

論理回路（3）

順序回路

2進カウンタ

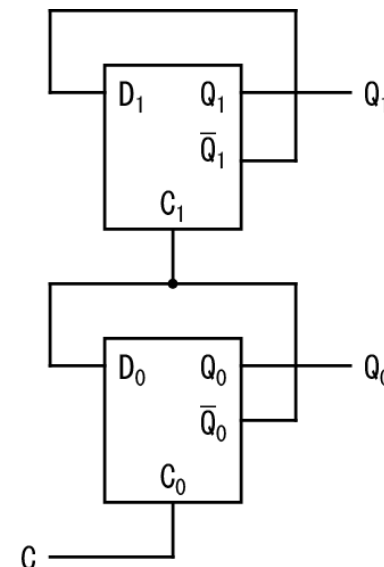
- ▶ 2進カウンタ
 - ▶ n 個のフリップフロップによって n ビットの二進数を表現させる順序回路.
 - ▶ クロックパルスをカウントすることにより, n ビットの二進数を増加あるいは減少させる.
 - ▶ 非同期カウンタと同期カウンタに大別できる.

非同期カウンタ

▶ 非同期カウンタ

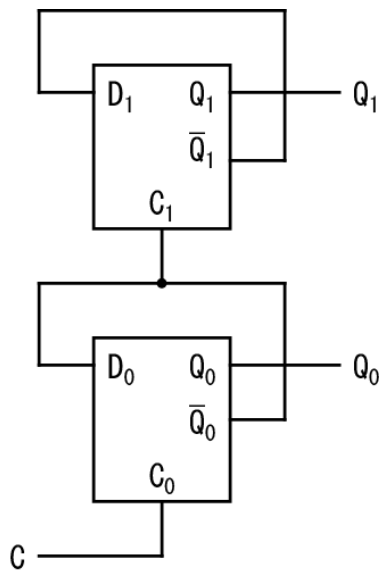
- ▶ フリップフロップのクロック入力として, 前段のフリップフロップの出力を使うカウンタ.
- ▶ カウントパルスが最下位ビットから最上位ビットまで順番に伝播するため, 各フリップフロップの伝播時間が, カウンタ全体の動作時間を左右する.

【例】 2ビット2進カウンタ
(クロックの立ち上がりをトリガとする
Dフリップフロップを用いた構成例)

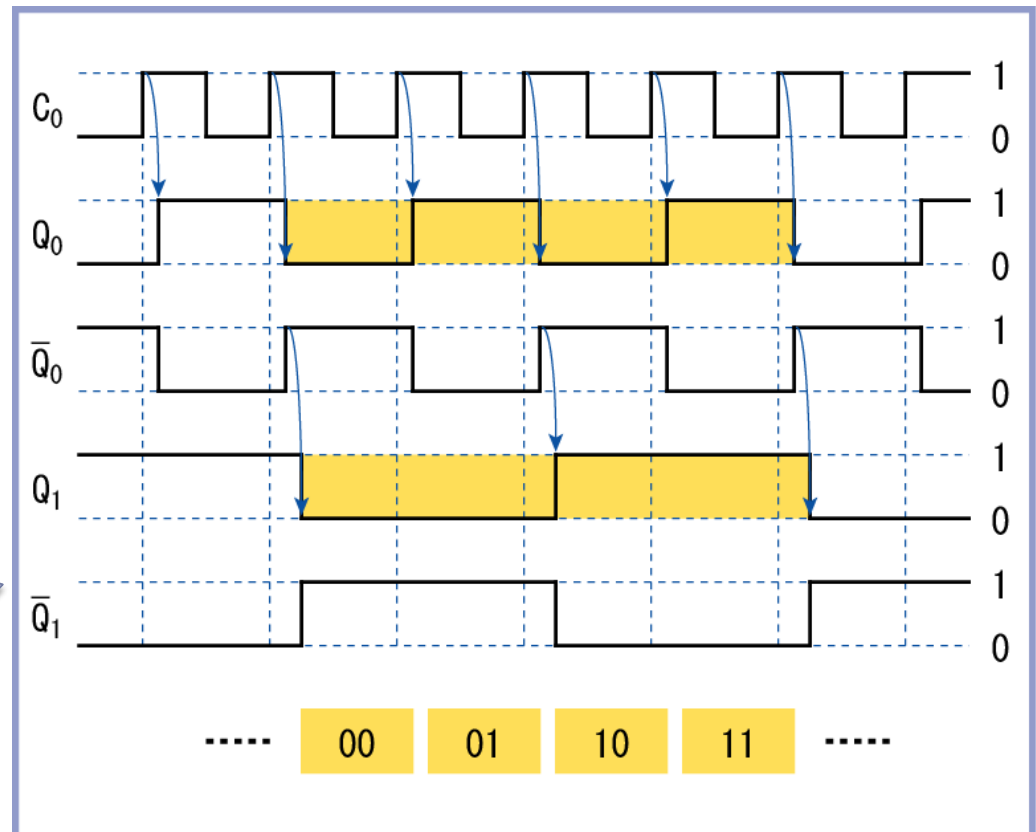


非同期カウンタの動作例

- ▶ 2ビット2進カウンタ
(クロックの立ち上がりトリガとするDフリップフロップを用いた構成例)



非同期カウンタでは、クロックが最下位ビットから最上位ビットまで順番に伝播していくため、最終出力が確定するまでに時間がかかる。



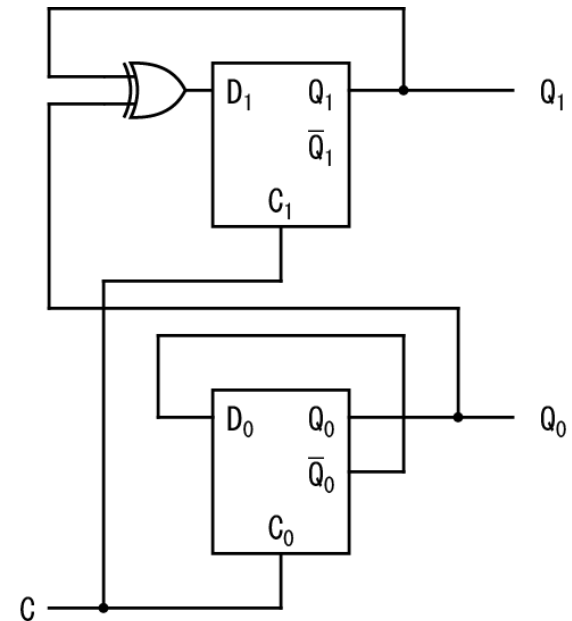
同期カウンタ

▶ 同期カウンタ

- ▶ カウンタを構成するすべてのフリップフロップに同一のクロックを入力し、すべてのフリップフロップが同時に状態変化するカウンタ。

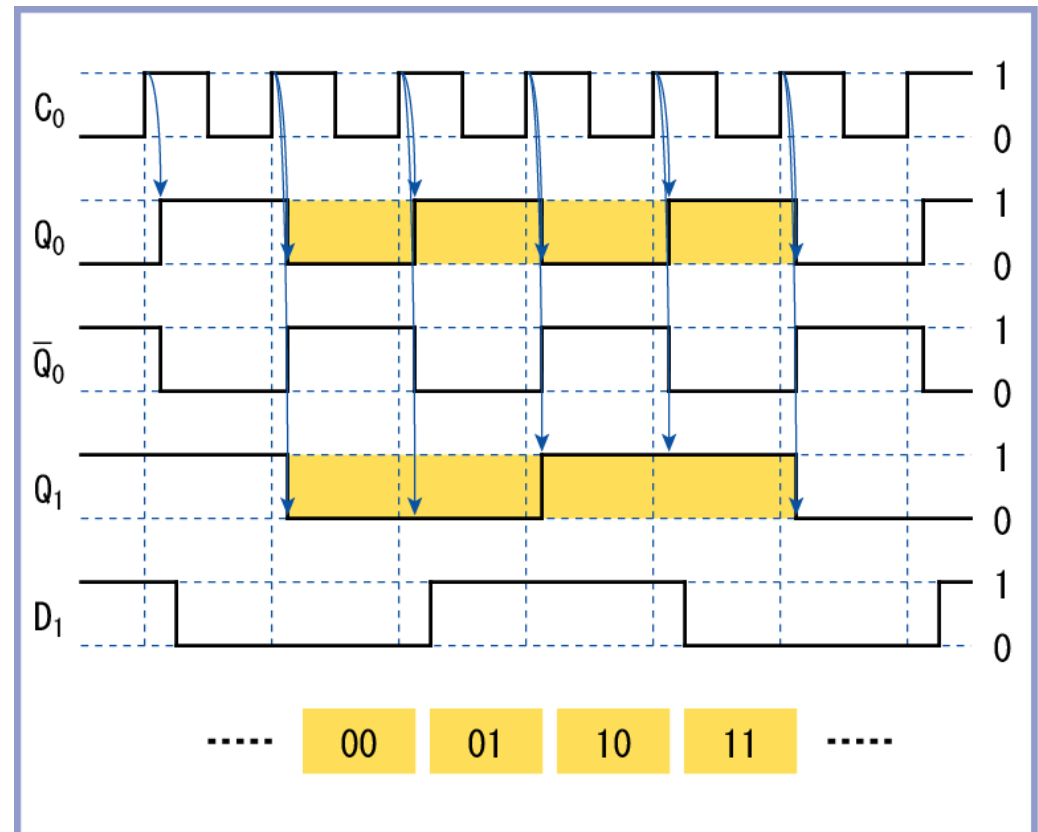
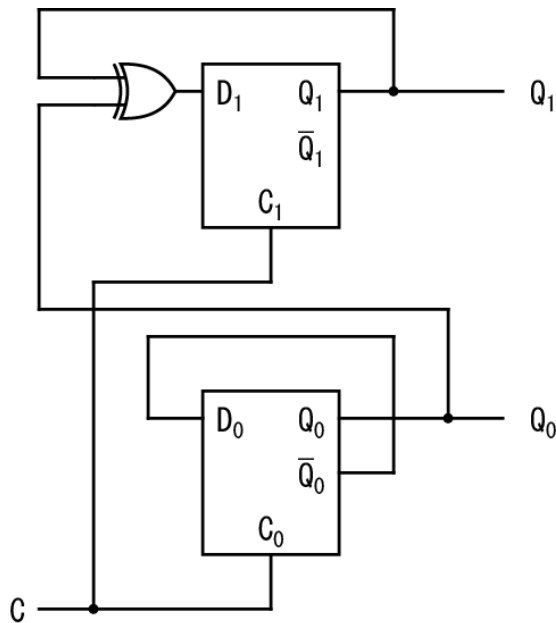
【例】 2ビット2進カウンタ

(クロックの立ち上がりをトリガとする
Dフリップフロップを用いた構成例)



同期カウンタの動作例

- 2ビット2進カウンタ
(クロックの立ち上がりトリガとするDフリップフロップを用いた構成例)

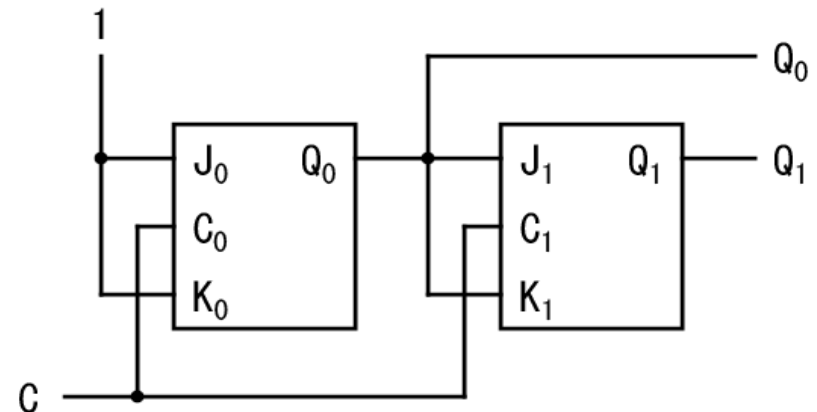


演習問題

問題1

- 下左表は, JKフリップフロップの真理値表である. JKフリップフロップは, クロックCが, 1から0になるごとに(すなわち, クロックの立下りをトリガとして), 真理値表に示した動作をする. ここで, Q_n , Q_{n+1} は, あるクロック周期nでの出力, その直後のクロック周期n+1での出力を, それぞれ表す.
- 下右図に示したJKフリップフロップを2個用いて構成した回路の状態変化を示せ. (タイミングチャートのひな型を次ページに示す.)

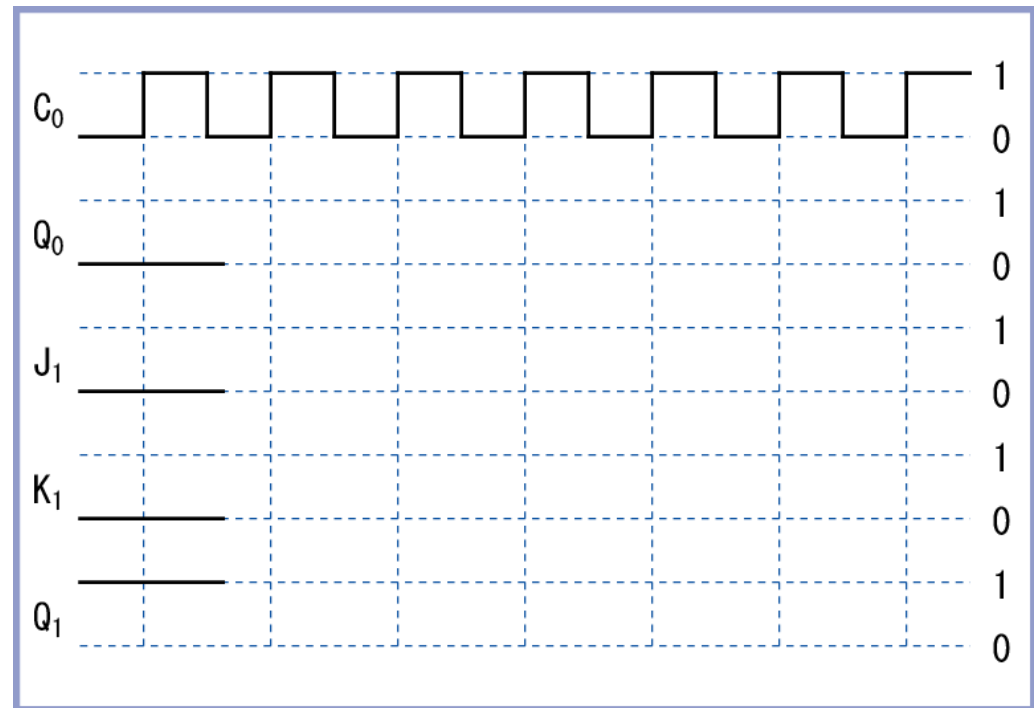
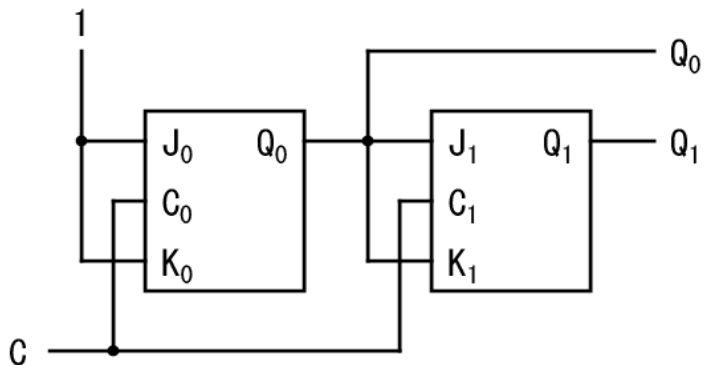
J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	\bar{Q}_n



演習問題

▶ 問題1(続き)

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$



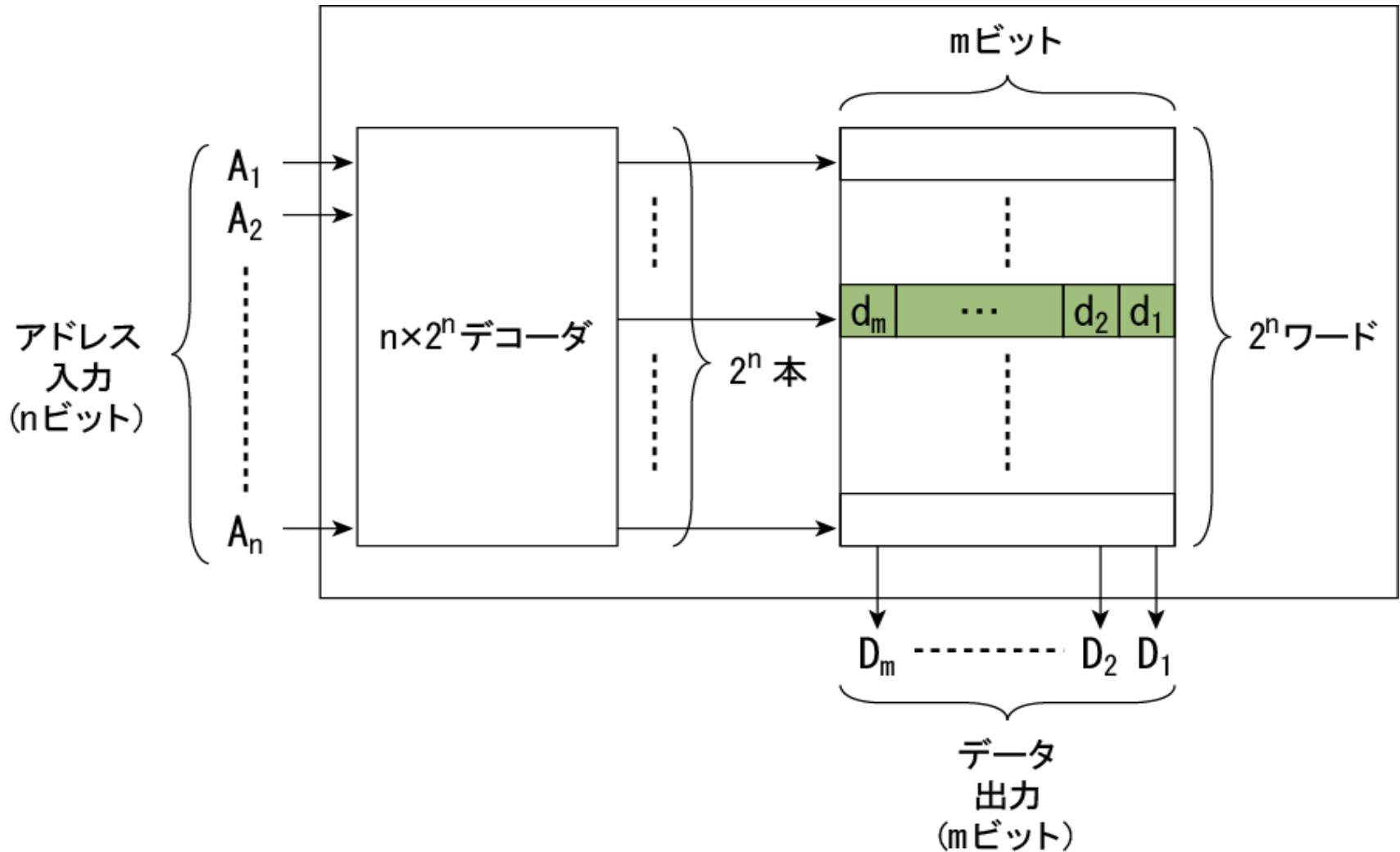
論理回路の実現方法

論理回路の実現方法

- ▶ 論理回路の実現方法
 - ▶ 基本論理素子を組み合わせて、論理回路を実現する.
 - ▶ メモリを用いて、論理回路を実現する.
 - ▶ プログラム可能論理素子を用いて、論理回路を実現する.

以下、メモリを用いた論理回路の実現方法、
プログラム可能論理素子を用いた論理回路の実現方法を説明する.

[補足] メモリの概略構成



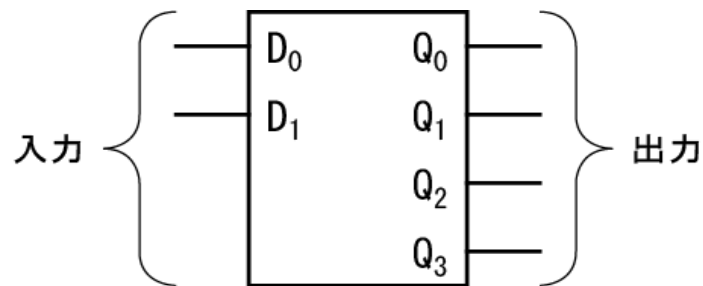
メモリによる組み合わせ回路

- ▶ 論理素子による組み合わせ論理回路
 - ▶ 入力値の組み合わせに対して出力値を生成するものであり, 基本論理素子を用いて構成することができる.
 - ▶ 複雑な組み合わせ論理回路を設計することは, 必ずしも容易ではない.

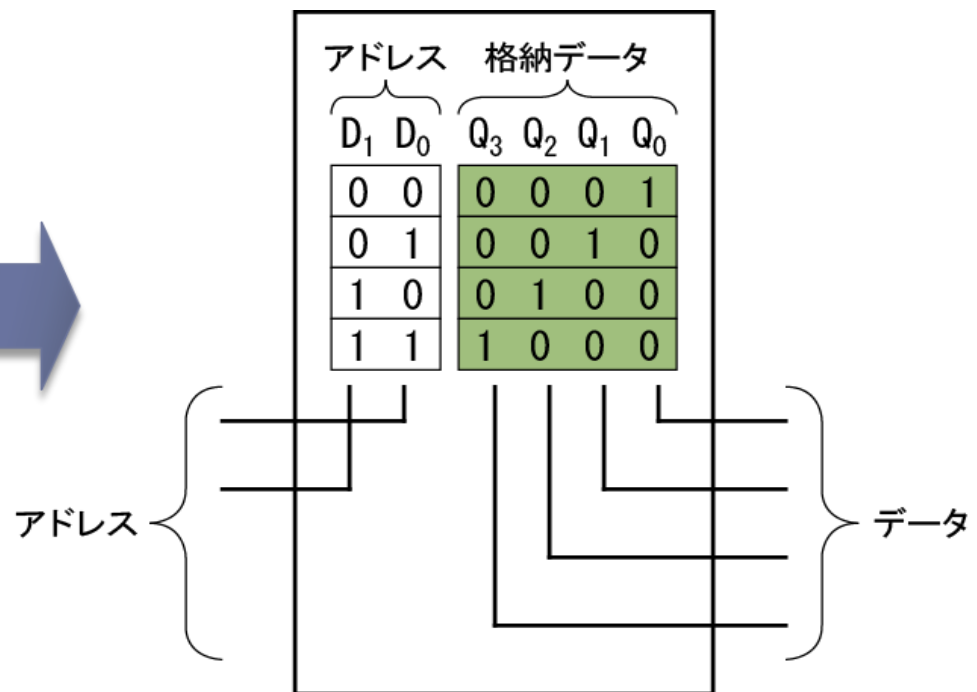
- ▶ メモリによる組み合わせ論理回路
 - ▶ 入力値の組み合わせに対する出力値を, あらかじめメモリに格納しておくことにより, 組み合わせ回路を構成することができる.
(すなわち, メモリに与えるアドレスを入力, 読み出すデータを出力とみなして, メモリを組み合わせ回路として用いる.)

メモリによる組み合わせ回路の例

▶ 【例】 2×4デコーダ



D ₁	D ₀	Q ₃	Q ₂	Q ₁	Q ₀
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0



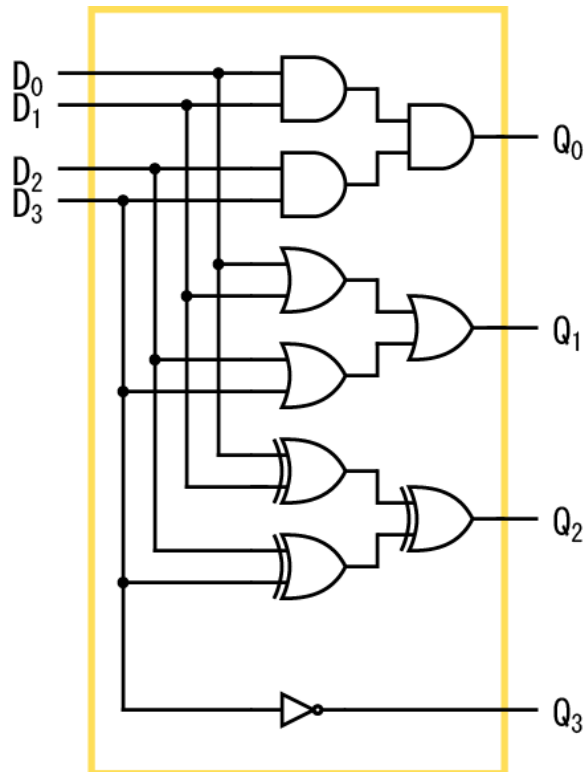
メモリによる組み合わせ回路の特徴

- ▶ メモリによる組み合わせ回路
 - ▶ 長所
 - ▶ 構成が簡単.
 - ▶ メモリを書き換えるだけで, 様々な組み合わせ回路を構成できる.
 - ▶ 短所
 - ▶ 不要な組み合わせに対してもメモリを使用してしまう.

演習問題

問題2

- 下図の組み合わせ回路を、メモリを用いて実現したい。メモリに格納すべき値を、右表に記入せよ。



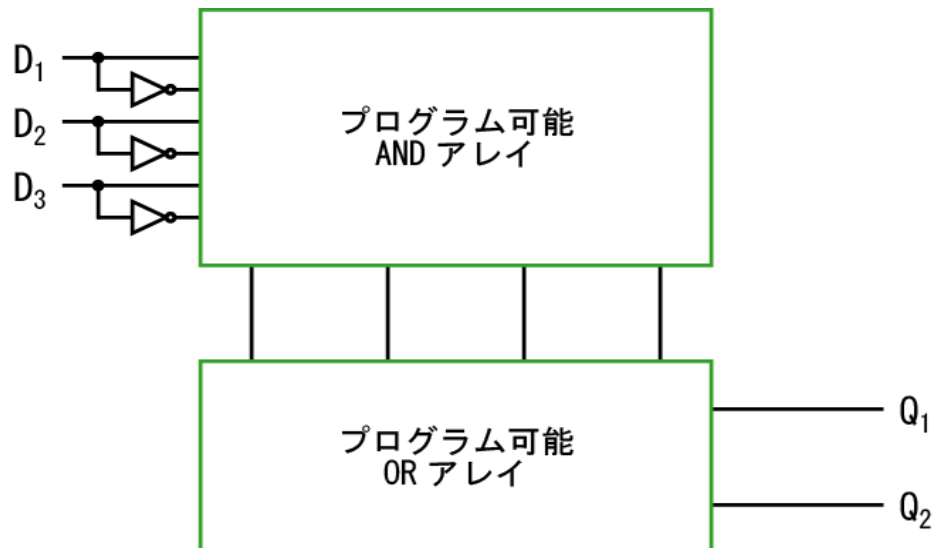
アドレス $D_3 D_2 D_1 D_0$	格納データ $Q_3 Q_2 Q_1 Q_0$
0 0 0 0	
0 0 0 1	
0 0 1 0	
0 0 1 1	
0 1 0 0	
0 1 0 1	
0 1 1 0	
0 1 1 1	
1 0 0 0	
1 0 0 1	
1 0 1 0	
1 0 1 1	
1 1 0 0	
1 1 0 1	
1 1 1 0	
1 1 1 1	

プログラム可能論理素子

- ▶ プログラム可能論理回路(再構成可能論理回路)
 - ▶ メモリや論理素子を規則的に並べたハードウェア素子を用意しておき, 論理素子間の接続情報をプログラムなどで与えることによって, 構成や再構成のできる論理回路.
 - ▶ カスタム(特注の)論理回路を, ソフトウェア的に設計・製作できる.

プログラム可能ロジックアレイ

- ▶ PLA (Programmable Logic Array)
 - ▶ PLAは、入力群とそれに対応する論理否定、そして、以下のような2段の論理で構成される。
 - ▶ 第1段は、プログラム可能なANDのアレイであり、複数の論理積が構成される。
 - ▶ 第2段は、プログラム可能なORのアレイであり、複数の論理和が構成される。

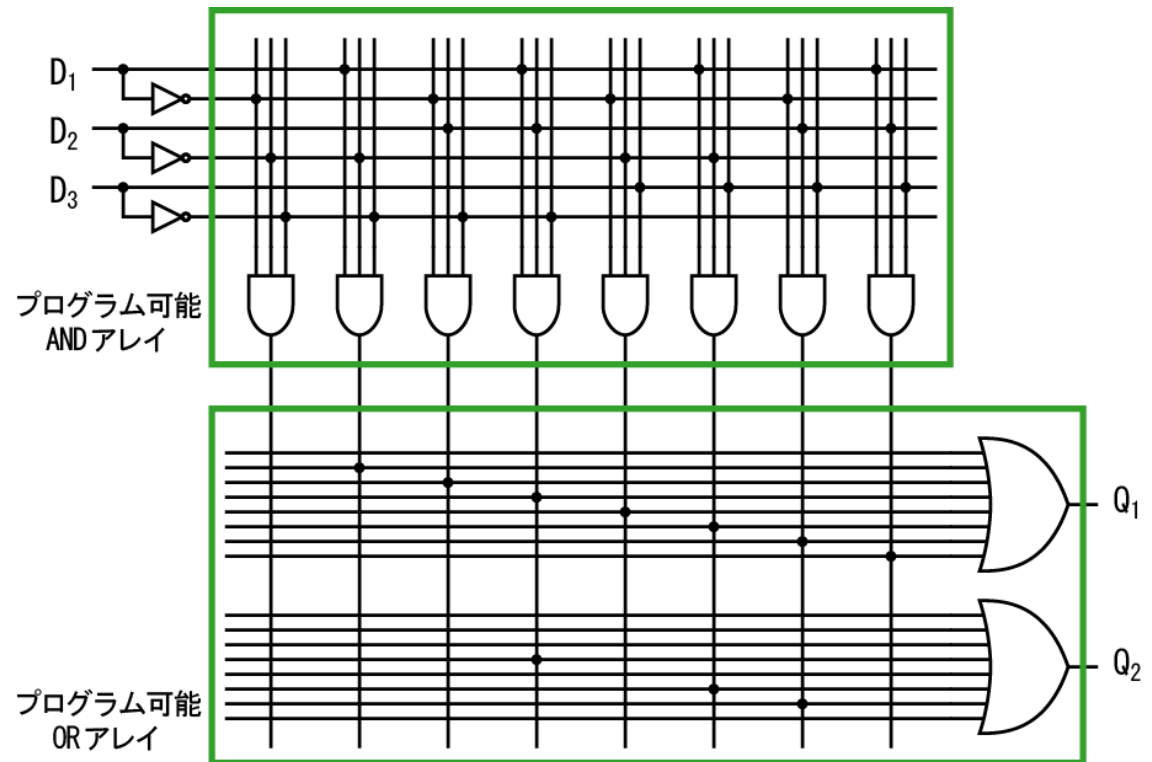


プログラム可能ロジックアレイの例

▶ PLAによる回路例

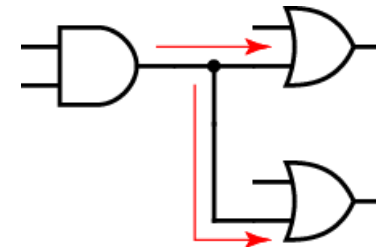
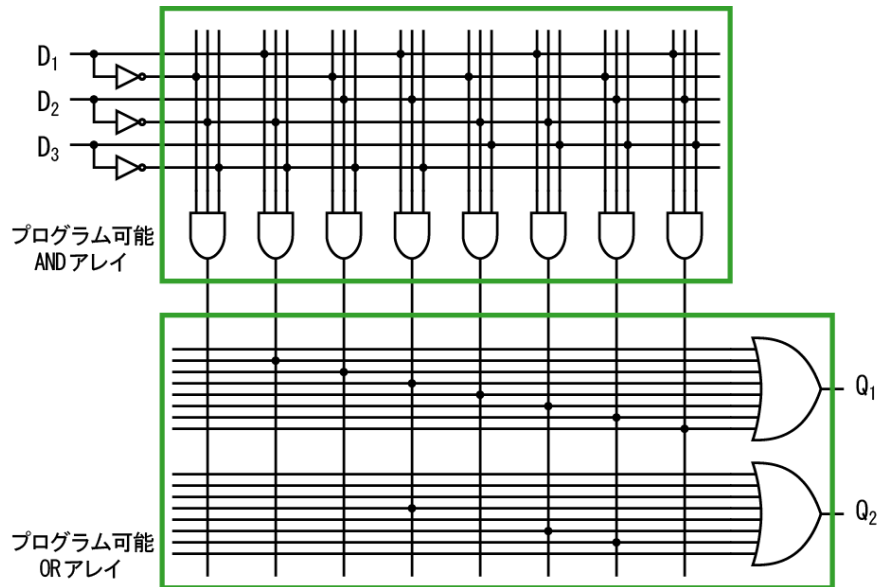
(注) 下図において, 未接続配線は, 論理値 0 とする.

D ₁	D ₂	D ₃	Q ₁	Q ₂
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
1	1	0	1	1
0	0	1	1	0
1	0	1	1	1
0	1	1	1	1
1	1	1	1	0



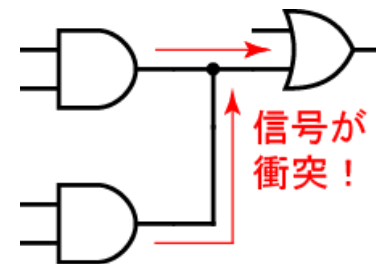
いかなる組み合わせ論理回路も, ANDとORとNOTだけで構成できる.

[補足] 論理素子の入力と出力について



正

1つの出力が、
複数の入力へ



誤

複数の出力が、
一つの入力へ

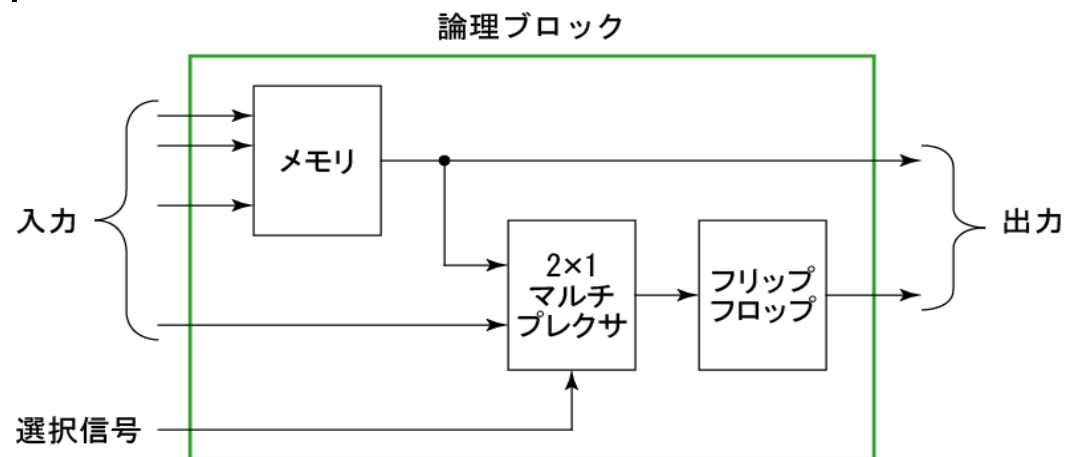
ゲートアレイ

- ▶ ゲートアレイ
 - ▶ NAND素子あるいはNOR素子のいずれかだけで、どのような論理回路も構成できる.
 - ▶ ゲートアレイは、NAND素子あるいはNOR素子のいずれかだけを、規則正しく並べただけの未配線のICである.
 - ▶ 配線情報は、回路設計者が後で指定し、それを工場でゲートアレイに組み込む(配線する).
 - ▶ ゲートアレイは、配線情報だけがプログラム可能である半注文生産の論理素子である.

フィールド・プログラマブル・ゲートアレイ

- ▶ FPGA (Field Programmable Gate Array)
 - ▶ 配線情報の組み込みが使用者の現場(フィールド)で行えるゲートアレイのことをFPGAという.
 - ▶ フィールドでの配線情報の組み込みを実現するために、「論理ブロック」と呼ぶNAND素子やNOR素子よりも機能が高い論理回路を規則正しく並べてある.
 - ▶ フィールドでの配線情報の組み込みができる一方で、ゲートアレイに比べると配線の自由度は低く、使用効率が低くなる.
 - ▶ ハードウェアの高速性とソフトウェアの柔軟性とを兼備した「柔軟かいハードウェア」といえる.

- ▶ 論理ブロックの例



演習問題

問題3

- 下左図の真理値表で示される組み合わせ回路を，下右図のPLAにて構成したい。接続する必要のある交点に，黒丸を記入せよ。なお，未接続配線は，論理値0とする。

D ₁	D ₂	D ₃	Q ₁	Q ₂	Q ₃
0	0	0	1	0	0
1	0	0	1	1	0
0	1	0	1	0	0
1	1	0	0	1	0
0	0	1	1	1	0
1	0	1	1	0	0
0	1	1	1	0	0
1	1	1	0	0	1

