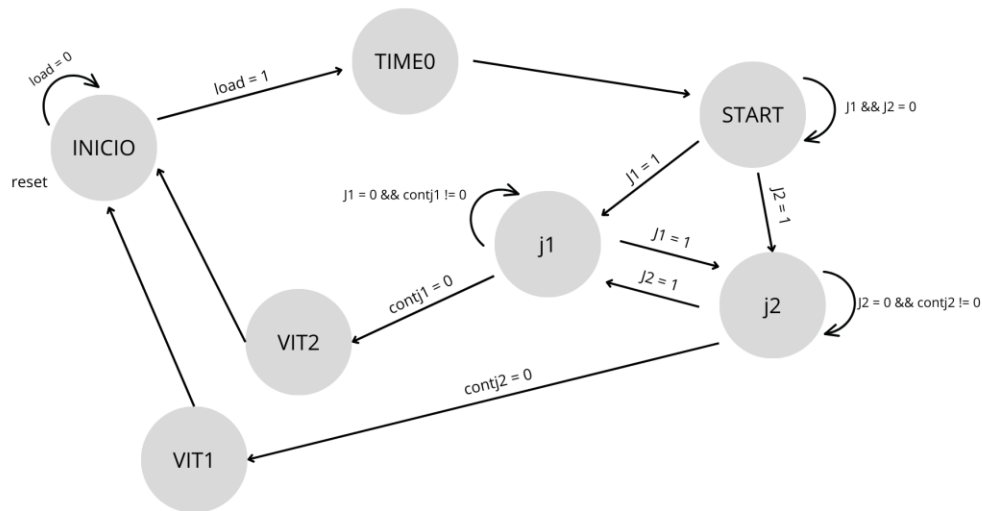


PARTE 1



| EA | winJ1 | winJ2 | CONDIÇÃO | PE |
|--------|-------|-------|---------------|--------|
| INICIO | 0 | 0 | load = 1 | TIME0 |
| TIME0 | 0 | 0 | - | START |
| START | 0 | 0 | se J1 = 1 | j1 |
| START | 0 | 0 | se J2 = 1 | j2 |
| START | 0 | 0 | - | START |
| j1 | 0 | 0 | se J1 = 1 | j2 |
| j1 | 0 | 0 | se J1 = 0 | j1 |
| j1 | 0 | 0 | se contj1 = 0 | VIT2 |
| j2 | 0 | 0 | se J2 = 1 | j1 |
| j2 | 0 | 0 | se J2 = 0 | j2 |
| j2 | 0 | 0 | se contj2 = 0 | VIT1 |
| VIT1 | 1 | 0 | - | INICIO |
| VIT2 | 0 | 1 | - | INICIO |

A máquina de estado (FSM) que controla o circuito do relógio de xadrez segue o seguinte funcionamento:

O ponto de partida é no estágio “INICIO”, quando ‘load = 0’ (sinal de transição) permanece no estado atual, entretanto ao ativar a condição ‘load = 1’, o próximo estágio passa a ser “TIME0”, responsável por inicializar ‘init_time’. Feito isso, o cronômetro atribui o estágio “START” com os sinais de ‘J1’ e ‘J2’ desligados. Conforme um jogador (seja J1 = 1 ou J2 = 1) iniciar sua jogada seu cronômetro começa a regredir.

Quando for a vez do “j1”, o relógio permanece decrescendo (J1 = 0 && contj1 != 0) e a troca só ocorre quando ele apertar o botão para interromper a contagem e habilitar a regressão do cronômetro do seu adversário (J1 = 1), passando para o estágio “j2”. Assim, o estado atual permanece (J2 = 0 && contj2 != 0) até que ele aperte o botão novamente (J2 = 1).

No momento que o relógio (contj1 = 0) alcançar o ponto zero, o estágio “VIT2” entra em ativação, declarando vitória ao segundo jogador. Entretanto, se ‘contj2 = 0’, o estágio passa a ser “VIT1”, e a vitória é do primeiro jogador. A partida se encerra após a vitória (seja VIT1 ou VIT2) de algum jogador, e para reiniciar (reset) o cronômetro o próximo estado é novamente “INICIO”.

PARTE 2

Exemplo de uma partida (tb.vhdl):

| | |
|----------------------------------------------------------|-------------------------------------------------------|
| (t => 4, load=>'1', init=>x"20", j1=>'0', j2 =>'0') | partida de 20 minutos |
| (t => 10, load=>'0', init=>x"00", j1=>'1', j2 =>'0') | jogador 1 começa a partida 10 ciclos depois |
| (t => 60, load=>'0', init=>x"00", j1=>'1', j2 =>'0') | jogador 1 joga por 60 ciclos (1 min) contj1: 19min |
| (t => 120, load=>'0', init=>x"00", j1=>'0', j2 =>'1') | jogador 2 joga por 120 ciclos (2 min) contj2: 18min |
| (t => 300, load=>'0', init=>x"00", j1=>'1', j2 =>'0') | jogador 1 joga por 300 ciclos (5 min) contj1: 14min |
| (t => 360, load=>'0', init=>x"00", j1=>'0', j2 =>'1') | jogador 2 joga por 360 ciclos (6 min) contj2: 12min |
| (t => 180, load=>'0', init=>x"00", j1=>'1', j2 =>'0') | jogador 1 joga por 180 ciclos (3 min) contj1: 11min |
| (t => 60, load=>'0', init=>x"00", j1=>'0', j2 =>'1') | jogador 2 joga por 60 ciclos (1 min) contj2: 11min |
| (t => 240, load=>'0', init=>x"00", j1=>'1', j2 =>'0') | jogador 1 joga por 240 ciclos (4 min) contj1: 7min |
| (t => 420, load=>'0', init=>x"00", j1=>'0', j2 =>'1') | jogador 2 joga por 420 ciclos (7 min) contj2: 4min |
| (t => 300, load=>'0', init=>x"00", j1=>'1', j2 =>'0') | jogador 1 joga por 300 ciclos (5 min) contj1: 2 min |
| (t => 180, load=>'0', init=>x"00", j1=>'0', j2 =>'1') | jogador 2 joga por ciclos (180 min) contj2: 1 min |
| (t => 10000, load=>'0', init=>x"00", j1=>'0', j2=>'0')); | último comando coloca todos os valores em zero |