Conway’s Game of Life

Author: B1030400012 謝承翰

規則

- 細胞只有死亡或存活
- 若細胞存活
  - 當周圍存活細胞 <2, 死亡
  - 當周圍存活細胞為 2或3, 繼續存活
  - 當周圍存活細胞 >3, 死亡
- 若細胞死亡
  - 當周圍存活細胞 =3, 開始存活

程式實現

Cell物件屬性:

- (x, y)座標
- 目前狀態
- 下個狀態

Cell物件方法:

- draw
- detect (決定下個狀態)
- transform (狀態轉移)

程式實現

改變下個狀態

轉移至下個狀態

```
def detect(self, n):
        match n:
                case 0|1:
                        self.nexAlive = False
                        return
                case 2|3:
                        if not self.isalive and n == 3:
                                self.nexAlive = True
                        if self.isalive:
                                self.nexAlive = self.isalive
                        return
                case 4|5|6|7|8:
                        self.nexAlive = False
                        return

def transform(self, canv):
        if self.isalive == self.nexAlive and self.drawed:
                return
        self.isalive = self.nexAlive
        self.draw(canv)
```

程式實現

判斷周圍存活細胞的數量

決定下個轉移狀態

轉移狀態

```
shift = [(-1,-1),(-1,0),(-1,1),(0,-1), (0,1),(1,-1),(1,0),(1,1)]

def run(Game, canv, width, height, dt = 0.01):
        for i in range(height):
                for j in range(width):
                        n = 0
                        for di, dj in shift:
                                nexi = i + di
                                nexj = j + dj
                                if nexi < 0 or nexi >= height:
                                        continue
                                if nexj < 0 or nexj >= width:
                                        continue
                                if Game[nexi][nexj].isalive:
                                        n += 1
                        Game[i][j].detect(n)

        for i in range(height):
                for j in range(width):
                        Game[i][j].transform(canv)

        canv.after(int(dt * 1000), run, Game, canv, width, height)
```

最終結果

有趣的小知識

Conway’s Game of Life是Turing complete

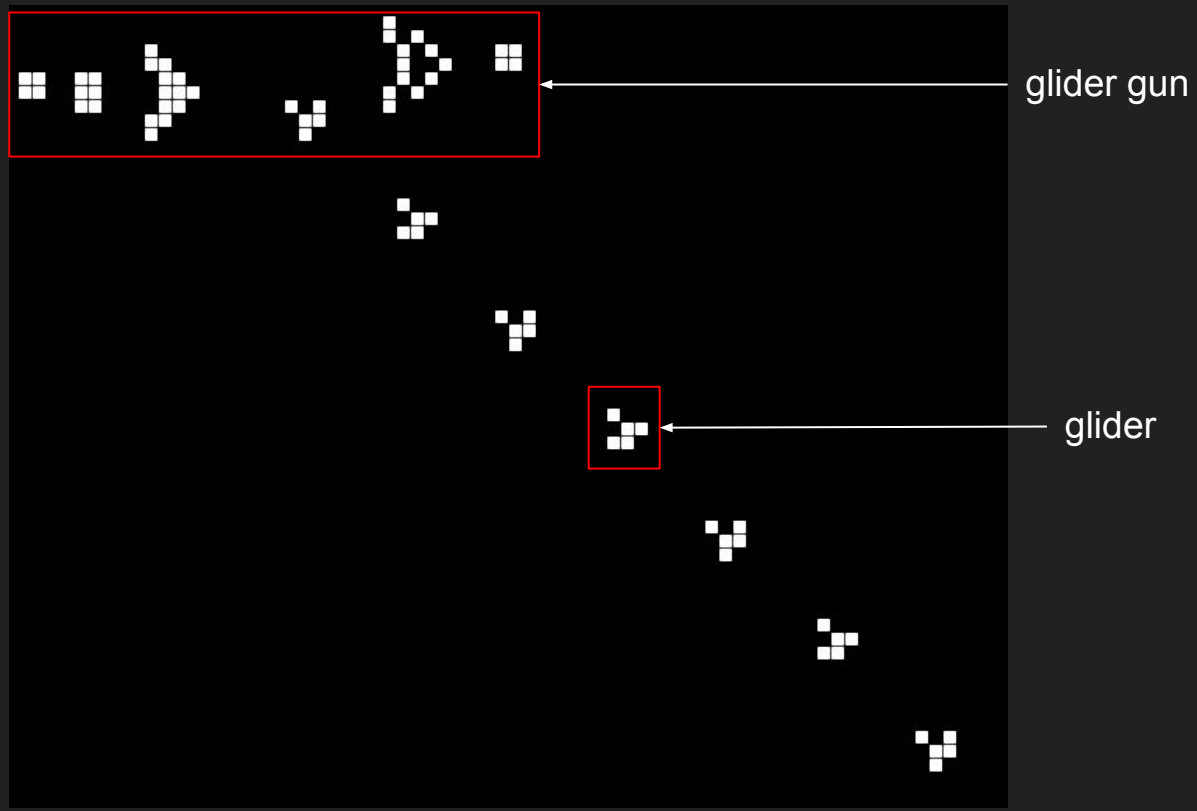
什麼是Turing Complete?

最簡單的解答: 對一個可計算問題, 對於任意輸入, 只要保證有答案, 那一個Turing Complete的系統就一定可以算出結果。

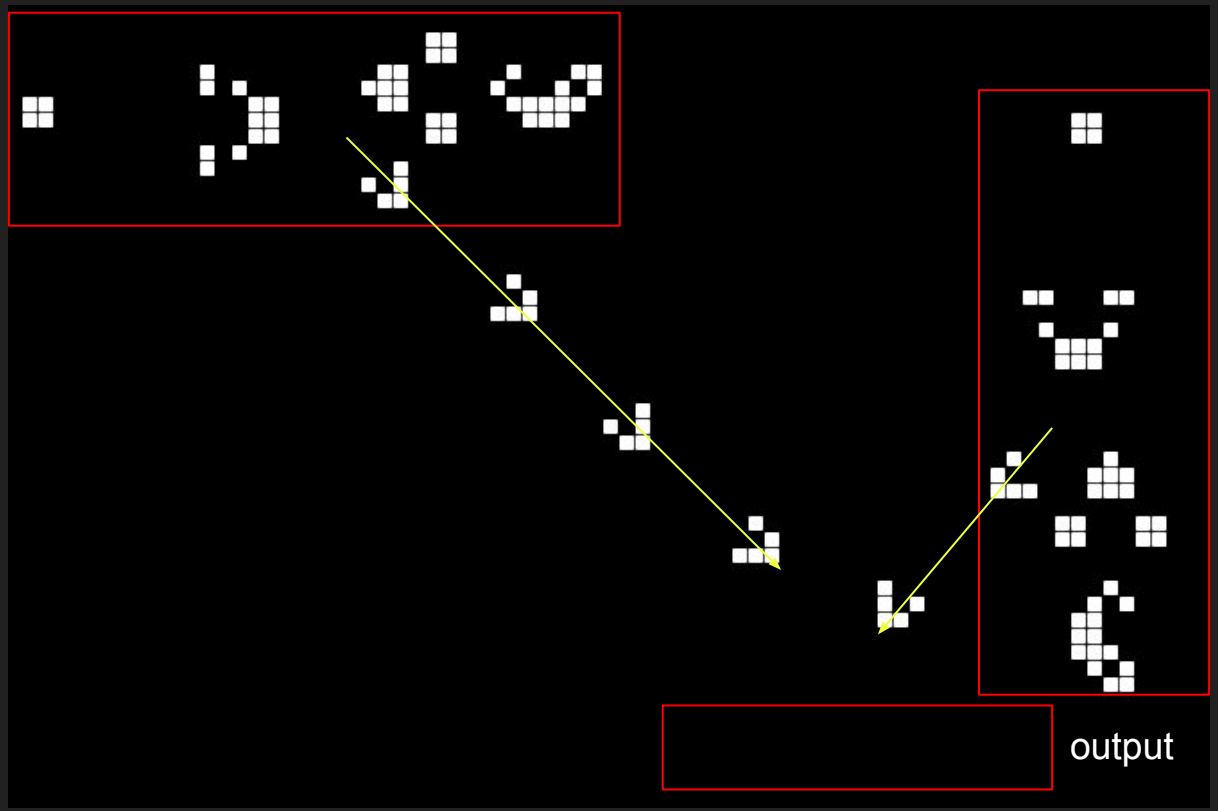
例子: general-purpose的程式語言(C/C++, Java, Python等)、通用型電腦.....

證明Conway’s Game of Life是Turing Complete

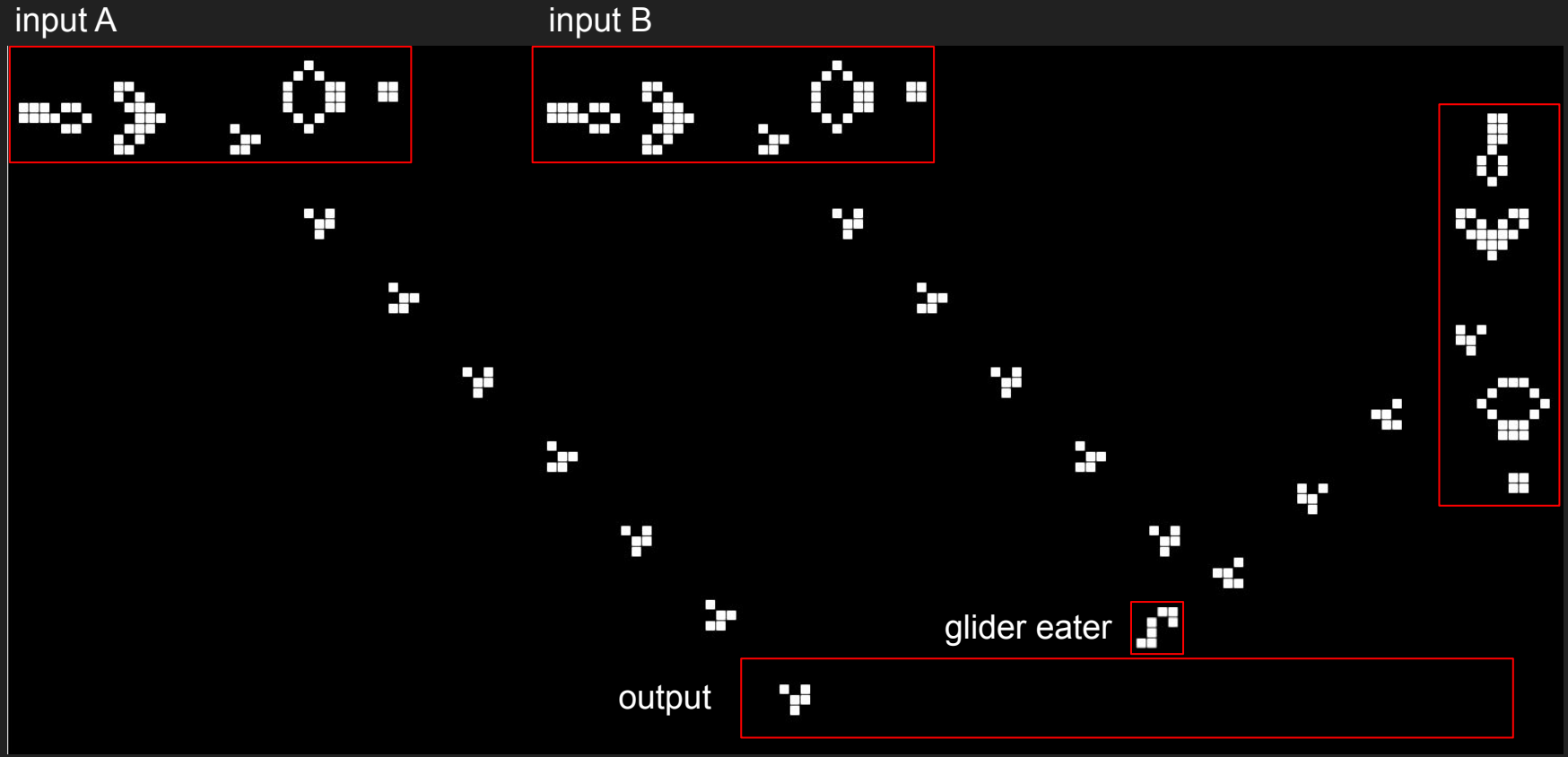
Signal



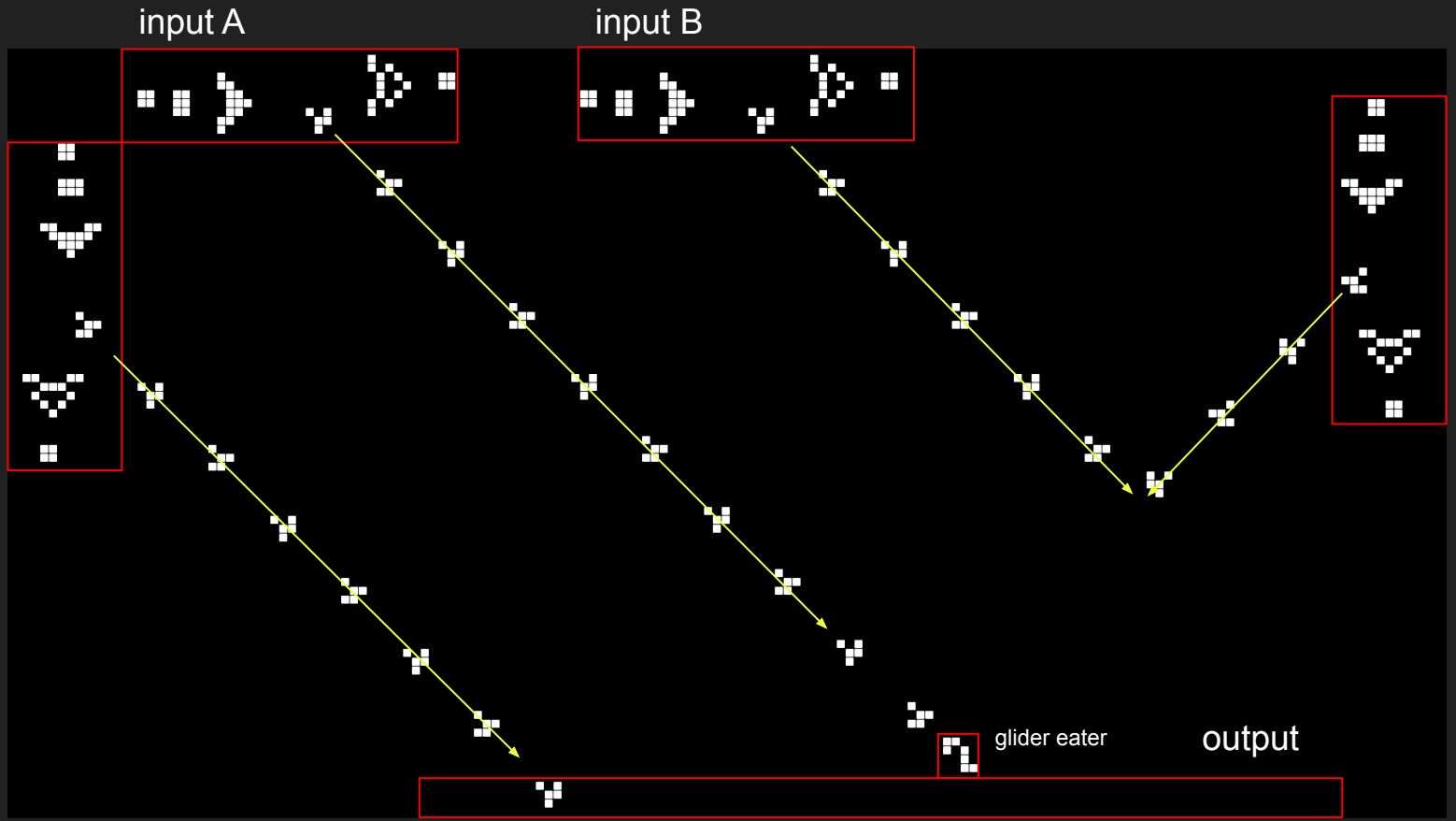
NOT Gate



AND gate



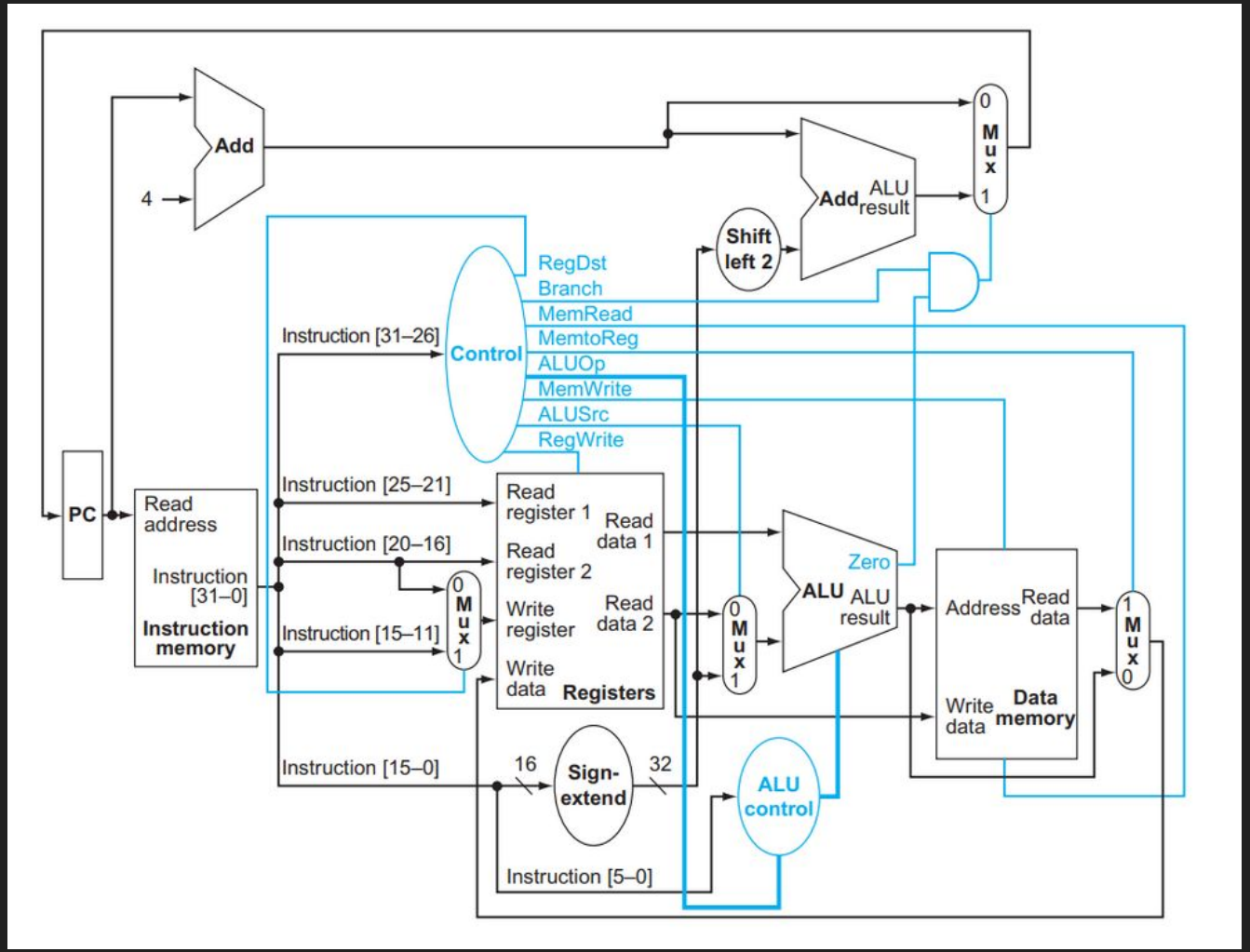
OR gate



證明Conway’s Game of Life是Turing Complete

將這些Logic gate組合起來，就有Latch、Flip-Flops、register、adder、multiplexor、ALU、CU。

然後再把signal接好，就有一台電腦出現在Conway’s Game of Life。


Add
4
0
M
u
x
1
Add
ALU result
Shift left 2
RegDst
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Control
Instruction [31–26]
PC
Read address
Instruction [31–0]
Instruction memory
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
Instruction [15–0]
Instruction [5–0]
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
Zero
ALU
ALU result
Address
Read data
Write data
Data memory
16
Sign-extend
32
ALU control

Life in Life

報告結束