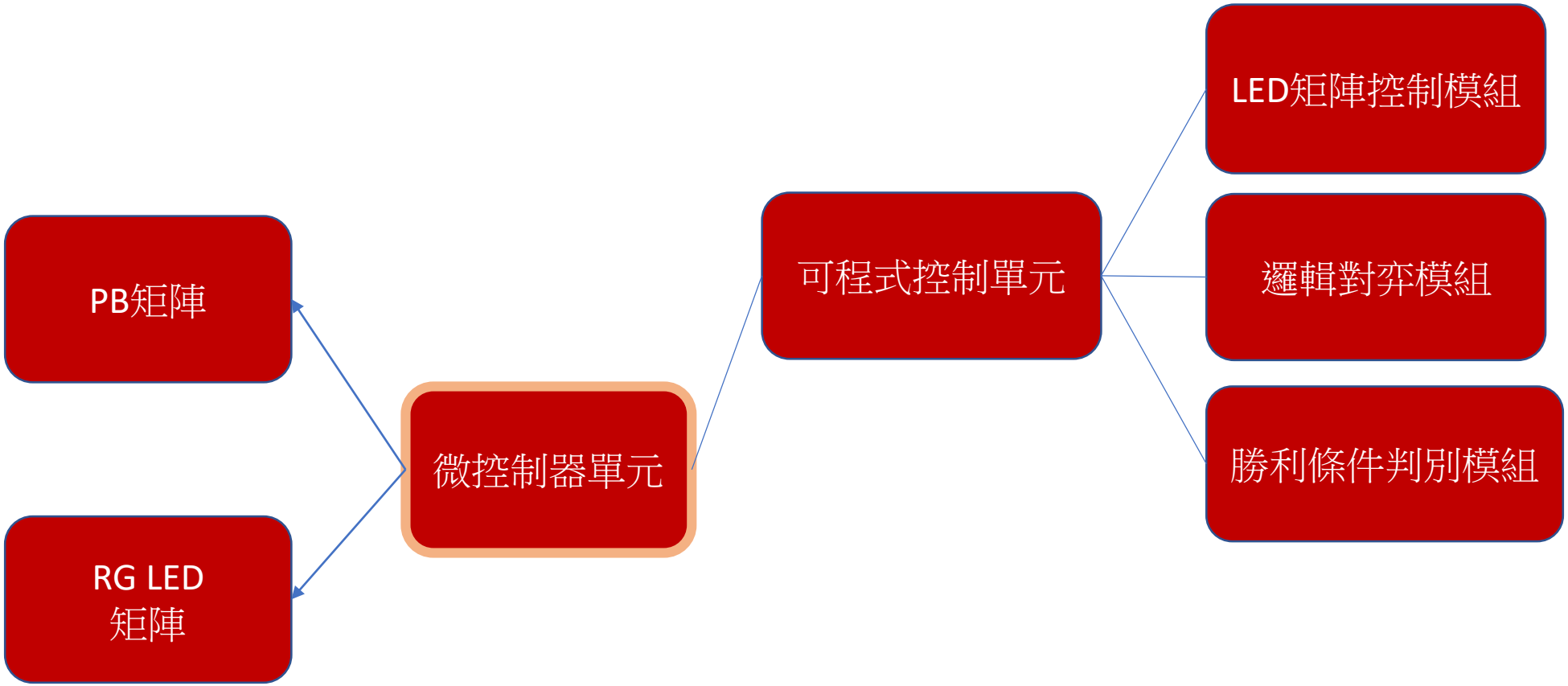
目錄

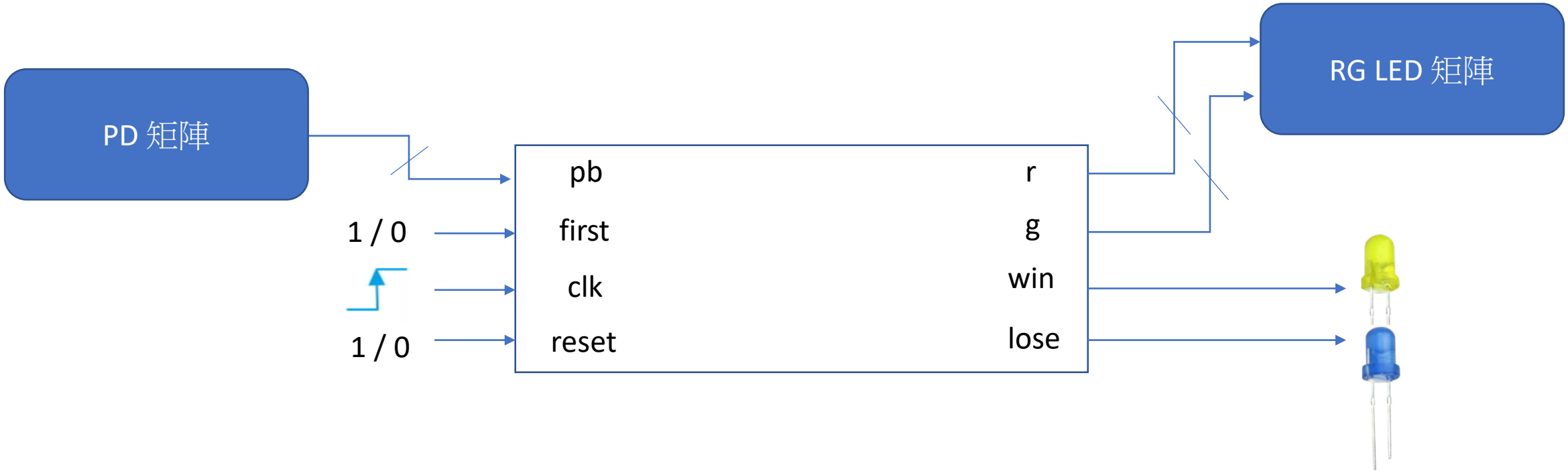
摘要

本論文主要研究方向即是製作一圈圈叉叉Tic-tac-toe系統。該系統架構可分為LED矩陣控制模組、勝利條件判別模組、邏輯對弈模組等子系統。電路設計包含LED矩陣、4x4鍵盤開關等元件。該系統除具備基礎之玩家對玩家（Player versus player）模式外，更有與電路邏輯對弈模式，分別可以選擇普通、超難模式，以及先後手。

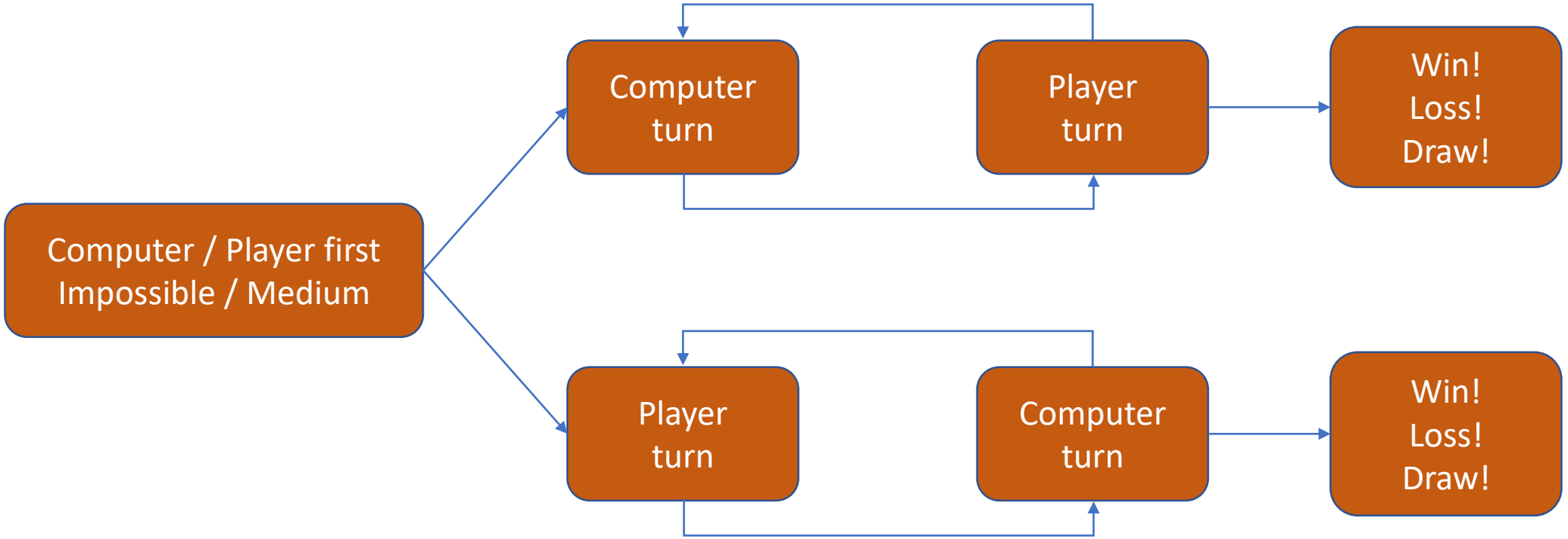
系統架構

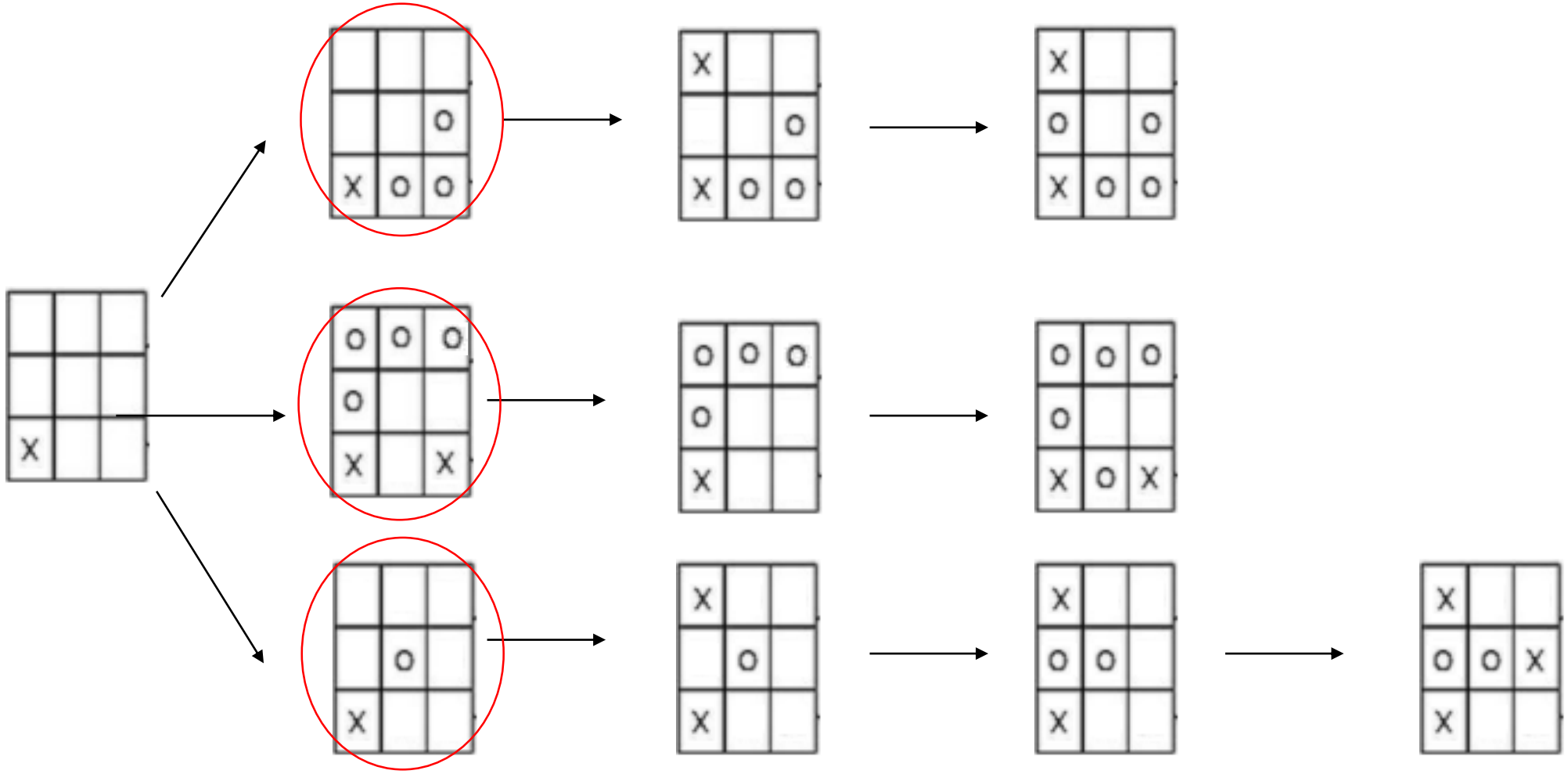


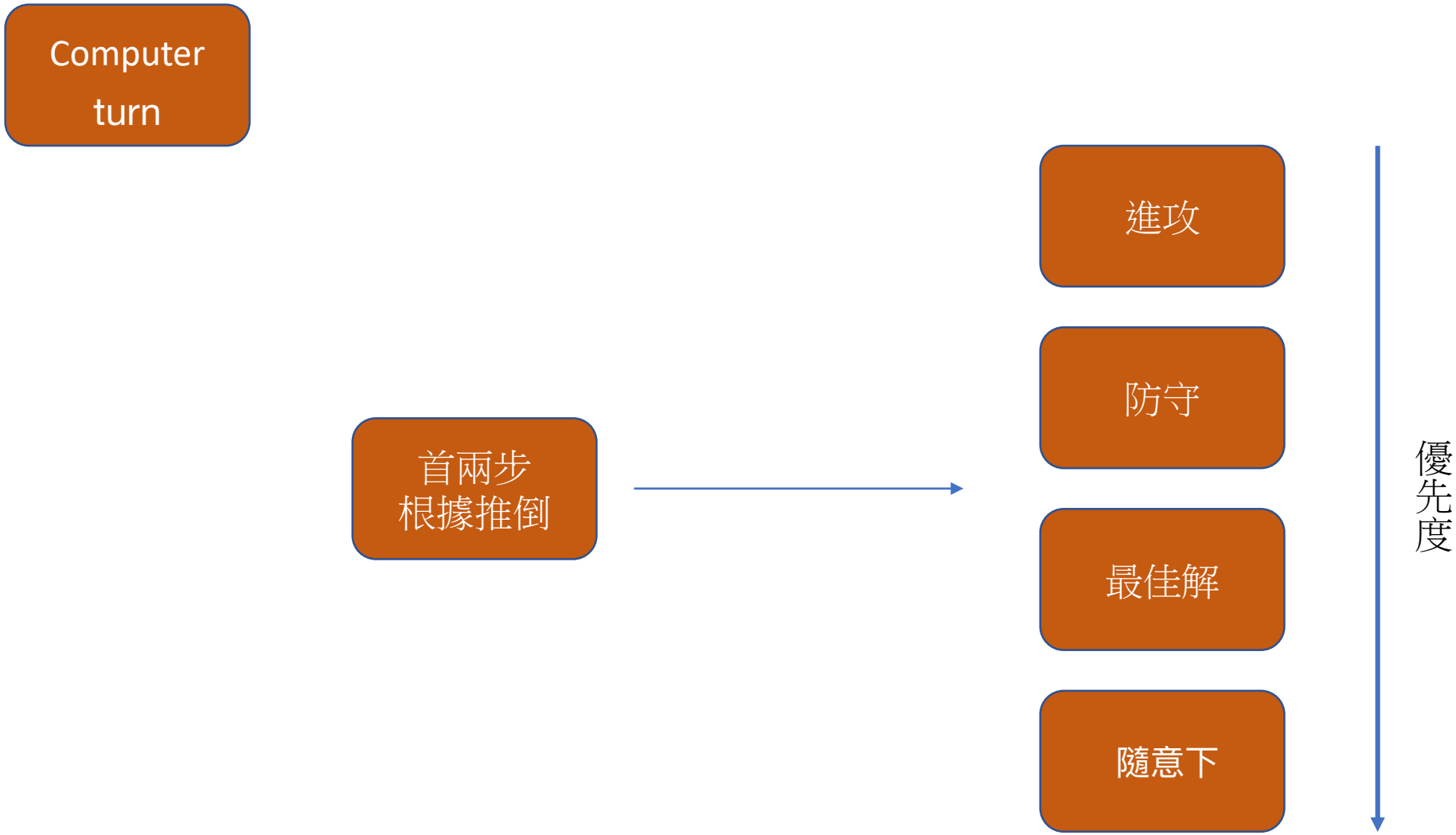
系統設計規劃



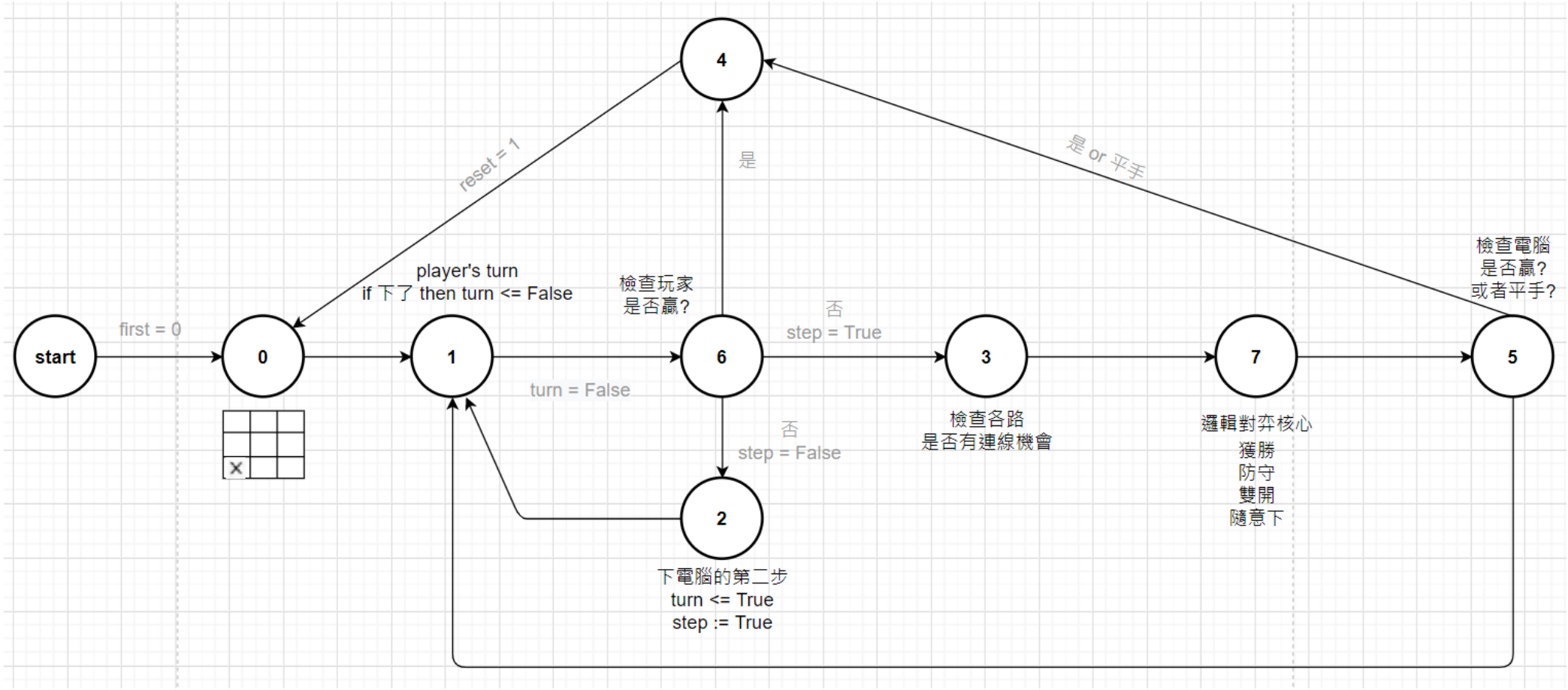
流程圖


Computer / Player first
Impossible / Medium
Computer turn
Player turn
Win!
Loss!
Draw!
Player turn
Computer turn
Win!
Loss!
Draw!

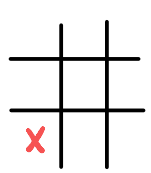


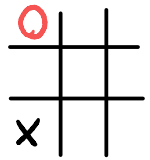

Computer turn
首兩步
根據推倒
進攻
防守
最佳解
隨意下
優先度

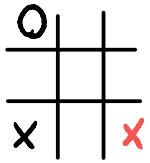
狀態圖


start
first = 0
0
player's turn
if 下了 then turn <= False
1
turn = False
檢查玩家
是否贏?
6
是
4
reset = 1
否
step = True
3
檢查各路
是否有連線機會
否
step = False
2
下電腦的第二步
turn <= True
step := True
7
邏輯對弈核心
獲勝
防守
雙開
隨意下
是 or 平手
檢查電腦
是否贏?
或者平手?
5

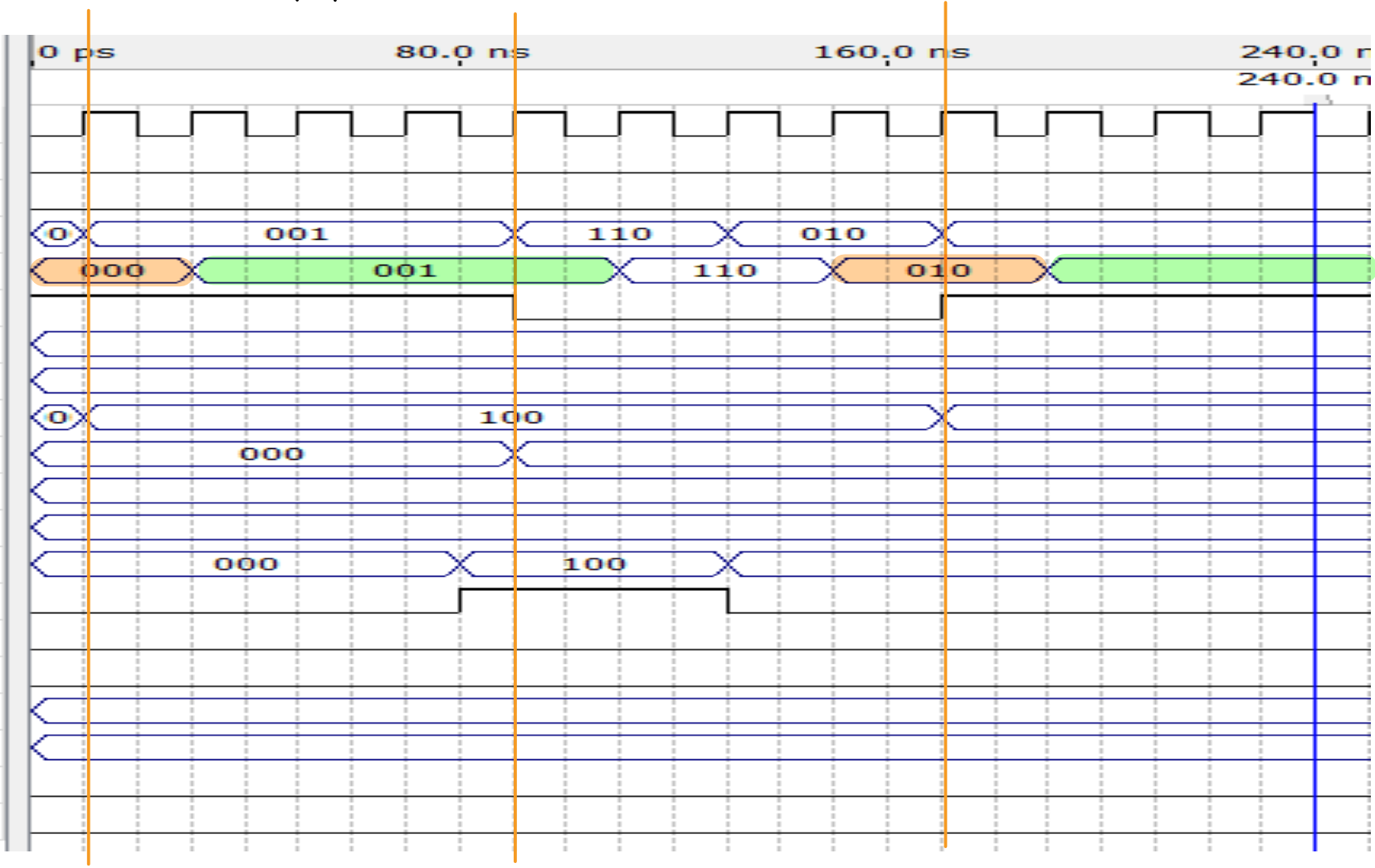
實驗成果

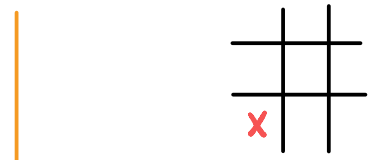


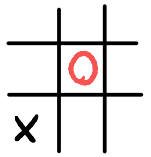


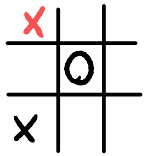


	Name	Value at 240.0 ns
in	clk	B 0
in	first	B 0
in	reset	B 0
R >	nextstate	B 001
R >	State	B 001
out	tuuurn	B 1
out >	red0	B 000
out >	red1	B 000
out >	red2	B 101
out >	green0	B 100
out >	green1	B 000
out >	green2	B 000
in ∨	pb0	B 000
in	pb00	B 0
in	pb01	B 0
in	pb02	B 0
in >	pb1	B 000
in >	pb2	B 000
out	win	B 0
out	lose	B 0

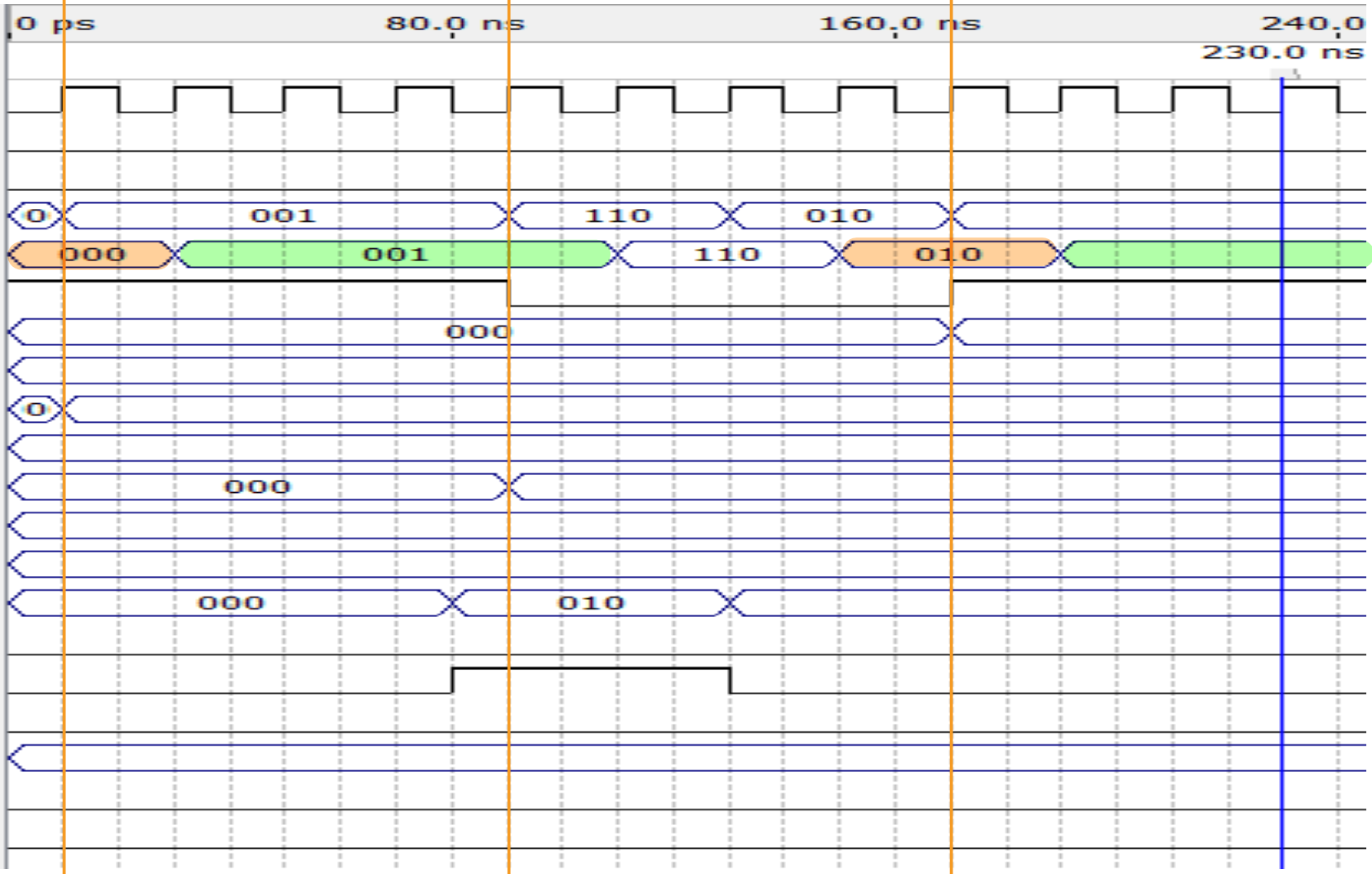


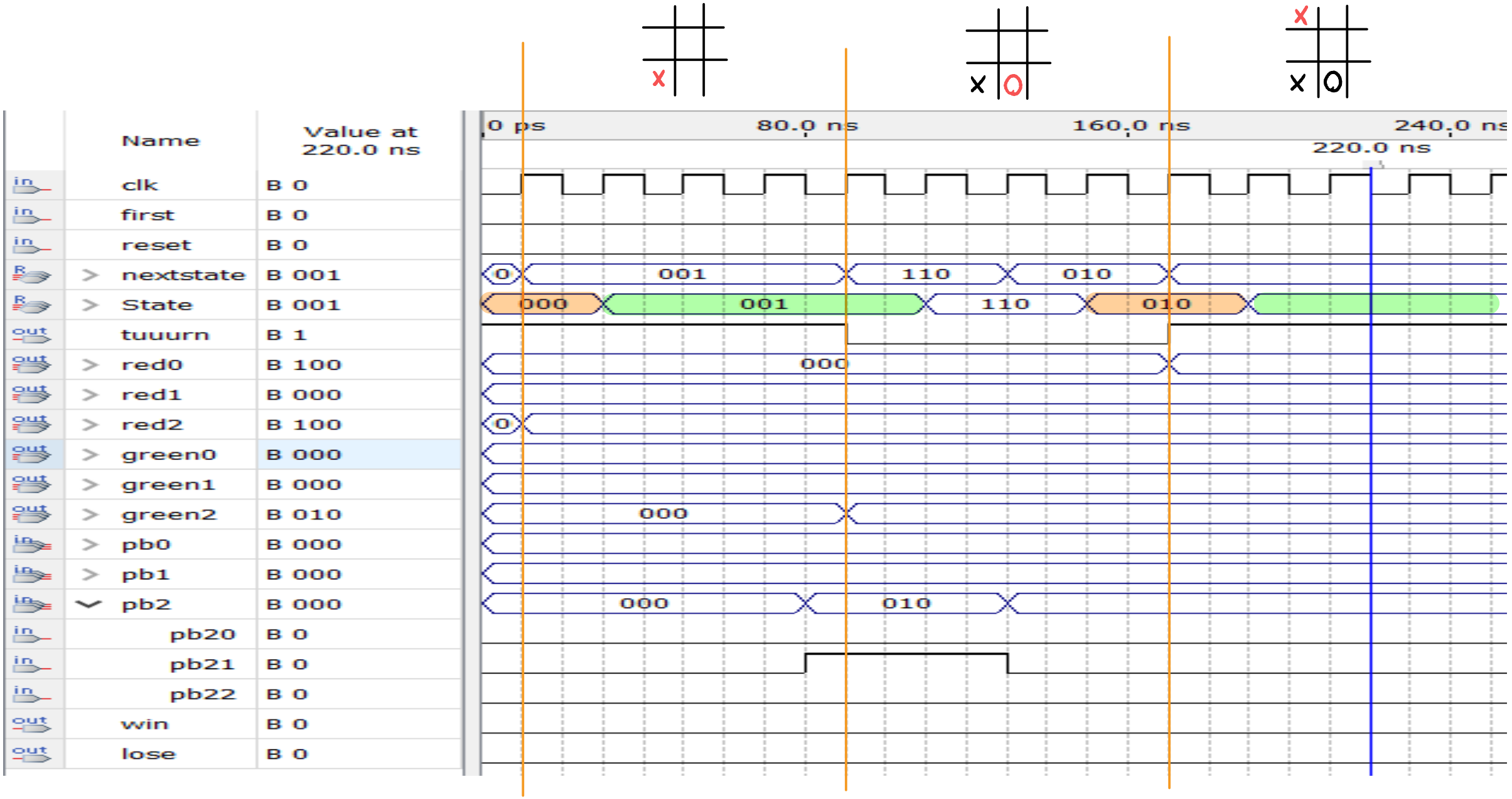


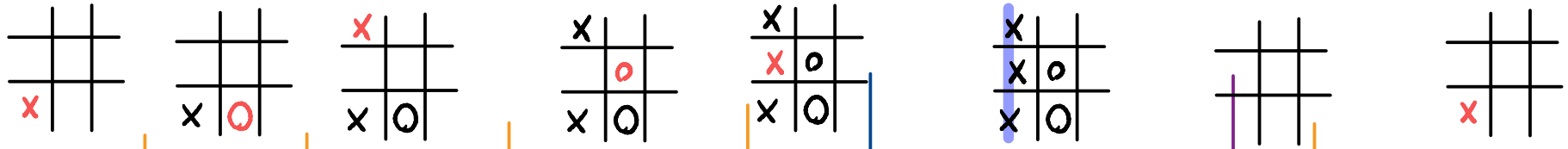




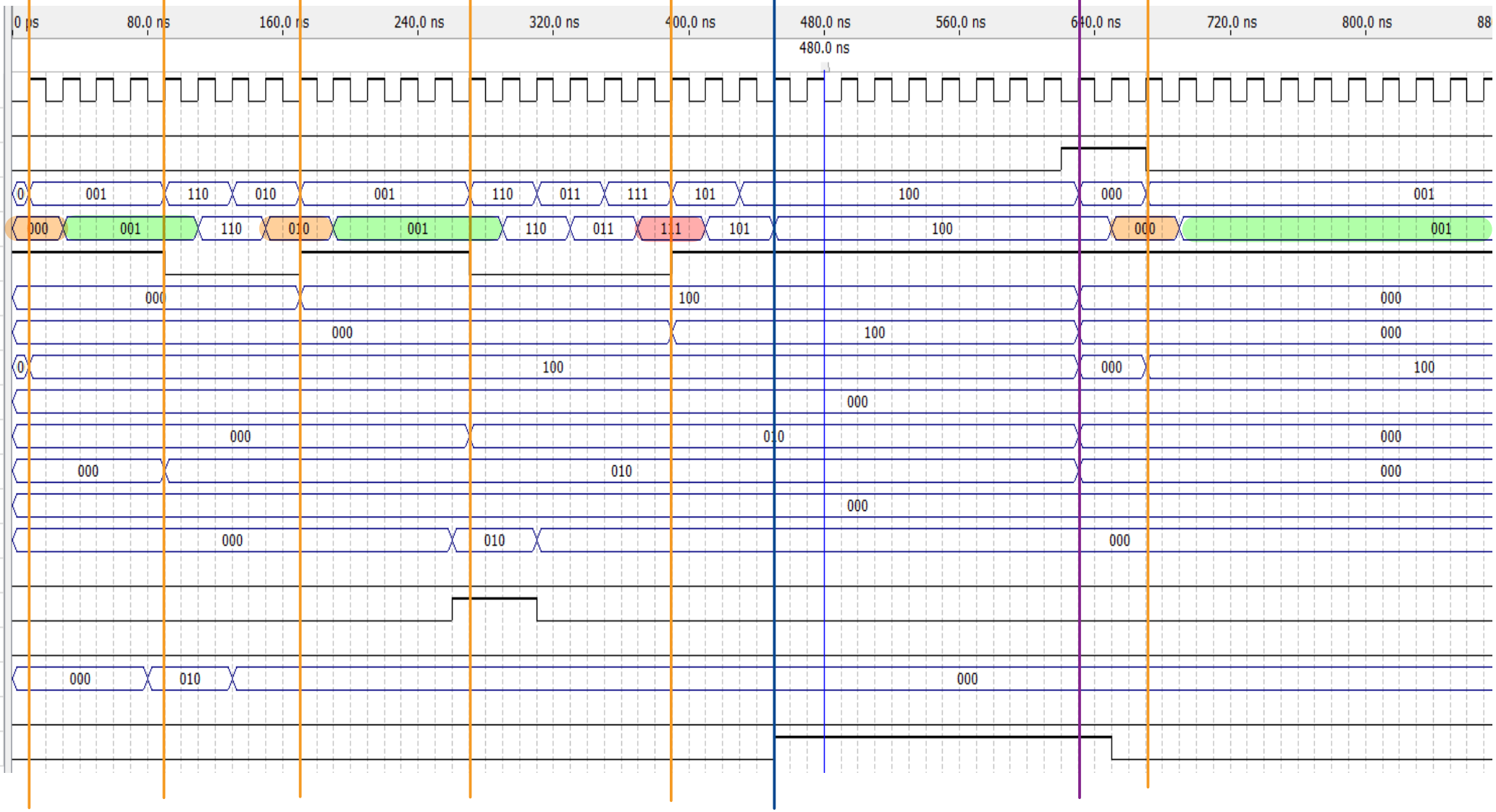
	Name	Value at 230.0 ns
in	clk	B 1
in	first	B 0
in	reset	B 0
R	> nextstate	B 001
R	> State	B 001
out	tuuurn	B 1
out	> red0	B 100
out	> red1	B 000
out	> red2	B 100
out	> green0	B 000
out	> green1	B 010
out	> green2	B 000
in	> pb0	B 000
in	⌄ pb1	B 000
in	pb10	B 0
in	pb11	B 0
in	pb12	B 0
in	> pb2	B 000
out	win	B 0
out	lose	B 0



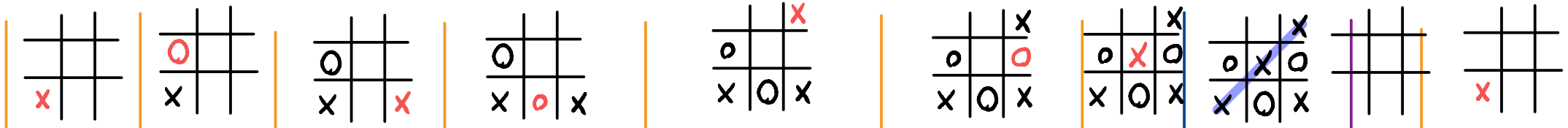

Name
Value at 220.0 ns
clk B 0
first B 0
reset B 0
nextstate B 001
State B 001
tuuurn B 1
red0 B 100
red1 B 000
red2 B 100
green0 B 000
green1 B 000
green2 B 010
pb0 B 000
pb1 B 000
pb2 B 000
pb20 B 0
pb21 B 0
pb22 B 0
win B 0
lose B 0
0 ps
80.0 ns
160.0 ns
240.0 ns
220.0 ns
0
001
110
010
000
001
110
010
000
0
000
000
010



Name	Value at 480.0 ns
clk	B 0
first	B 0
reset	B 0
nextstate	B 100
State	B 100
tuuurn	B 1
red0	B 100
red1	B 100
red2	B 100
green0	B 000
green1	B 010
green2	B 010
pb0	B 000
pb1	B 000
pb10	B 0
pb11	B 0
pb12	B 0
pb2	B 000
win	B 0
lose	B 1

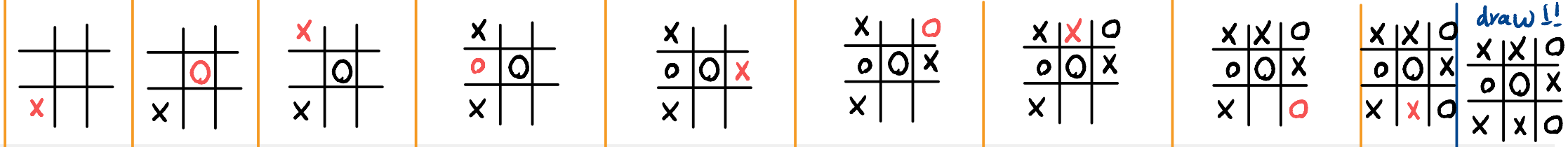


①攻擊 > 防守 → ②失敗

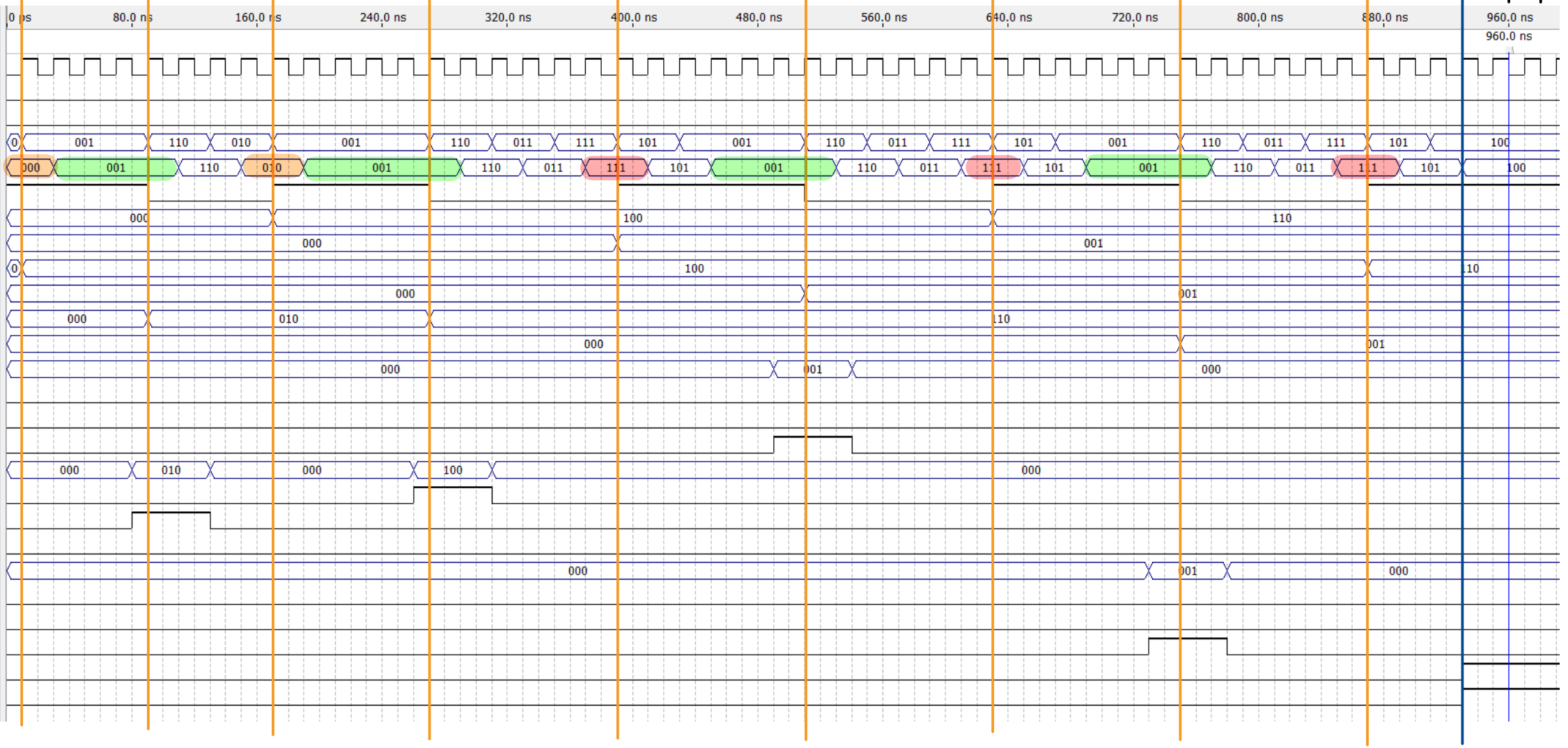


Name	Value at 750.0 ns
clk	B 1
first	B 0
reset	B 0
nextstate	B 100
State	B 100
tuuurn	B 1
red0	B 001
red1	B 010
red2	B 101
green0	B 000
green1	B 101
green2	B 010
pb0	B 000
pb00	B 0
pb01	B 0
pb02	B 0
pb1	B 000
pb10	B 0
pb11	B 0
pb12	B 0
pb2	B 000
pb20	B 0
pb21	B 0
pb22	B 0
win	B 0
lose	B 1

①雙開 → ②攻擊 → ③失敗



Name	Value at 960.0 ns
clk	B 0
first	B 0
reset	B 0
nextstate	B 100
State	B 100
tuuurn	B 1
red0	B 110
red1	B 001
red2	B 110
green0	B 001
green1	B 110
green2	B 001
pb0	B 000
pb00	B 0
pb01	B 0
pb02	B 0
pb1	B 000
pb10	B 0
pb11	B 0
pb12	B 0
pb2	B 000
pb20	B 0
pb21	B 0
pb22	B 0
win	B 1
lose	B 1



①防守 → ②隨意下 → ③平手

附錄

玩家下棋

```
if pb00='1' and p(0)(0) = '0' then
   nextstate <= 6;
   rg(0)(0) <= '1';
   turn <= False;
elsif pb01='1' and p(0)(1) = '0' then
   nextstate <= 6;
   rg(0)(1) <= '1';
   turn <= False;
elsif pb02='1' and p(0)(2) = '0' then
   nextstate <= 6;
   rg(0)(2) <= '1';
   turn <= False;
elsif pb10='1' and p(1)(0) = '0' then
   nextstate <= 6;
   rg(1)(0) <= '1';
   turn <= False;
elsif pb11='1' and p(1)(1) = '0' then
   nextstate <= 6;
   rg(1)(1) <= '1';
   turn <= False;
elsif pb12='1' and p(1)(2) = '0' then
   nextstate <= 6;
   rg(1)(2) <= '1';
   turn <= False;
elsif pb20='1' and p(2)(0) = '0' then
   nextstate <= 6;
   rg(2)(0) <= '1';
   turn <= False;
elsif pb21='1' and p(2)(1) = '0' then
   nextstate <= 6;
   rg(2)(1) <= '1';
   turn <= False;
elsif pb22='1' and p(2)(2) = '0' then
   nextstate <= 6;
   rg(2)(2) <= '1';
   turn <= False;
end if;
```

攻撃

```
-----------------------------------------------------------------
--to attack
--- -
if rr(0)="011" and p(0)(0)='0' then nextstate <= 5; turn <= True; rr(0)(0) <= '1';
elsif rr(0)="101" and p(0)(1)='0' then nextstate <= 5; turn <= True; rr(0)(1) <= '1';
elsif rr(0)="110" and p(0)(2)='0' then nextstate <= 5; turn <= True; rr(0)(2) <= '1';
elsif rr(1)="011" and p(1)(0)='0' then nextstate <= 5; turn <= True; rr(1)(0) <= '1';
elsif rr(1)="101" and p(1)(1)='0' then nextstate <= 5; turn <= True; rr(1)(1) <= '1';
elsif rr(1)="110" and p(1)(2)='0' then nextstate <= 5; turn <= True; rr(1)(2) <= '1';
elsif rr(2)="011" and p(2)(0)='0' then nextstate <= 5; turn <= True; rr(2)(0) <= '1';
elsif rr(2)="101" and p(2)(1)='0' then nextstate <= 5; turn <= True; rr(2)(1) <= '1';
elsif rr(2)="110" and p(2)(2)='0' then nextstate <= 5; turn <= True; rr(2)(2) <= '1';
--- |
elsif (rr(0)(0)='0' and rr(1)(0)='1' and rr(2)(0)='1') and p(0)(0)='0' then nextstate <= 5; turn <= True; rr(0)(0) <= '1';
elsif (rr(0)(0)='1' and rr(1)(0)='0' and rr(2)(0)='1') and p(1)(0)='0' then nextstate <= 5; turn <= True; rr(1)(0) <= '1';
elsif (rr(0)(0)='1' and rr(1)(0)='1' and rr(2)(0)='0') and p(2)(0)='0' then nextstate <= 5; turn <= True; rr(2)(0) <= '1';
elsif (rr(0)(1)='0' and rr(1)(1)='1' and rr(2)(1)='1') and p(0)(1)='0' then nextstate <= 5; turn <= True; rr(0)(1) <= '1';
elsif (rr(0)(1)='1' and rr(1)(1)='0' and rr(2)(1)='1') and p(1)(1)='0' then nextstate <= 5; turn <= True; rr(1)(1) <= '1';
elsif (rr(0)(1)='1' and rr(1)(1)='1' and rr(2)(1)='0') and p(2)(1)='0' then nextstate <= 5; turn <= True; rr(2)(1) <= '1';
elsif (rr(0)(2)='0' and rr(1)(2)='1' and rr(2)(2)='1') and p(0)(2)='0' then nextstate <= 5; turn <= True; rr(0)(2) <= '1';
elsif (rr(0)(2)='1' and rr(1)(2)='0' and rr(2)(2)='1') and p(1)(2)='0' then nextstate <= 5; turn <= True; rr(1)(2) <= '1';
elsif (rr(0)(2)='1' and rr(1)(2)='1' and rr(2)(2)='0') and p(2)(2)='0' then nextstate <= 5; turn <= True; rr(2)(2) <= '1';
--- \
elsif (rr(0)(0)='0' and rr(1)(1)='1' and rr(2)(2)='1') and p(0)(0)='0' then nextstate <= 5; turn <= True; rr(0)(0) <= '1';
elsif (rr(0)(0)='1' and rr(1)(1)='0' and rr(2)(2)='1') and p(1)(1)='0' then nextstate <= 5; turn <= True; rr(1)(1) <= '1';
elsif (rr(0)(0)='1' and rr(1)(1)='1' and rr(2)(2)='0') and p(2)(2)='0' then nextstate <= 5; turn <= True; rr(2)(2) <= '1';
--- /
elsif (rr(2)(0)='0' and rr(1)(1)='1' and rr(0)(2)='1') and p(2)(0)='0' then nextstate <= 5; turn <= True; rr(2)(0) <= '1';
elsif (rr(2)(0)='1' and rr(1)(1)='0' and rr(0)(2)='1') and p(1)(1)='0' then nextstate <= 5; turn <= True; rr(1)(1) <= '1';
elsif (rr(2)(0)='1' and rr(1)(1)='1' and rr(0)(2)='0') and p(0)(2)='0' then nextstate <= 5; turn <= True; rr(0)(2) <= '1';
```

防守

```
if (rr(0)="100" and p(0)(1)='0' and p(0)(2)='0')
or (rr(0)="010" and p(0)(0)='0' and p(0)(2)='0')
or (rr(0)="001" and p(0)(0)='0' and p(0)(1)='0') then l1 := 1;
else l1 := 0; end if;
if (rr(1)="100" and p(1)(1)='0' and p(1)(2)='0')
or (rr(1)="010" and p(1)(0)='0' and p(1)(2)='0')
or (rr(1)="001" and p(1)(0)='0' and p(1)(1)='0') then l2 := 1;
else l2 := 0; end if;
if (rr(2)="100" and p(2)(1)='0' and p(2)(2)='0')
or (rr(2)="010" and p(2)(0)='0' and p(2)(2)='0')
or (rr(2)="001" and p(2)(0)='0' and p(2)(1)='0') then l3 := 1;
else l3 := 0; end if;

if (rr(0)(0)='1' and rr(1)(0)='0' and rr(2)(0)='0' and p(1)(0)='0' and p(2)(0)='0')
or (rr(0)(0)='0' and rr(1)(0)='1' and rr(2)(0)='0' and p(0)(0)='0' and p(2)(0)='0')
or (rr(0)(0)='0' and rr(1)(0)='0' and rr(2)(0)='1' and p(0)(0)='0' and p(1)(0)='0') then l4 := 1;
else l4 := 0; end if;
if (rr(0)(1)='1' and rr(1)(1)='0' and rr(2)(1)='0' and p(1)(1)='0' and p(2)(1)='0')
or (rr(0)(1)='0' and rr(1)(1)='1' and rr(2)(1)='0' and p(0)(1)='0' and p(2)(1)='0')
or (rr(0)(1)='0' and rr(1)(1)='0' and rr(2)(1)='1' and p(0)(1)='0' and p(1)(1)='0') then l5 := 1;
else l5 := 0; end if;
if (rr(0)(2)='1' and rr(1)(2)='0' and rr(2)(2)='0' and p(1)(2)='0' and p(2)(2)='0')
or (rr(0)(2)='0' and rr(1)(2)='1' and rr(2)(2)='0' and p(0)(2)='0' and p(2)(2)='0')
or (rr(0)(2)='0' and rr(1)(2)='0' and rr(2)(2)='1' and p(0)(2)='0' and p(1)(2)='0') then l6 := 1;
else l6 := 0; end if;

if (rr(0)(0)='1' and rr(1)(1)='0' and rr(2)(2)='0' and p(1)(1)='0' and p(2)(2)='0')
or (rr(0)(0)='0' and rr(1)(1)='1' and rr(2)(2)='0' and p(0)(0)='0' and p(2)(2)='0')
or (rr(0)(0)='0' and rr(1)(1)='0' and rr(2)(2)='1' and p(0)(0)='0' and p(1)(1)='0') then l7 := 1;
else l7 := 0; end if;
if (rr(2)(0)='1' and rr(1)(1)='0' and rr(0)(2)='0' and p(1)(1)='0' and p(0)(2)='0')
or (rr(2)(0)='0' and rr(1)(1)='1' and rr(0)(2)='0' and p(2)(0)='0' and p(0)(2)='0')
or (rr(2)(0)='0' and rr(1)(1)='0' and rr(0)(2)='1' and p(2)(0)='0' and p(1)(1)='0') then l8 := 1;
else l8 := 0; end if;
```

```
-------------------------------------------------------------------
--to defend
--- -
elsif rg(0)="011" and p(0)(0)='0' then nextstate <= 5; turn <= True; rr(0)(0) <= '1';
elsif rg(0)="101" and p(0)(1)='0' then nextstate <= 5; turn <= True; rr(0)(1) <= '1';
elsif rg(0)="110" and p(0)(2)='0' then nextstate <= 5; turn <= True; rr(0)(2) <= '1';
elsif rg(1)="011" and p(1)(0)='0' then nextstate <= 5; turn <= True; rr(1)(0) <= '1';
elsif rg(1)="101" and p(1)(1)='0' then nextstate <= 5; turn <= True; rr(1)(1) <= '1';
elsif rg(1)="110" and p(1)(2)='0' then nextstate <= 5; turn <= True; rr(1)(2) <= '1';
elsif rg(2)="011" and p(2)(0)='0' then nextstate <= 5; turn <= True; rr(2)(0) <= '1';
elsif rg(2)="101" and p(2)(1)='0' then nextstate <= 5; turn <= True; rr(2)(1) <= '1';
elsif rg(2)="110" and p(2)(2)='0' then nextstate <= 5; turn <= True; rr(2)(2) <= '1';
--- |
elsif (rg(0)(0)='0' and rg(1)(0)='1' and rg(2)(0)='1') and p(0)(0)='0' then nextstate <= 5; turn <= True; rr(0)(0) <= '1';
elsif (rg(0)(0)='1' and rg(1)(0)='0' and rg(2)(0)='1') and p(1)(0)='0' then nextstate <= 5; turn <= True; rr(1)(0) <= '1';
elsif (rg(0)(0)='1' and rg(1)(0)='1' and rg(2)(0)='0') and p(2)(0)='0' then nextstate <= 5; turn <= True; rr(2)(0) <= '1';
elsif (rg(0)(1)='0' and rg(1)(1)='1' and rg(2)(1)='1') and p(0)(1)='0' then nextstate <= 5; turn <= True; rr(0)(1) <= '1';
elsif (rg(0)(1)='1' and rg(1)(1)='0' and rg(2)(1)='1') and p(1)(1)='0' then nextstate <= 5; turn <= True; rr(1)(1) <= '1';
elsif (rg(0)(1)='1' and rg(1)(1)='1' and rg(2)(1)='0') and p(2)(1)='0' then nextstate <= 5; turn <= True; rr(2)(1) <= '1';
elsif (rg(0)(2)='0' and rg(1)(2)='1' and rg(2)(2)='1') and p(0)(2)='0' then nextstate <= 5; turn <= True; rr(0)(2) <= '1';
elsif (rg(0)(2)='1' and rg(1)(2)='0' and rg(2)(2)='1') and p(1)(2)='0' then nextstate <= 5; turn <= True; rr(1)(2) <= '1';
elsif (rg(0)(2)='1' and rg(1)(2)='1' and rg(2)(2)='0') and p(2)(2)='0' then nextstate <= 5; turn <= True; rr(2)(2) <= '1';
--- \
elsif (rg(0)(0)='0' and rg(1)(1)='1' and rg(2)(2)='1') and p(0)(0)='0' then nextstate <= 5; turn <= True; rr(0)(0) <= '1';
elsif (rg(0)(0)='1' and rg(1)(1)='0' and rg(2)(2)='1') and p(1)(1)='0' then nextstate <= 5; turn <= True; rr(1)(1) <= '1';
elsif (rg(0)(0)='1' and rg(1)(1)='1' and rg(2)(2)='0') and p(2)(2)='0' then nextstate <= 5; turn <= True; rr(2)(2) <= '1';
--- /
elsif (rg(2)(0)='0' and rg(1)(1)='1' and rg(0)(2)='1') and p(2)(0)='0' then nextstate <= 5; turn <= True; rr(2)(0) <= '1';
elsif (rg(2)(0)='1' and rg(1)(1)='0' and rg(0)(2)='1') and p(1)(1)='0' then nextstate <= 5; turn <= True; rr(1)(1) <= '1';
elsif (rg(2)(0)='1' and rg(1)(1)='1' and rg(0)(2)='0') and p(0)(2)='0' then nextstate <= 5; turn <= True; rr(0)(2) <= '1';
```

雙開

```
-----------------------------------------------------------------------
--for best position
elsif (l1+l4+l7)>1 and p(0)(0)='0' then nextstate <= 5; turn <= True; rr(0)(0) <= '1'; --nex
elsif (l1+l5)>1 and p(0)(1)='0' then nextstate <= 5; turn <= True; rr(0)(1) <= '1';
elsif (l1+l6+l8)>1 and p(0)(2)='0' then  nextstate <= 5; turn <= True; rr(0)(2) <= '1';
elsif (l2+l4)>1 and p(1)(0)='0' then  nextstate <= 5; turn <= True; rr(1)(0) <= '1';
elsif (l2+l5+l7+l8)>1 and p(1)(1)='0' then  nextstate <= 5; turn <= True; rr(1)(1) <= '1';
elsif (l2+l6)>1 and p(1)(2)='0' then  nextstate <= 5; turn <= True; rr(1)(2) <= '1';
elsif (l3+l4+l8)>1 and p(2)(0)='0' then  nextstate <= 5; turn <= True; rr(2)(0) <= '1';
elsif (l3+l5)>1 and p(2)(1)='0' then  nextstate <= 5; turn <= True; rr(2)(1) <= '1';
elsif (l3+l6+l7)>1 and p(2)(2)='0' then  nextstate <= 5; turn <= True; rr(2)(2) <= '1';
```

隨意下

```
-----------------------------------------------------------------------
--arbitrary
elsif (p(0)(0)='0') then nextstate <= 5; turn<=true; rr(0)(0) <= '1';
elsif (p(0)(1)='0') then nextstate <= 5; turn<=true; rr(0)(1) <= '1';
elsif (p(0)(2)='0') then nextstate <= 5; turn<=true; rr(0)(2) <= '1';

elsif (p(1)(0)='0') then nextstate <= 5; turn<=true; rr(1)(0) <= '1';
elsif (p(1)(1)='0') then nextstate <= 5; turn<=true; rr(1)(1) <= '1';
elsif (p(1)(2)='0') then nextstate <= 5; turn<=true; rr(1)(2) <= '1';

elsif (p(2)(0)='0') then nextstate <= 5; turn<=true; rr(2)(0) <= '1';
elsif (p(2)(1)='0') then nextstate <= 5; turn<=true; rr(2)(1) <= '1';
elsif (p(2)(2)='0') then nextstate <= 5; turn<=true; rr(2)(2) <= '1';
else nextstate <= 4;
end if;
```