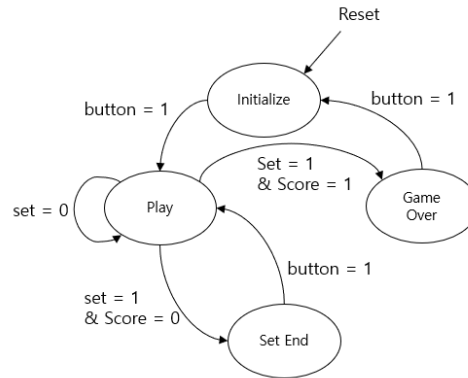


Pong Game Control Module

i. State Diagram



ii. State assignment & Transition Table

State	XY
Initialize	00
Play	01
Set End	11
Game Over	10

Present State		Inputs			Next State		Output	
State	XY	B	Set	Score	Next	XY+	S	G
Initialize	00	0	X	X	Initialize	00	1	1
		1	X	X	Play	01	1	1
Play	01	X	0	X	Play	01	0	0
		X	1	0	Set End	11	0	0
		X	1	1	Game Over	10	0	0
Set End	11	1	X	X	Play	01	1	0
		0	X	X	Set End	11	1	0
Game Over	10	1	X	X	Initialize	00	0	1
		0	X	X	Game Over	10	0	1

Input: button, set_over, score

(Score: 점수가 7점이 될 경우 1이 되도록 Adder의 c_out을 이용해 구성)

Output: set reset(S: paddle과 공 위치/속도 초기화), game reset(G: score 초기화)

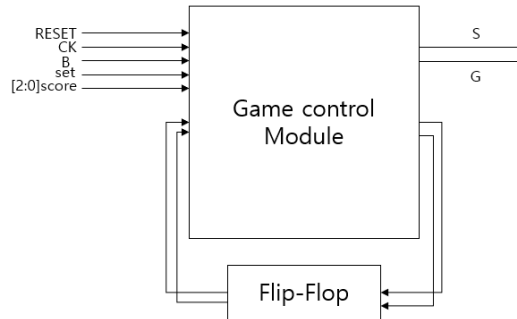
iii. Flip-Flop Input Equation and Implementation

$$X+ = X \& B + X' \& Y \& \text{Set}$$

$$Y+ = X' \& Y' \& B + X' \& Y \& \text{Set} \& \text{Score}$$

$$S = (X \wedge Y)'$$

$$G = Y'$$



Game Control Module은 위의 diagram과 같이 대략적으로 구성할 것이다. Game Control Module의 input으로는 RESET, CK, B, set, [2:0]score가 있다. 현재 구성은 1인용으로 생각하고 구성하였다.

flip_flop.v, adder.v, game_control.v 세 개의 파일을 통해 구성하였다. 'adder.v'에서는 half Adder, full Adder를 만든 뒤 [2:0]score를 입력 받아 계산하는 'threebitAdder'를 구성하였다. 'threebitAdder'에서 score와 1을 더하고, C_OUT을 통해 점수가 7이상인지 판별한다.

'flip_flop.v'에서는 D flip-flop을 구현하고 이를 사용했다. 아래는 game_control module의 Code와 Schematic이다.

```
module game_control(
    input RESET_N, input CK,
    input [1:0]game_state, input B, input set, input [2:0]score,
    output X, output X_, output Y, output Y_, output S, output G
);
    wire [2:0]trash;
    wire score_;
    threebitAdder(score, trash, score_);

    edgeTriggeredDFF X_state(RESET_N, game_state[1]&B | (~game_state[1])&game_state[0]&set, CK, X, X_);
    edgeTriggeredDFF Y_state(RESET_N, (~game_state[1])&(~game_state[0])&B | (~game_state[1])&game_state[0]&set&score_, CK, Y, Y_);

    assign S = ~(game_state[0]^game_state[1]);
    assign G = ~game_state[0];
endmodule
```

