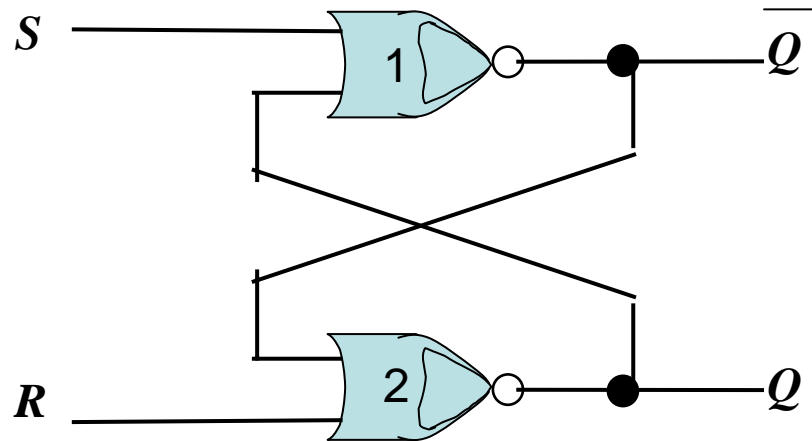


A	B	A+B	$\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

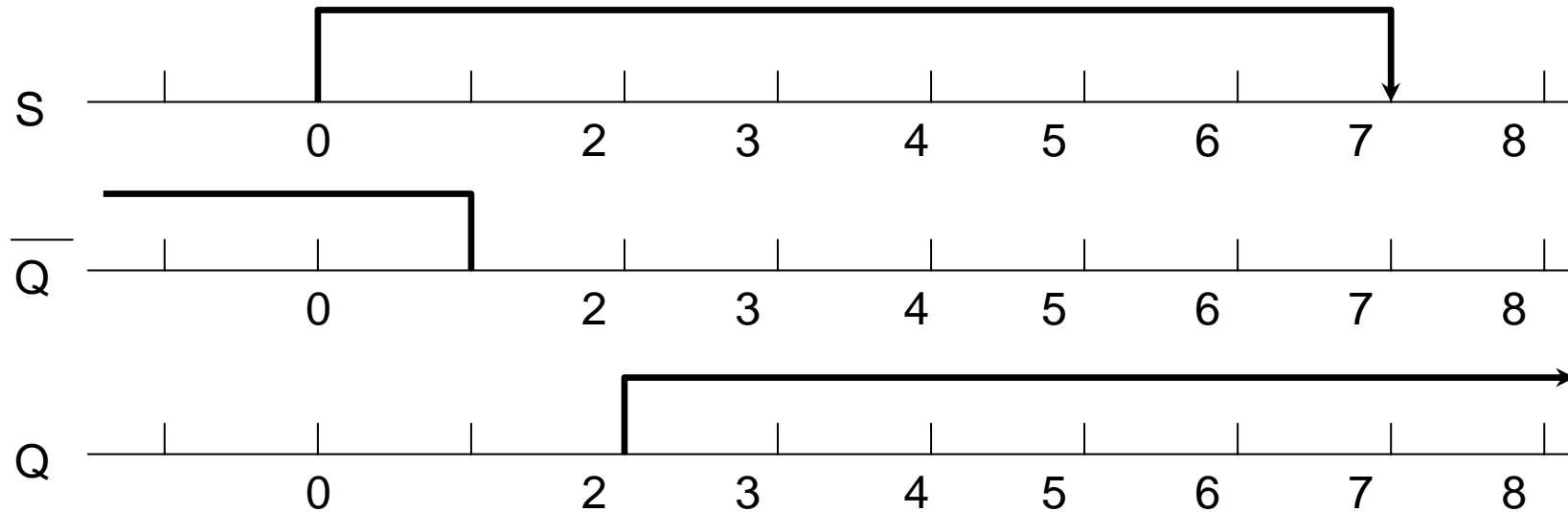


# SRフリップフロップ

flip-flop

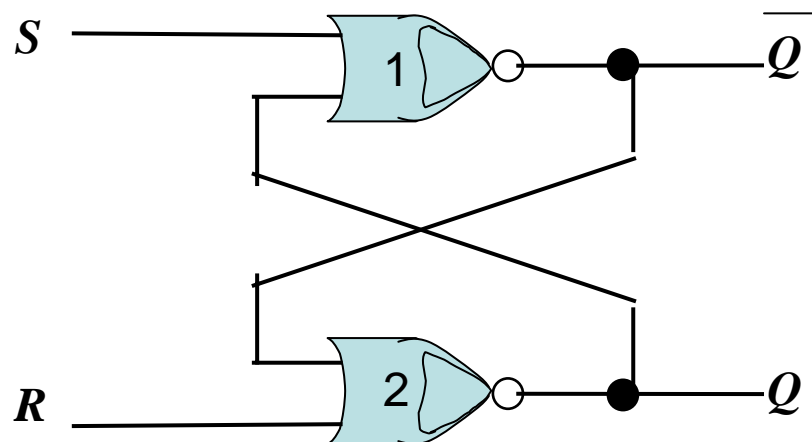


A	B	A+B	$\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

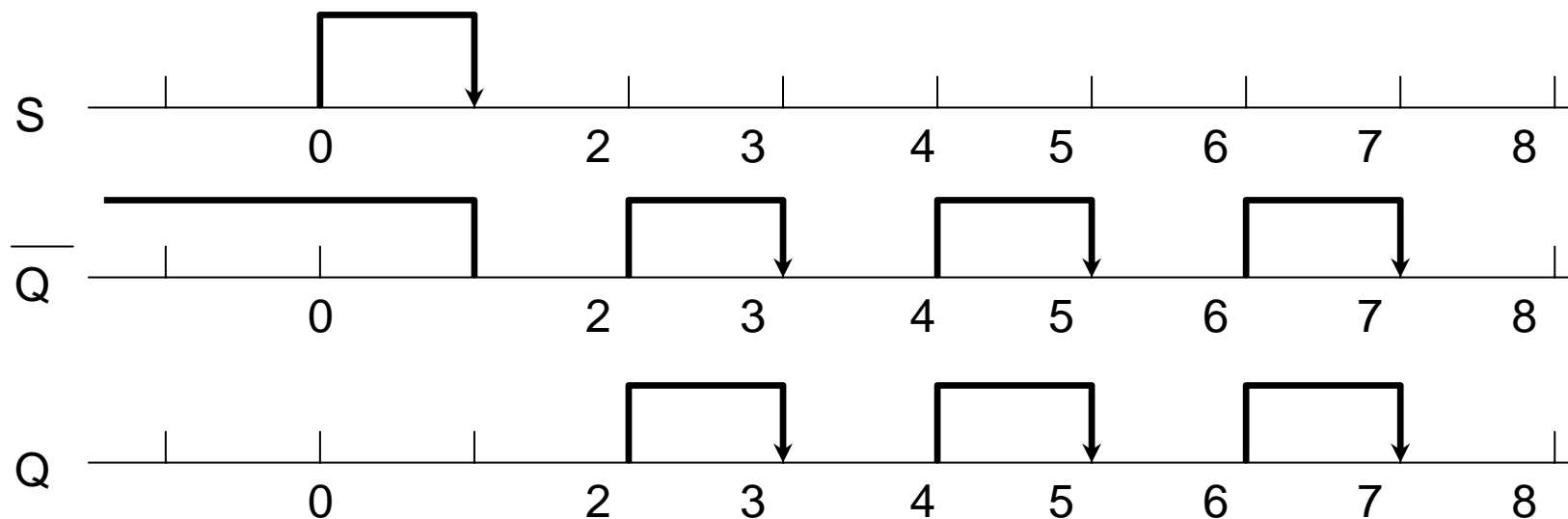


# SRフリップフロップ

flip-flop

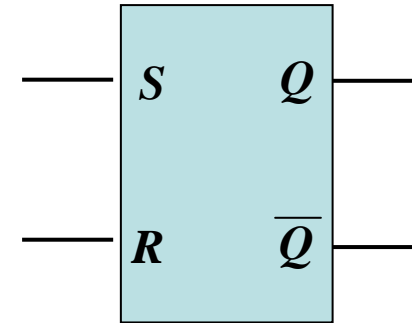
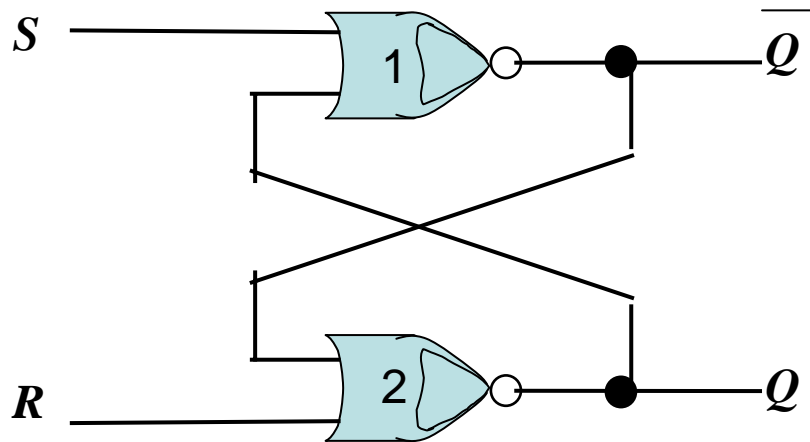


A	B	A+B	$\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0



# SRフリップフロップ

flip-flop



SRフリップフロップの展開特性表

$S$	$R$	$Q$	$Q^+$	$\overline{Q}^+$
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	-	-
1	1	1	-	-

SRフリップフロップの入力要求表

$Q$	$Q^+$	$S$	$R$
0	0	0	-
0	1	1	0
1	0	0	1
1	1	-	0

# Dフリップフロップ

flip-flop

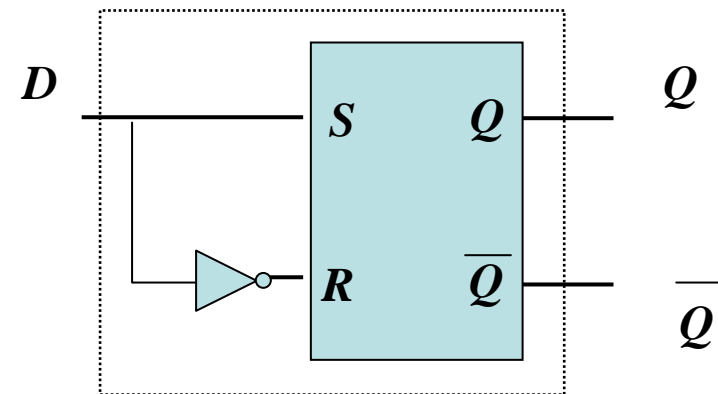
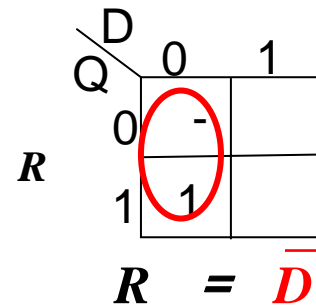
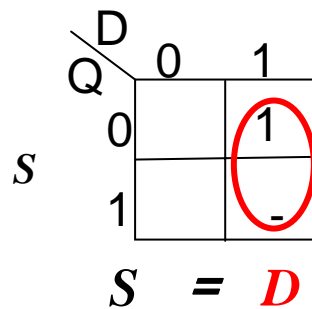
現状態が何であってもD入力を次状態とする

Dフリップフロップの展開特性表

$D$	$Q$	$Q^+$
0	0	0
0	1	0
1	0	1
1	1	1

Dフリップフロップの入力要求表

$Q$	$Q^+$	$D$	$S$	$R$
0	0	0	0	-
0	1	1	1	0
1	0	0	0	1
1	1	1	-	0



# Tフリップフロップ

flip-flop

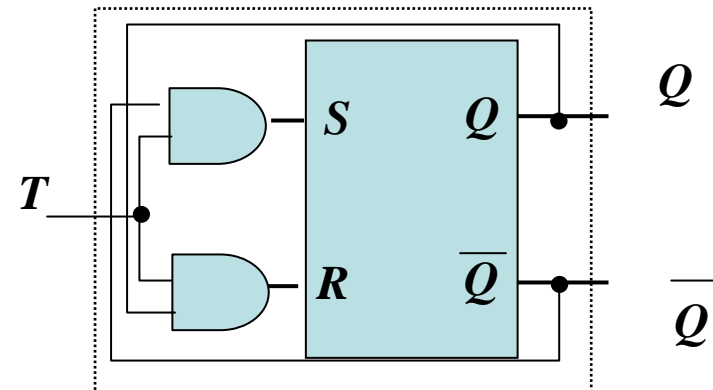
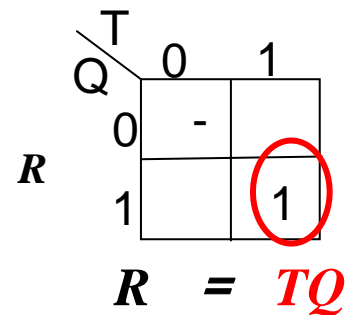
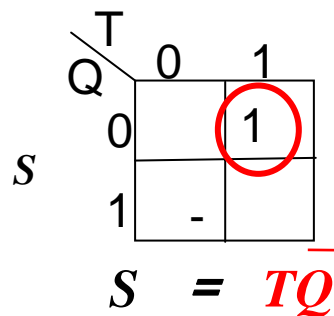
- T入力が“0”の時現状態Qは不変
- T入力が“1”の時現状態の否定が次状態

Tフリップフロップの展開特性表

$T$	$Q$	$Q^+$
0	0	0
0	1	1
1	0	1
1	1	0

Tフリップフロップの入力要求表

$Q$	$Q^+$	$T$	$S$	$R$
0	0	0	0	-
0	1	1	1	0
1	0	1	0	1
1	1	0	-	0



# JKフリップフロップ

flip-flop

- J,Kともに“0”のときは、現状態Qは不変
- Jが“0”Kが“1”のときは現状が何であっても次状態は“0”
- Jが“1”Kが“0”のときは現状が何であっても次状態は“1”
- J,Kともに“1”のときは、時現状態の否定が次状態

JKフリップフロップの展開特性表

$J$	$K$	$Q$	$Q^+$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

JKフリップフロップの入力要求表

$Q$	$Q^+$	$J$	$K$	$S$	$R$
0	0	0	-	0	-
0	1	1	-	1	0
1	0	-	1	0	1
1	1	-	0	-	0

JK Q	00	01	11	10
0			1	1
1	-			-

$$S = J\bar{Q}$$

JK Q	00	01	11	10
0	-	-		
1		1	1	

$$R = KQ$$



# JKフリップフロップ

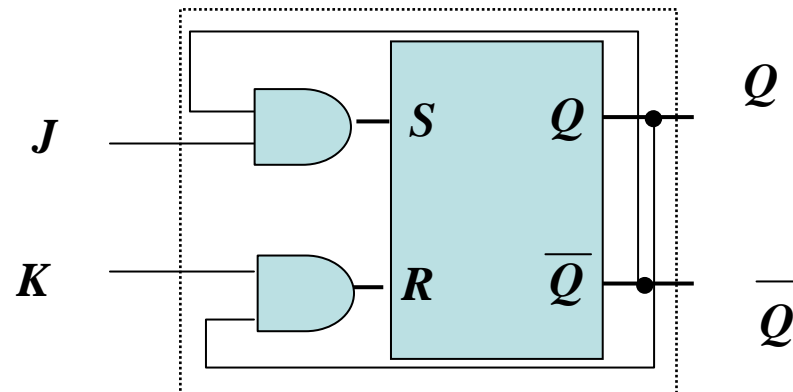
flip-flop

		JK			
		00	01	11	10
Q	0			1	1
	1	-			-

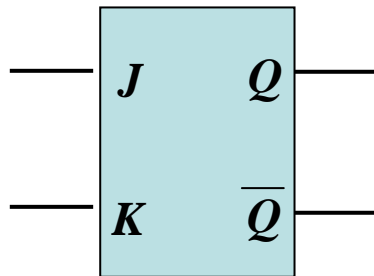
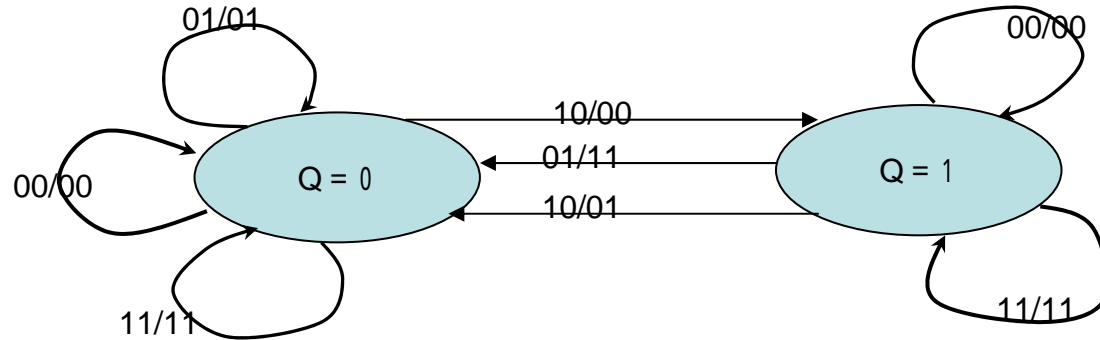
$$S = J\bar{Q}$$

		JK			
		00	01	11	10
Q	0	-	-		
	1		1	1	

$$R = KQ$$



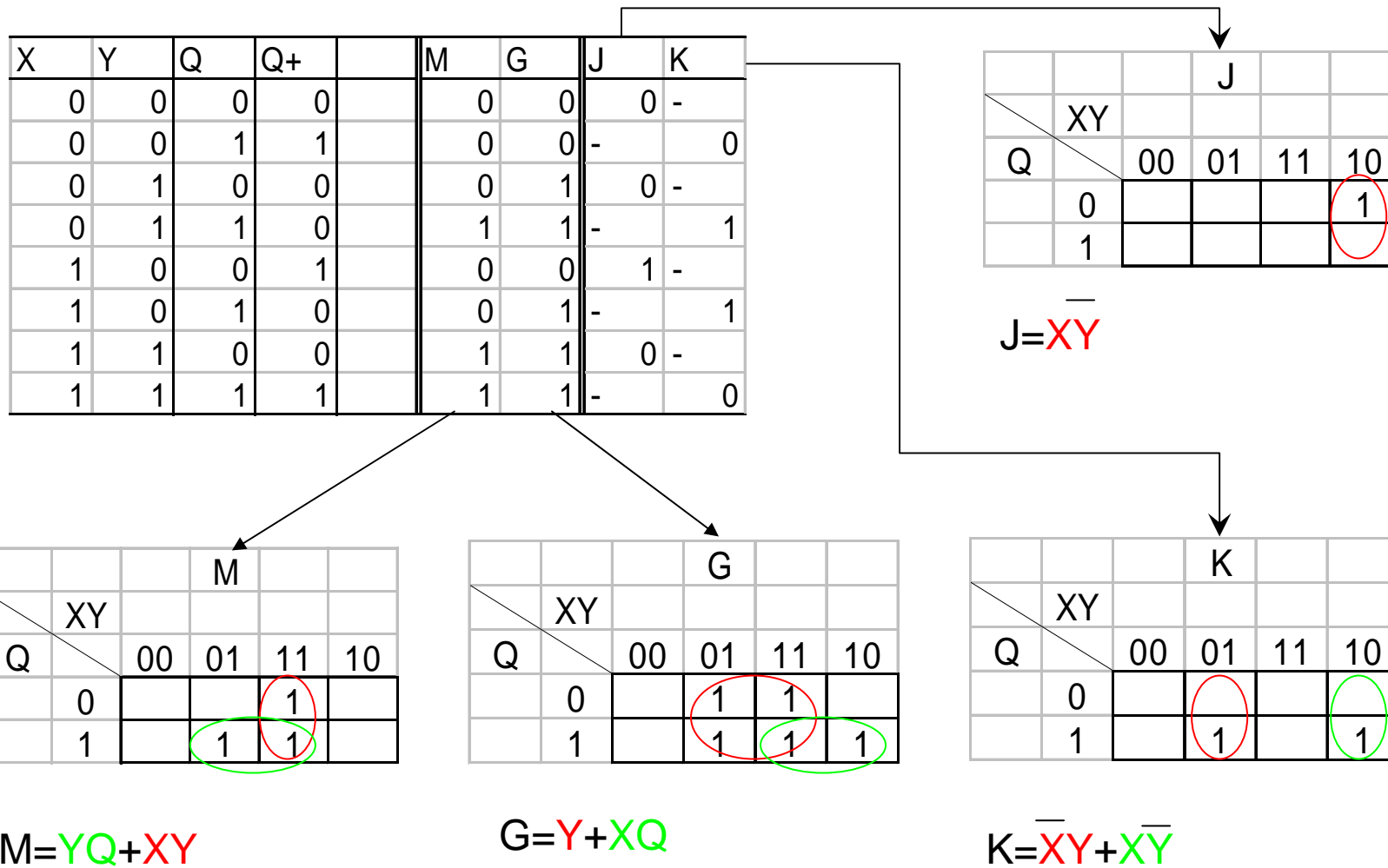
# 自動販売機の状態遷移表



$Q$	$Q^+$	$J$	$K$
0	0	0	-
0	1	1	-
1	0	-	1
1	1	-	0

X	Y	Q	Q+		M	G	J	K
0	0	0	0		0	0	0	-
0	0	1	1		0	0	-	0
0	1	0	0		0	1	0	-
0	1	1	0		1	1	-	1
1	0	0	1		0	0	1	-
1	0	1	0		0	1	-	1
1	1	0	0		1	1	0	-
1	1	1	1		1	1	-	0

# 自動販売機のカルノー図

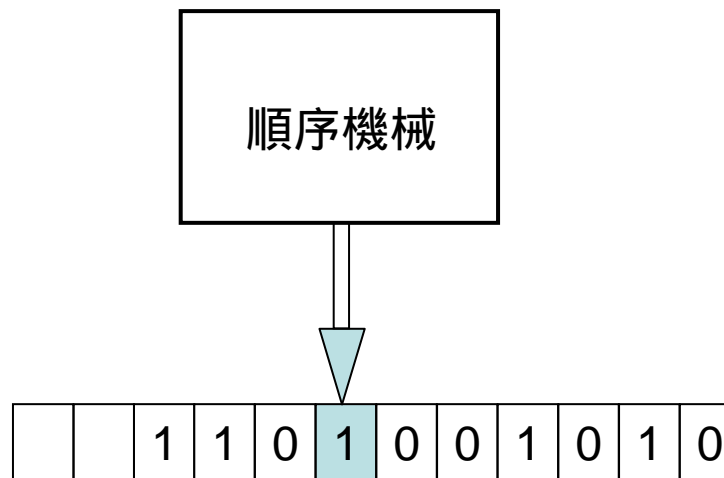


# 3.3.3 コンピュータの状態モデル

オートマトン (automaton): 有限個の状態、有限個の入力、状態遷移関数、初期状態、最終状態の5項によって定義する計算機械

順序機械: 有限オートマトンを定義する5項のうち、最終状態を出力に置き換え、新たに、出力関数を加えた6項で定義する計算機械

順序回路: 順序機械を実際のハードウェア機構として実現したもの



Alan Mathison Turing

チューリング機械  
Turing Machine

<http://ja.wikipedia.org/wiki/>

# Outline

1. コンピュータとはなにか
2. デジタル情報の世界
3. 論理回路からコンピュータまで1
4. 論理回路からコンピュータまで2
5. プログラム基礎1
6. プログラム基礎2
7. データ構造とアルゴリズム1
8. データ構造とアルゴリズム2
9. コンピュータネットワーク
10. 情報倫理
11. さまざまな情報処理
12. コンピュータ科学の諸問題

# 4 . 論理回路からコンピュータまで2

## 4.1 コンピュータのシステム設計

### 4.1.1 バスとインタフェース

### 4.1.2 割り込み機能

### 4.1.3 記憶の階層

## 4.2 オペレーティングシステム (OS)

### 4.2.1 OSとは

### 4.2.2 OSの構造

### 4.2.3 OSの役割

### 4.2.4 OSによる管理

## 4.3 並行処理

### 4.3.1 並行処理と並列処理

### 4.3.2 マルチプログラミングとマルチタスク

### 4.3.3 コンピュータシステムの利用形態

# 4.1 コンピュータのシステム設計

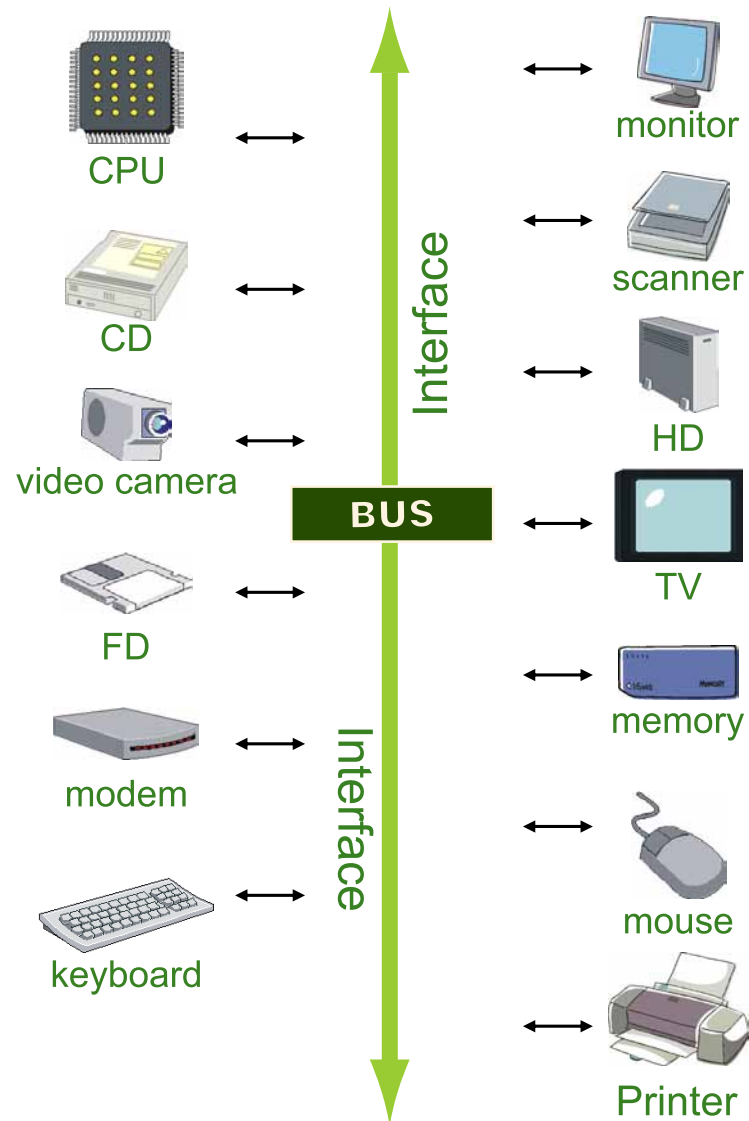
4.1.1 バスとインタフェース

4.1.2 割り込み機能

4.1.3 記憶の階層

# 4.1.1 バスとインターフェース

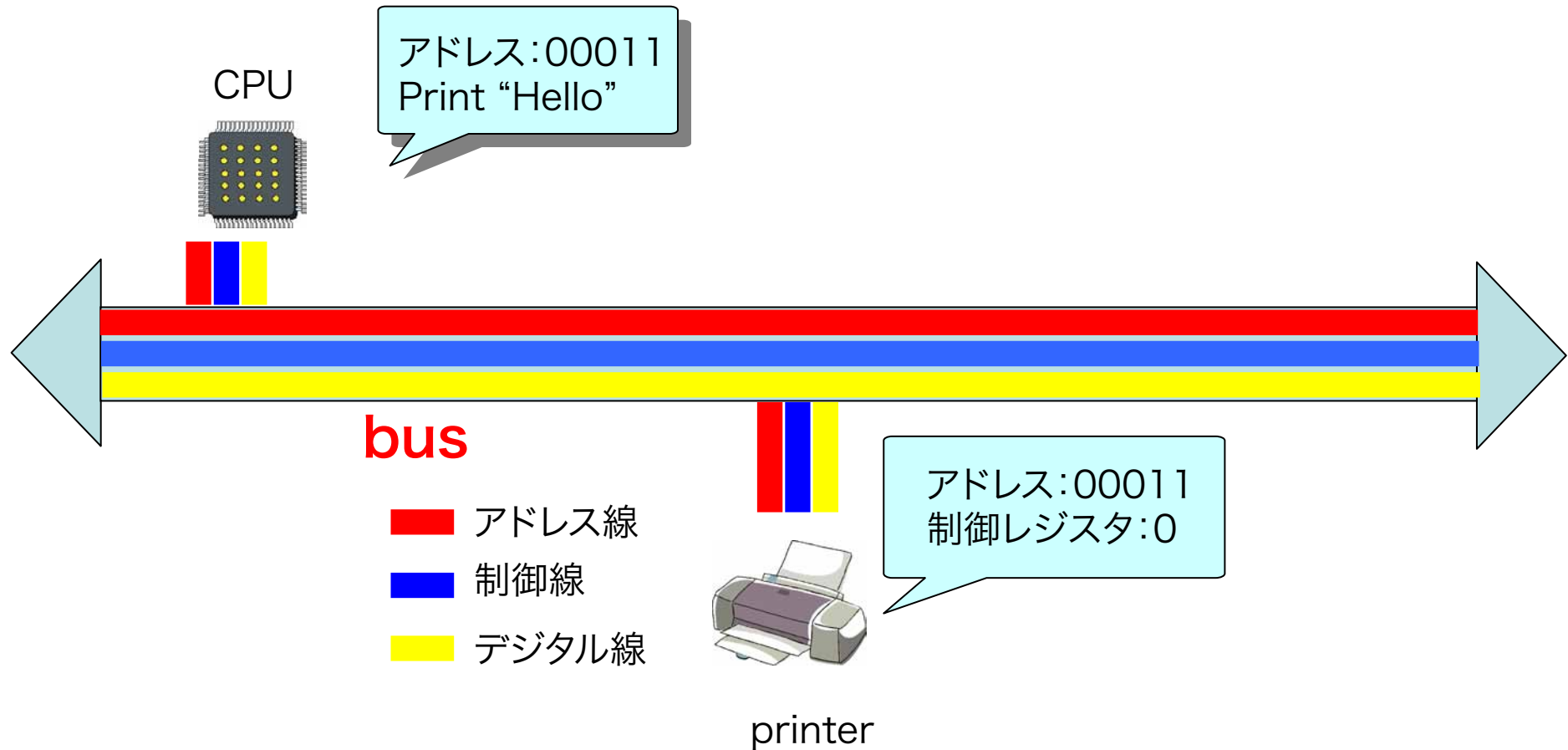
## バス構成コンピュータシステムの例



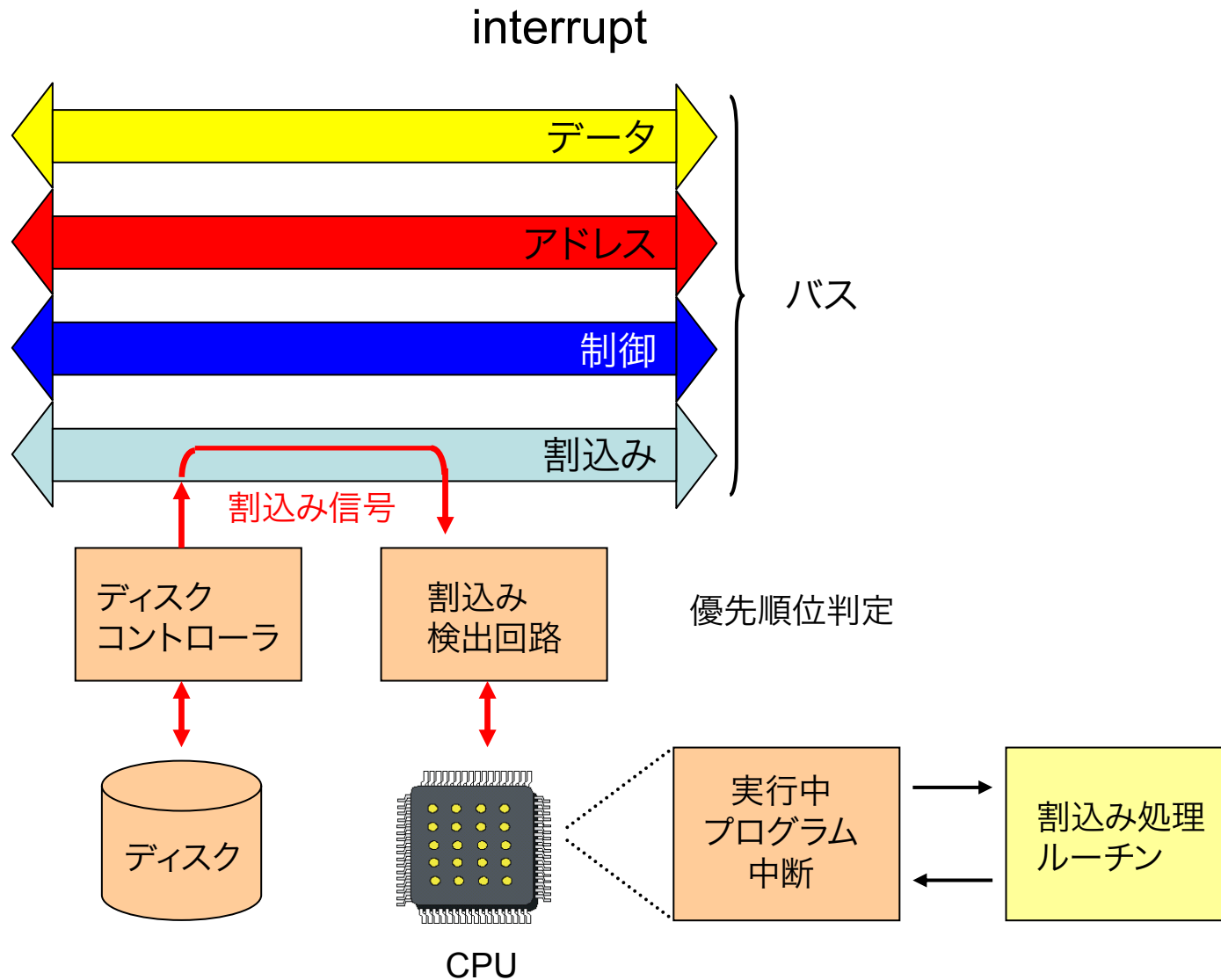


# 4.1.1 バスとインタフェース

## プリンタへの命令

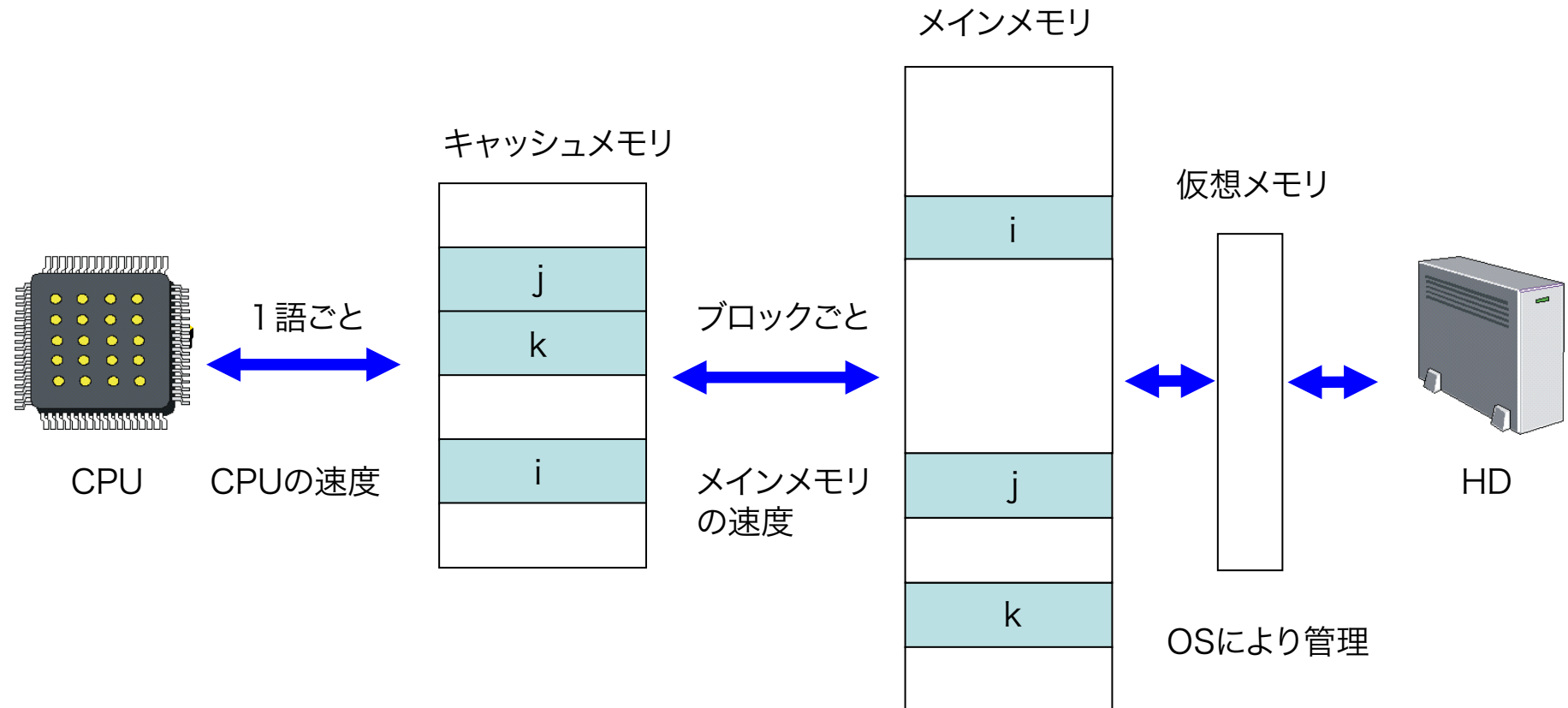


## 4.1.2 割り込み機能



# 4.1.3 記憶の階層

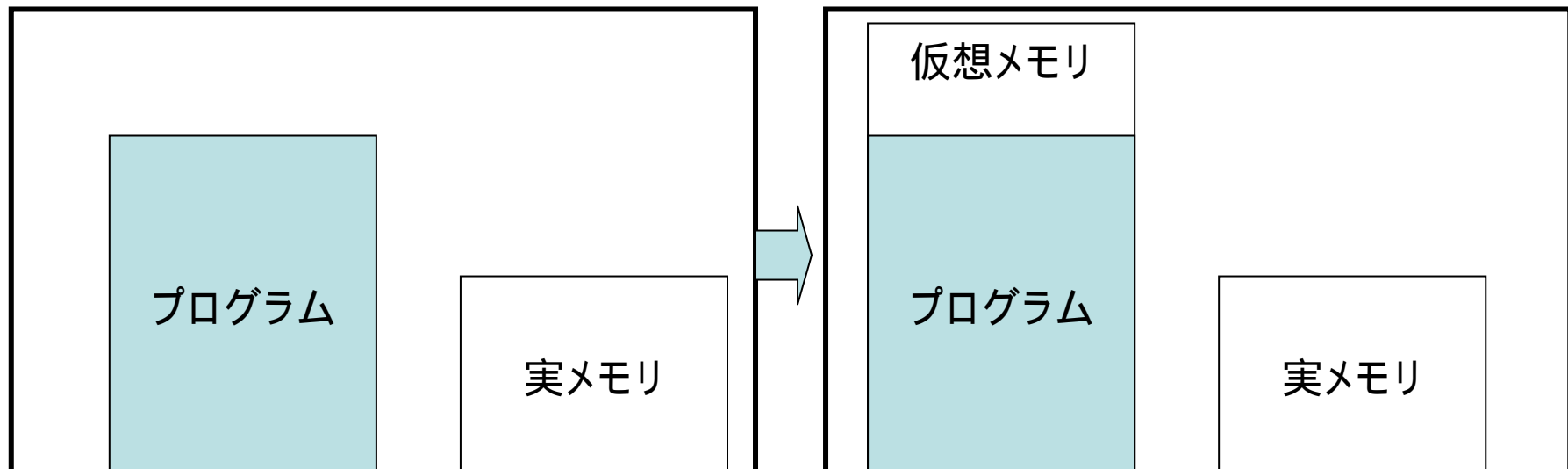
Memory hierarchy



# 仮想メモリ

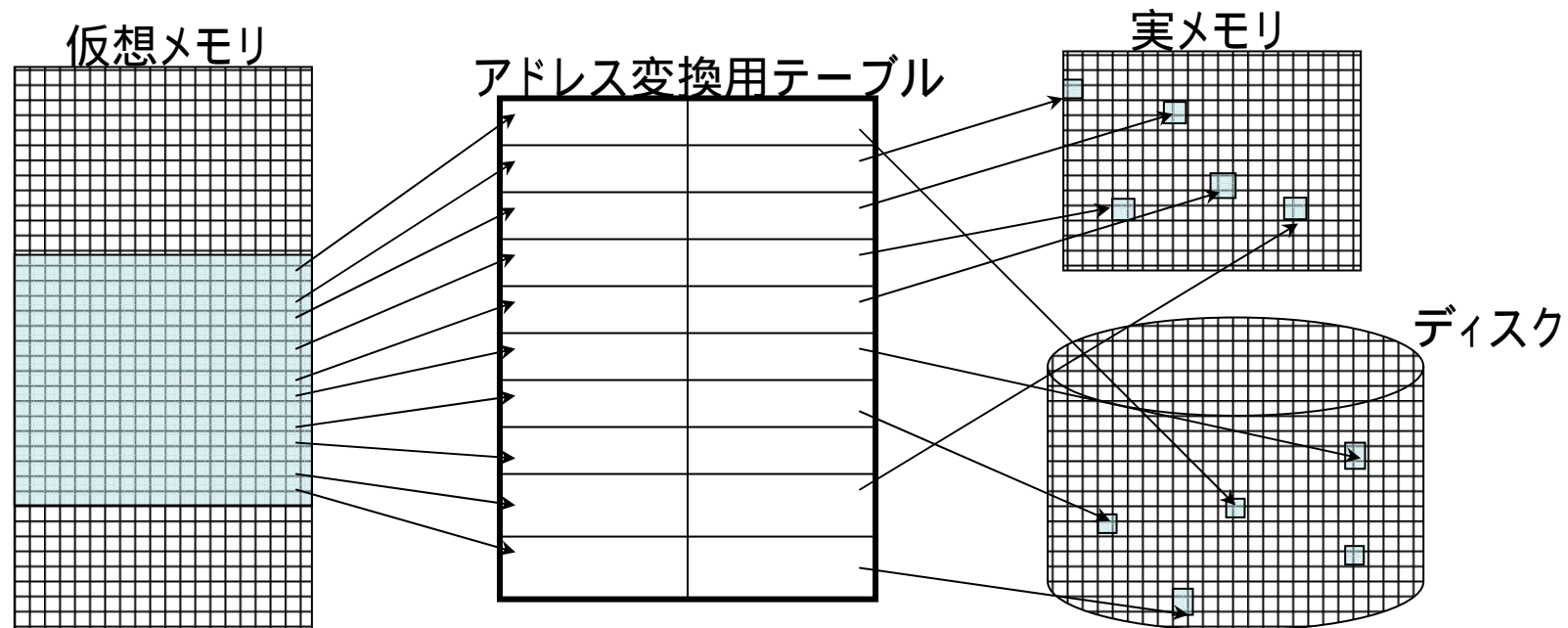
## 仮想メモリの必要性

- 利用可能な実メモリに全部入りきらない
- プログラムをいくつかのより小さなものに分割するという手法で対処
- プログラム開発に要する時間が増え、プログラマーの生産性を下げる



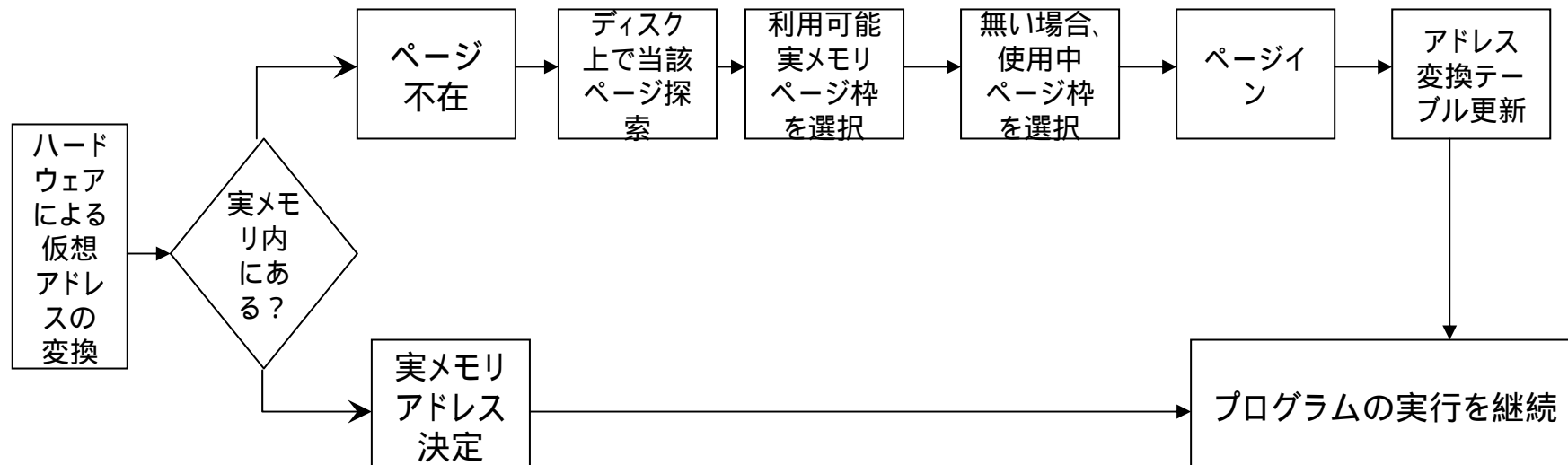
# ページング

- プログラムのうちある部分を活動状態にする必要が生じたときこれをディスクから実メモリに持ち込む
- 稼動状態にあった部分を不要になった段階で実メモリからはきだしてディスクに戻す
  - 仮想メモリをページという単位に分ける
  - 実メモリはページ枠に分割される
  - あるプログラムの仮想メモリページは非活動状態のときディスクに記憶されたままになっている
  - アドレス変換用テーブルが実メモリ上に常駐し仮想メモリアドレスを、実メモリアドレスか、またはディスク上アドレスに結び付けている



# ページ不在

- 必要なページが実メモリになくディスクにある場合、ページ不在といい、ハードウェアにより検出される
- ページ不在が起こった場合、アドレス変換用テーブルを使ってディスク上参照されている当該アドレスを含んだページを仮想メモリ上で見つけ出しそれを実メモリの利用可能なページ枠に運び入れる。
  - 使われていない実メモリページ枠は、利用可能ページ枠として管理されている
    - 利用できるページ枠が無い場合には現在使用中のものからどれかを一つ選びそのページ枠に入っているページを入れ替える。
    - ページ枠内に入ったとき以降変更されている場合ディスク上に書きだす(ページアウト)



## 4.2 オペレーティングシステム (OS)

4.2.1 OSとは

4.2.2 OSの構造

4.2.3 OSの役割

4.2.4 OSによる管理

## 4.2.2 OSの構造

アプリケーションプログラム

コンパイラ   リンカ   ロード  
エディタ  
サービスプログラム   処理プログラム

ジョブ管理   通信管理  
ファイル管理   入出力管理

プロセス管理   記憶管理  
ハードウェア管理  
核(カーネル)

Operating System



## 4.2.1 OSとは

**OSとは**

ED2

**Operating System の略**

**コンピュータハードウェアを**

**動作させるための**

**基本的な**

**ソフトウェア**

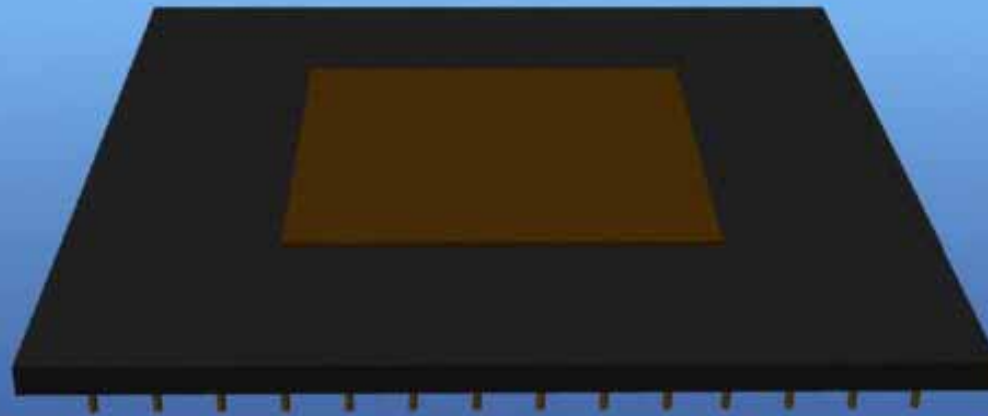


## 4.2.1 OSとは

### OSの種類

ED2

CPUの種類によって  
使用できるOSが異なる



## 4.2.1 OSとは

### OSの種類

ED2

**x86系 = Windows**

x86 プロセッサを  
用いたコンピュータ

||

**AT互換機**



## 4.2.1 OSとは

### OSの種類

ED2

x86系 = Windows  
PowerPC系 = MacOS



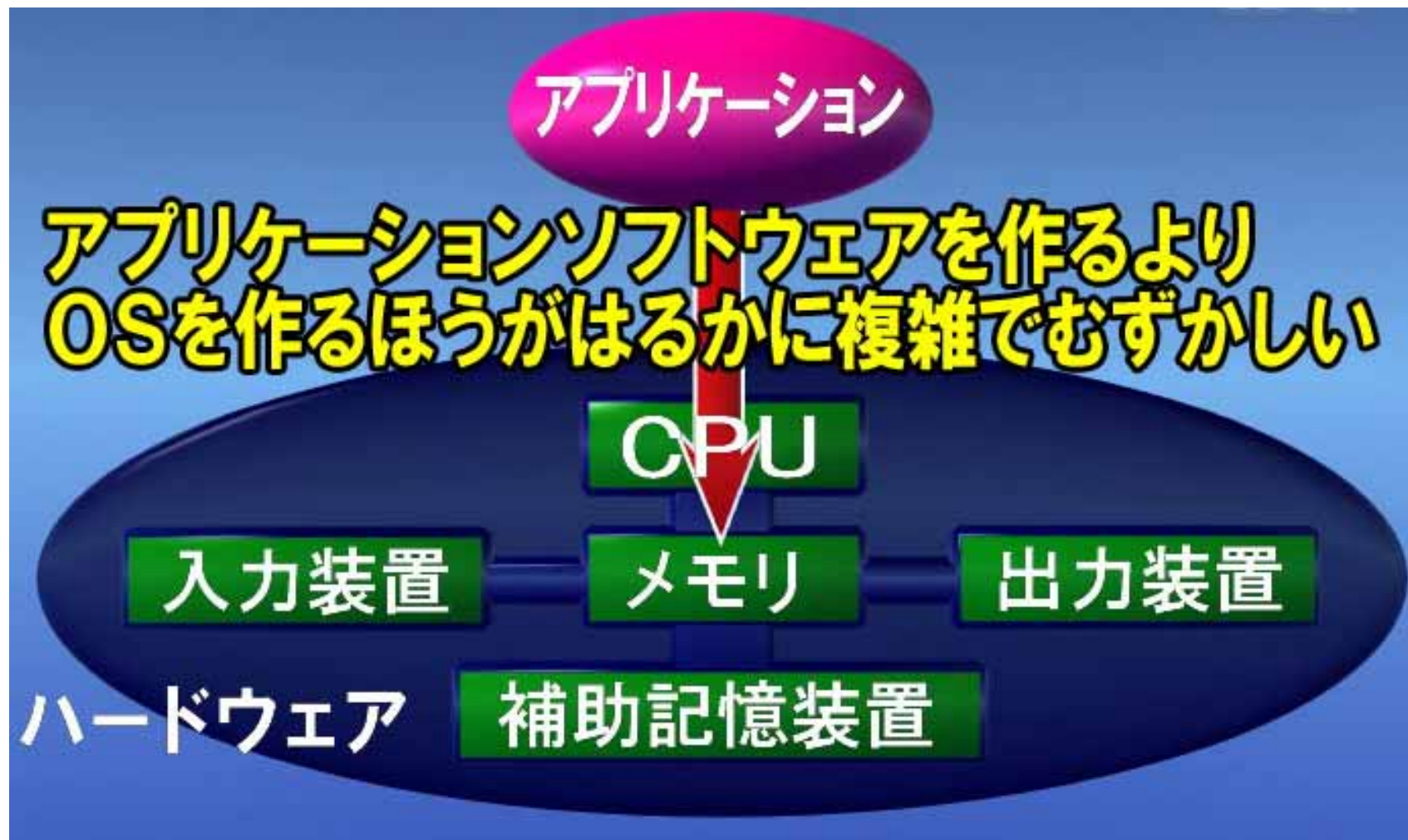
### OSの種類

ED2

ワークステーション = UNIX



## 4.2.1 OSとは



## 4.2.3 OSの役割

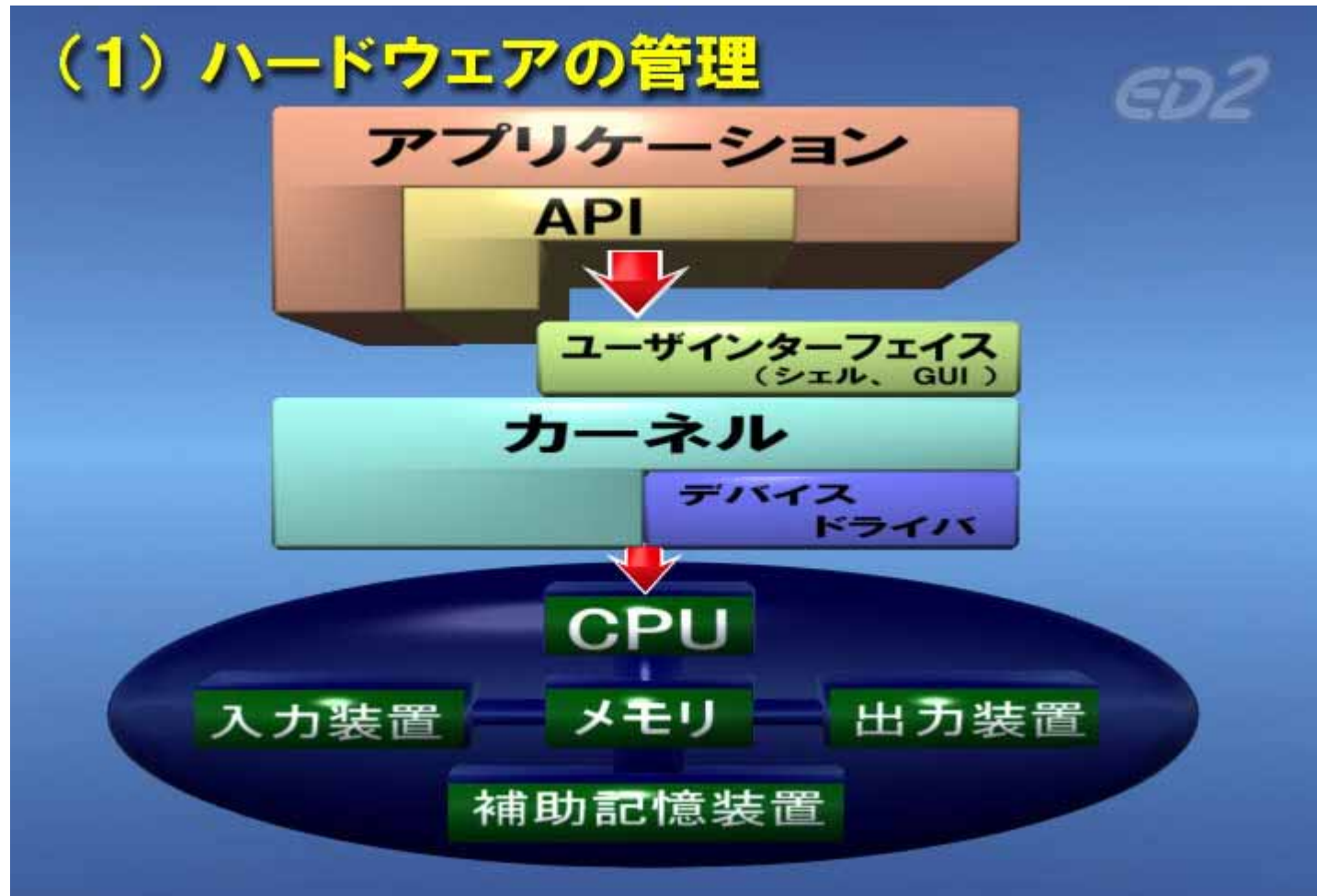
- (1) ハードウェア管理
- (2) 基本機能提供
- (3) ファイルシステム提供
- (4) ネットワーク機能提供
- (5) その他



## 4.2.3 OSの役割

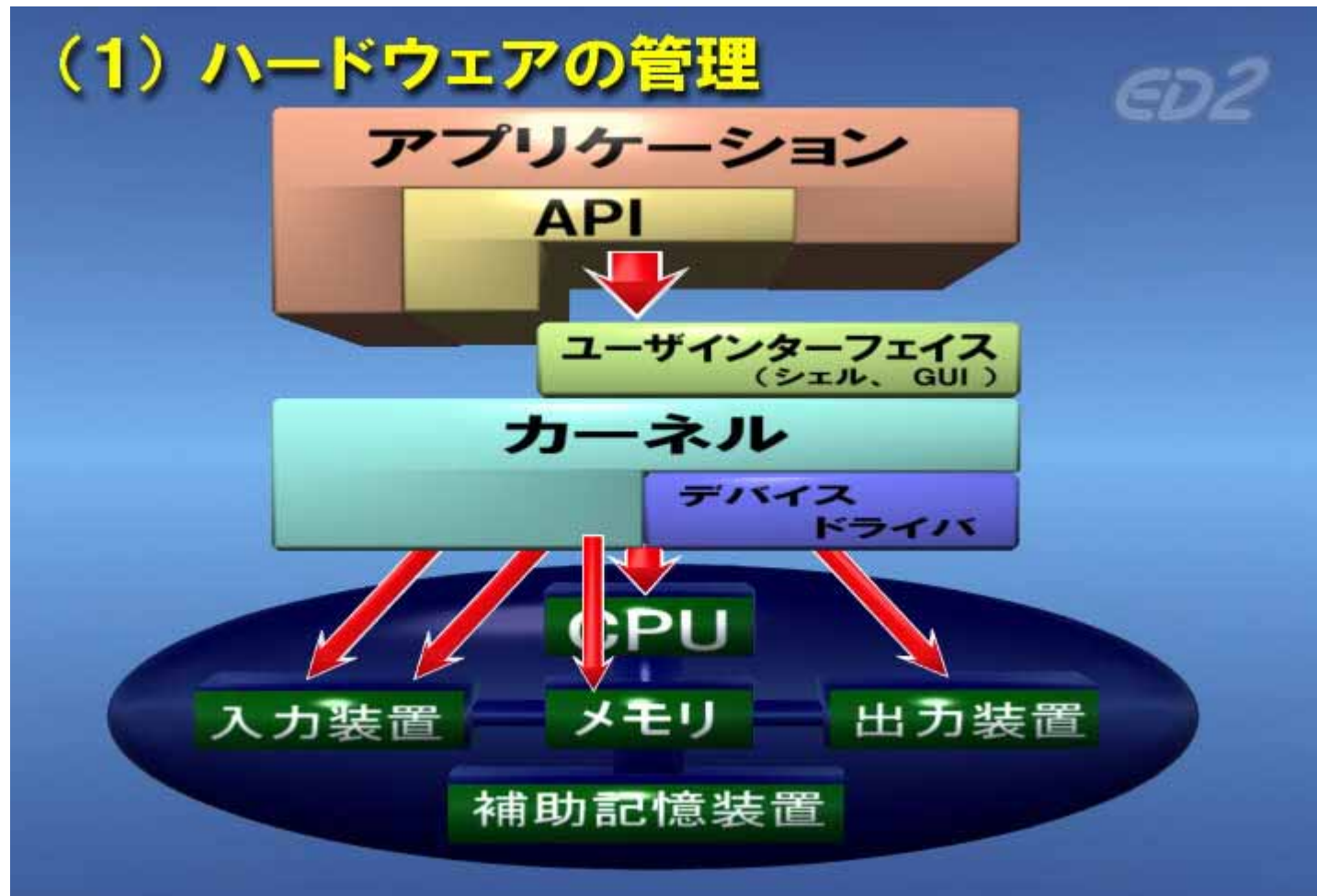


## 4.2.3 OSの役割

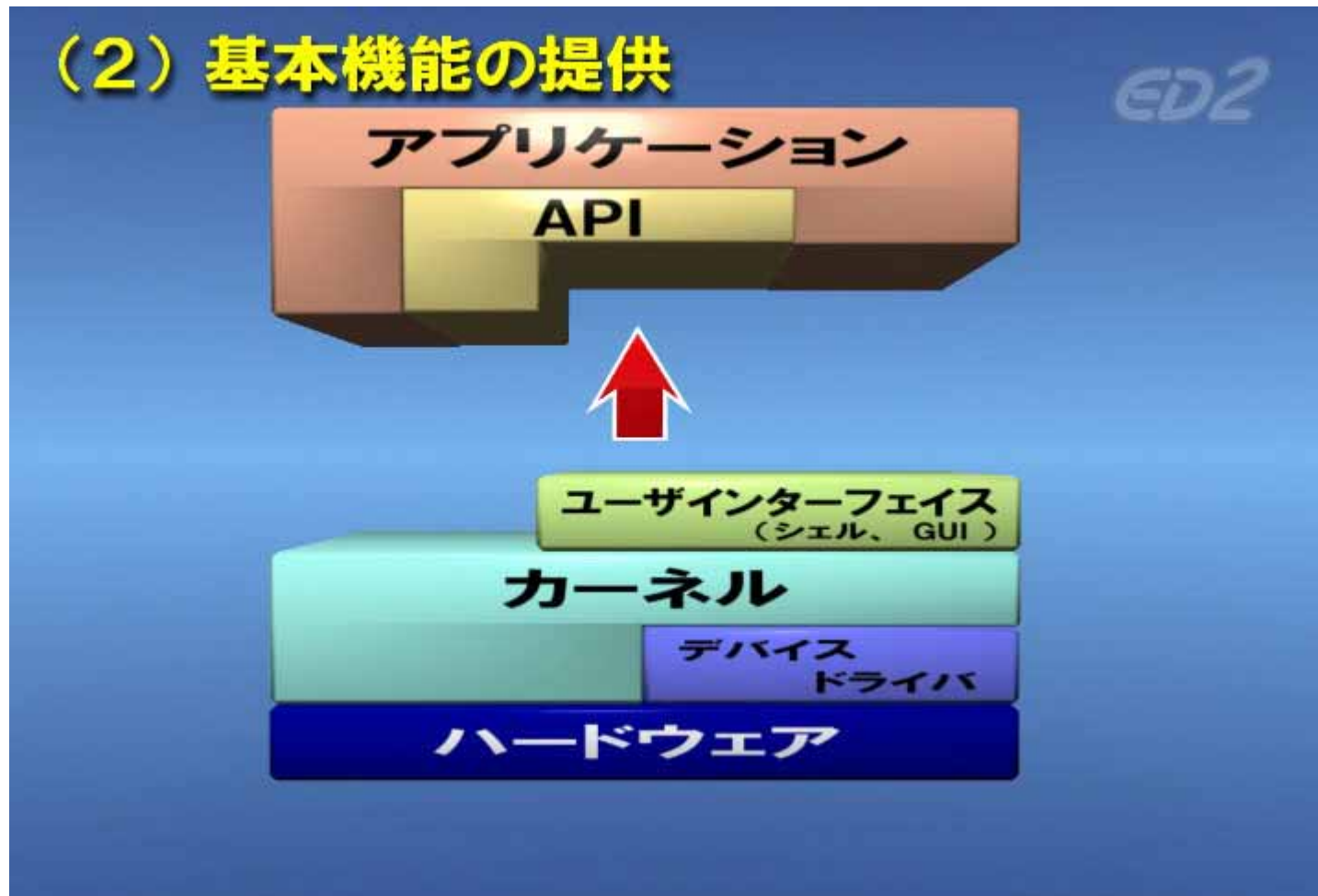




## 4.2.3 OSの役割



## 4.2.3 OSの役割




## 4.2.3 OSの役割

### (2) 基本機能の提供

ED2

アプリケーション

API

メニュー表示、メニューへの応答、印刷  
などの、基本的、 共通的功能を提供

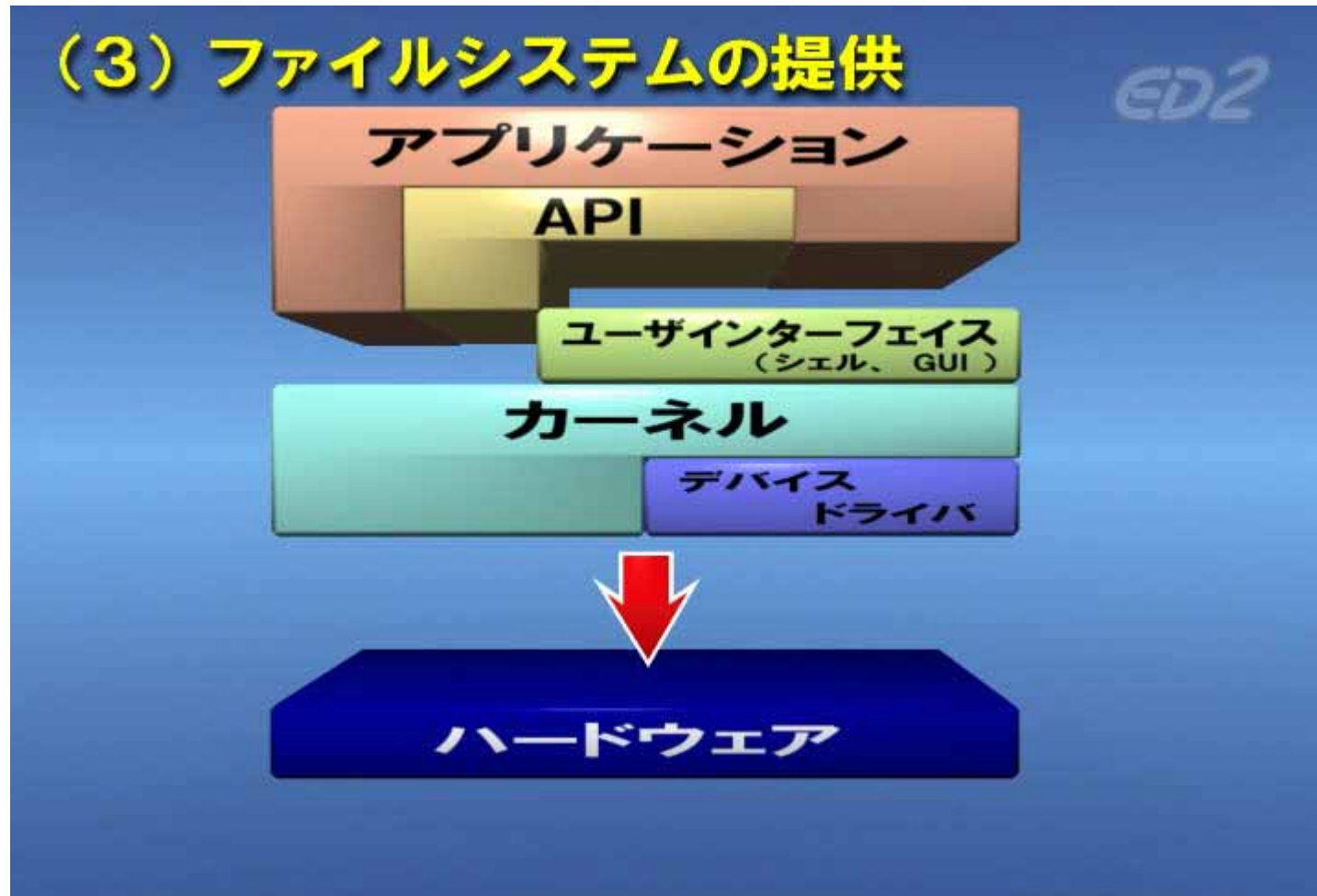
ユーザインターフェイス  
(シェル、GUI)

カーネル

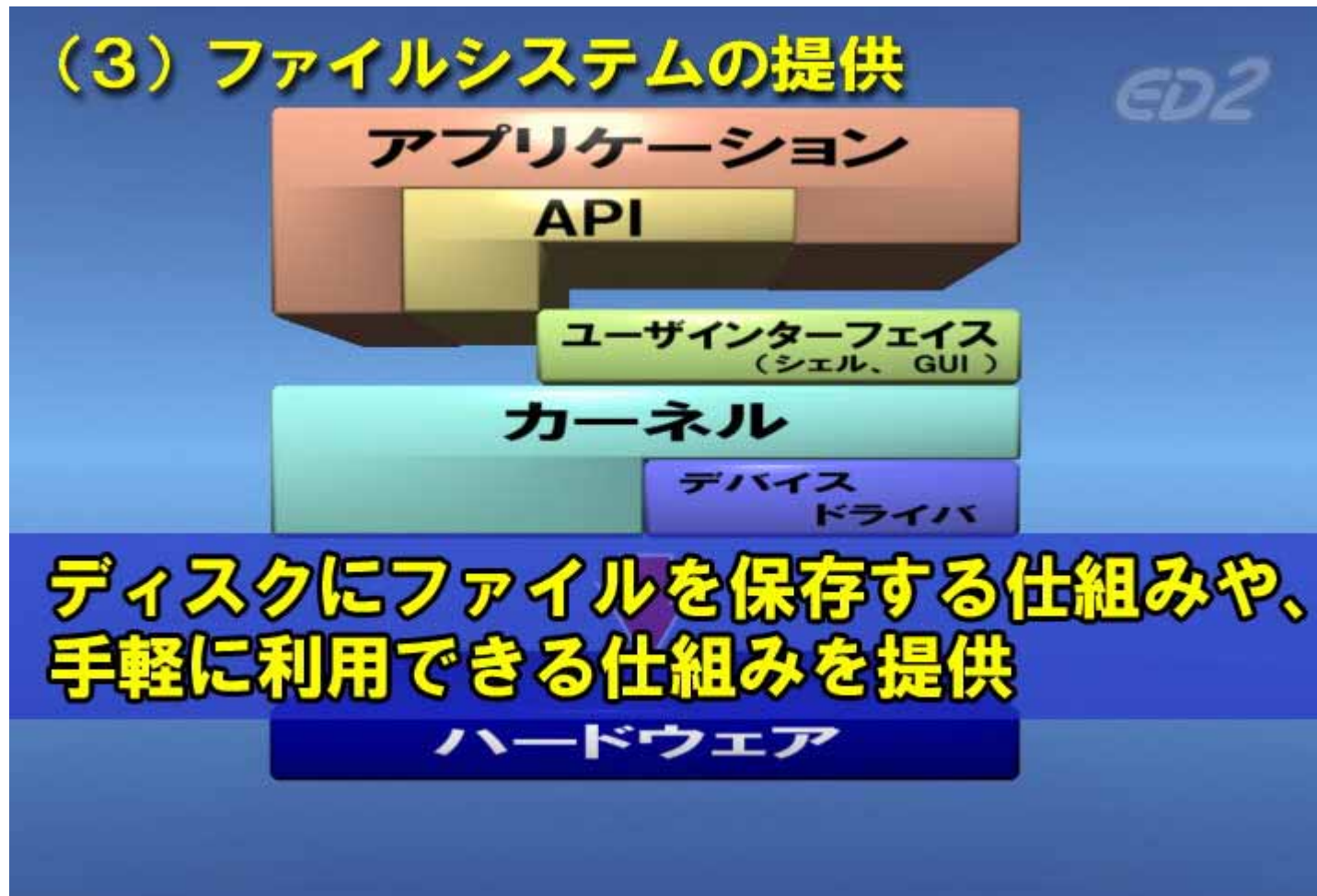
デバイス  
ドライバ

ハードウェア

## 4.2.3 OSの役割

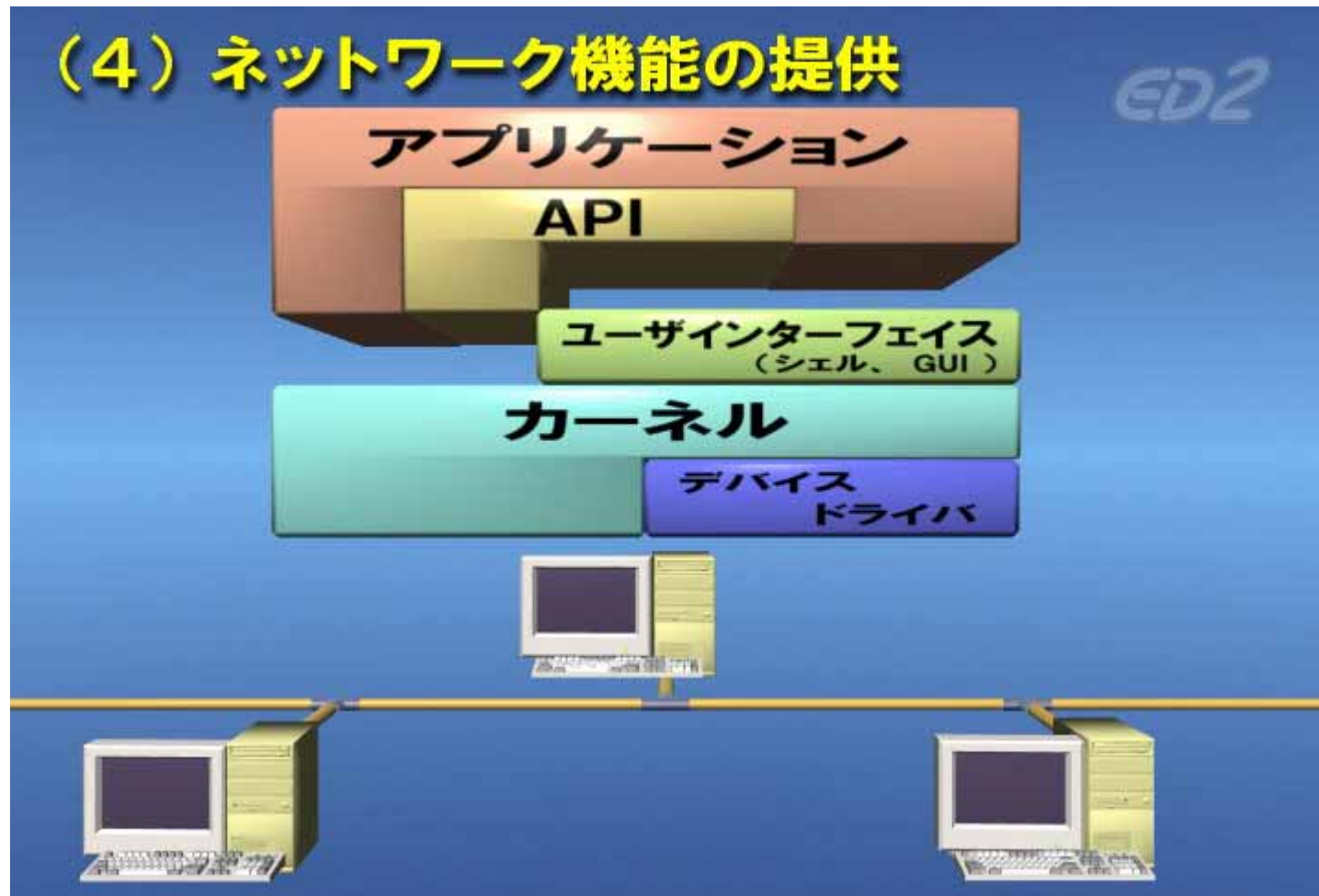


## 4.2.3 OSの役割

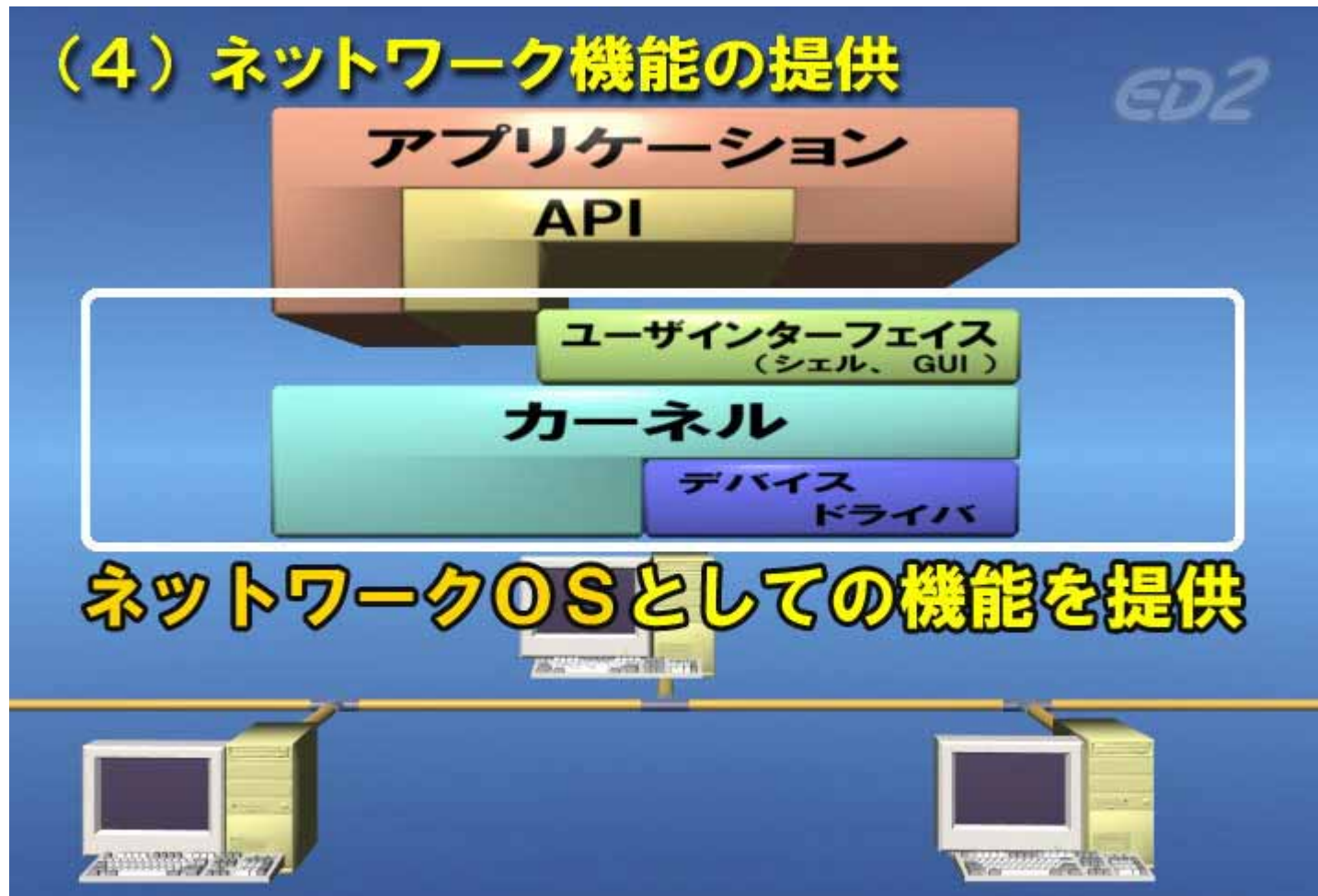




## 4.2.3 OSの役割



## 4.2.3 OSの役割



## 4.2.4 OSによる管理

### 代表的なOS

Unix, Linux, windows, Macintosh, etc.

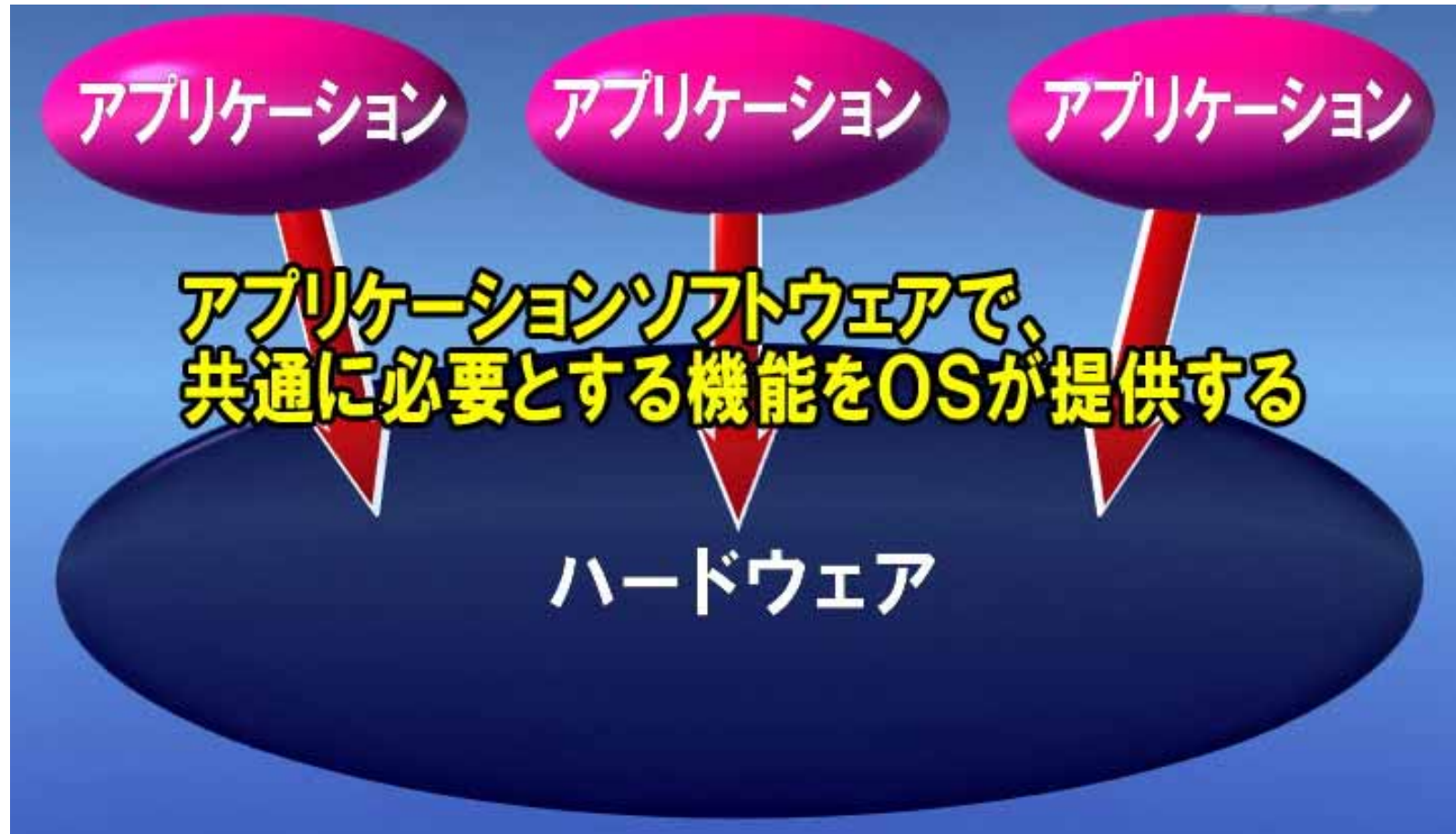
### OSによるファイル管理



Windowsファイルエクスプローラの例



## 4.2.4 OSによる管理



## 4.3 並行処理

4.3.1 並行処理と並列処理

4.3.2 マルチプログラミングとマルチタスク

4.3.3 コンピュータシステムの利用形態

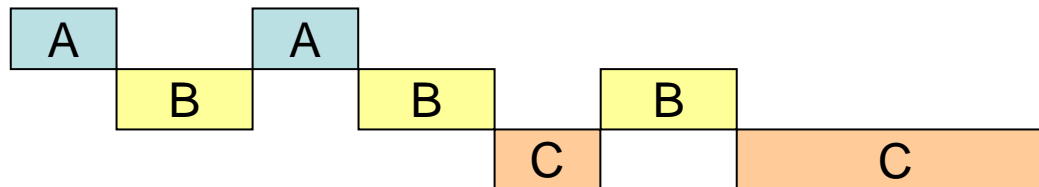
## 4.3.1 並行処理と並列処理

Concurrent processing & parallel processing

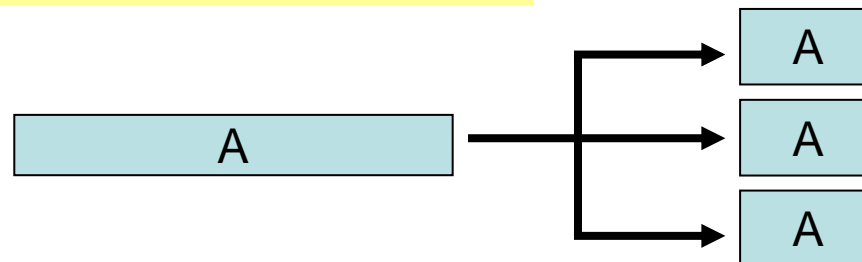
直列処理 (serial processing)



並行処理 (concurrent processing)

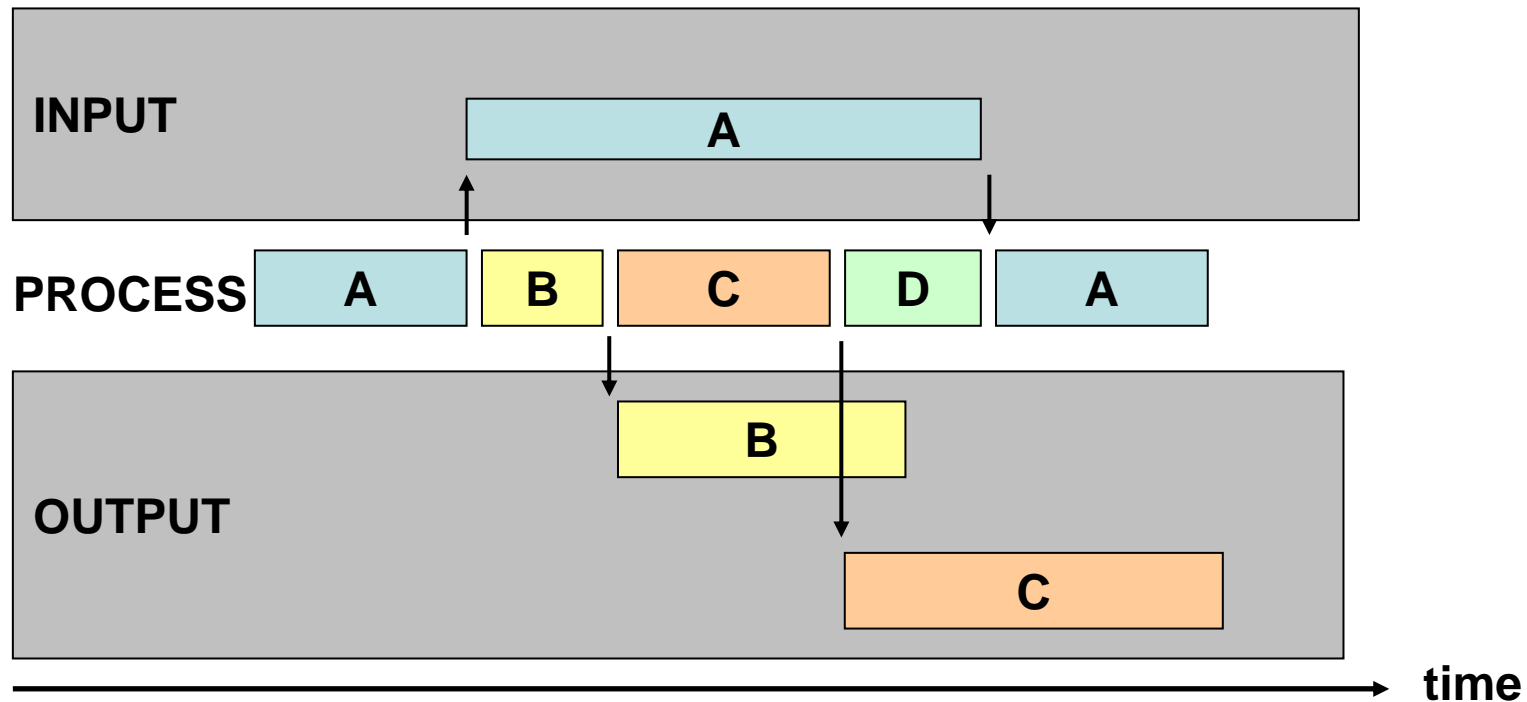


並列処理 (parallel processing)



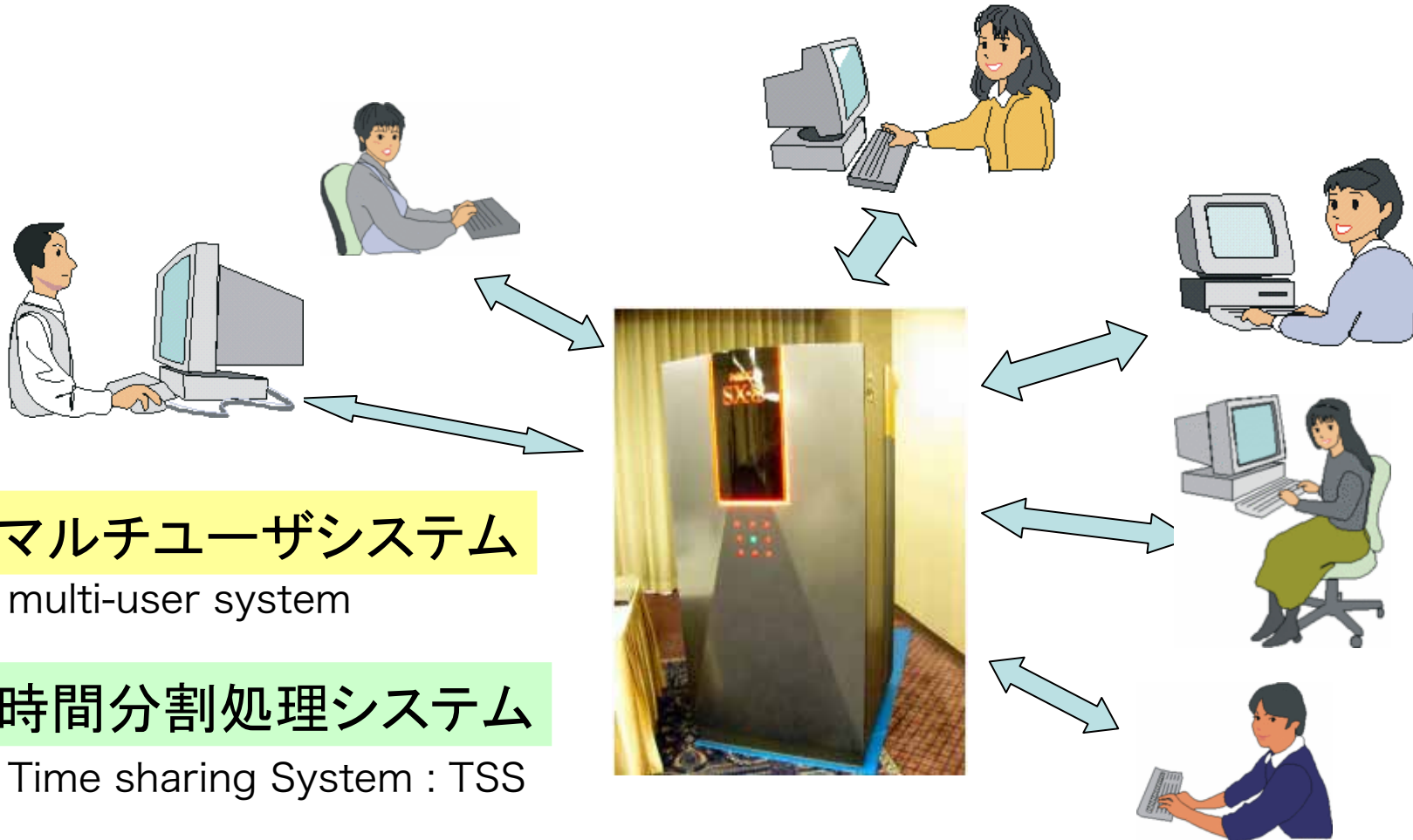
## 4.3.2 マルチプログラミングとマルチタスク

Multiprogramming & multitask



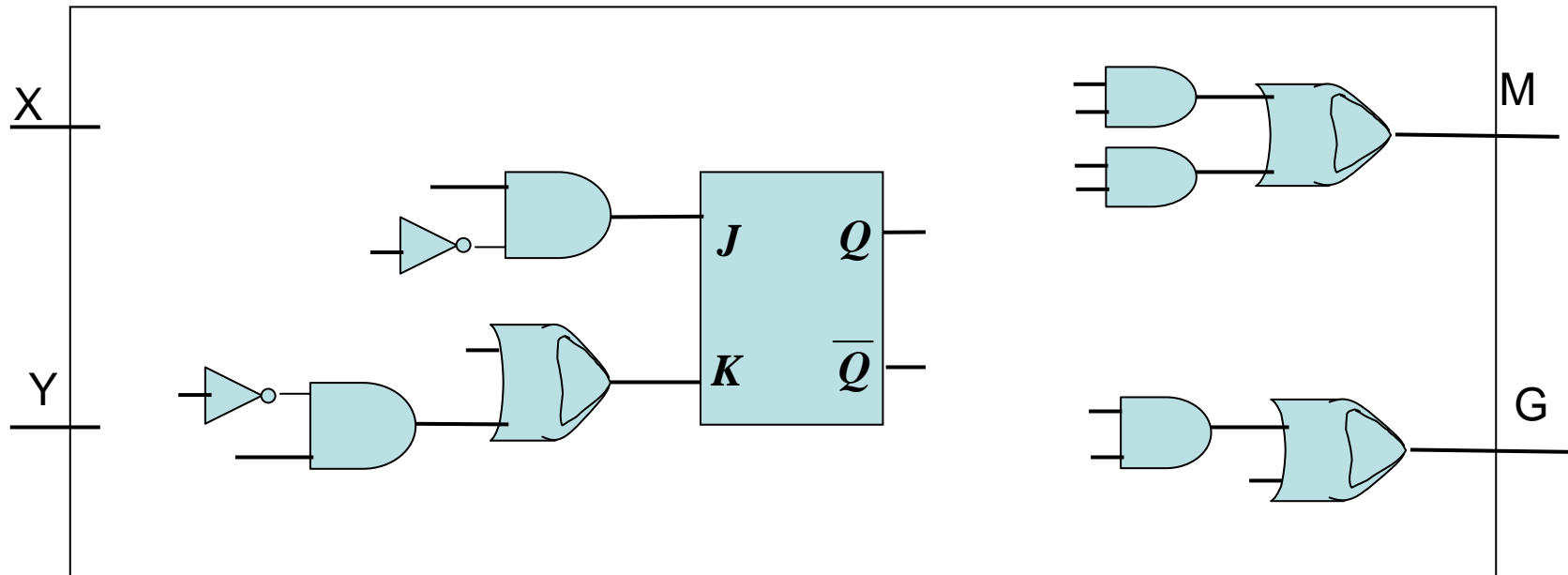
マルチプログラミングの概念図

## 4.4.3 コンピュータシステムの利用形態



# 小テスト(氏名: )

(1) 前述の自動販売機をJKフリップフロップで構成せよ。



(2) 講義に関する感想等を述べよ。