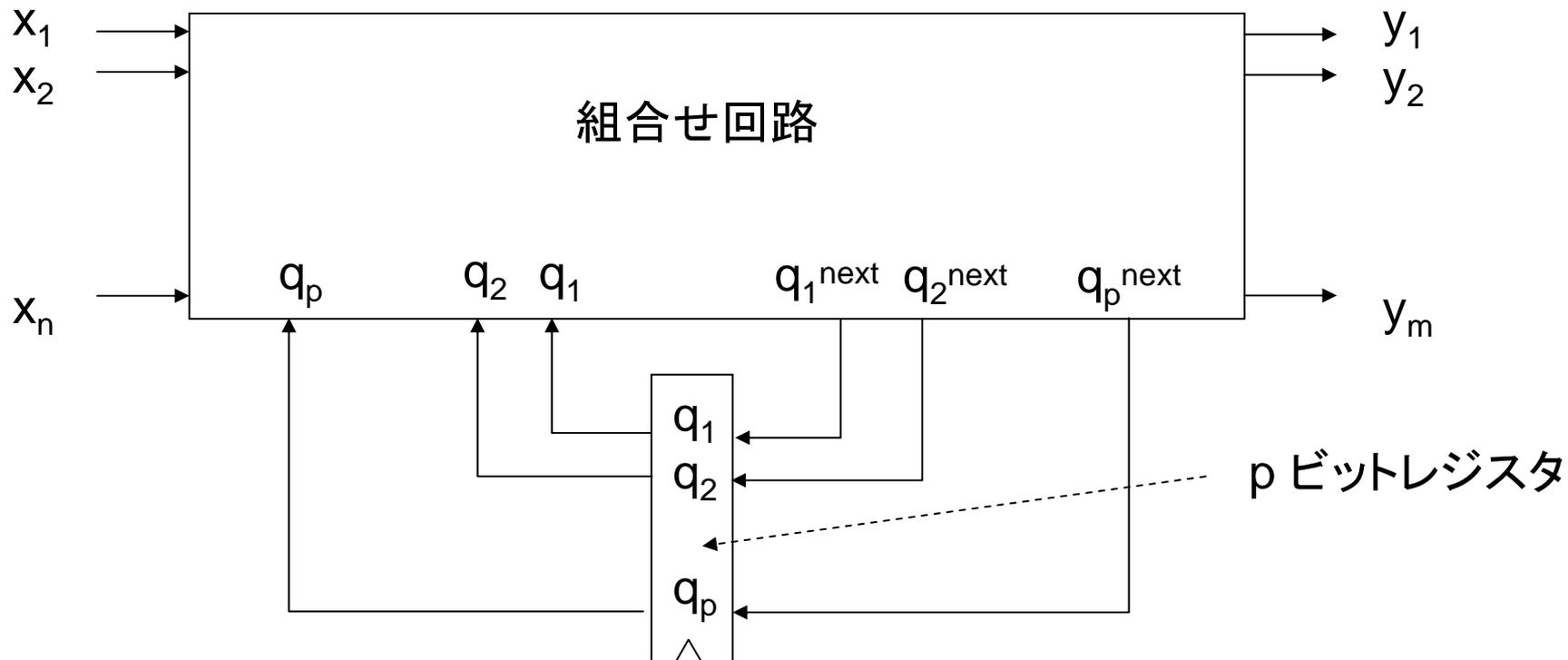


順序回路(2)

順序回路の設計

- 組合せ論理回路の設計法
 - 構造や規則性に着目した手設計(先人の知恵を使う)
 - 入力・出力の関係に基づく自動合成(カルノー図など)
- 順序回路の設計法
 - 構造や規則性に着目した手設計(前回の各例)
 - 入力・出力・状態の関係に基づく自動合成

同期式順序回路の入力・出力・状態の関係



$$q_i(t+1) = f_i(q_1(t), q_2(t), \dots, q_p(t), x_1(t), x_2(t), \dots, x_n(t))$$

$i = 1, 2, \dots, p$

f_i : 状態遷移関数

$$y_j(t) = g_j(q_1(t), q_2(t), \dots, q_p(t), x_1(t), x_2(t), \dots, x_n(t))$$

$j = 1, 2, \dots, m$

g_j : 出力関数

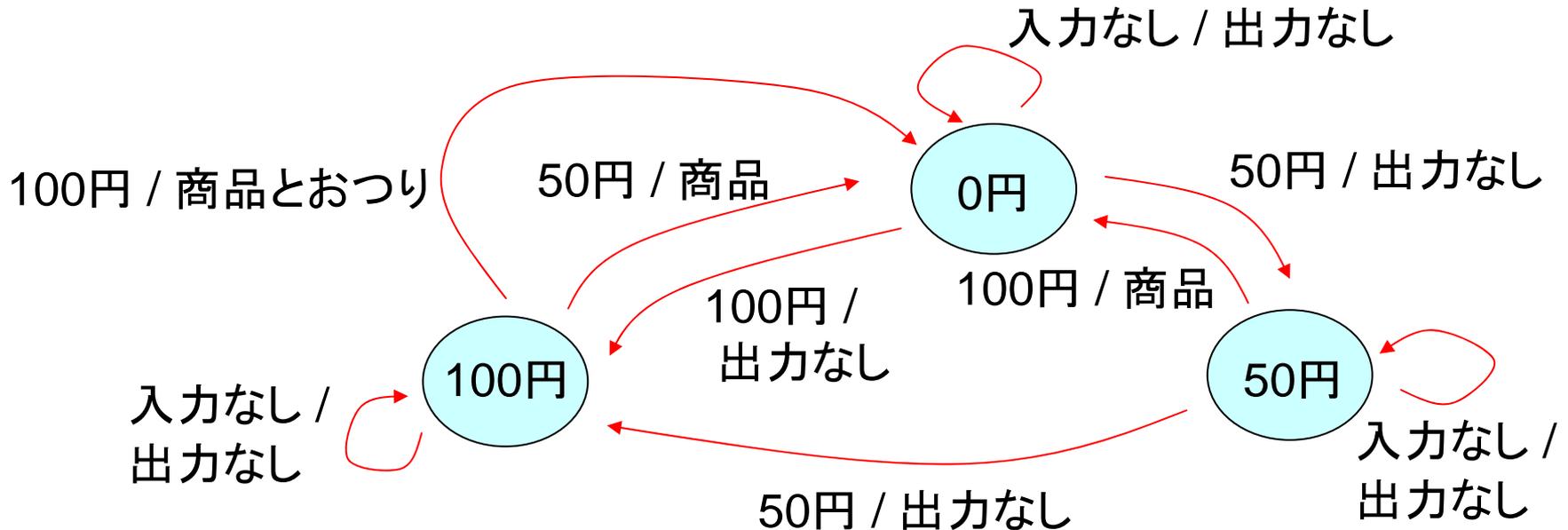
有限状態機械

- 取り得る状態の数が有限であるようなシステム(回路)
- すべての状態を列挙して, どんな入力のあるときにどの状態からどの状態に遷移するのか, そのとき何が出力されるのかを考え尽くすことができる
- 設計手順
 - 入力・出力・状態の関係を状態遷移図で表す
 - 入力・出力・状態に2進符号を割り当てる
 - 状態遷移表・出力表を作成する
 - 論理回路に置き換える(必要ならば簡単化する)

例1: レトロな自動販売機

50円, 100円の2種類の硬貨を受けつけ, 150円の商品1種類を販売する自動販売機を順序回路として設計したい. クロック周波数は十分に高く, 毎時刻1枚を超える硬貨が検出されることはないとする. 商品やおつりも十分に速く出すことができるとする. (そんな都合のよい機械があるかどうかはこの際考えない)

- 入力: { なし, 50円投入, 100円投入 }
- 出力: { なし, 商品排出, おつり50円排出, 商品とおつり50円排出 }
- 状態: { 累積金額0円, 累積金額50円, 累積金額100円 }



符号割り当て (例1)

- 入力: なし, 50円投入, 100円投入

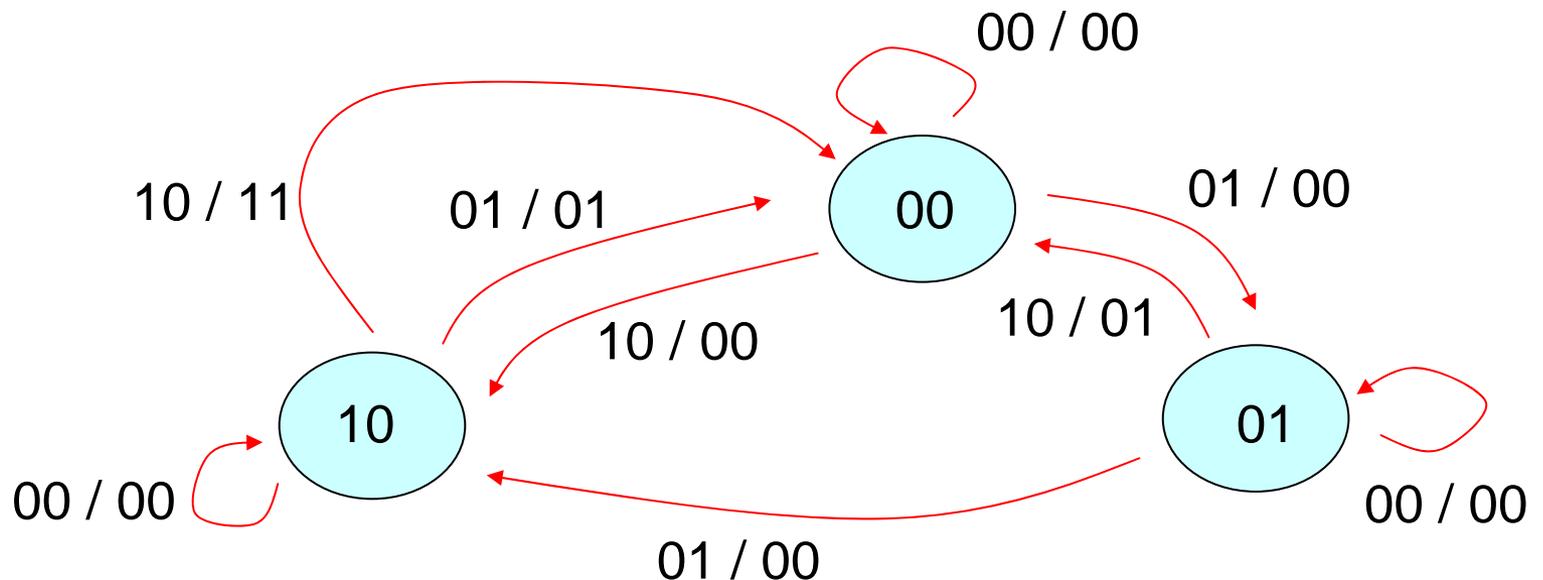
00 01 10

- 出力: なし, 商品排出, おつり50円排出, 商品とおつり50円排出

00 01 10 11

- 状態: 累積金額0円, 累積金額50円, 累積金額100円

00 01 10



状態遷移表・出力表 (例1)

入力×状態
→ 状態遷移先
の表を状態遷移表,

入力×状態
→ 出力
の表を出力表と呼ぶ.

両方合わせて状態
遷移表と呼ぶことも
多い.

q_1	q_0	x_1	x_0	q_1^{next}	q_0^{next}	y_1	y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0
0	0	1	1	*	*	*	*
0	1	0	0	0	1	0	0
0	1	0	1	1	0	0	0
0	1	1	0	0	0	0	1
0	1	1	1	*	*	*	*
1	0	0	0	1	0	0	0
1	0	0	1	0	0	0	1
1	0	1	0	0	0	1	1
1	0	1	1	*	*	*	*
1	1	0	0	*	*	*	*
1	1	0	1	*	*	*	*
1	1	1	0	*	*	*	*
1	1	1	1	*	*	*	*

カルノー図による簡単化 (例1)

q_1^{next}		$x_1 x_0$			
		00	01	11	10
$q_1 q_0$	00	0	0	*	1
	01	0	1	*	0
	11	*	*	*	*
	10	1	0	*	0

q_0^{next}		$x_1 x_0$			
		00	01	11	10
$q_1 q_0$	00	0	1	*	0
	01	1	0	*	0
	11	*	*	*	*
	10	0	0	*	0

y_1		$x_1 x_0$			
		00	01	11	10
$q_1 q_0$	00	0	0	*	0
	01	0	0	*	0
	11	*	*	*	*
	10	0	0	*	1

y_0		$x_1 x_0$			
		00	01	11	10
$q_1 q_0$	00	0	0	*	0
	01	0	0	*	1
	11	*	*	*	*
	10	0	1	*	1

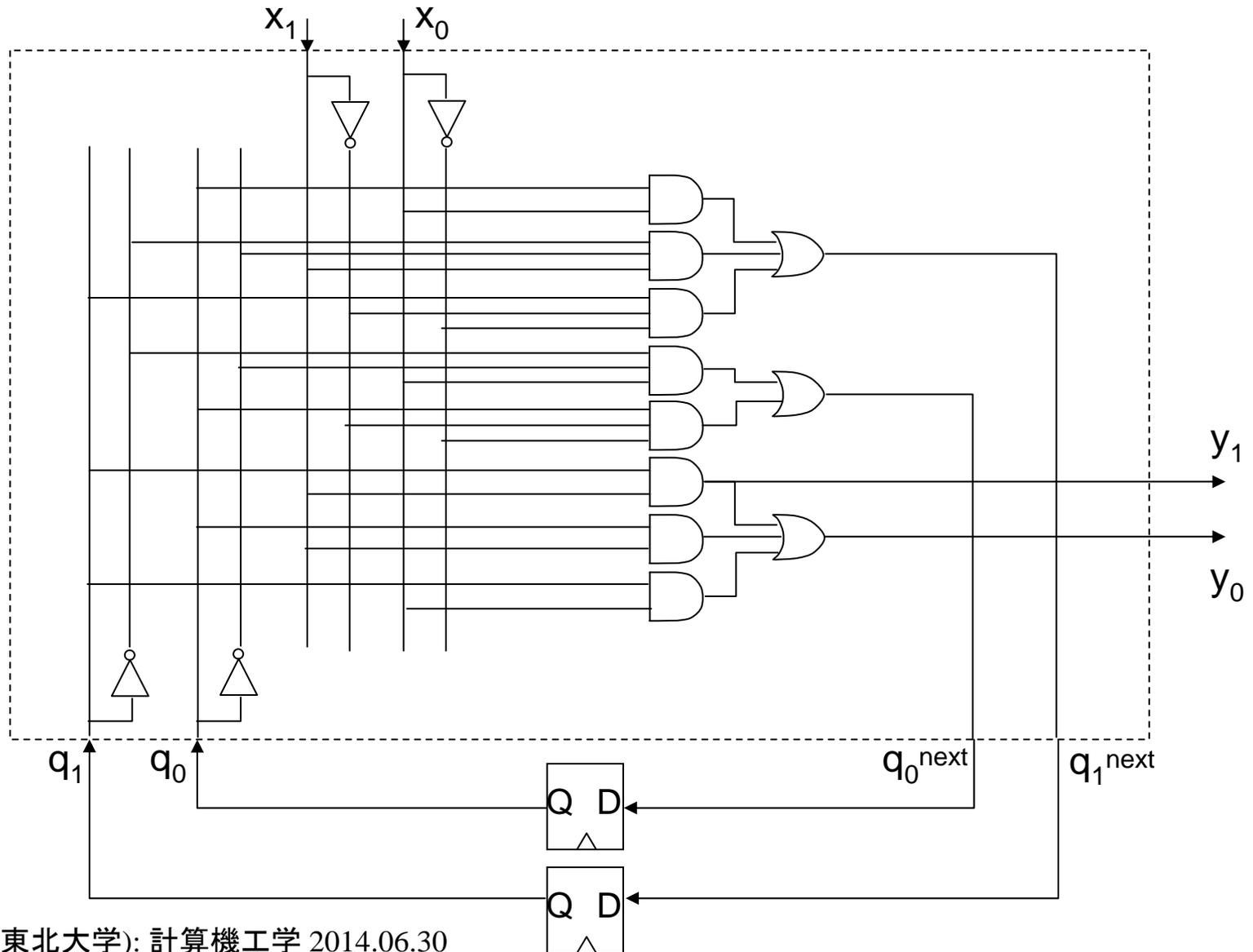
$$q_1^{next} = q_0 x_0 + \overline{q_1} \overline{q_0} x_1 + q_1 \overline{x_1} \overline{x_0}$$

$$q_0^{next} = \overline{q_1} \overline{q_0} x_0 + q_0 \overline{x_1} \overline{x_0}$$

$$y_1 = q_1 x_1$$

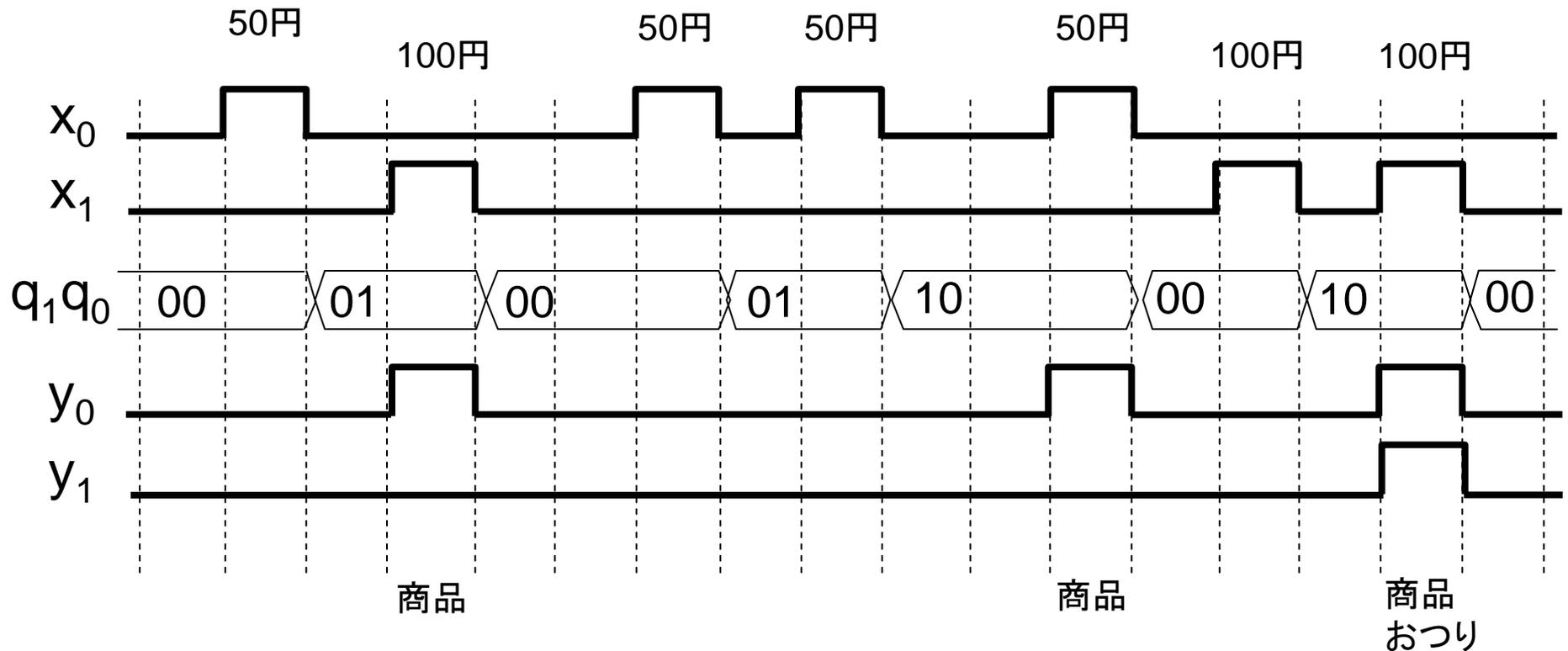
$$y_0 = q_1 x_1 + q_0 x_1 + q_1 x_0$$

完成図 (例1)

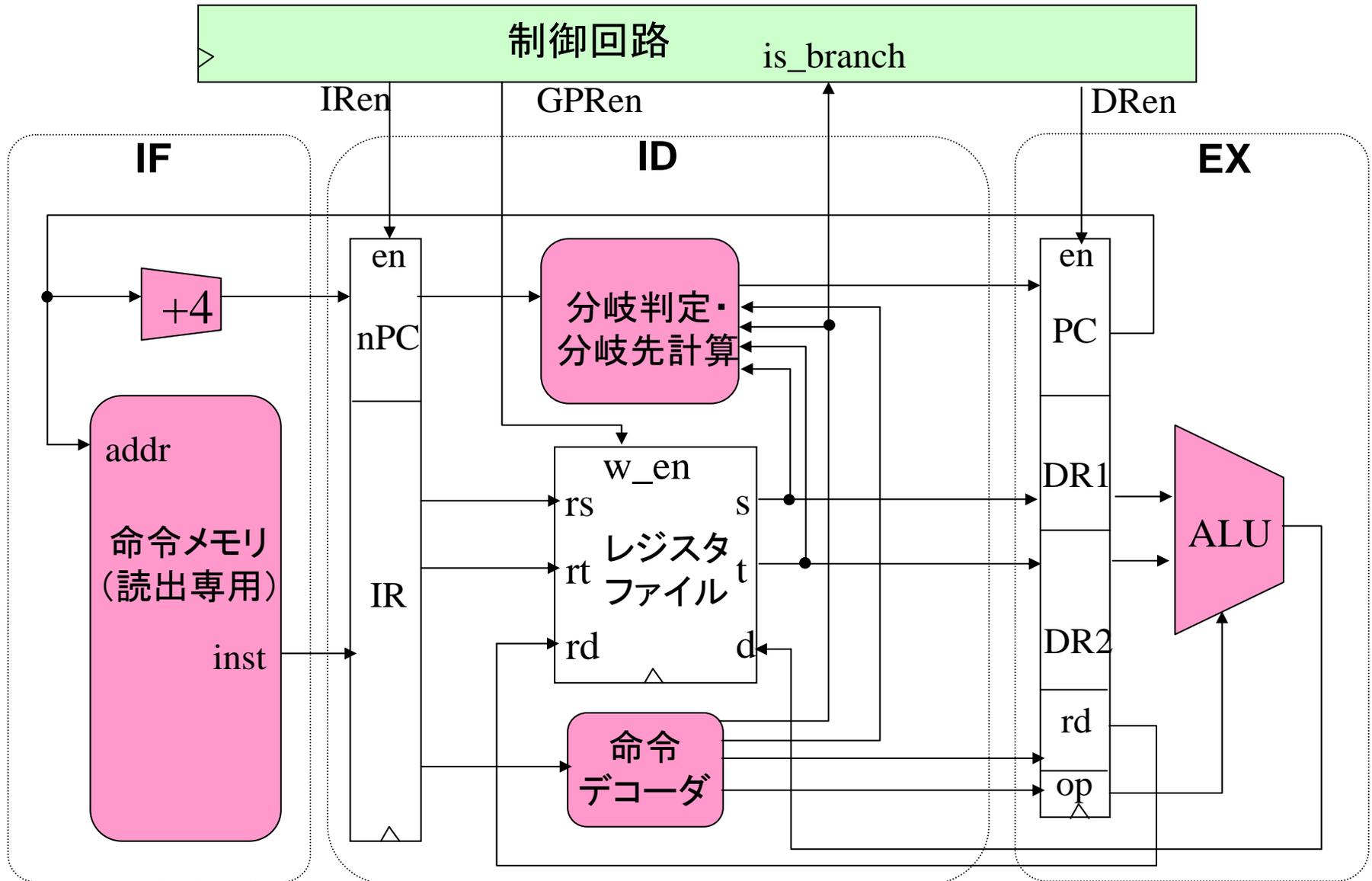


タイミングチャートの例 (例1)

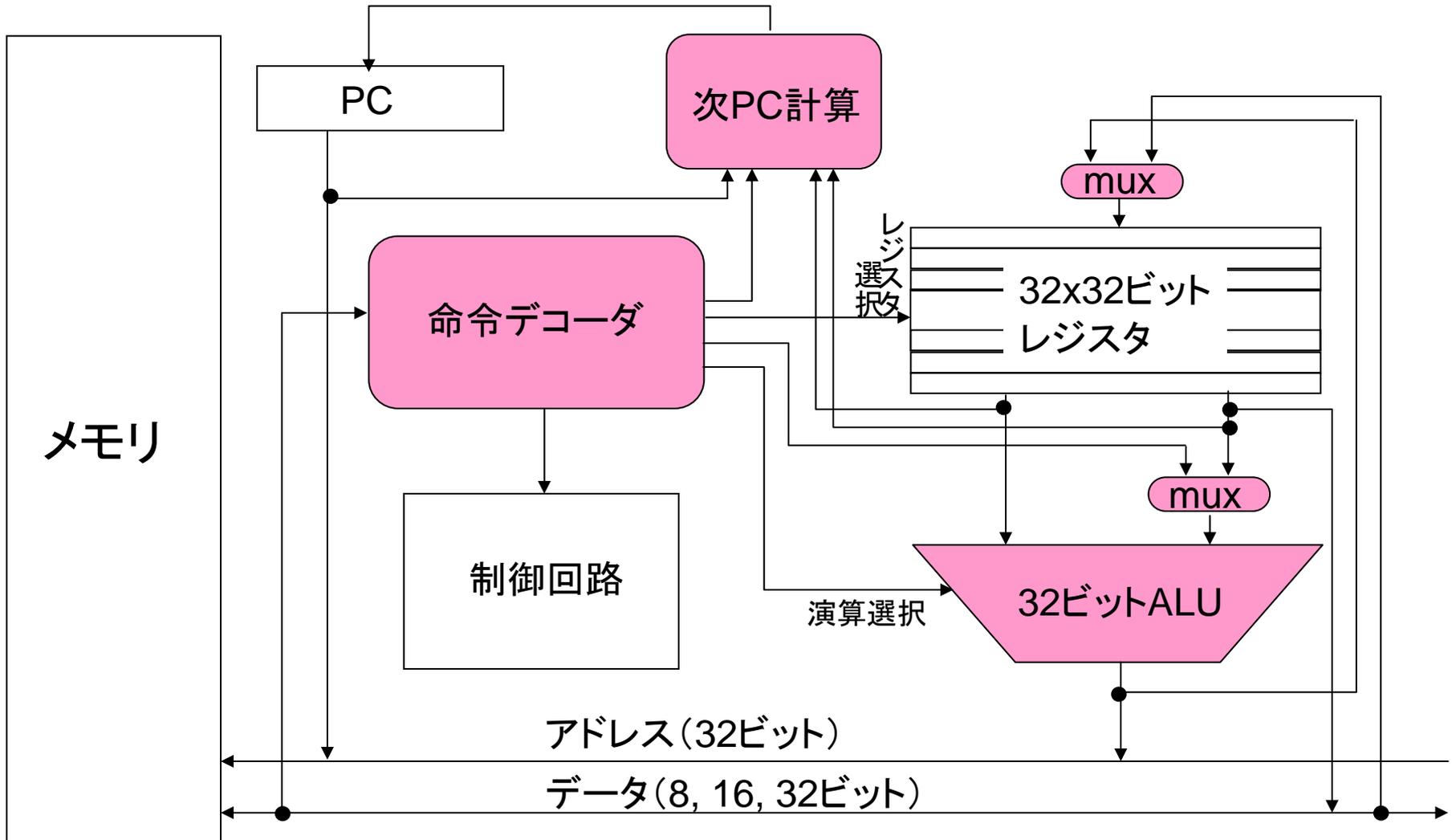
初期状態を 00 とする. クロック信号は省略



例2: 簡易版MIPSの制御回路



復習: MIPSの構造

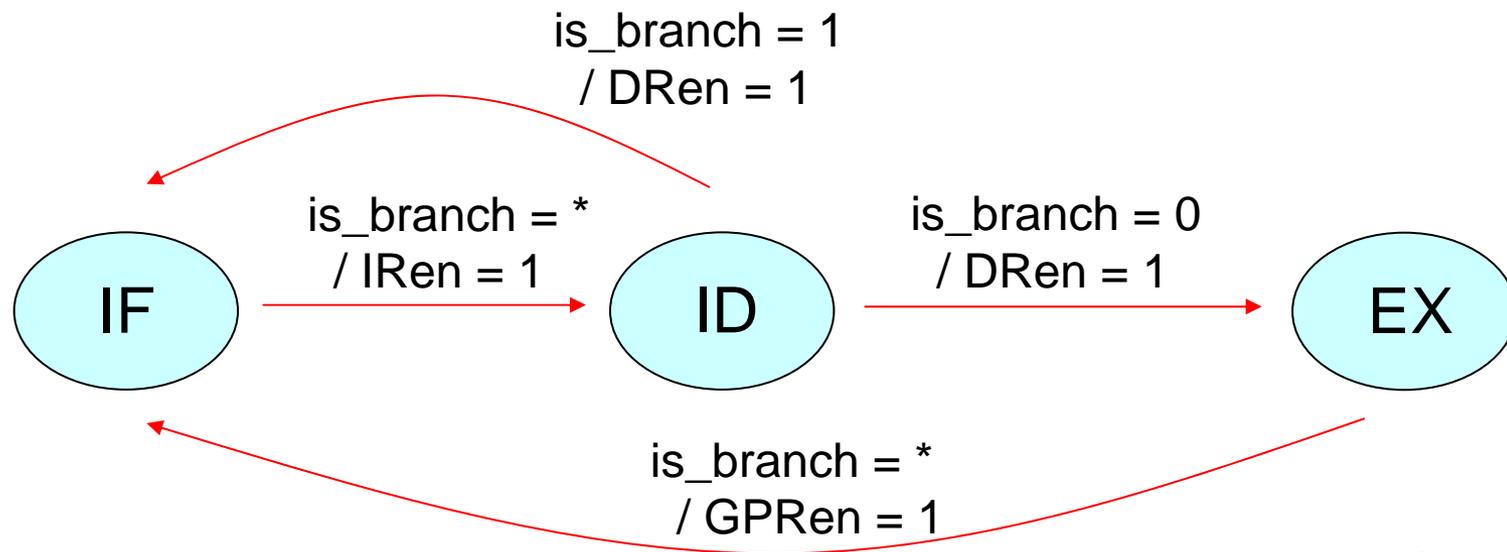


状態遷移図 (例2)

演算命令と分岐命令だけの簡略版MIPSの制御回路を設計したい。

- IF (Instruction Fetch): 命令読出し, 後続命令アドレス計算
- ID (Instruction Decode): 命令デコード, レジスタ読出し, 分岐判定
- EX (EXecution): 演算実行, レジスタ書き込み

の各ステージを適切な順序で動作させるために, イネーブル信号 IRen, DRen, GPRen を出力する. 命令デコードの結果生成される is_branch 信号 (分岐命令なら1, 演算命令なら0) を入力とする.



※記載のない出力は 0

状態遷移表・出力表 (例2)

符号割り当て:

IF: $q_1q_0 = 00$

ID: $q_1q_0 = 01$

EX: $q_1q_0 = 10$

q_1	q_0	is_branch	q_1^{next}	q_0^{next}	IRen	DRen	GPren
0	0	0	0	1	1	0	0
0	0	1	0	1	1	0	0
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	0
1	0	0	0	0	0	0	1
1	0	1	0	0	0	0	1
1	1	0	*	*	*	*	*
1	1	1	*	*	*	*	*

カルノー図による簡単化 (例2)

q_1^{next}		$q_1 q_0$			
		00	01	11	10
is_branch	0	0	1	*	0
	1	0	0	*	0

q_0^{next}		$q_1 q_0$			
		00	01	11	10
is_branch	0	1	0	*	0
	1	1	0	*	0

IRen		$q_1 q_0$			
		00	01	11	10
is_branch	0	1	0	*	0
	1	1	0	*	0

DRen		$q_1 q_0$			
		00	01	11	10
is_branch	0	0	1	*	0
	1	0	1	*	0

GPRen		$q_1 q_0$			
		00	01	11	10
is_branch	0	0	0	*	1
	1	0	0	*	1

$$q_1^{\text{next}} = q_0 \cdot \overline{\text{is_branch}}$$

$$q_0^{\text{next}} = \overline{q_1} \overline{q_0}$$

$$\text{IRen} = \overline{q_1} \overline{q_0}$$

$$\text{DRen} = q_0$$

$$\text{GPRen} = q_1$$

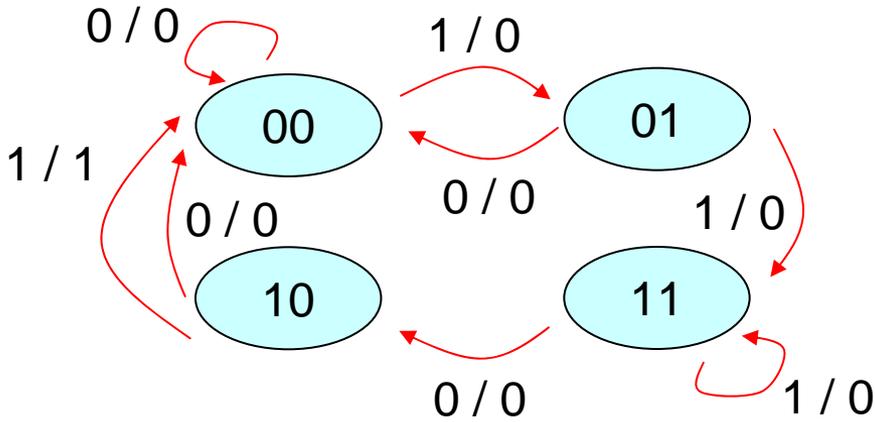
練習問題

入力信号系列 x の中に 1101 という並びが現れたら $y = 1$ を出力する順序回路を設計する. その状態遷移表, 出力表は右下の通りである.

1. 状態遷移図をかけ. かかれた図に基づいて, 状態 $(q_1, q_0) = (0, 0), (0, 1), (1, 0), (1, 1)$ がそれぞれ何を表しているか説明せよ.
2. カルノー図で表し, できるだけ簡単な順序回路を設計せよ.

q_1	q_0	x	q_1^{next}	q_0^{next}	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	1	0	0	1
1	1	0	1	0	0
1	1	1	1	1	0

解答例



q_1^{next}		$q_1 q_0$			
		00	01	11	10
x	0	0	0	1	0
	1	0	1	1	0

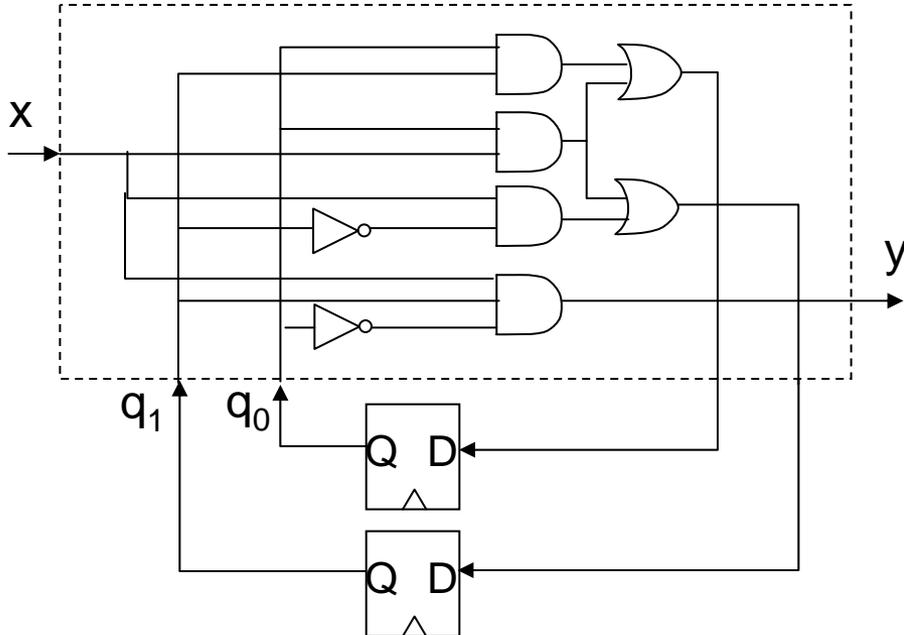
q_0^{next}		$q_1 q_0$			
		00	01	11	10
x	0	0	0	0	0
	1	1	1	1	0

y		$q_1 q_0$			
		00	01	11	10
x	0	0	0	0	0
	1	0	0	0	1

$$q_1^{next} = q_0x + q_1q_0$$

$$q_0^{next} = \overline{q_1}x + q_0x$$

$$y = q_1 \overline{q_0}x$$



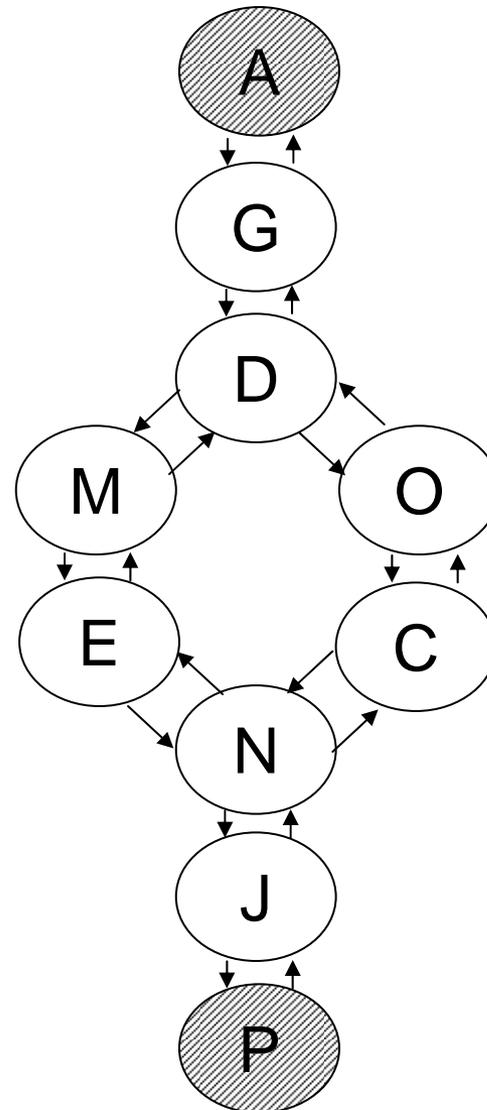
例題 (おまけ)

ある人が狼, 羊, 牧草とともに旅をしていたところ, 川にさしかかった. 小さな舟を漕いで渡るしかない. 舟には, 漕ぎ手である人のほか, 狼, 羊, 牧草のいずれか高々1つしか乗せるスペースがない. ただし, 人がいないと狼は羊を食べてしまい, また羊は牧草を食べてしまう. 人, 狼, 羊, 牧草すべて無事に川を渡るにはどうすればよいか.

- (1) こちらの岸と向こう岸に存在し得る人, 狼, 羊, 牧草の組合せを列挙せよ. これが有限状態機械の状態となる. (渡っている途中の状況は, 状態としてなくてよい. 理由を考えよ)
- (2) 状態遷移図を作成せよ. ただし入力, 出力を考える必要はない. 状態に2進符号を割り当てる必要もない. (ある状態からはどの状態に遷移し得るかのみを考慮する)
- (3) 状態遷移図をもとに, 最短で向こう岸に渡るための手順を示せ. 答えは一つとは限らない. (ヒント: 初期状態から順に, あるいは最終状態から逆にたどるとよい. あるいは状態遷移図を整理しながら考えてもよい)

例題(おまけ) 解答例

状態	こちら岸	向こう岸
A	人狼羊草	- ϕ
(B	狼羊草	- 人)
C	人羊草	- 狼
D	人狼草	- 羊
E	人狼羊	- 草
(F	羊草	- 人狼)
G	狼草	- 人羊
(H	狼羊	- 人草)
(I	人草	- 狼羊)
J	人羊	- 狼草
(K	人狼	- 羊草)
(L	人	- 狼羊草)
M	狼	- 人羊草
N	羊草	- 人狼草
O	草	- 人狼羊
P	ϕ	- 人狼羊草



最短経路は二つ

参考

- 今回学んだ Mealy 型の有限状態機械のほかに、出力が状態にだけ依存する(入力に直接依存しない) Moore 型の有限状態機械がある。
 - 注) 一昨年度までの講義では Moore 型を採用していた。
 - Moore 型も Mealy 型も表現能力は同じである。ただし、Moore 型は入力が出力に反映されるのに1時刻以上かかる。
 - 入力が出力に直接影響するのが好ましくない場合は Moore 型で設計する必要がある。
- 今回は、状態遷移図から安直に回路を生成した、実際は
 - より状態数の少ない有限状態機械で表せるかも知れない
 - 符号の割り当て方によって、回路がもっと簡単になるかも知れないなどといったことも考える必要がある。