

Assignment #3

Due date: 4/29

Simple State Machine

- 눈치게임 (Hunch Game) Verilog module을 구현하시오.



I/O Interface

- 1-bit input CLK (clock)
- 1-bit input RST (active high, reset)
- 1-bit input A, B, C (0: sit, 1: stand)
- 3-bit outputs WINNER_DISP[2:0] – winner display
- 1-bit output A_DISP, B_DISP, C_DISP – standing display

Rules

1. 아무도 일어나지 않는 경우 계속 진행
2. 동시에 2명이 일어나면 남은 사람 Win
3. 혼자 앉아 있는 상태인 사람이 생기면 그 사람을 제외한 두 사람 Win
4. 세 명이 동시에 일어나면 Draw
5. 한번 일어난 사람의 다음 입력은 무시
 - Ex) 처음 A만 일어났을 때 (A: 1, B: 0, C: 0) 다음 입력에서 A의 신호는 고려 x
6. 승부가 결정 나면 다시 처음으로 step1으로 돌아가 게임 시작

Displays

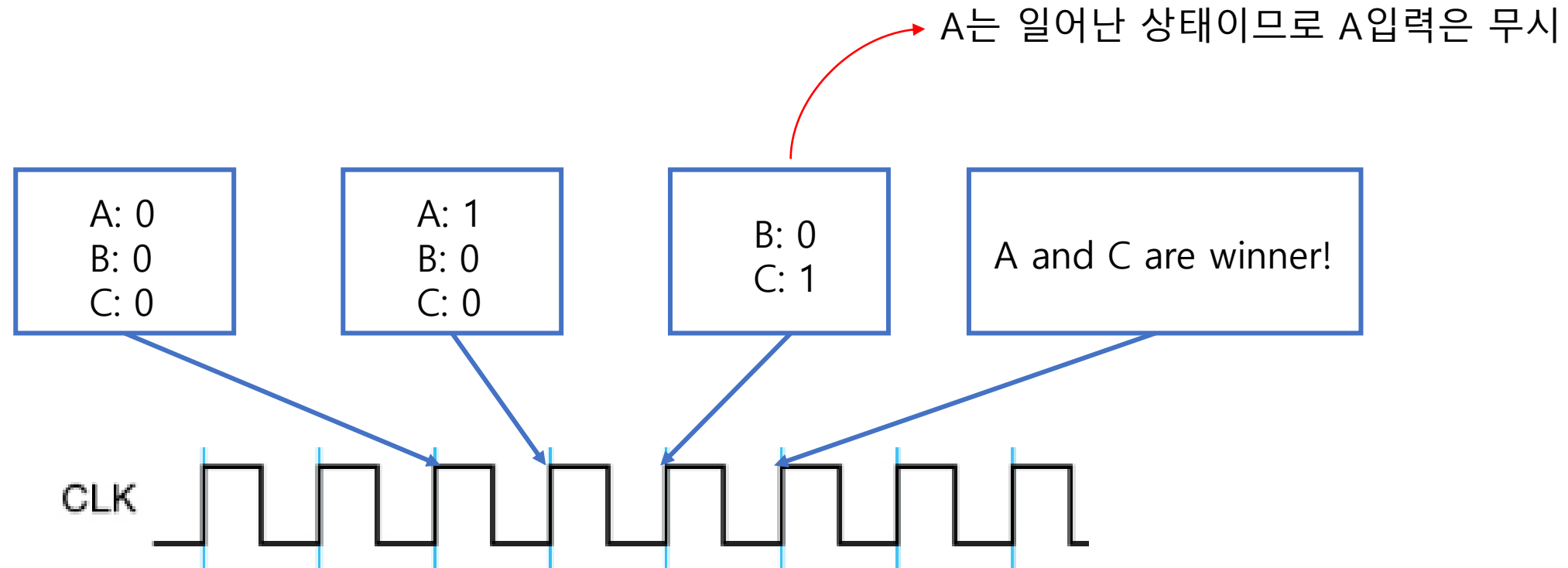
- A_DISP, B_DISP, C_DISP: 현재 일어난 상태 (0: sit, 1: stand)
- WINNER_DISP[2:0]:

WINNER	0	1	2
A	0	0	1
B	0	1	0
A, B	0	1	1
C	1	0	0
C, A	1	0	1
C, B	1	1	0
DRAW	1	1	1

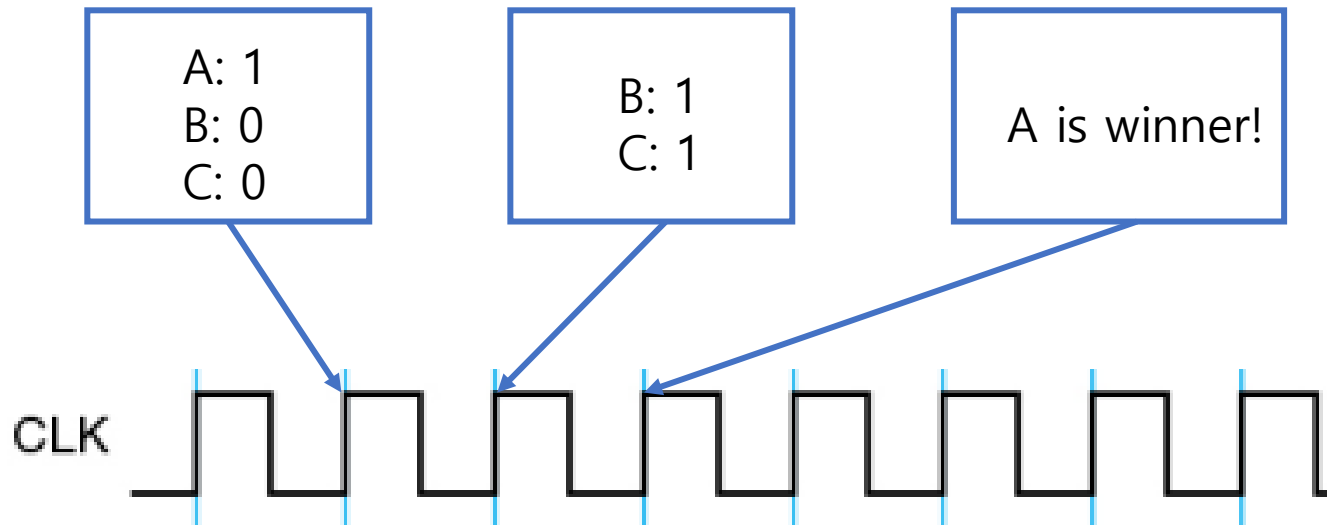
States Name

NAME	DESCRIPTION
S_INIT	Initial state
S_AS	Standing only A state
S_BS	Standing only B state
S_CS	Standing only C state
S_AW	Winner is A
S_BW	Winner is B
S_CW	Winner is C
S_ABW	Winners are A and B
S_ACW	Winners are A and C
S_BCW	Winners are B and C
S_DRAW	Draw

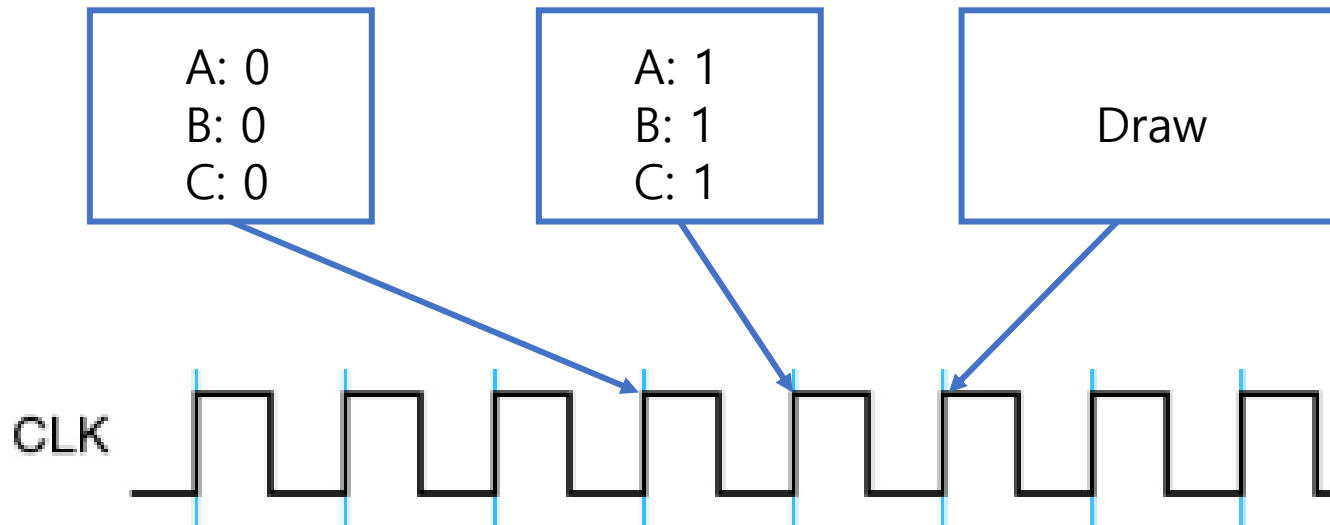
Game 진행과정 (예제 1)



Game 진행과정 (예제 2)



Game 진행과정 (예제 3)



Requirement

- Design a game module (HGame.v)
 - Input
 - CLK, RST
 - A, B, C
 - Output
 - WINNER_DISP[2:0]
 - A_DISP, B_DISP, C_DISP
 - 필요한 요소
 - Main FSM module (next state logic)
 - Decoder module (output logic)
- Design a testbench module (HGameTb.v)
 - 테스트 벤치는 가능한 경우의 수를 모두 테스트 할 수 있어야 합니다.

Requirement

- Design a display module (Display.v)
 - Input
 - CLK, RST
 - WINNER_DISP[2:0]
 - A_DISP, B_DISP, C_DISP
 - 현재 상태 및 게임 결과 출력

승부가 결정되지 않은 경우

[#<clock tick>] A: <sit or stand> B: <sit or stand> C: <sit or stand>

승부가 결정된 경우

[#<clock tick>] A: <sit or stand> B: <sit or stand> C: <sit or stand> Result: Winner is <winner>

[#<clock tick>] A: <sit or stand> B: <sit or stand> C: <sit or stand> Result: Draw