

専門基礎科目 - 必修科目

GB10804 論理回路

1

システム情報系情報工学域 山口佳樹

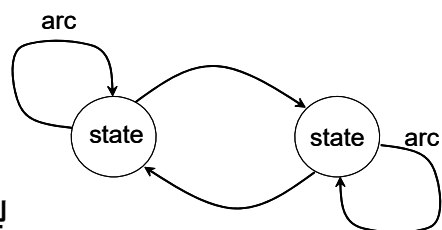
専門基礎科目 - 必修科目 (情報科学類)
基礎科目 - 関連科目 (情報メディア創成学類)
基礎科目 - 関連科目 (工学システム学類)

1

状態遷移図

- 取りうるべき状態と状態から状態への変化を表現したグラフ

- 状態 (state)
- 方向のある線(arc)



- ムーア型とミーリー型

2

順序回路の設計 (フリップフロップ プログラミング)

1. 状態遷移図(ミリーグラフ)
 - 必要があれば、状態遷移図の簡単化
2. 状態遷移表の作成
3. 状態に2進数を割り当てる
4. 組み合わせ論理回路による設計
 - 次の状態
 - 出力

3

例題

- $\rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 3 \rightarrow \dots$ と数えるような**カウンタ**を、フリップフロッププログラミングで設計せよ。
 - ただし、サイコロカウンタと同様に、入力信号 $S=H(1)$ の時は動作し、 $L(0)$ の時は止まっている(同じ値でいる)ものとする。
 - (ヒント: 状態は5つしかないが、状態を表すのに3bit必要。カルノー図にはdon't careが出て来る)
 - 解答は、状態を表す3bitのそれぞれの論理式を導くまでも構いませんが、余裕のある人は回路図まで考えてみてください。

4

カルノー図 現在の状態 (Current State) から次の状態 (Next State) を導く。

N ₂					N ₁					N ₀				
SC ₂₄ C ₁ C ₀	00	01	11	10	SC ₂₄ C ₁ C ₀	00	01	11	10	SC ₂₄ C ₁ C ₀	00	01	11	10
00					00					00				
01					01					01				
11					11					11				
10					10					10				

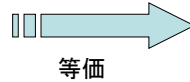
5

状態の簡単化について

6

状態の簡単化

状態遷移図
(状態数 x)

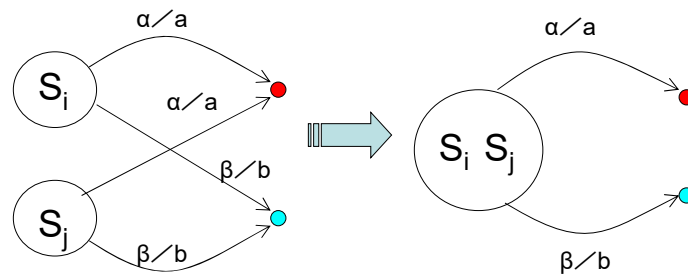


等価

簡単な状態遷移図
(状態数 y)

$x > y$

- すべての入力の組み合わせに対して、出力が同じ
- すべての入力の組み合わせに対して、遷移先が同じ



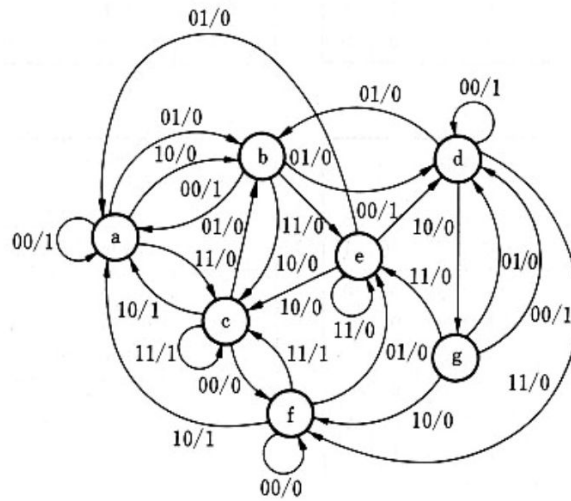
7

状態の簡単化の方法

- 状態図より遷移表A0をつくる
- 遷移表A0を用いて, すべての入力の各々に対して同じ出力を与える状態を一つにまとめて組分けを行い表A1を作る
- 表A1の各組のなかで, すべての入力の各々に対して遷移先が同じである状態は等価であるから, 一つの状態にまとめ、新しい遷移表A2を作る
- 新しい遷移表の各組の中で, すべての入力に対して遷移表が等しくないものがあるか？
 - YES
 - その組には等価でない状態が存在するので, これを別の組にあるよう組分けを変更し新しい遷移表をつくり、この操作を繰り返す
 - NO
 - 各組はすべて等価な状態だけとなり, 状態の簡単化は終了。改めて状態を割り当てれば, 状態数最小の順序回路が得られる。

8

最初の状態



9

A0表

S \ 入力	N(次の状態)				出力			
	00	01	10	11	00	01	10	11
a	a	b	b	c	1	0	0	0
b	a	d	c	e	1	0	0	0
c	f	b	a	c	0	0	1	1
d	d	b	g	f	1	0	0	0
e	d	a	c	e	1	0	0	0
f	f	e	a	c	0	0	1	1
g	d	d	f	e	1	0	0	0

10

A1表

S'(新)	S	N(オリジナル)				N'(新状態)				出力			
		00	01	10	11	00	01	10	11	00	01	10	11
A	a	a	b	b	c	A	A	A	B	1	0	0	0
	d	d	b	g	f	A	A	A	B	1	0	0	0
	b	a	d	c	e	A	A	B	A	1	0	0	0
	e	d	a	c	e	A	A	B	A	1	0	0	0
	g	d	d	f	e	A	A	B	A	1	0	0	0
B	c	f	b	a	c	B	A	A	B	0	0	1	1
	f	f	e	a	c	B	A	A	B	0	0	1	1

11

A2表

S'(新)	S	N(オリジナル)				N'(新状態)				出力			
		00	01	10	11	00	01	10	11	00	01	10	11
A	a	a	b	b	c	A	B	B	C	1	0	0	0
	d	d	b	g	f	A	B	B	C	1	0	0	0
B	b	a	d	c	e	A	A	C	B	1	0	0	0
	e	d	a	c	e	A	A	C	B	1	0	0	0
	g	d	d	f	e	A	A	C	B	1	0	0	0
C	c	f	b	a	c	C	B	A	C	0	0	1	1
	f	f	e	a	c	C	B	A	C	0	0	1	1

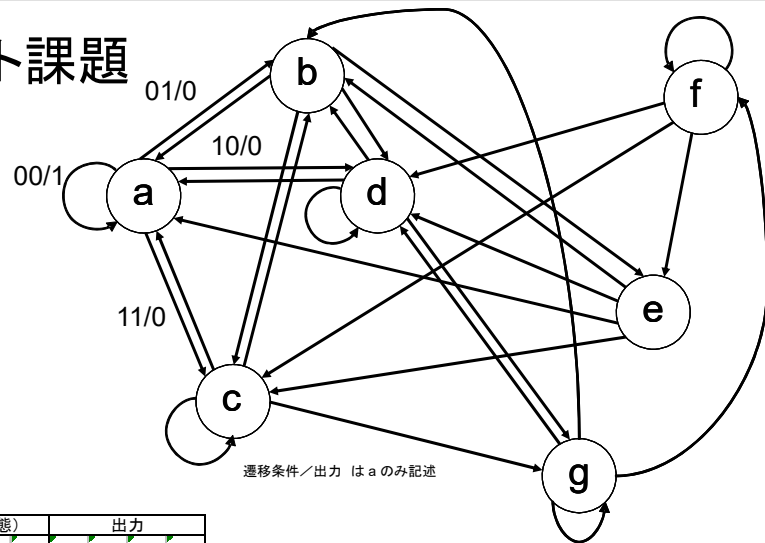
12

簡単化された状態遷移

```
graph LR; A((A)) -- "00/1" --> A; A -- "10/0" --> B((B)); B -- "01/0" --> A; B -- "11/0" --> B; B -- "10/0" --> C((C)); C -- "01/0" --> B; C -- "00/0" --> C; C -- "11/1" --> C; A -- "10/1" --> C; C -- "11/0" --> A;
```

状態遷移図の簡単化

レポート課題



A0表

入力 S	N(次の状態)				出力			
	00	01	10	11	00	01	10	11
a	a	b	d	c	1	0	0	0
b	e	c	a	d	0	0	1	0
c	c	b	g	a	1	0	0	0
d	d	b	g	a	0	0	1	0
e	b	a	c	d	0	0	1	0
f	f	e	d	c	1	0	0	0
g	g	b	d	f	0	0	1	0

上記の状態遷移図について簡略化を適用し、最小の状態遷移図を解答しなさい。A0表は左に示す。

レポートは manaba より提出すること。

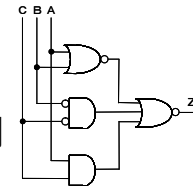
15

メモリによる組み合わせ回路の実現

16

組み合わせ回路の実現

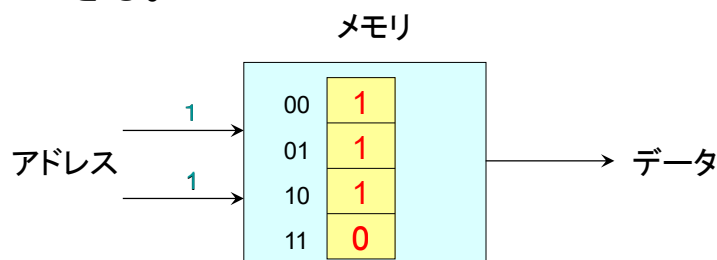
- 論理ゲートによる実現
 - TTLやCMOSデバイスによる実装
 - トランジスタによるインバータ回路を利用
- メモリによる実現
 - 全ての入力に対する出力の値をメモリに蓄えておく
 - 入力の数が大きくなると、多くのメモリが必要



17

メモリとは

- 機能的には、1次元の配列
- アドレスを与えると、そのアドレスに格納された値を読み出したり、書き換えたりすることができる。



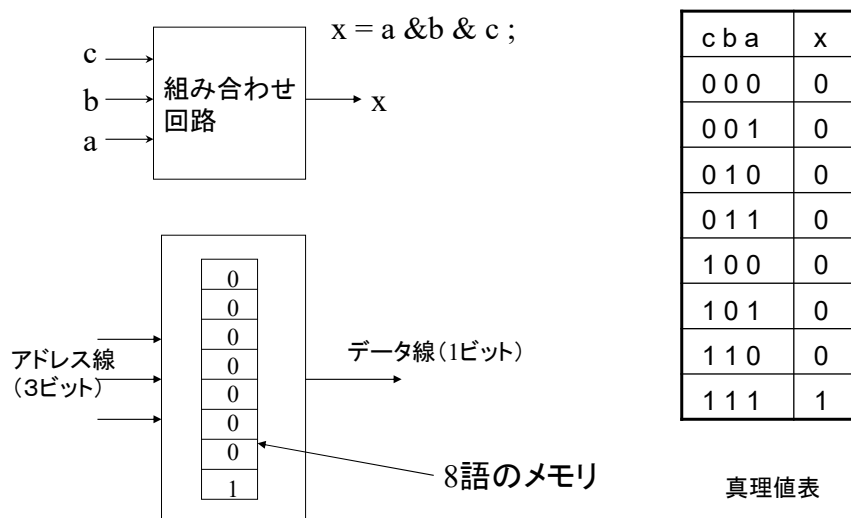
18

メモリによる組み合わせ回路の実現

- 入力の全ての組み合わせに対して、組み合わせ回路の出力値を求めておく(真理値表)
 - 入力の値 → アドレスの番地とする
 - 出力の値 → その入力に対するメモリの値

19

メモリによる組み合わせ回路の実現

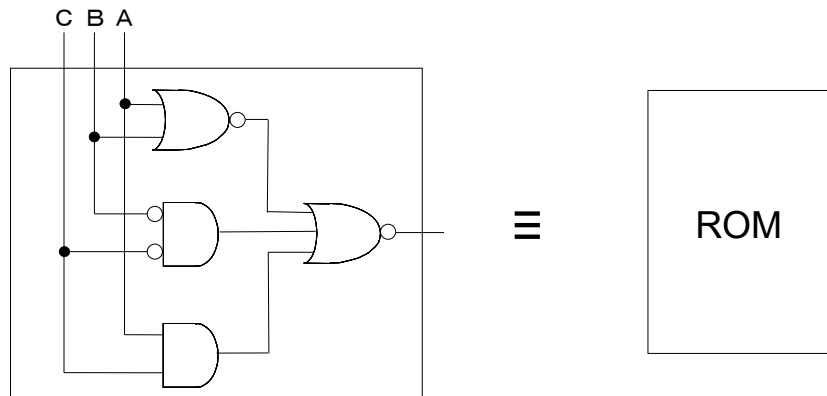


20

問題

次の回路と等価なROM(Read Only Memory)を考える

- (1) 何ビットのメモリが必要か？
- (2) アドレス線、データ線は何ビットか？
- (3) メモリの内容は？



21

順序回路のクリティカルパス

22

同期回路と非同期回路

- 同期回路
 - すべての回路が、同一のクロック信号に同期して動作する回路
- 非同期回路
 - 個々の素子やモジュールの動作が統一したクロックに依らずに、各が様々なタイミングで動作する回路

23

同期回路と非同期回路の比較

	同期回路	非同期回路
利点	<ul style="list-style-type: none">•動作の再現性がある•大規模な回路の設計に適する	<ul style="list-style-type: none">•タイミング設計が不要•低消費電力
欠点	<ul style="list-style-type: none">•クロック信号の幅(クロックの周波数)の決め方や、クロックの分配に最新の注意を払う必要がある(クリティカルパスによるタイミング設計)•消費電力が高くなる	<ul style="list-style-type: none">•ハザードなどにより動作の再現性に乏しい•大規模回路に適さない

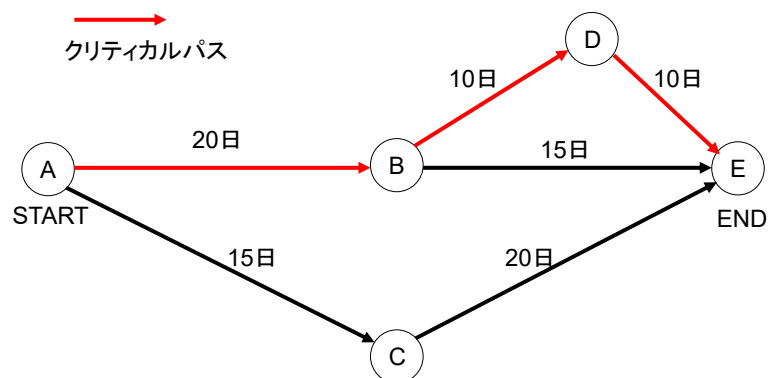
24

クリティカルパス(最長経路)

- 生産工程やプロジェクトなどで、お互いに従属関係(前工程が終わらないと次工程に進めない)にある複数の作業のうち、開始から終了までをつなぐ時間的余裕のない一連の作業の集まりを言う
- 論理回路では、最も長く信号の伝送遅延が大きい配線経路

25

クリティカルパスの例

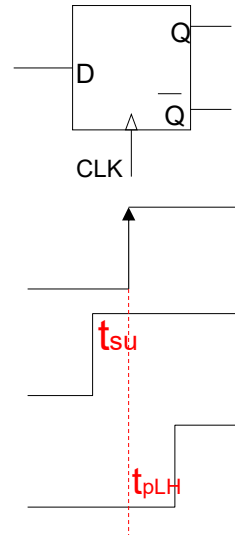


PERT (Program Evaluation and Review Technique) ネットワーク図

26

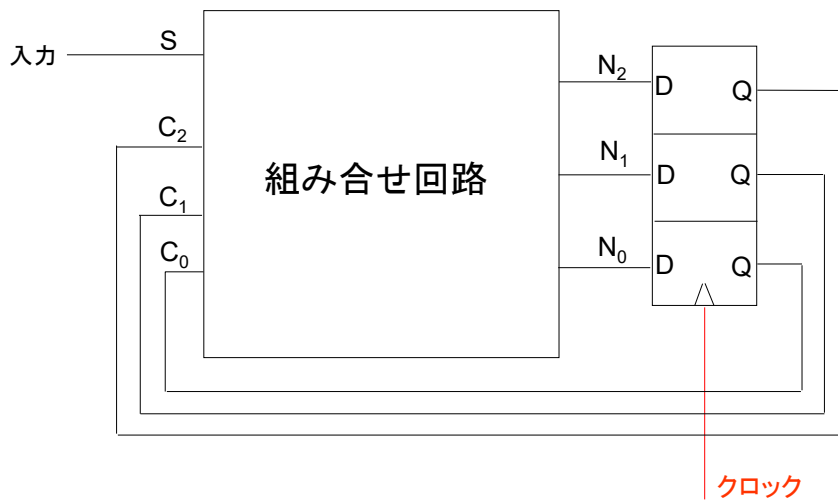
フリップフロップの動特性

- ゲートと同様の遅れ t_{pLH} t_{pHL}
 - CLK, (clear, preset) からの遅れ (FFの遅延)
- セットアップタイム t_{su}
 - クロックが立ち上がる前に入力が安定である時間
- ホールドタイム t_h
 - クロック変化後入力が安定である時間 (通常は0)



27

順序回路の構成例



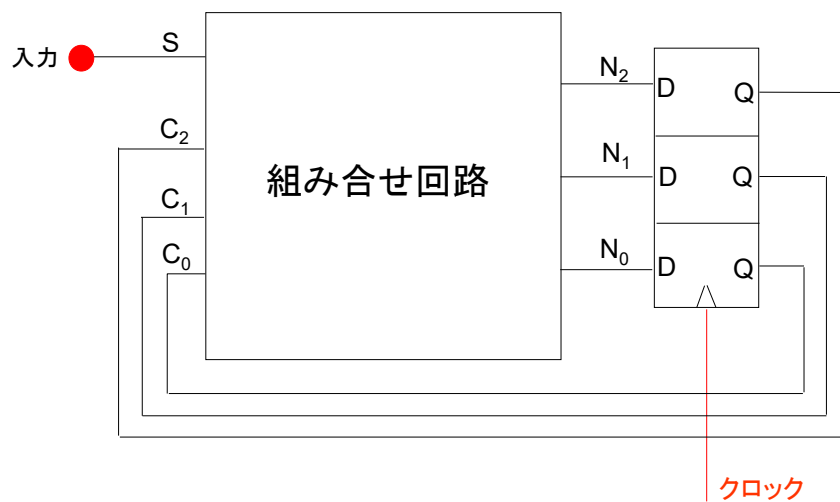
28

最大遅延とクリティカルパス

- ①入力からの遅延
- ②FF出力からの遅延

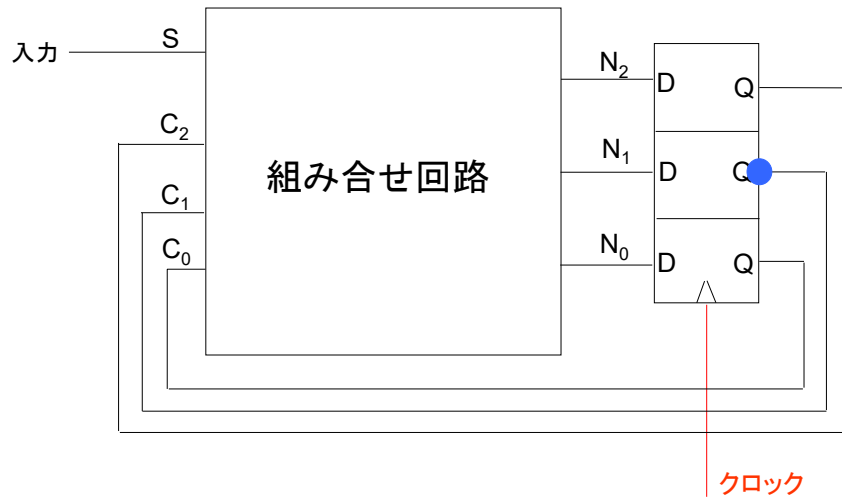
29

入力からの遅延



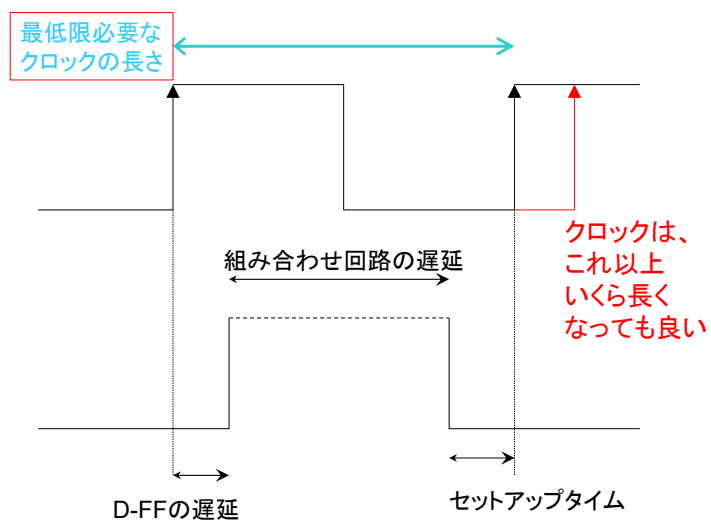
30

FF出力からの遅延



31

順序回路の遅延とタイミング設計



32

最大遅延とクリティカルパス

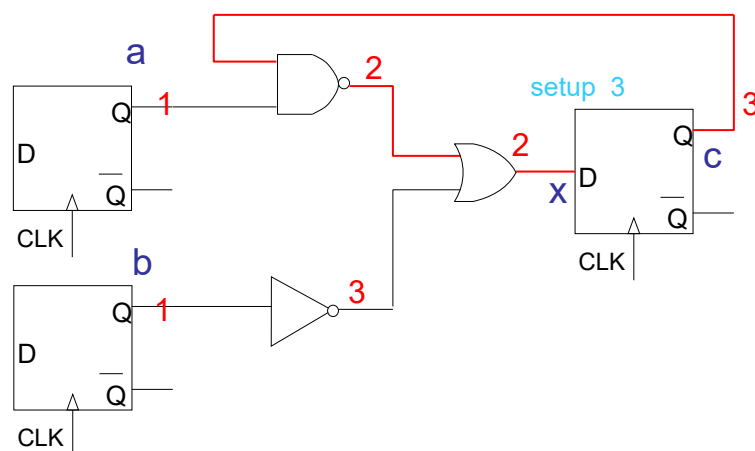
- ①入力からの遅延
(組み合わせ回路の遅延 A) + (D-FFのセットアップタイム)
- ②FF出力からの遅延
(D-FFの遅延) + (組み合わせ回路の遅延 A') + (D-FFのセットアップタイム)

最大遅延は、①と②の大きい方
そのときの、パスがクリティカルパス

⇒ 同期回路の最大周波数が決定される

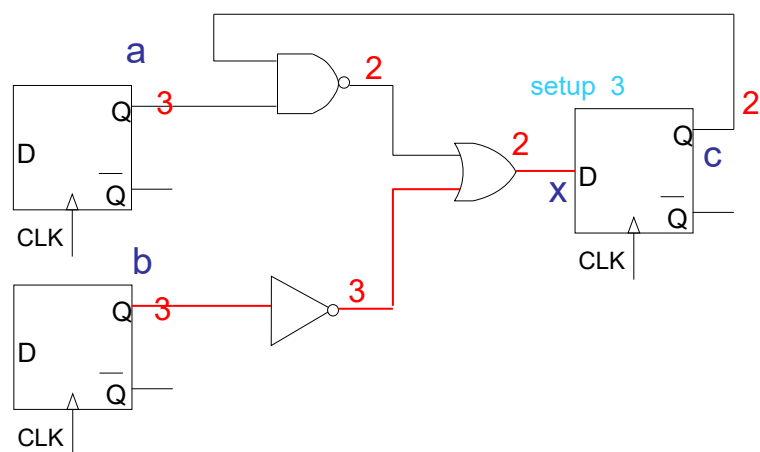
33

クリティカルパス 例1



34

クリティカルパス 例2



35