

PROJETO 2 - TAREFAS DA SEMANA 1

Vinícius de Barros Silva

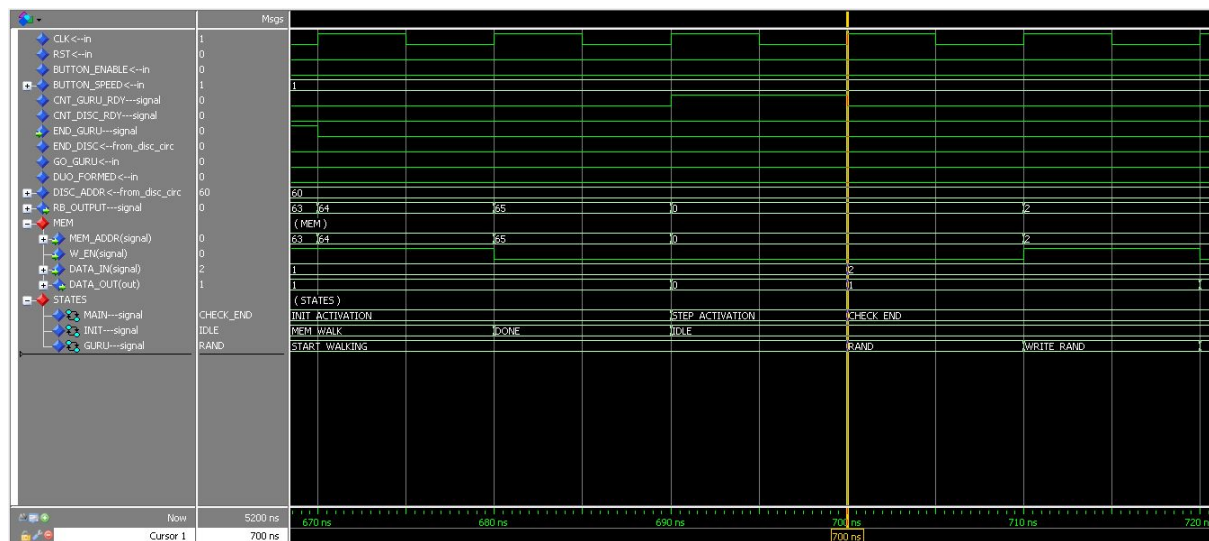
10335913

a) Para a FSM Guru

1c) captura: imagem do Wave onde fique evidente os dois estados de espera da FSM Guru e de suas relações com os sinais respectivos que os liberam

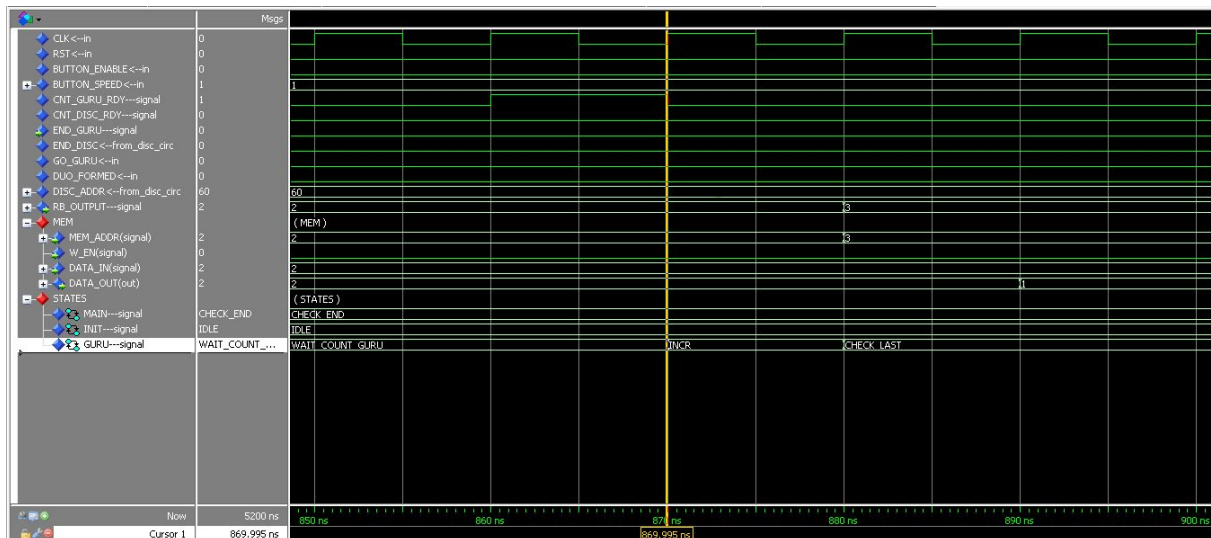
1t) texto: para cada evento acima, explicar o ocorrido de acordo com a interação do circuito com o ambiente externo.

START WALKING



Após a FSM INIT terminar de percorrer a memória e chegar ao estado DONE , a FSM MAIN passa ao estado STEP_ACTIVATION onde o sinal fsm_sg_start = 1 acontece e a FSM GURU passa do estado de espera START_WALKING para RAND. Durante esse processo o usuário não teve contato com o jogo, apenas o hardware está fazendo a inicialização para interação com o jogador.

WAIT_COUNT_GURU

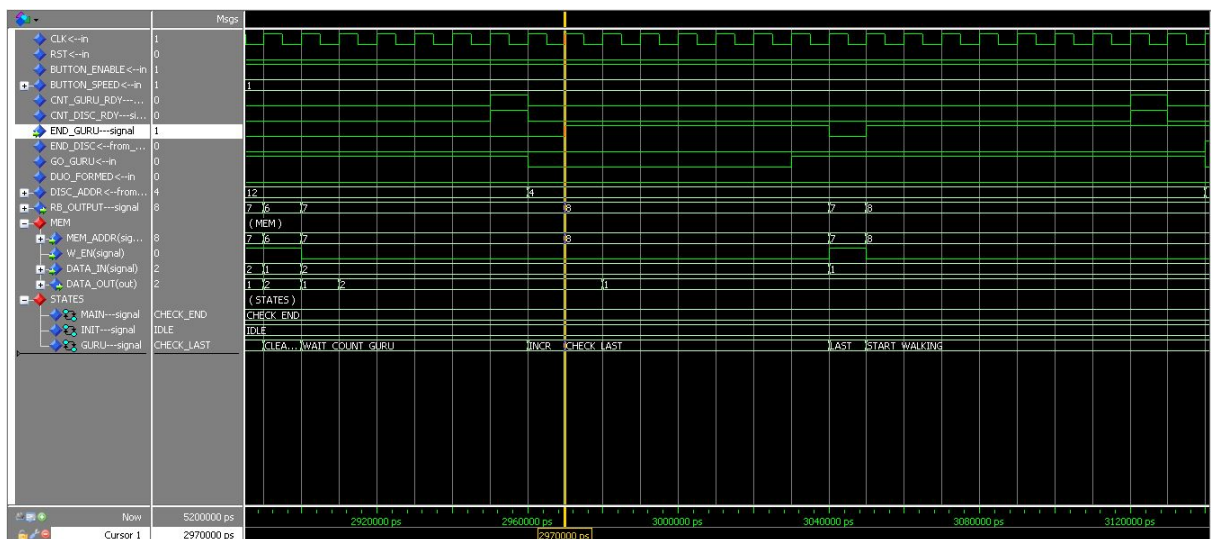


Observa-se na imagem a FSM GURU no estado WAIT_COUNT_GURU estado em que a máquina espera pelo sinal de disparo cnt_guru_rdy = '1' (sinal vindo de STEP_COUNTER para contar ciclos de relógio e controlar os passos), quando isso ocorre, no ciclo seguinte a FSM GURU vai ao estado INCR. O usuário interage com o button handler indicando o disparo do discípulo apenas, então nesse momento o usuário observa o posicionamento do guru na tela.

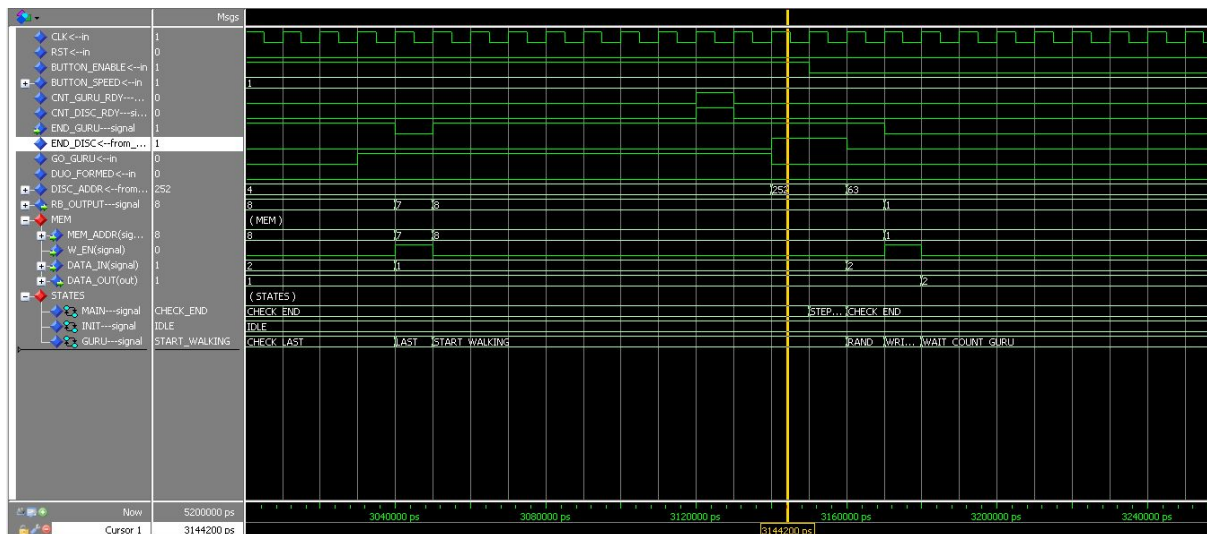
2c) captura: imagem dos eventos em SIM-2 e SIM-3 com os estados de finalização da iteração e os sinais que os determinam.

2t) texto: Inclua o trecho do código VHDL da FSM Main onde estes eventos são determinados e explicar o ocorrido

SIM-2 :



Momento em que END_GURU = 1 e a FSM GURU passa ao estado LAST indicando que houve overflow, isto é o guru passou pela borda do tabuleiro.



Momento em que END_DISC = 1 e a FSM DISCÍPULO chega ao fim da rodada após passar pela casa 4. Vale notar que nessa simulação o guru termina antes do discípulo, então o guru aguarda o discípulo terminar para uma nova rodada.

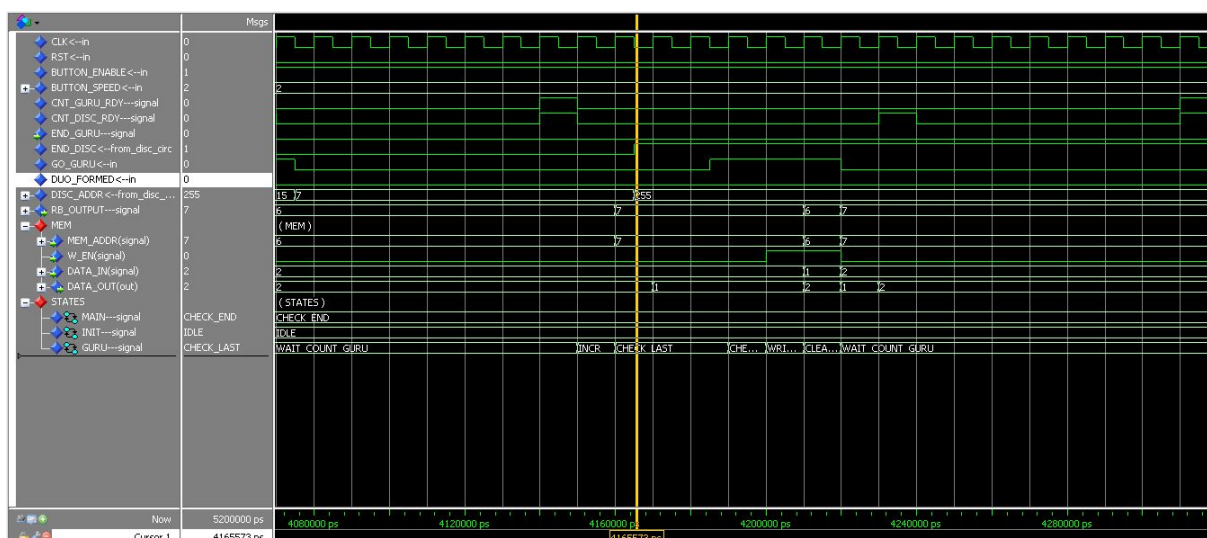
```

when CHECK_END      => if (end_of_guru = '0') OR (end_of_disc = '0') then
    NEXT_STATE <= CHECK_END;
    elsif (end_of_guru = '1') AND (end_of_disc = '1') AND (duo_formed = '0') then
    NEXT_STATE <= STEP_ACTIVATION;
    else
    ---- (end_of_guru = '1') AND (end_of_disc = '1') AND (duo_formed = '1')
    NEXT_STATE <= HIT_POINT;
    end if;

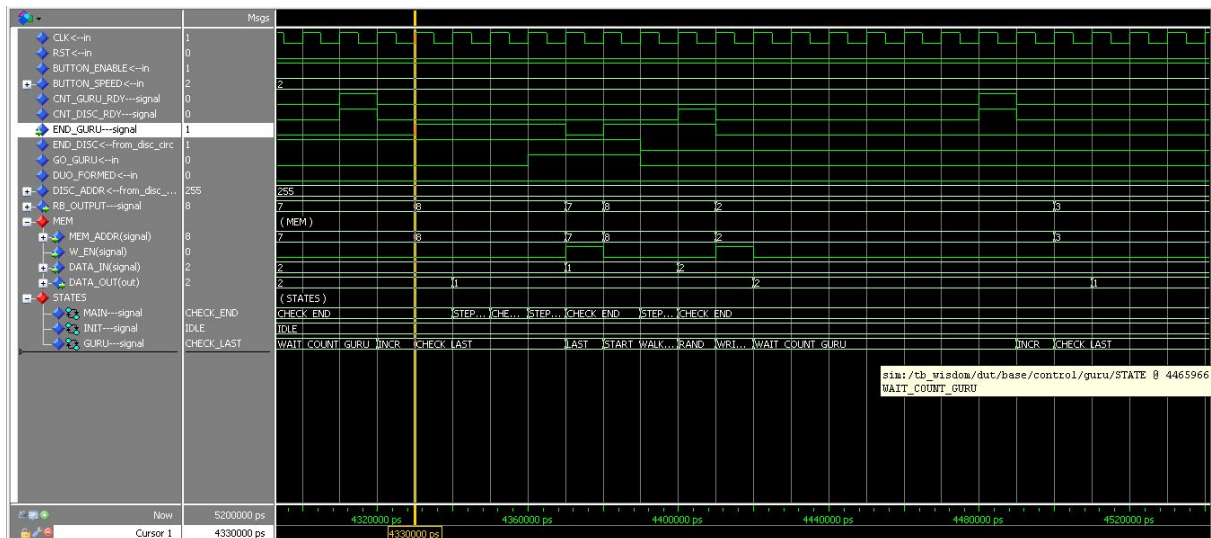
```

Trecho de código que determina o término da rodada, para esse caso o guru termina e assim indica, a fsm aguarda enquanto o discípulo não termina, até que o discípulo envia o sinal de término e duo não formado, assim a FSM MAIN troca do estado CHECK_END para STEP_ACTIVATION.

SIM-3:



Momento em que o discípulo chega ao final e o sinal END_DISC = '1', porém o guru ainda não terminou de passar pelo tabuleiro, pode-se notar que o guru estava na casa 6 enquanto o discípulo estava na casa 7, quando o discípulo sai da casa 7 só aí o guru vai para casa 7, não havendo encontro por tanto e a FSM aguarda o guru terminar para uma nova partida.



Nesse momento pode-se notar o guru chegando a casa 8 e indicando termino com o sinal `END_GURU = 1`, assim a FSM MAIN indica o término da partida e entra no estado `STEP_ACTIVATION` para uma próxima rodada.

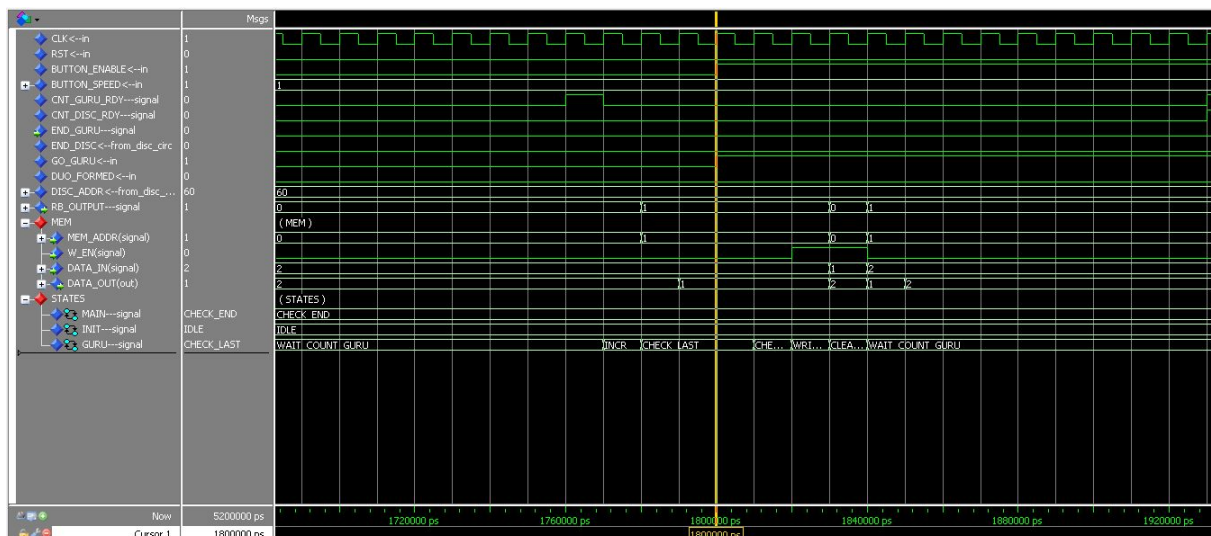
```
when CHECK_END => if (end_of_guru = '0') OR (end_of_disc = '0') then
    NEXT_STATE <= CHECK_END;
elsif (end_of_guru = '1') AND (end_of_disc = '1') AND (duo_formed = '0') then
    NEXT_STATE <= STEP_ACTIVATION;
else
    ---- (end_of_guru = '1') AND (end_of_disc = '1') AND (duo_formed = '1')
    NEXT_STATE <= HIT_POINT;
end if;
```

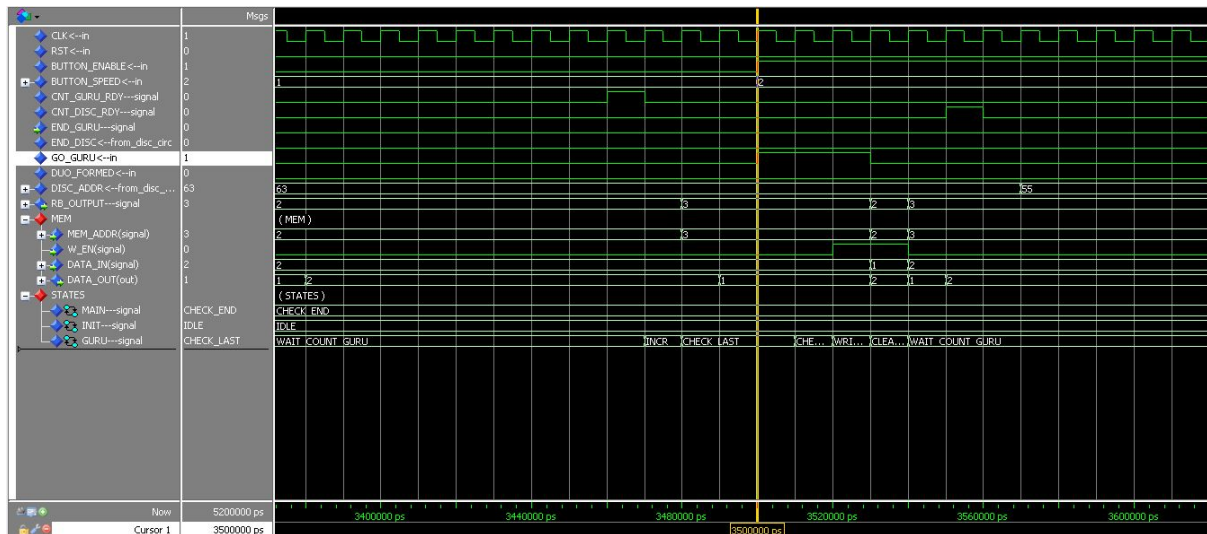
O trecho responsável pelo término da SIM3 é o mesmo responsável por organizar o término da SIM2. Onda a FSM recebe o sinal `end_of_disc = 1` e aguarda até que o sinal `end_of_guru = 1` com duo não formado.

b) Para a interação com o Circuit Disciple

3c) captura: imagem dos eventos em SIM-2 e SIM-3 onde o sinal enable é ativado. Indicar o primeiro efeito que isto causa nos sinais provenientes do circuito do discípulo.

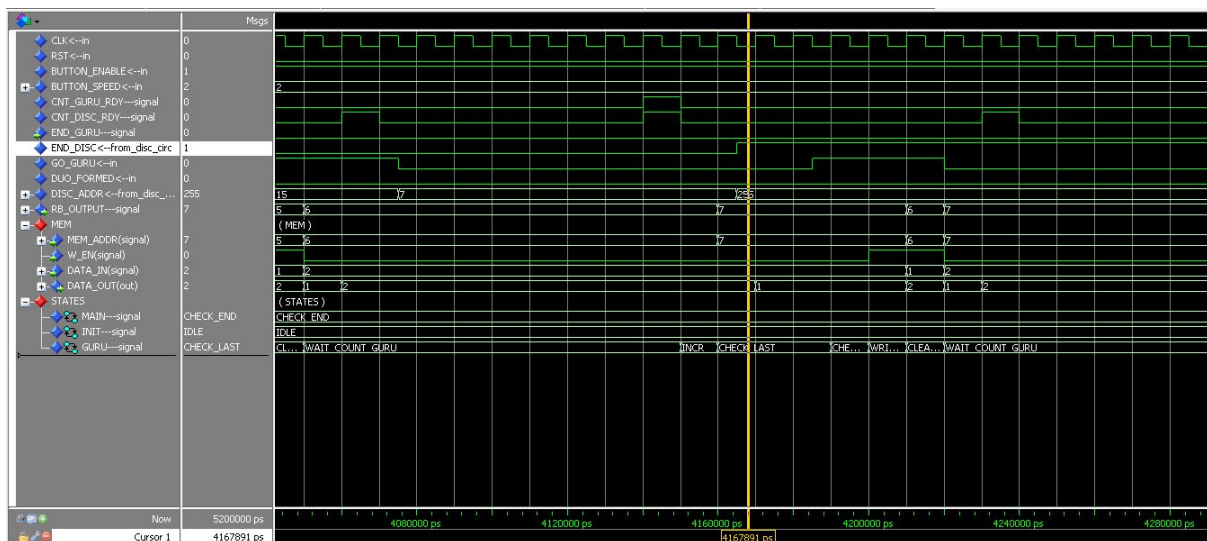
3t) Explique a correlação entre o enable e o sinal do discípulo.



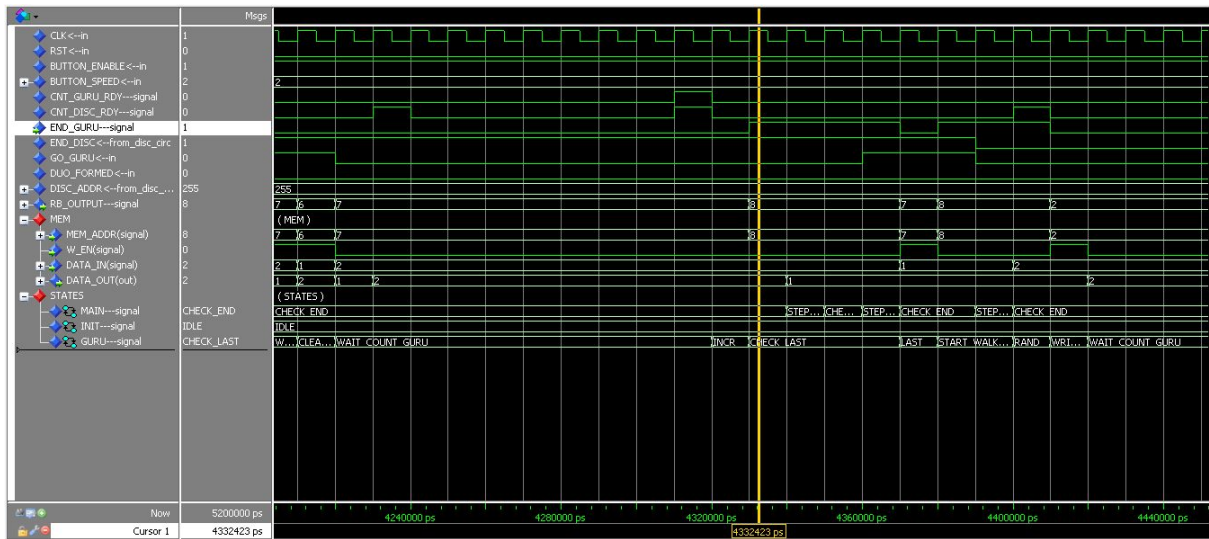


Seguem as imagens onde ocorre ativação do enable na SIM-2 e na SIM-3 respectivamente. O sinal de enable é ativado quando o jogador pressiona algum dos botões de velocidade, junto ao enable a velocidade é atualizada, paralelamente ocorre o disparo do sinal go_guru. Já quando o discípulo termina e dá o sinal end_dis = 1, no ciclo seguinte enable = 0.

4c) captura: imagem do evento SIM-3, com a chegada do sinal end_of_disciple e a geração do end_of_guru. Identifique o valor dos respectivos endereços na memória que causaram a ativação destes sinais.



Chegada do sinal end_od_disc, causado após endereço 255 ocorrer, que ocorre depois do endereço 7, o qual se encontra na borda do tabuleiro para o discípulo.



Geração do sinal `end_of_guru` após o guru ultrapassar da posição 7 para posição 8 (isto é quando acusa posição 8 overflow), indicando o fim do tabuleiro.

4t) Faça um cálculo aproximado do instante que deveria ter ocorrido o enable para que houvesse o encontro entre o guru e o discípulo e em que casa (endereço) isto ocorreria.

O encontro poderia acontecer na casa 7, o discípulo entrou na casa 7 no instante 4075 ns e o guru entrou na casa 7 no instante 4160 ns, portanto houve uma diferença de 85 ns. Tendo em vista que o disparo do discípulo havia acontecido em 3500 ns, para haver o encontro o discípulo deveria ser disparado aproximadamente em 3585 ns.