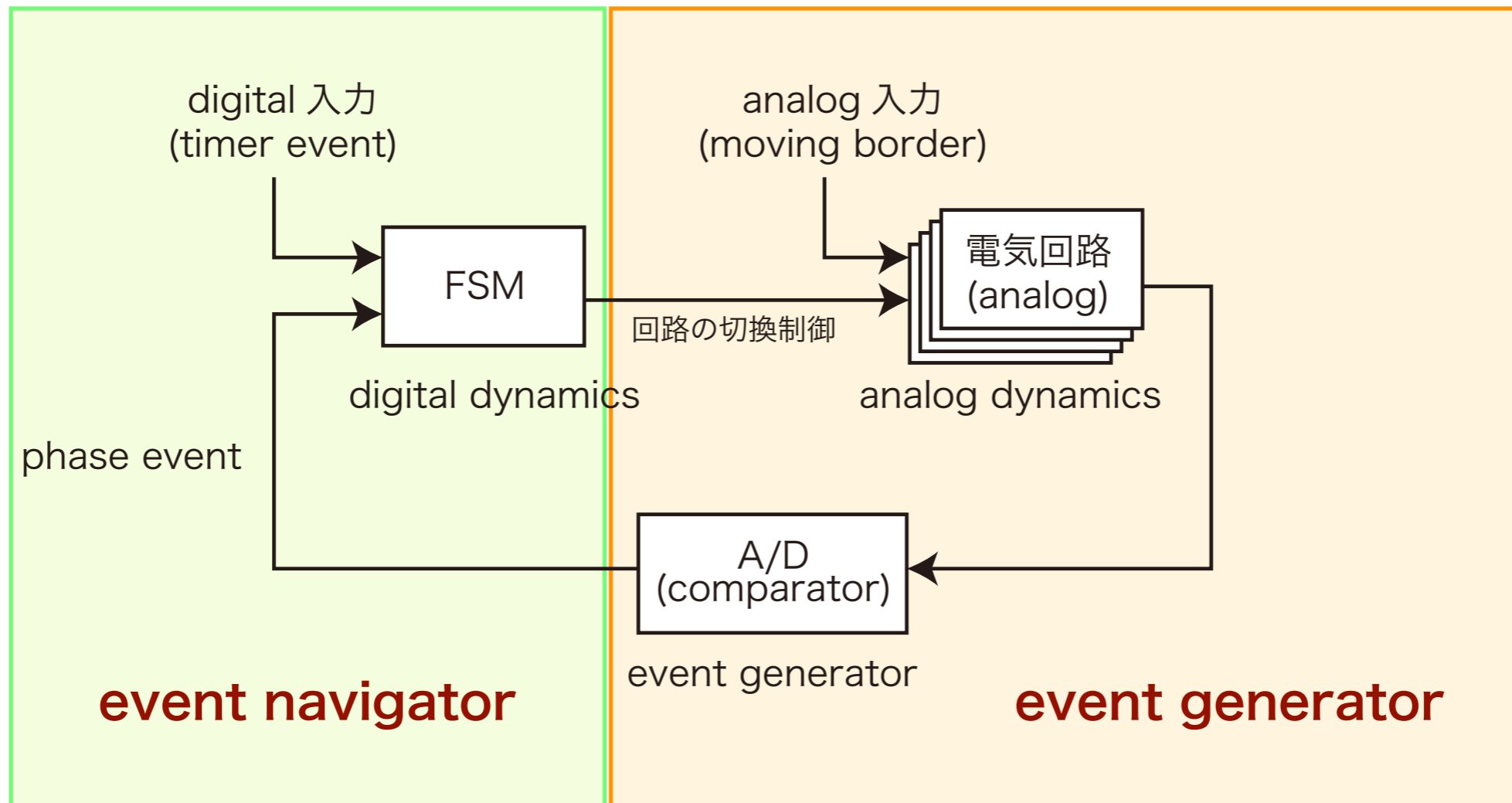


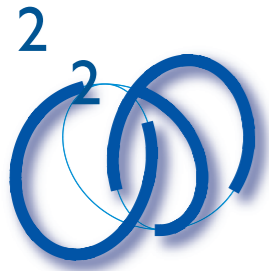


# Hybrid 回路の定性論 (3)



川上 博

2014(H26).02.03



# 話の流れ

---

## 1. Hybrid回路の例

- ◎ スイッチ, コンパレータ, 論理素子を含むRLC回路
- ◎ これまでに研究した回路との関連性

## 2. Hybrid回路の数学モデル

- ◎ FSMとODE混合系の定式化法

## 3. Hybrid回路の解析法

- ◎ 貼合わされた(FSM)相空間の流れ(ODE)に関する幾何学

## 4. Hybrid回路の解析プログラム

- ◎ FSMと力学系を実現するdesign patternをつくる



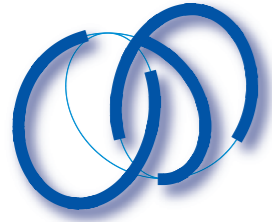
# 今日の内容

---

## 3. Hybrid回路の解析プログラム

目標：LEDホタルの結合系をシミュレートしたい  
ハイブリッド系を系統的にプログラムしたい

1. event drivenな手続き
2. mode transition diagramをプログラムにする
  - ◎ mode数とmode遷移数を低減するには？
3. State patternを使う
  - ◎ 基本となるhybrid oscillatorをobjectにする



**event drivenな手続きとする**



# event drivenな手続き

---

(analog状態の進展)

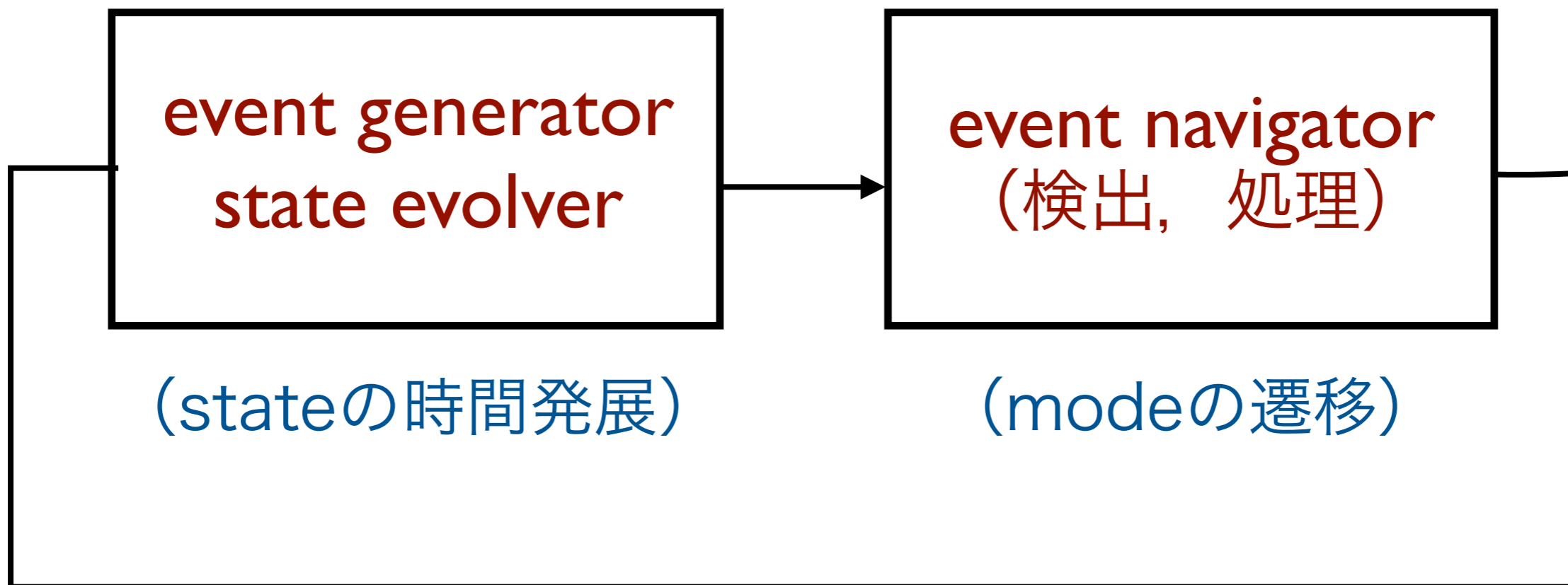
**event generator  
state evolver**

(stateの時間発展)

(digital状態の進展)

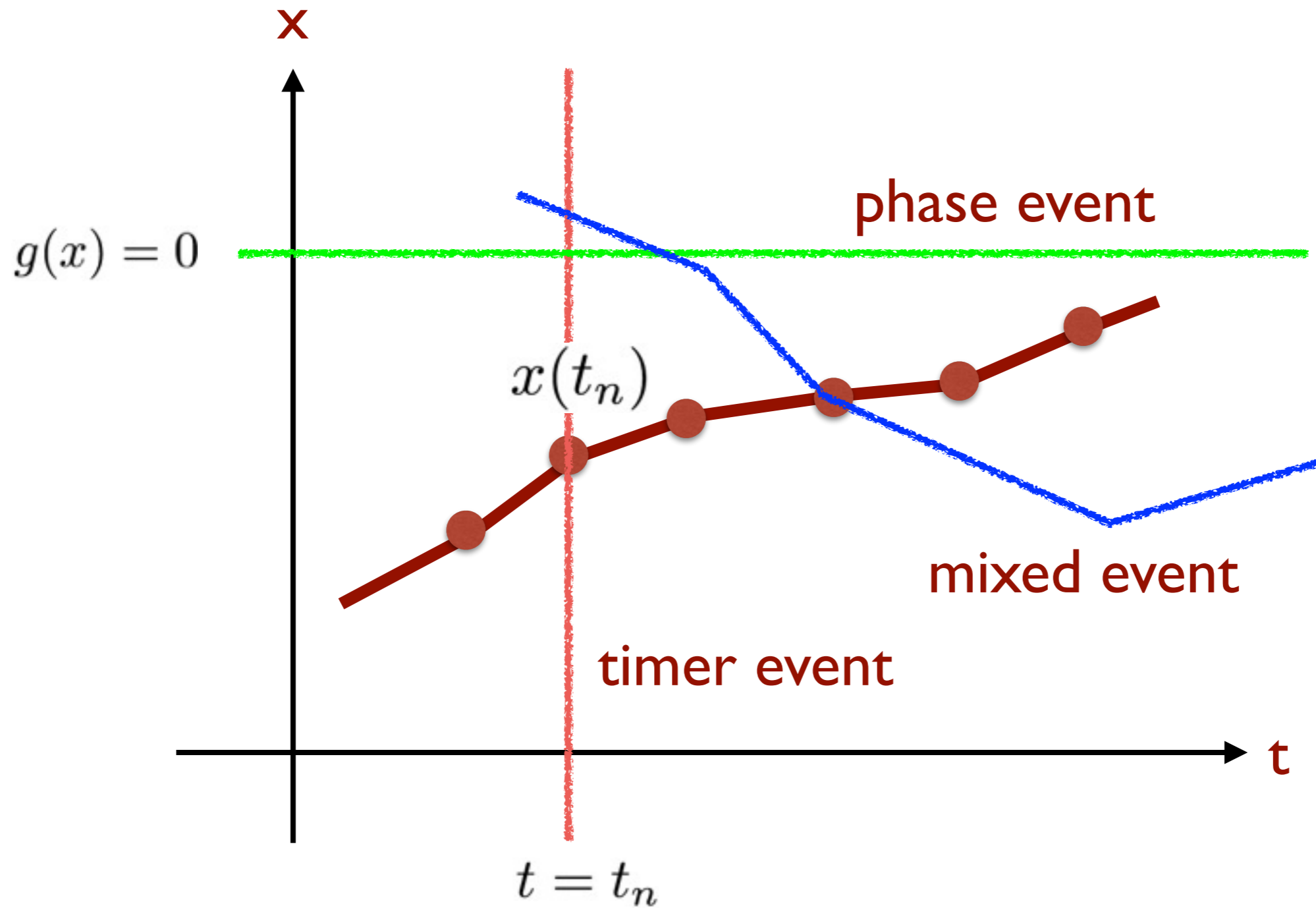
**event navigator  
(検出, 処理)**

(modeの遷移)





# eventの種類





# timer event

---

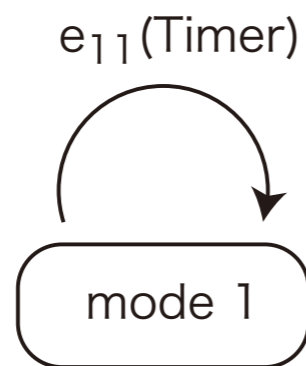
**eventの起こる時刻**：digital modeやanalog stateと独立して  
特定の時刻に発生する

**処理内容**：timer event枝のみからなるtransition diagramを  
つくって，各枝に対応した手続きを書く

**mode数を減らす工夫**：時間的に隣り合わせたmodeを合体  
できないか等の検討をする



# model

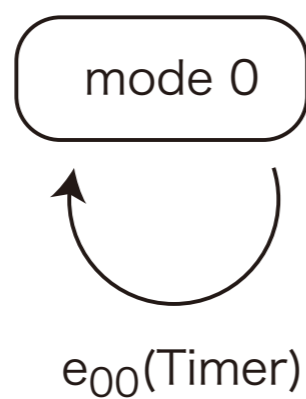


$e_{01} : A_{11} \dashrightarrow D_{01}$

$e_{11} : A_{10} \dashrightarrow D_{10}$

$e_{01}(\text{Phase})$

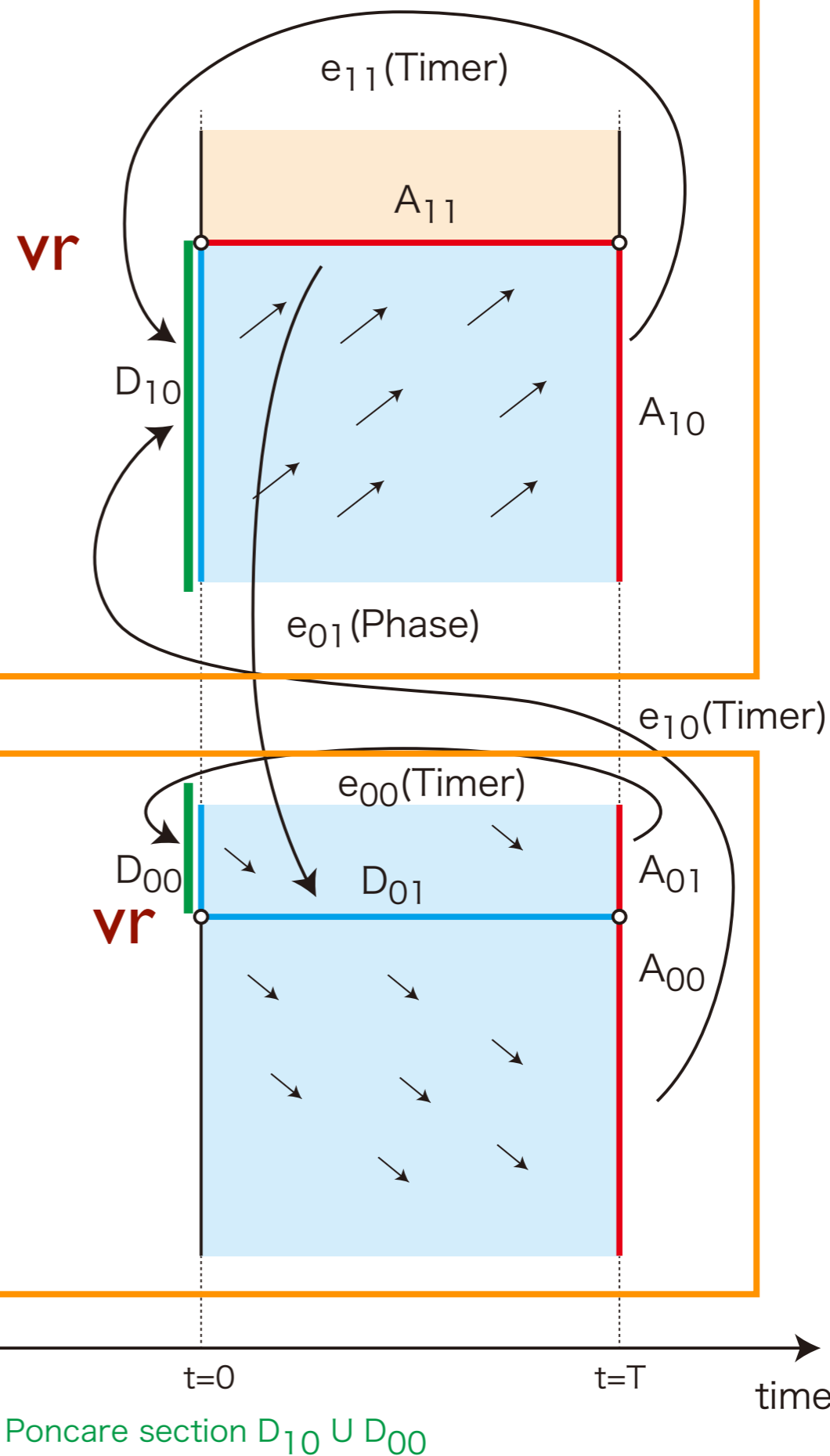
$e_{10}(\text{Timer})$



$e_{00} : A_{01} \dashrightarrow D_{00}$

$e_{10} : A_{00} \dashrightarrow D_{10}$

# mode0





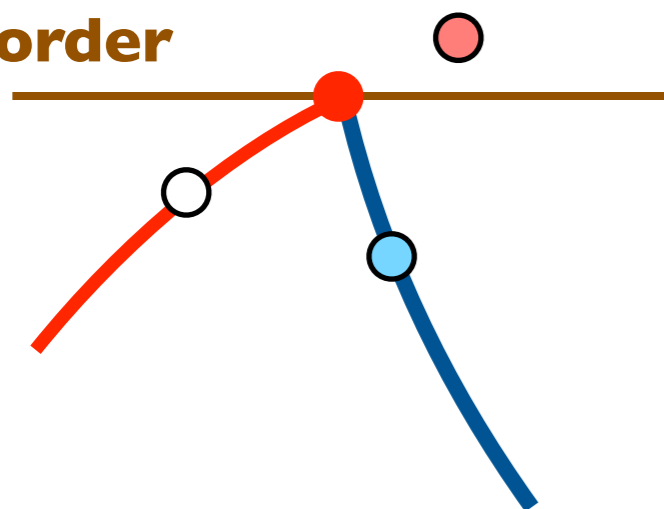


```

for (count=0; count<NN; count++) {
  tt=h*(double)count;
  if (count==0){ //timer event
    mode=1;
  }
  // one step evolution:start
  currentPts(a, y);
  runge(1,h,x,0.0, mode);
  if (x[0]>alpha0ff){ // phase event
    h1=(alpha0ff-y[1])/(-y[1]+1.0);
    y[1]=alpha0ff;
    nextPts(y, u, h1);
    drawTrajectory(a, u);
    x[0]=alpha0ff;
    currentPts(a, y);
    mode=0; // mode transition
    h2=h-h1;
    runge(1, h2, x, 0.0, mode);
    nextPts(y, u, h2);
    drawTrajectory(a, u);
  }
  nextPts(y, u, h);
  drawTrajectory(a, u);
  // one step evolution:end
}

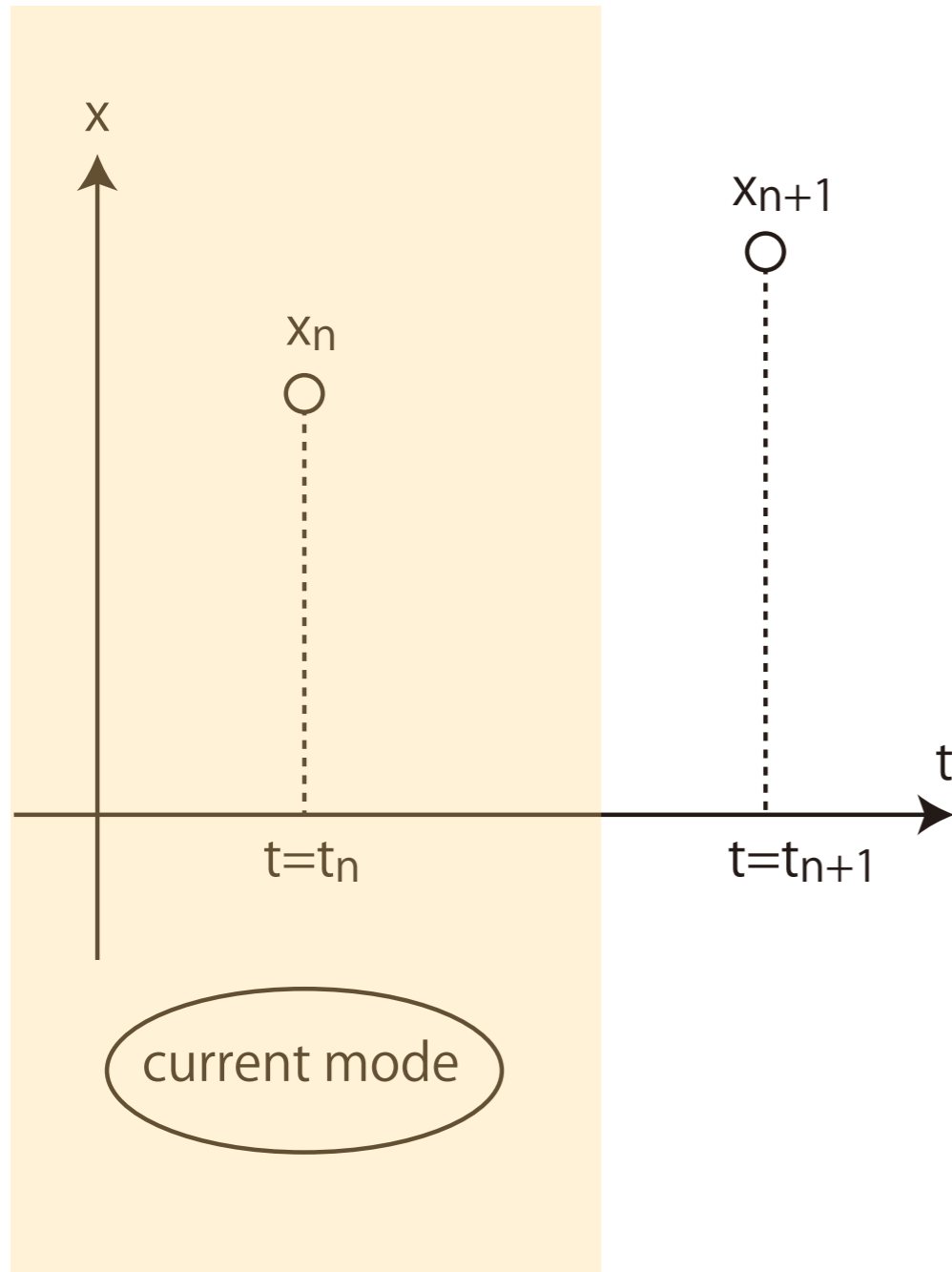
```

border

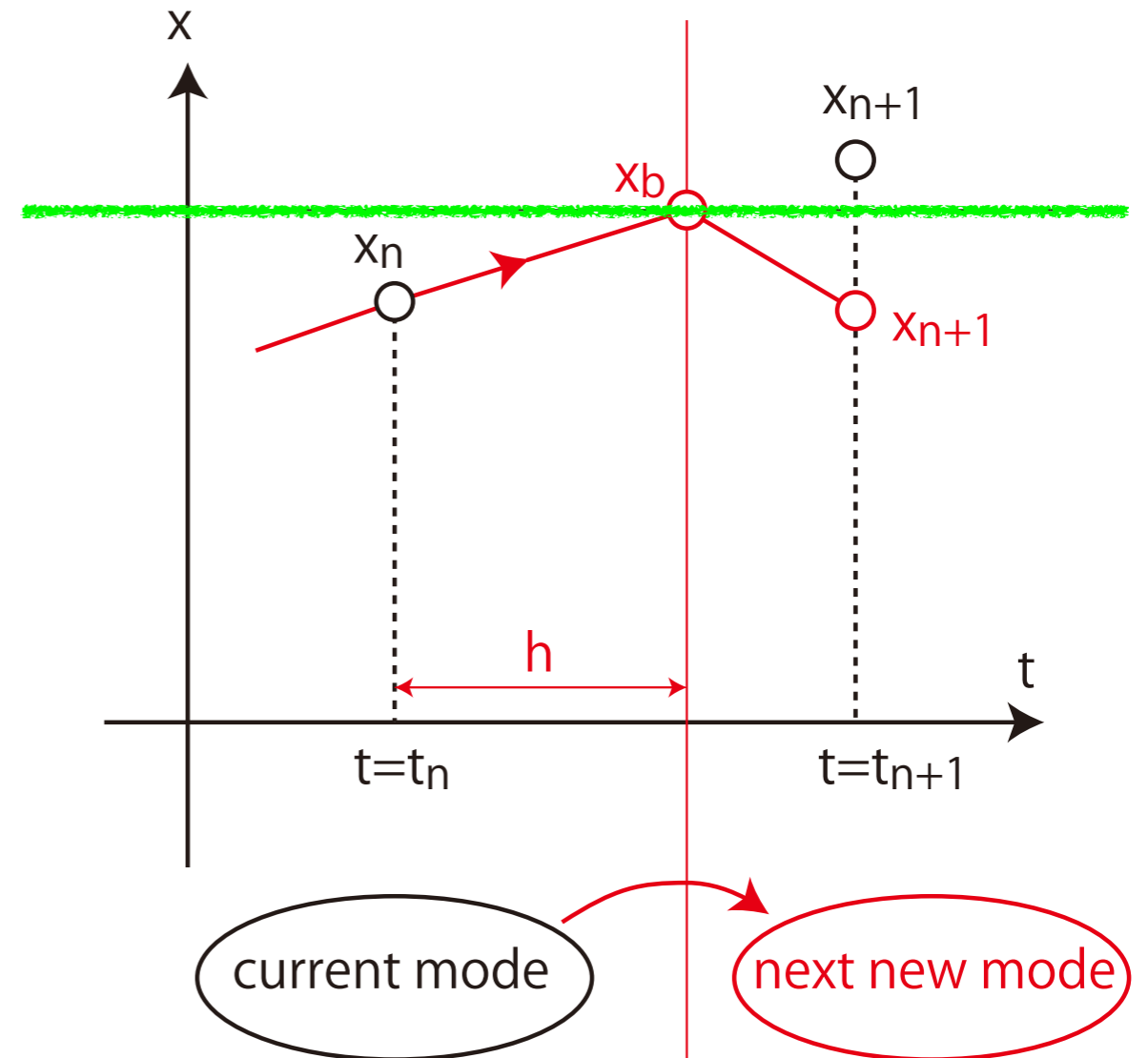




# phase event



phase eventが起こっているか

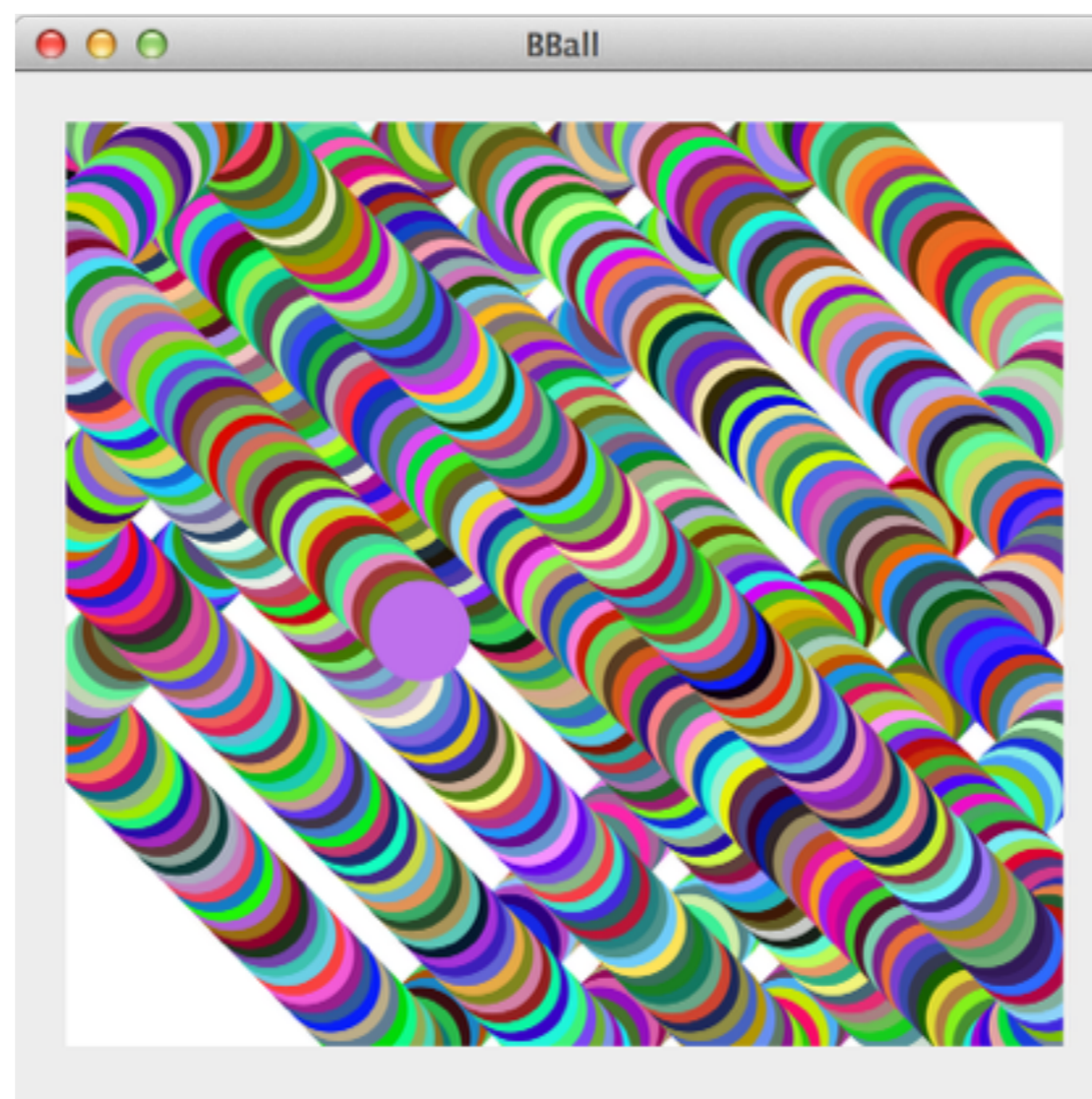


起こった場合の処理

11

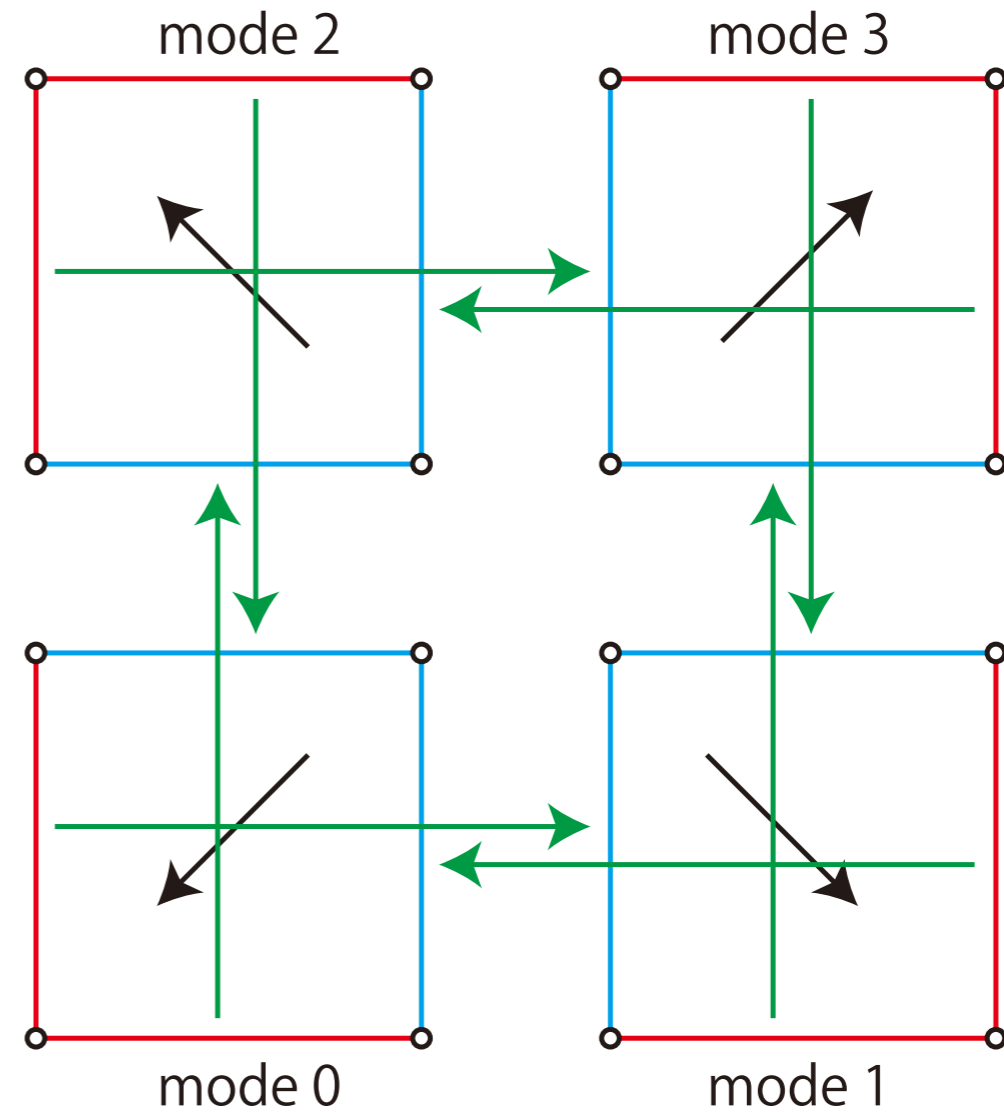
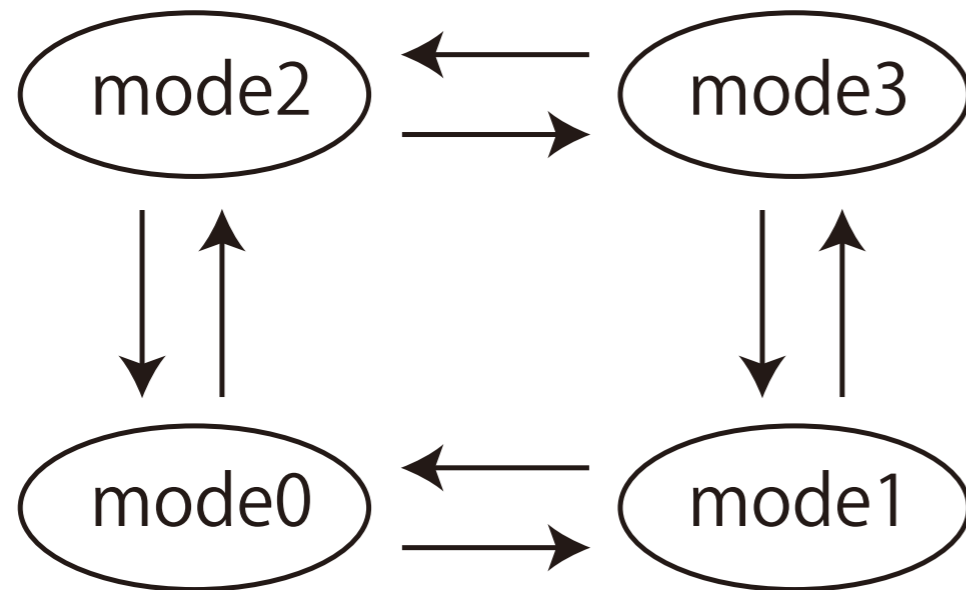


## 簡単な例：机上の球の運動





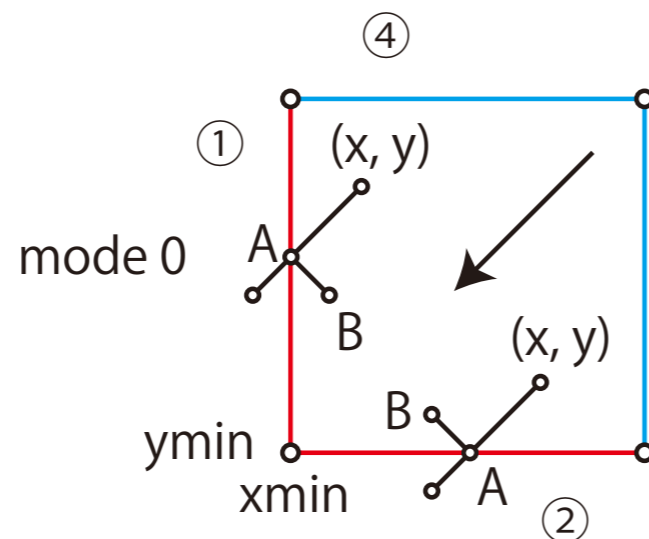
# Uncoupled SW Oscillator

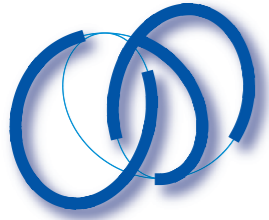




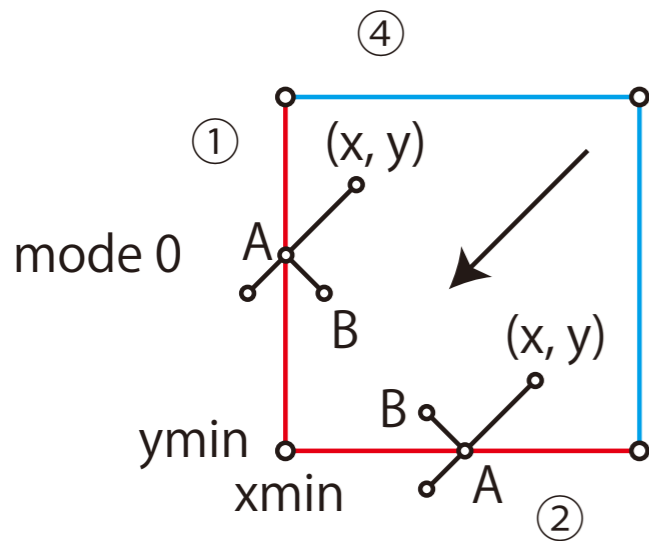
# 1. event-mode matrix

transition	mode0	mode1	mode2	mode3
$x > x_{max}$		mode0		mode2
$x < x_{min}$	mode1		mode3	
$y > y_{max}$			mode0	mode1
$y < y_{min}$	mode2	mode3		





# 1. mode-event matrix



```

switch (mod) {
  case 0:
    if (x[0] < xmin) {
      dx = -dx;
      mod = 1;
    }
    if (x[1] < ymin) {
      dy = -dy;
      mod = 2;
    }
    break;
  case 1:
    if (x[0] > xmax) {
      dx = -dx;
      mod = 0;
    }
    if (x[1] < ymin) {
      dy = -dy;
      mod = 3;
    }
    break;
  case 2:
    if (x[0] < xmin) {
      dx = -dx;
      mod = 3;
    }
    if (x[1] > ymax) {
      dy = -dy;
      mod = 0;
    }
    break;
  case 3:
    if (x[0] > xmax) {
      dx = -dx;
      mod = 2;
    }
    if (x[1] > ymax) {
      dy = -dy;
      mod = 1;
    }
    break;
  default:
    break;
}

```



## 2. primitive method

```
if(x[0]>xmax){dx=-dx;} //collision criteria
if(x[0]<xmin){dx=-dx;}
if(x[1]>ymax){dy=-dy;}
if(x[1]<ymin){dy=-dy;}

x1[0]=x[0]+dx; // ball dynamics on x-axis
x1[1]=x[1]+dy; // ball dynamics on y-axis
```

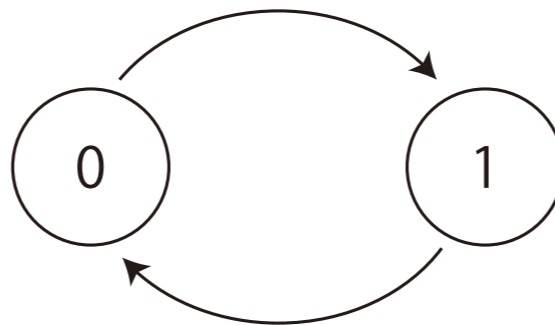
4つのphase eventは独立に生起する



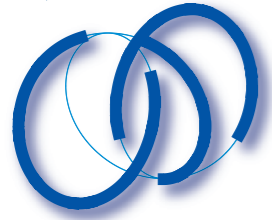
### 3. mode variable method

```
for (int i=0; i<2; i++) {  
    if (mod[i]==1 && x[i]>xmax[i]) {  
        dx[i]=-dx[i];  
        mod[i]=0;  
    }else if(mod[i]==0 && x[i]<xmin[i]){  
        dx[i]=-dx[i];  
        mod[i]=1;  
    }  
}  
  
x1[0]=x[0]+dx[0]; // ball dynamics on x-axis  
x1[1]=x[1]+dx[1]; // ball dynamics on y-axis
```

4つのmodeを2つの2値論理変数mod[0],mod[1]で表した





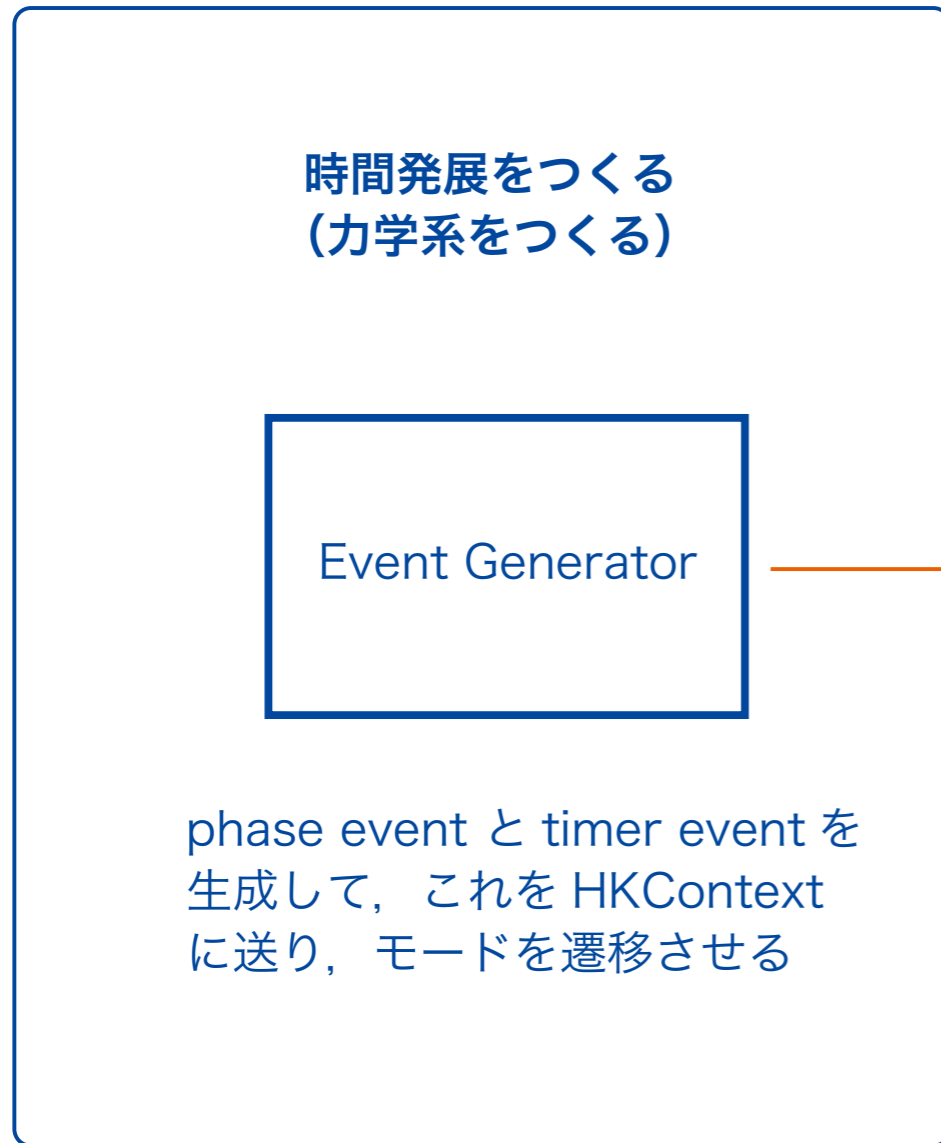


## 4. State patternを使う

```
next=[mode updateX:x];  
if (mode != next) {  
    // actions before transition  
    mode=next;  
    // actions after transition  
}  
  
x1[0]=x[0]+[mode dx]; // ball dynamics on x-axis  
x1[1]=x[1]+[mode dy]; // ball dynamics on y-axis
```



# Hybrid 系解析用 State Pattern

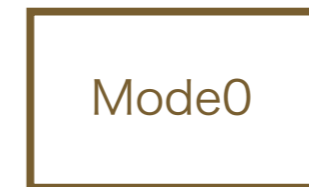
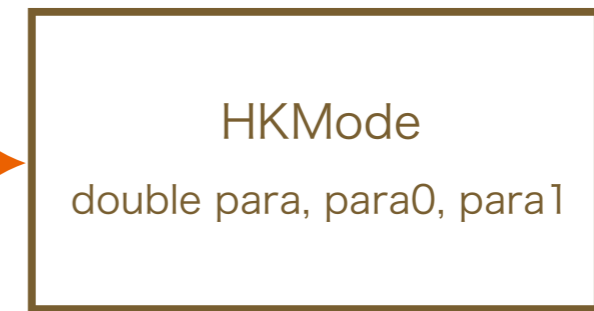


イベントを受け取って mode に受け渡す. モードの遷移はサブモードで行う.

パラメータ :

- border の切り換え
- 方程式の切り換え

パラメータの定義はすべて親の HKMode のインスタンス変数として定義しておく.



パラメータの設定はそれぞれのサブモードですべてのパラメータを設定し直す

そうするとモードが遷移するとパラメータも自動的に切り替わる



# Optically Controlled Oscillator

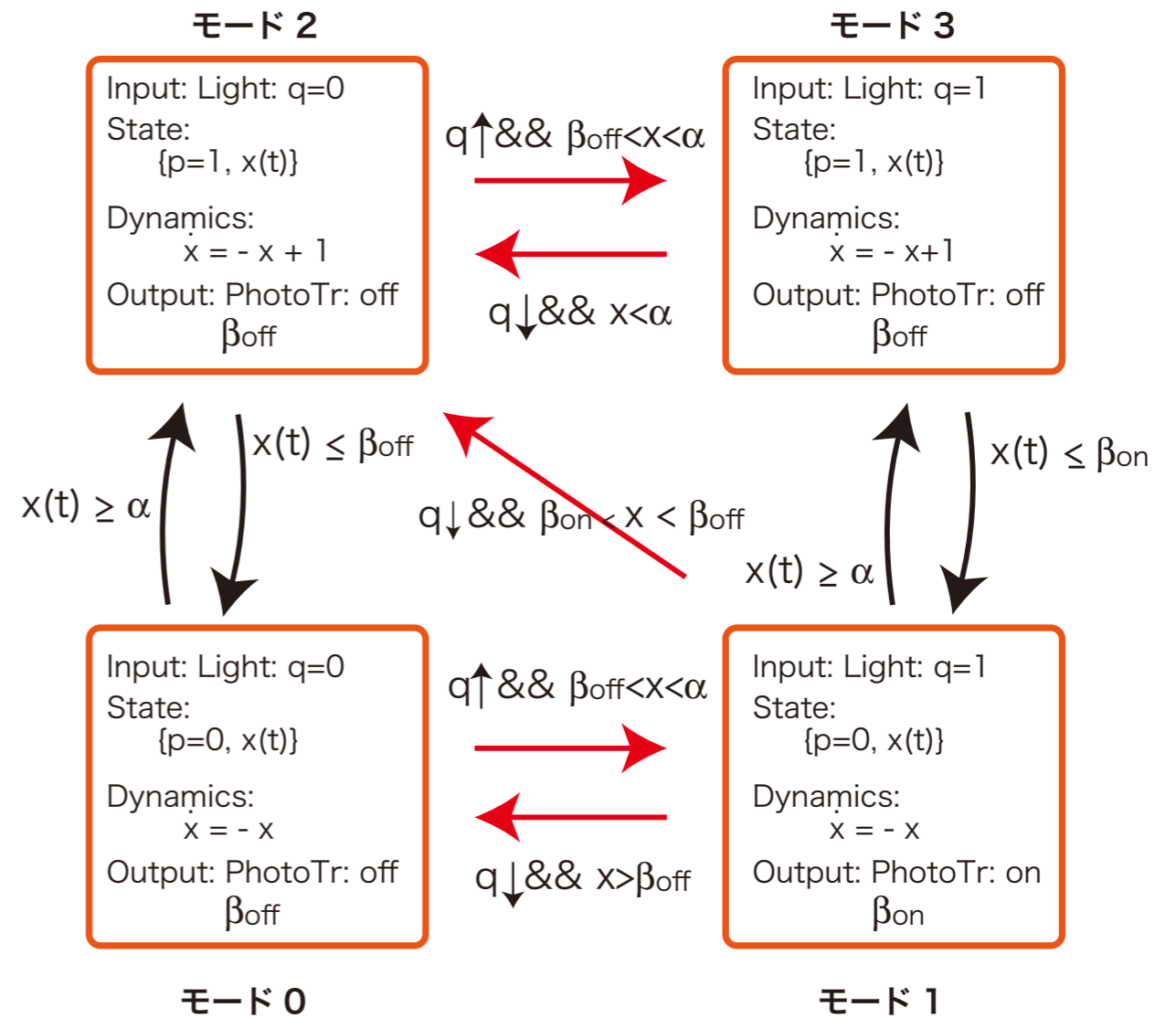
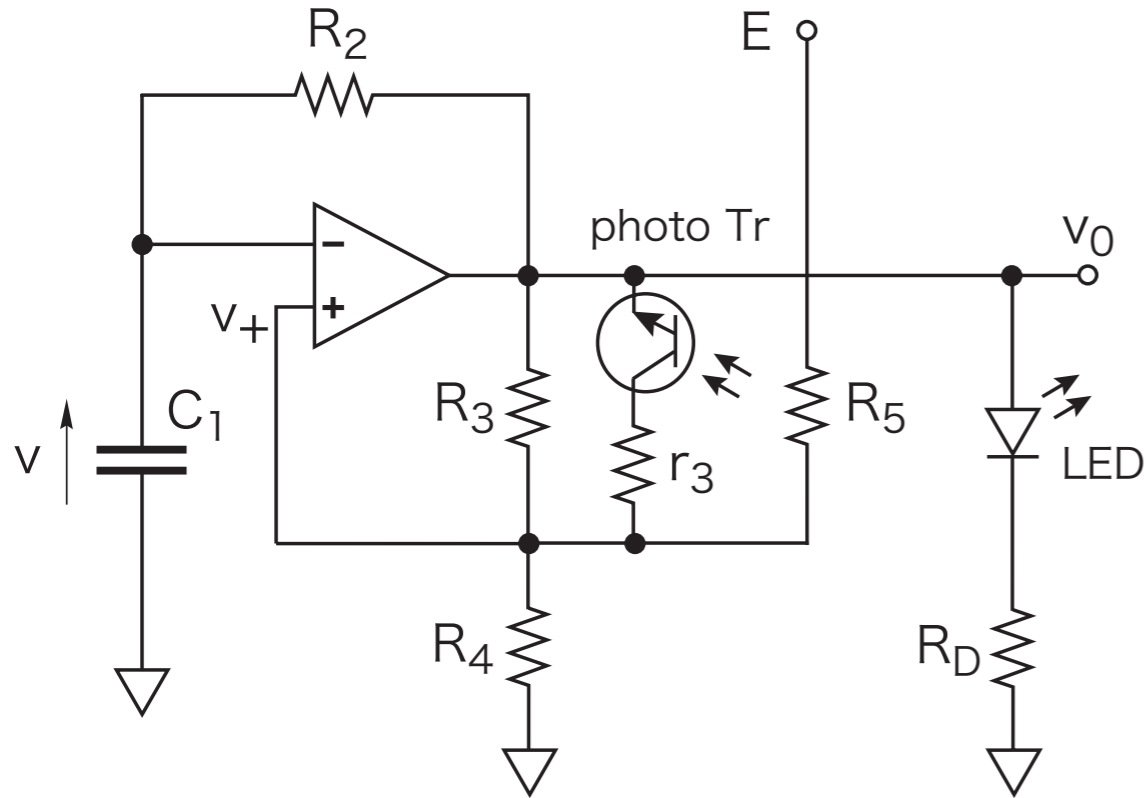
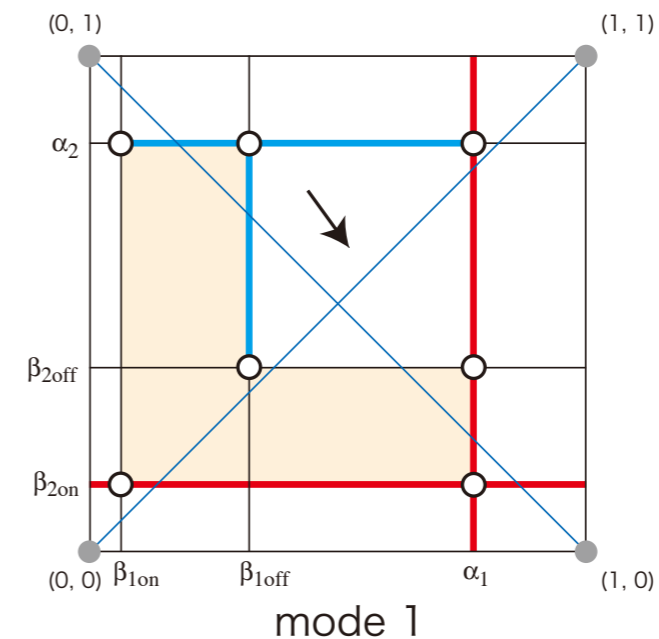
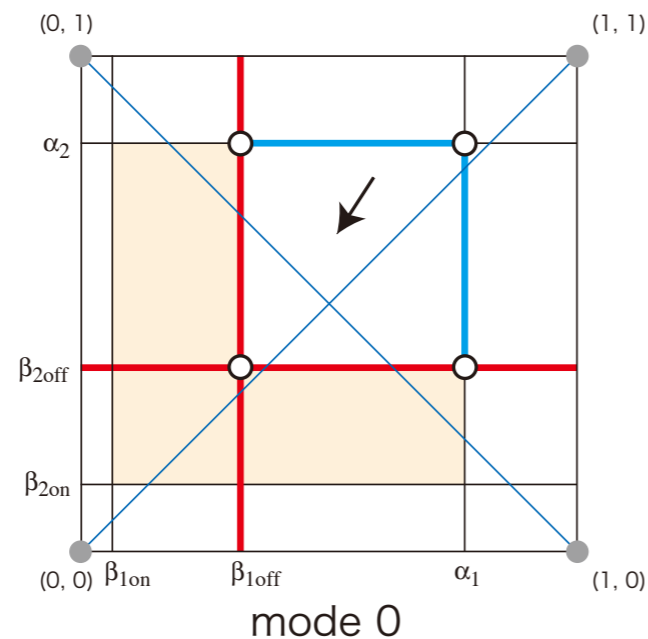
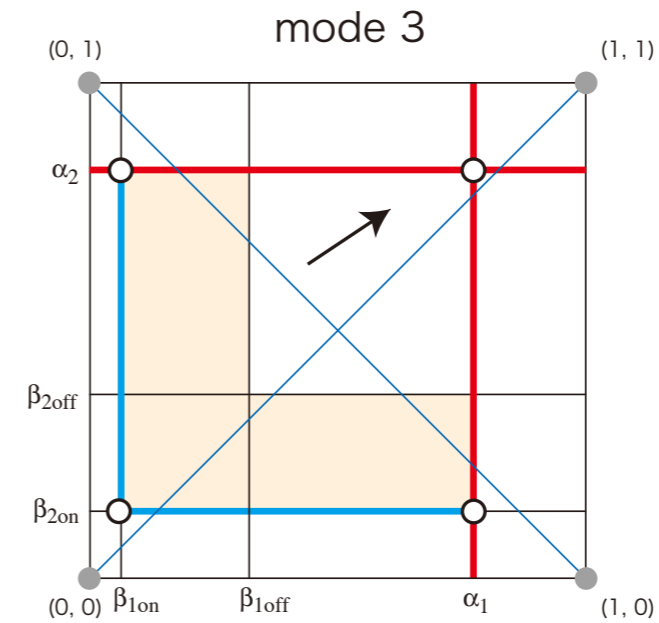
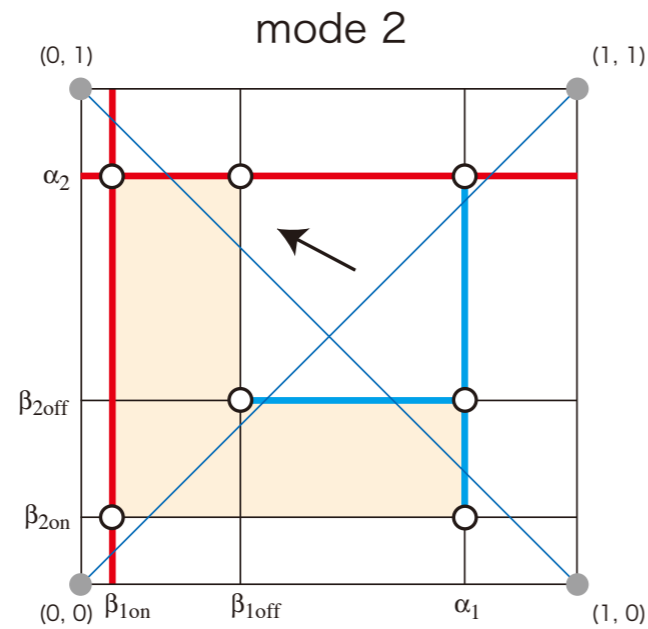


photo Tr		comparator : p	
		on:1	off:0
光入力 : q	on:1	off:0	on:1
	off:0	off:0	off:0



mode0 :  $(q, p) = (0, 0)$ ,  $dx/dt+x = 0$ ,  $dy/dt+y=0$

mode1 :  $(q, p) = (1, 0)$ ,  $dx/dt+x = 1$ ,  $dy/dt+y=0$ ,  $\beta_{2\_on}$

mode2 :  $(q, p) = (0, 1)$ ,  $dx/dt+x = 0$ ,  $dy/dt+y=1$ ,  $\beta_{1\_on}$

mode3 :  $(q, p) = (1, 1)$ ,  $dx/dt+x = 1$ ,  $dy/dt+y=1$



# mode variable method-coupled

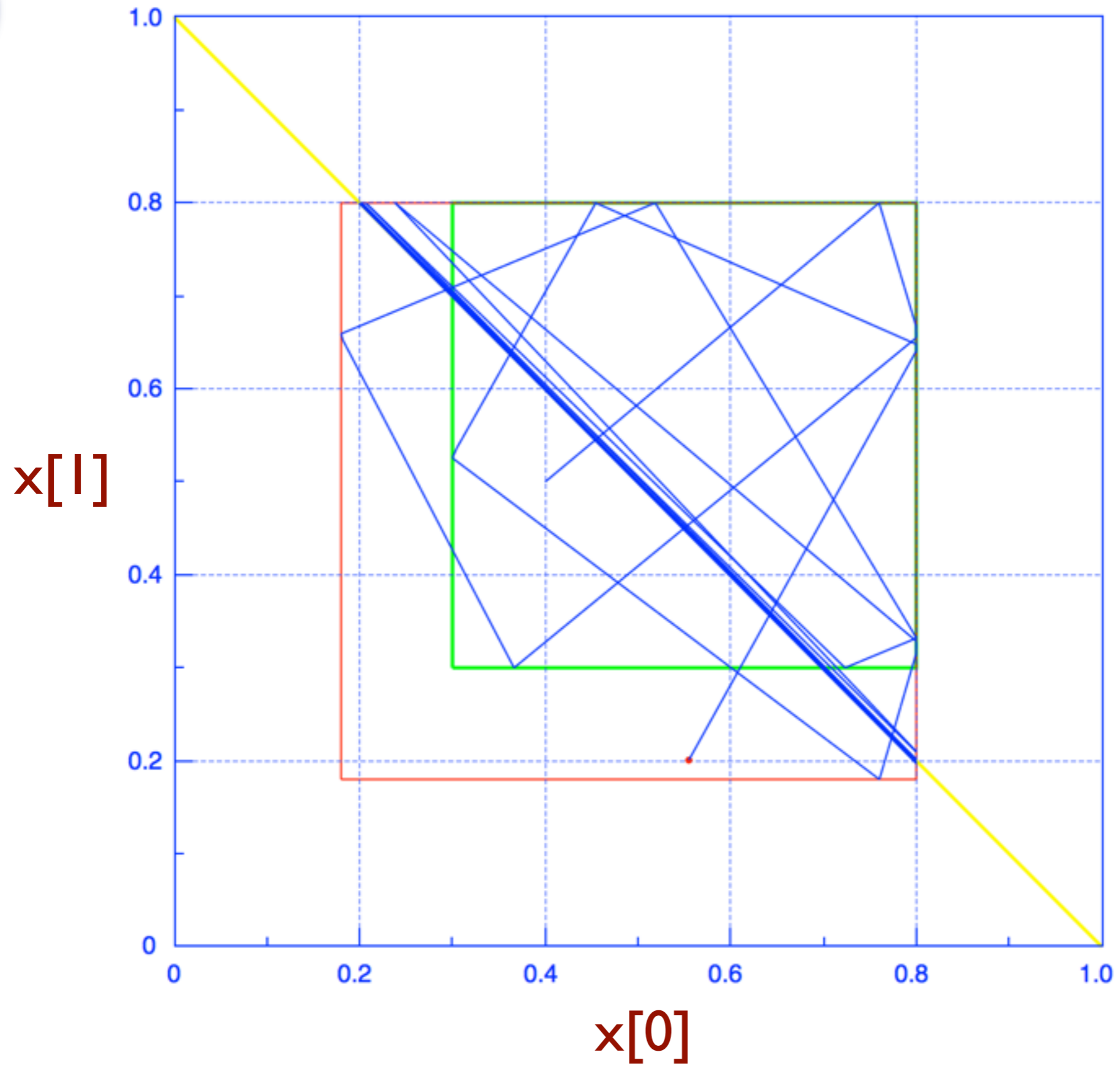
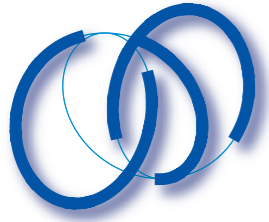
```

for (int i=0; i<2; i++) {
    if (mod[i]==1 && x[i]>xmax[i]) {
        dx[i]=-dx[i];
        mod[i]=0;
    }else if(mod[i]==0 && x[i]<xmin[i]){
        dx[i]=-dx[i];
        mod[i]=1;
    }
}

x1[0]=x[0]+dx[0];
x1[1]=x[1]+dx[1];

//coupling relation
if (q[0]==0 && q[1]==1) {
    bdrB[0]=bon;
}
else {
    bdrB[0]=boff;
    if (x[0]<bdrB[0]) {
        q[0]=1;
    }
}
if (q[1]==0 && q[0]==1) {
    bdrB[1]=bon;
}
else {
    bdrB[1]=boff;
    if (x[1]<bdrB[1]) {
        q[1]=1;
    }
}
}

```





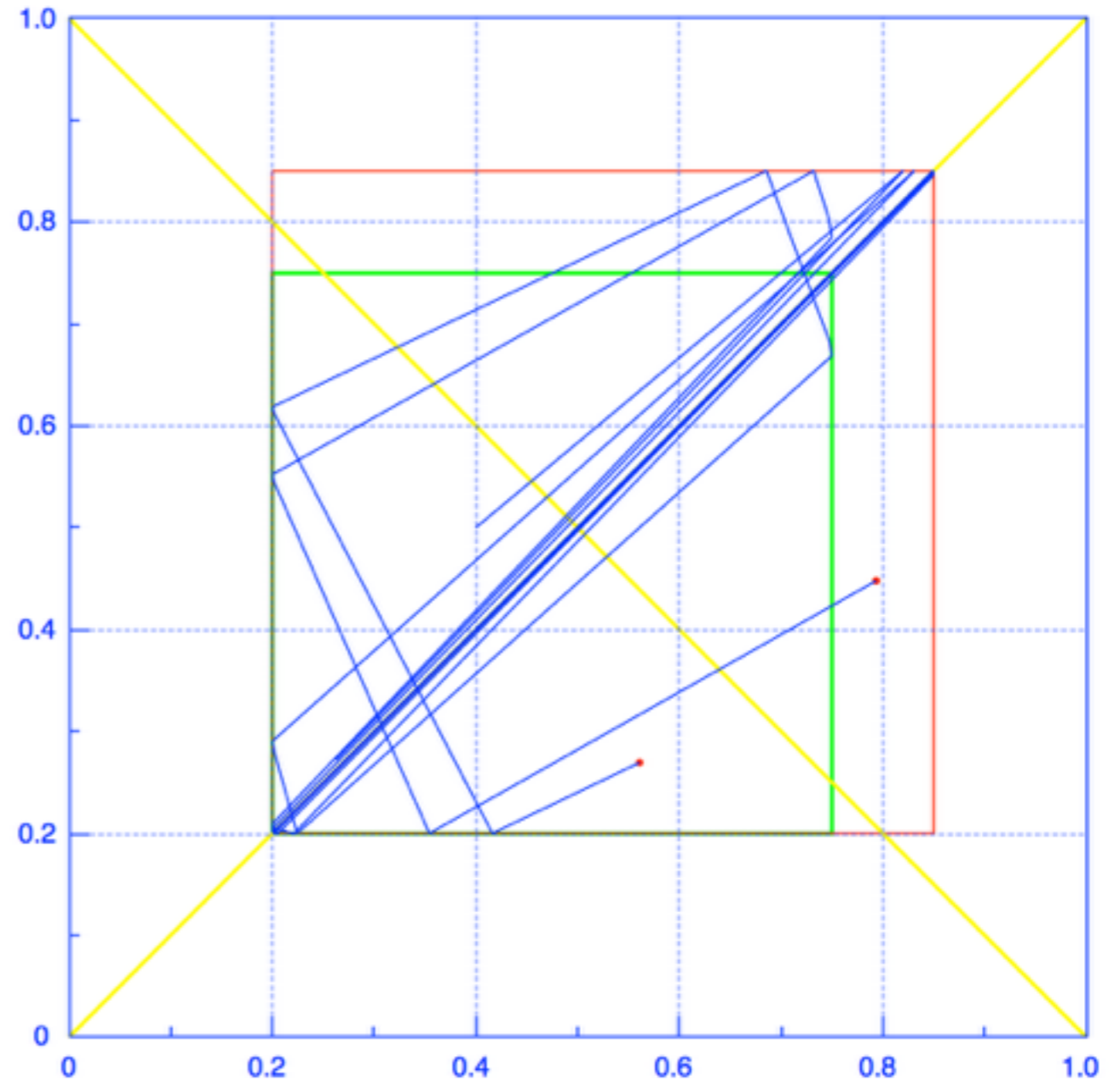
# mode variable method-coupled

```

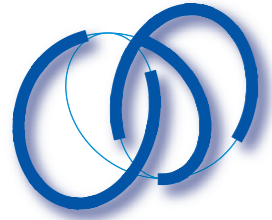
if (q[0]==1 && q[1]==1) {
    bdrA[0]=aon;
    bdrA[1]=aon;
}
else {
    bdrA[0]=aoff;
    bdrA[1]=aoff;
    if (x[0]>bdrA[0]) {
        q[0]=0;
    }
    if (x[1]>bdrA[1]) {
        q[1]=0;
    }
}

```

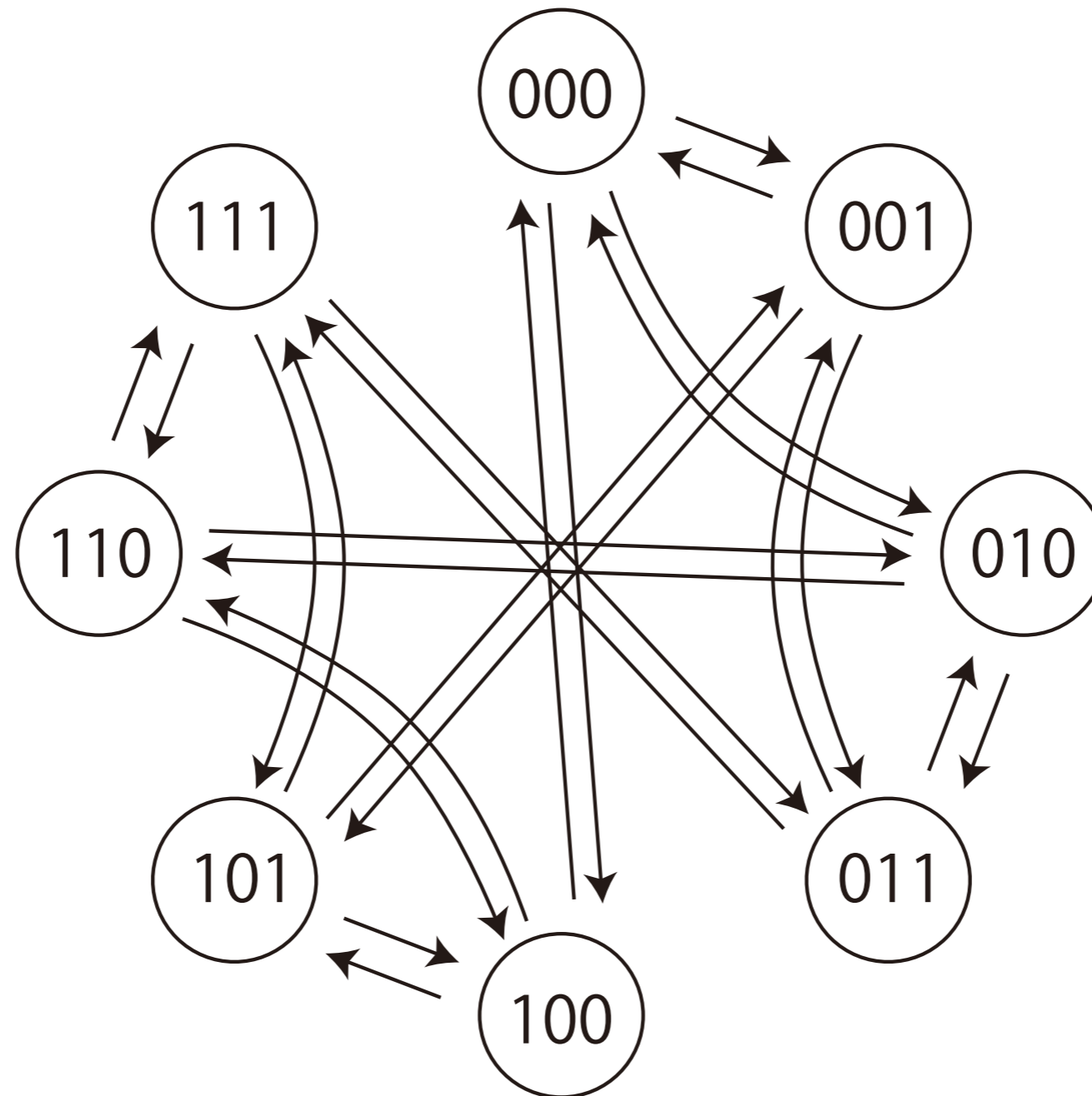
$x[1]$



$x[0]$



# LEDホテル 3 個結合のモード遷移図

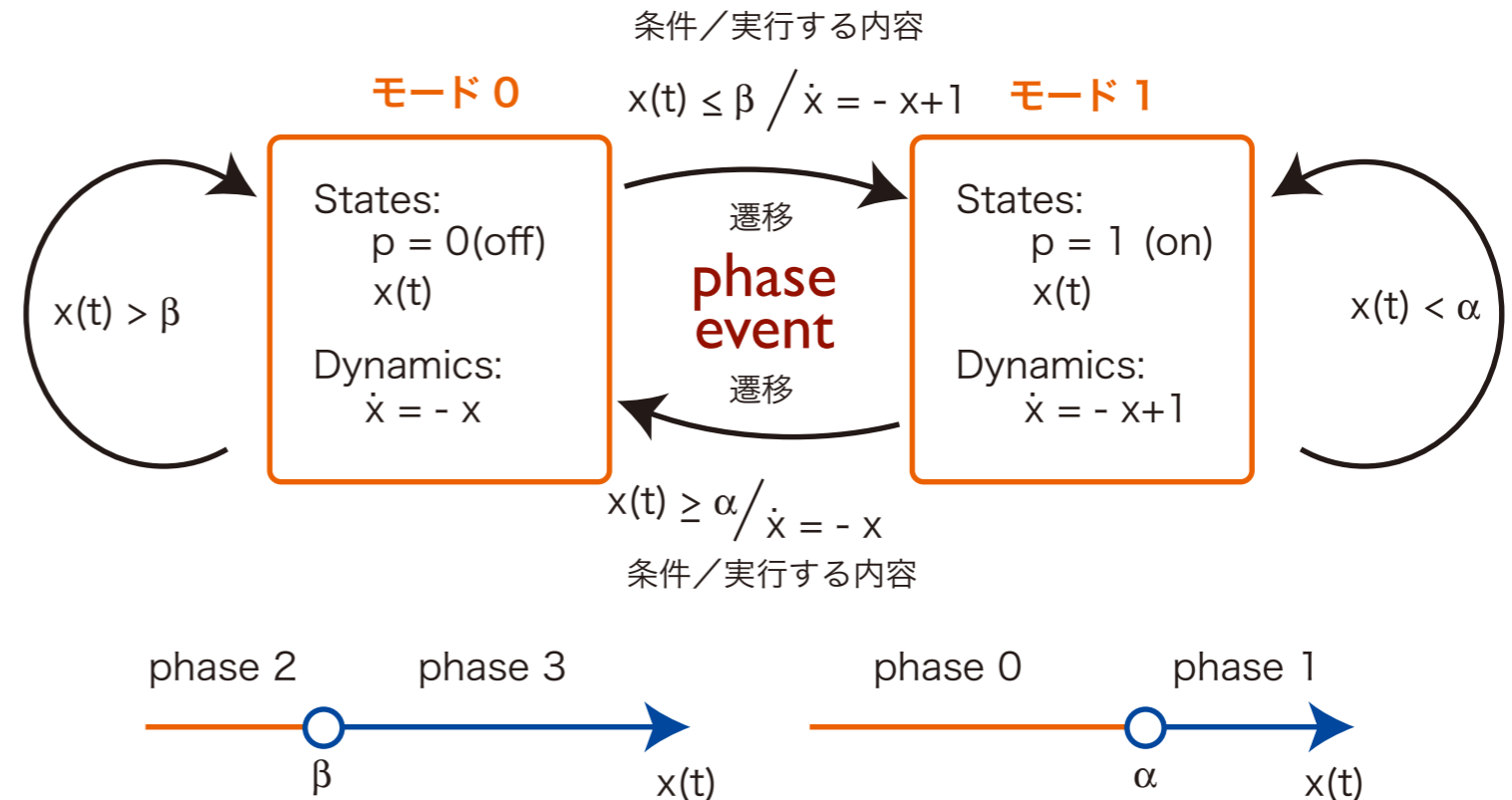
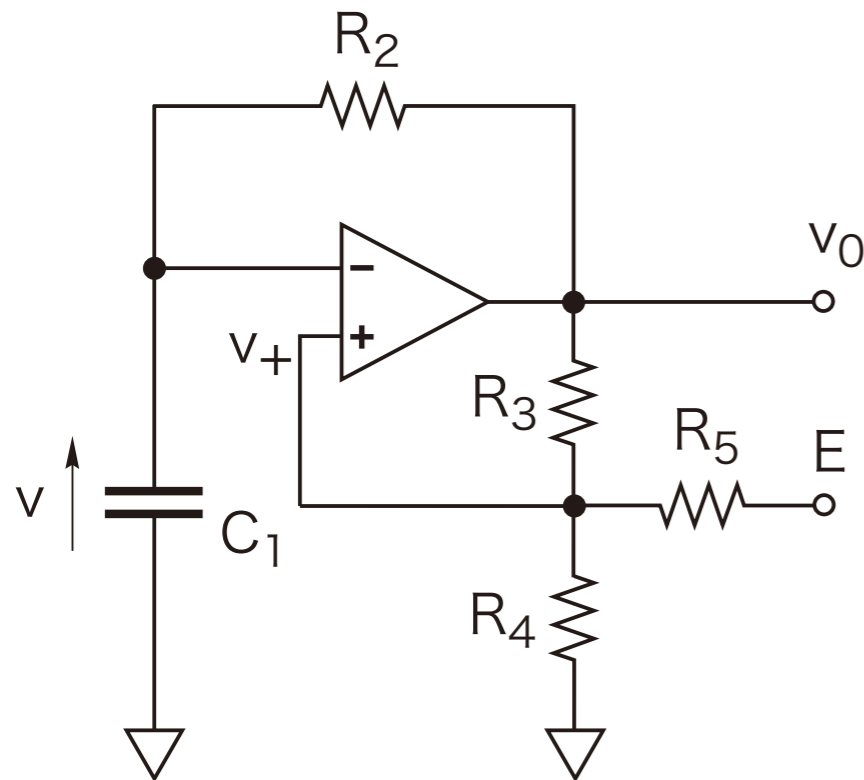


mode transition diagramが複雑になったらどうする？





# 1 bit Atomic Hybrid Oscillator



```

if (p==1 && x>bdrA) p=0;
if (p==0 && x<bdrB) p=1;
// with ODE containing p
// \dot{x} = -x + p;
  
```

2値論理変数を使ってmode transition diagramを表す



# まとめ

---

**論理変数を使う**：digital partのhardwareに還って考え直そう

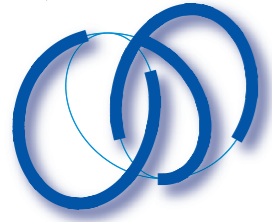
- ◎ 計算順序で結果が変わる
- ◎ bit操作を使って並列計算

## 3回全体のまとめ

**解析ツール**：何とか足場がみえてきた？

**今後の目標**：すこしmapの勉強が必要

**今後の目標**：環状結合SW Oscillatorの解析



# References

---

- [1] 高坂、上田、田原、川上、安部：Border-Collision分岐を呈する簡素な回路の実現と解析、電気学会論文誌 C, 平成14年11月号、pp.1908-1916.
- [2] T. Kousaka, T. Ueta and H. Kawakami, Bifurcation of Swotched Nonlinear Dynamical Systems, IEEE Trans. on Circuit and Systems II, Vol. 46, No. 7, July 1999, pp. 878-885.
- [3] 高坂拓司：断続動作特性を有する非線形力学系の分岐解析