

Vinícius de Barros Silva
Caio Sobreira Monteiro

10335913
9922314

PSI-3451 --- GUIA DE PREPARAÇÃO DE RELATÓRIO FINAL (PROJETO)
TURMA 2 (TARDE)

Instruções/Informações Gerais (ler com atenção)

De acordo com o Documento 1 do Projeto, este é finalizado com uma sequência de ações, iniciadas no início do semestre, descritas nas Partes A a D. A seguir, apresentamos algumas diretrizes para a confecção dos relatórios e da seleção de arquivos VHDL a serem entregues.

ATENÇÃO-1: Alguns de seus resultados do Projeto-1 serão exigidos dentro do wisdom circuit.

ATENÇÃO-2: Como será solicitada a apresentação cartas de tempos com a presença de sinais internos do projeto, recomenda-se que a simulação no Modelsim seja feita com a opção de “full visibility” (normalmente já realizado pelo NativeLink do Quartus).

ATENÇÃO-3: Para as diversas solicitações de simulação no Modelsim a seguir, as duplas precisarão apresentar explicitamente os sinais específicos, com a resolução adequada, e deverão alterar o arquivo run_sim.txt fornecido para tal.

ATENÇÃO-4: No Wave, todos os valores de sinais compostos (vetores) deverão ser apresentados como unsigned ou hexa, dependendo da necessidade; o formato binário só poderá ser adotado em análises especiais onde a localização de bits específicos é necessária.

ATENÇÃO-5: Recomenda-se às duplas alterarem o arquivo run_sim.text, fornecido junto aos arquivos de teste, criando versões específicas para cada uma das seções do relatório (com os sinais específicos relevantes), para um reúso posterior de fácil identificação.

APRESENTAÇÃO DE RESULTADOS E ARQUIVOS- A seguir, apresentamos uma lista dos resultados a serem apresentados, sendo que algumas partes já servem de gabarito (recomendado) para o relatório. Todos os arquivos anexados **devem estar identificados com a enumeração a seguir**, para não tê-los não considerados.

1. Resultados do projeto do LFSR escolhido pela dupla para o projeto do circuito do discípulo (repetição do Projeto 1)

Nome do aluno cujo número USP é utilizado para obtenção do polinômio característico (incluir aqui): Vinícius de Barros Silva

Número USP utilizado (incluir aqui): 10335913

Número USP mod 2048 em decimal (incluir aqui): 1705

Número USP mod 2048 em binário (incluir aqui): 110 1010 1001

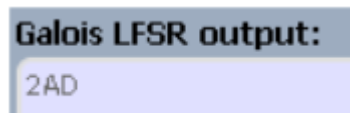
Número USP mod 2048 em hexadecimal (incluir aqui): 6A9

Polinômio característico (incluir aqui):

$$p(x) = x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^4 + x^1 + 1$$

A partir de simulações on-line realizadas com o programa gerador de números de lfsr-galois de leventozturk.com:

a) Inclua a imagem do primeiro número aleatório gerado, junto com a semente e o polinômio característico



b) Copie a tabela do Relatório do Projeto 1, com os 10 valores aleatórios gerados.

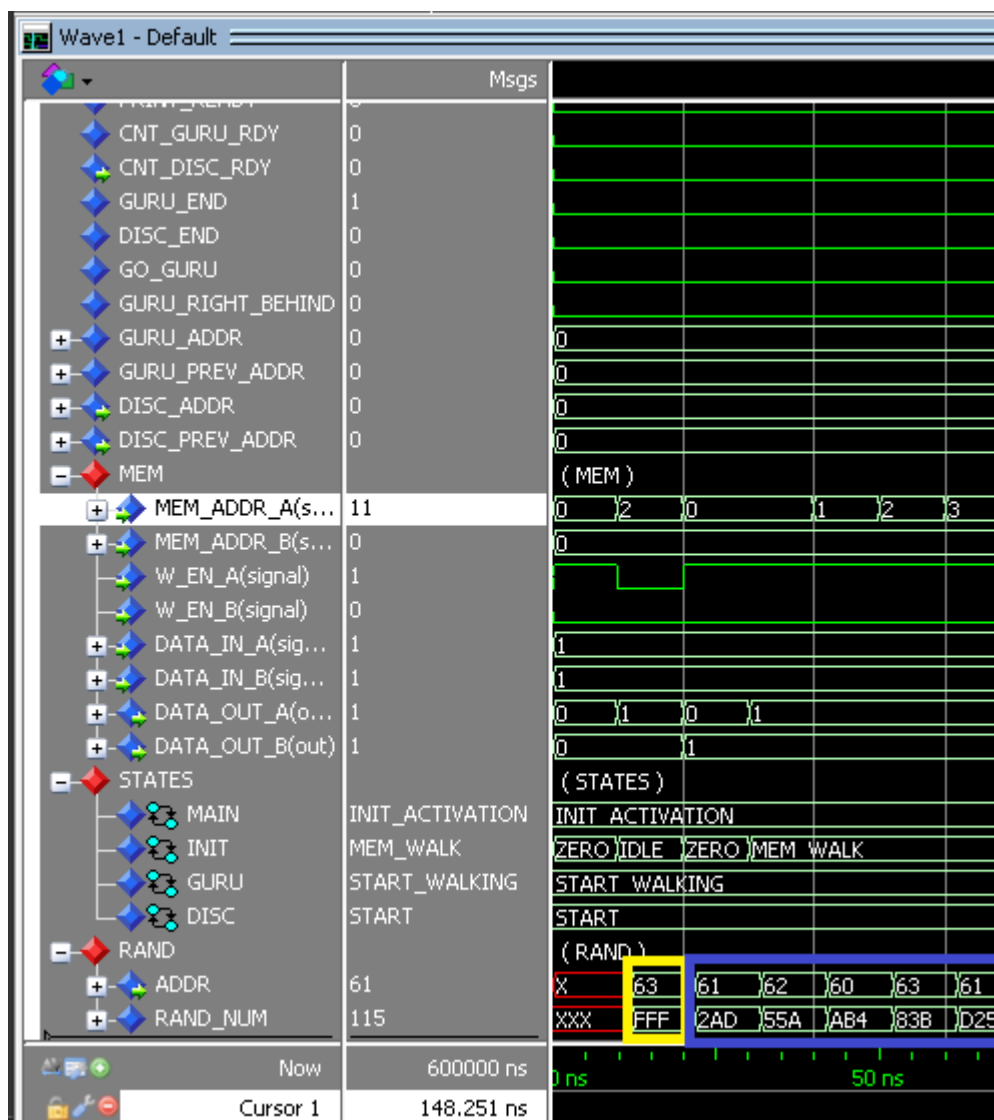
Tabela 1: Exemplo de resultados da simulação (10 ciclos)

No. de ciclos a partir da semente	Saída do LFSR (hex) na simulação software
1	2AD
2	55A
3	AB4
4	83B
5	D25
6	719
7	E32
8	137
9	26E
10	4DC

2. Análise do funcionamento do projeto do LFSR do Disciple Circuit dentro do Wisdom completo a partir de sua descrição VHDL e da sua visualização no Modelsim

- a) Copie a seguir imagem(ns) do ModelSim, evidenciando com anotações o valor da semente (após o reset) e os primeiros 5 valores aleatórios, todos juntos aos correspondentes endereços de memória formados no módulo num_gen.

Observação: Sugere-se alterar o arquivo run_sim.text, para possibilitar a visualização no Wave dos números aleatórios gerados pelo LFSR Galois no Wave do Modelsim, assim como dos correspondentes endereços aleatórios para a memória.



Retângulo amarelo: valor da semente.

Retângulo azul : primeiros 5 valores aleatórios.

- b) Compare os números aleatórios de a) com aqueles obtidos por software on-line (item 1 do Modelo de entrega do projeto 1)

Os 5 primeiros números aleatórios gerados foram :

FFF 2AD 55A AB4 83B D25

Os 5 primeiros números aleatórios gerados por software foram:

Galois LFSR output: 2AD	Galois LFSR output: 55A	Galois LFSR output: AB4
Galois LFSR output: 83B	Galois LFSR output: 719	Galois LFSR output: D25

Como pode ser observado os resultados obtidos com os números gerados aleatoriamente no discipule circuit é condizente com os números gerados via software.

ATENÇÃO: Todos os sinais citados acima devem ser RESSALTADOS na(s) própria(s) imagen(s)

3. Dados das Condições de Simulação Utilizadas para a Dupla (Cenários 7a, b, c e d do Documento da Semana Final)

a) Anexe o arquivo de estímulos do testbench onde a sequência de velocidades do discípulo está programada (denominar como anexo 3a).

b) Anexe o arquivo de mapas resultante da simulação com os estímulos do item a). Não realize edições nele, exceto a identificar a sequência de rodadas para cada cenário+velocidade (denominar como anexo 3b). Caso tenha rodadas extras, também identifique-os para uso no item c).

c) Caso tenham sido configuradas mais de 4 rodadas (para os 4 cenários solicitados), identifique-as de acordo com as anotações do arquivo do item b) e justifique a sua necessidade.

Não foram configuradas mais de 4 rodadas para fazer as situações necessárias.

4. Resultados das simulações do VHDL final (Wisdom completo) e do seu testbench no ModelSim para a condição de "guru termina e discípulo não se mexe".

a) Identifique o cenário de acordo com a anotação do arquivo de mapas do item 3b.

-- Rodada com o botão DE-pressionado e velocidade X1 (não importa - irrelevante) . Deve-se observar apenas a movimentação do GURU

* NEW ROUND *

-----NEXT STEP-----

|.>.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|....^..|

-----NEXT STEP-----

|..>.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|....^..|

-----NEXT STEP-----

|...>....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....^.....|

-----NEXT STEP-----

|....>...|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....^.....|

-----NEXT STEP-----

|....>..|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....^.....|

-----NEXT STEP-----

```
|.....>|
|.....|
|.....|
|.....|
|.....|
|.....|
|.....|
|.....|
|.....^..|
```

-----NEXT STEP-----

```
|.....>|
|.....|
|.....|
|.....|
|.....|
|.....|
|.....|
|.....|
|.....^..|
```

b) Inclua abaixo o trecho da arquitetura do módulo Stimuli_wisdom modificado com os comandos para se chegar ao "guru termina e discípulo não se mexe"

```
-- Rodada com o botao DE-pressionado e velocidade X1 (não importa - irrelevante) . Deve-se observar apenas a movimentacao do GURU
checkWC('0',1);

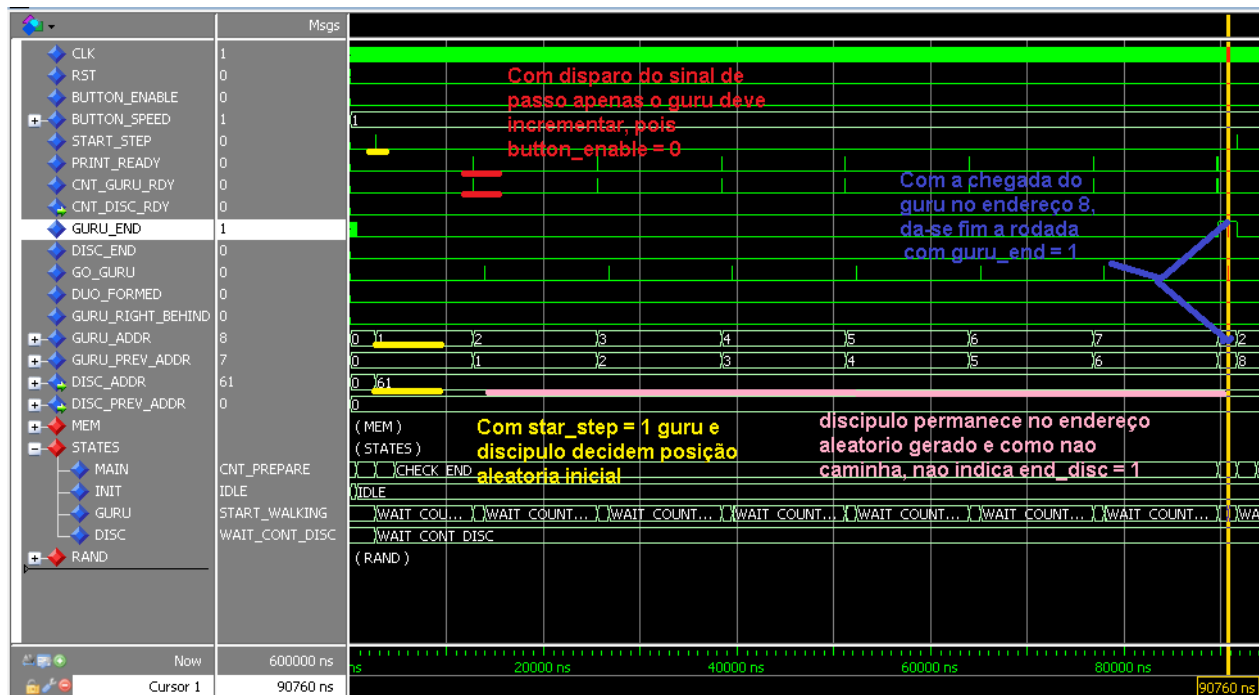
file_open (f_status, l_file, "wisdom_log.txt", write_mode);
write(msg,LF & LF & "-----" & LF & "*" Botão apertado com velocidade 1  *" & LF & "-----",right, 0);
writeline(l_file, msg);
file_close (l_file);

wait for 9500*CLK_PERIOD;
```

c) Descreva abaixo como os comandos do Stimuli_wisdom correlacionam-se com os resultados do mapa do item a).

Fazemos checkWC('0',1); assim definimos enable= '0' e velocidade = 1, dessa forma o discípulo não vai poder caminhar, pois precisa de enable = '1' para sua caminhada, mas o guru vai andar no tabuleiro. A velocidade definida é irrelevante para essa situação, já que o discípulo vai ficar parado.

d) Copie a seguir imagem(ns) do ModelSim, evidenciando com anotações, os sinais envolvidos na condição "guru termina e discípulo não se mexe" sendo atingida (enable , end_of_guru/endo_of_disc e endereços do guru/disciple, etc.).



ATENÇÃO: Todos os sinais citados acima devem ser RESSALTADOS na(s) própria(s) imagem(s). O arquivo run_sim.txt, já fornecido junto aos arquivos de teste, pode ter sido modificado para possibilitar a visualização acima.

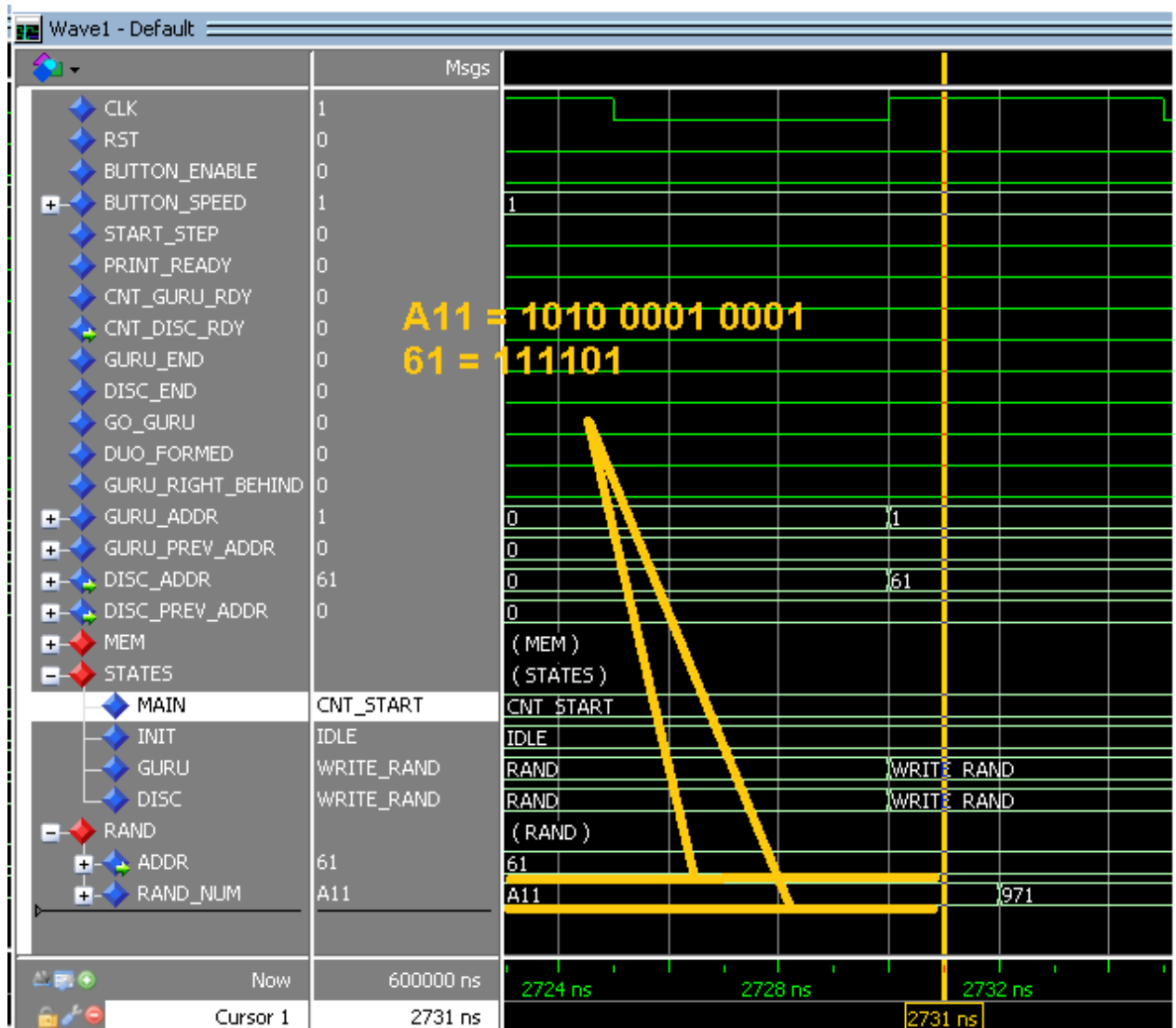
e) Faça uma explicação/justificativa detalhada de como os tempos da curva do item d) se correlacionam com o controle de tempo do trecho de código do item b) na obtenção do cenário "guru termina e discípulo não se mexe".

A cada passo do guru dado temos um intervalo de tempo, esse mesmo intervalo de tempo gera uma nova imagem. Assim vemos uma imagem para cada passo do guru.

f) Faça uma explicação/justificativa detalhada de como os sinais da curva do item d) evidenciam a condição pretendida.

Pode-se observar a transição das posições do guru até chegar na última casa do tabuleiro (posição 8 para o guru), junto a isso observa-se a posição inicial do discípulo permanecendo inalterada até o fim da rodada.

g) Copie a seguir imagem(ns) do ModelSim, evidenciando com anotações, o valor aleatório pela saída do módulo rand_num (para a posição inicial do discípulo), o qual deve ser compatível com o mapa de posições.



5. Resultados das simulações do VHDL final (Wisdom completo) e do seu testbench no ModelSim para a condição de "guru e discípulo terminam sem interação".

a) Identifique o cenário de acordo com a anotação do arquivo de mapas do item 3b.

-- -- Rodada com o botao pressionado e velocidade X4 . Deve-se observar a movimentacao do GURU e DISCIPULO sem interação

* NEW ROUND *

-----NEXT STEP-----

|>.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....^|

-----NEXT STEP-----

|>.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....^|

|.....|

-----NEXT STEP-----

|>.....|

|.....|

|.....|

|.....|

|.....|

|.....^|

|.....|

|.....|

-----NEXT STEP-----

|.>.....|

|.....|

|.....|

|.....|

|.....^|

|.....|

|.....|

|.....|

-----NEXT STEP-----

|.>.....|

|.....|

|.....|

|.....^|

|.....|

|.....|

|.....|

|.....|

-----NEXT STEP-----

|.>.....|

|.....|

|.....^.|

|.....|

|.....|

|.....|

|.....|

|.....|

-----NEXT STEP-----

|.>.....|

|.....^.|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

-----NEXT STEP-----

|.>...^.|

|.....|

|.....|

|.....|
|.....|
|.....|
|.....|
|.....|

-----NEXT STEP-----

|..>.....|
|.....|
|.....|
|.....|
|.....|
|.....|
|.....|
|.....|

-----NEXT STEP-----

|...>....|
|.....|
|.....|
|.....|
|.....|
|.....|
|.....|
|.....|

-----NEXT STEP-----

|...>...|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

-----NEXT STEP-----

|...>..|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

-----NEXT STEP-----

|...>..|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

-----NEXT STEP-----

|.....>|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

-----NEXT STEP-----

|.....>|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

b) Inclua abaixo o trecho da arquitetura do módulo Stimuli_wisdom modificado com os comandos para se chegar ao "guru e discípulo terminam sem interação".

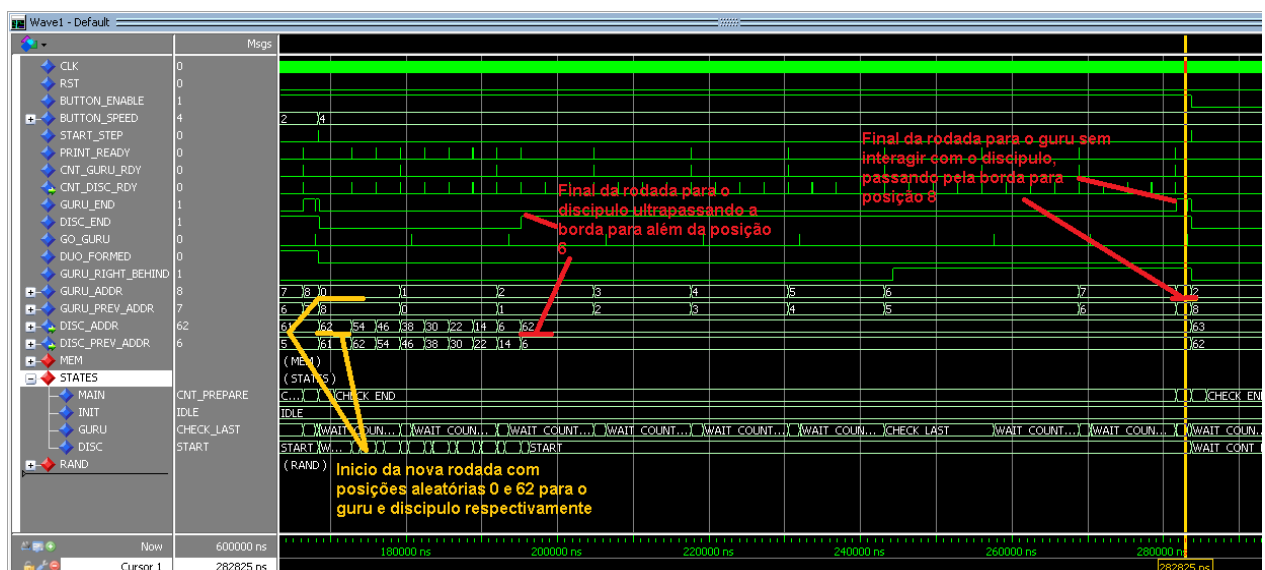
```
wait for 7346*CLK_PERIOD;

-- -- Rodada com o botao pressionado e velocidade X4 . Deve-se observar a movimentacao do GURU e DISCIPULO sem interaçao
checkWC('1',4); --na proxima rodada
wait for 11521*CLK_PERIOD;
```

c) Descreva abaixo como os comandos do Stimuli_wisdom correlacionam-se com os resultados do mapa do item a).

Fazemos checkWC('1',4); assim definimos enable= '1' e velocidade = 4, dessa forma o discípulo vai caminhar, pois enable = '1' permite o movimento do guru indicando que o jogador interage. Como a velocidade definida foi de 4, veremos o discípulo e o guru caminharem, mas o discípulo caminhar com velocidade 4x a do guru.

d) Copie a seguir imagem(ns) do ModelSim, evidenciando com anotações, os sinais envolvidos na condição "guru e discípulo terminam sem interação" sendo atingida (enable , end_of_guru/endo_of_disc e endereços do guru/disciple, etc.).



ATENÇÃO: Todos os sinais citados acima devem ser **RESSALTADOS** na(s) própria(s) imagem(s). O arquivo run_sim.txt, já fornecido junto aos arquivos de teste, pode ter sido modificado para possibilitar a visualização acima.

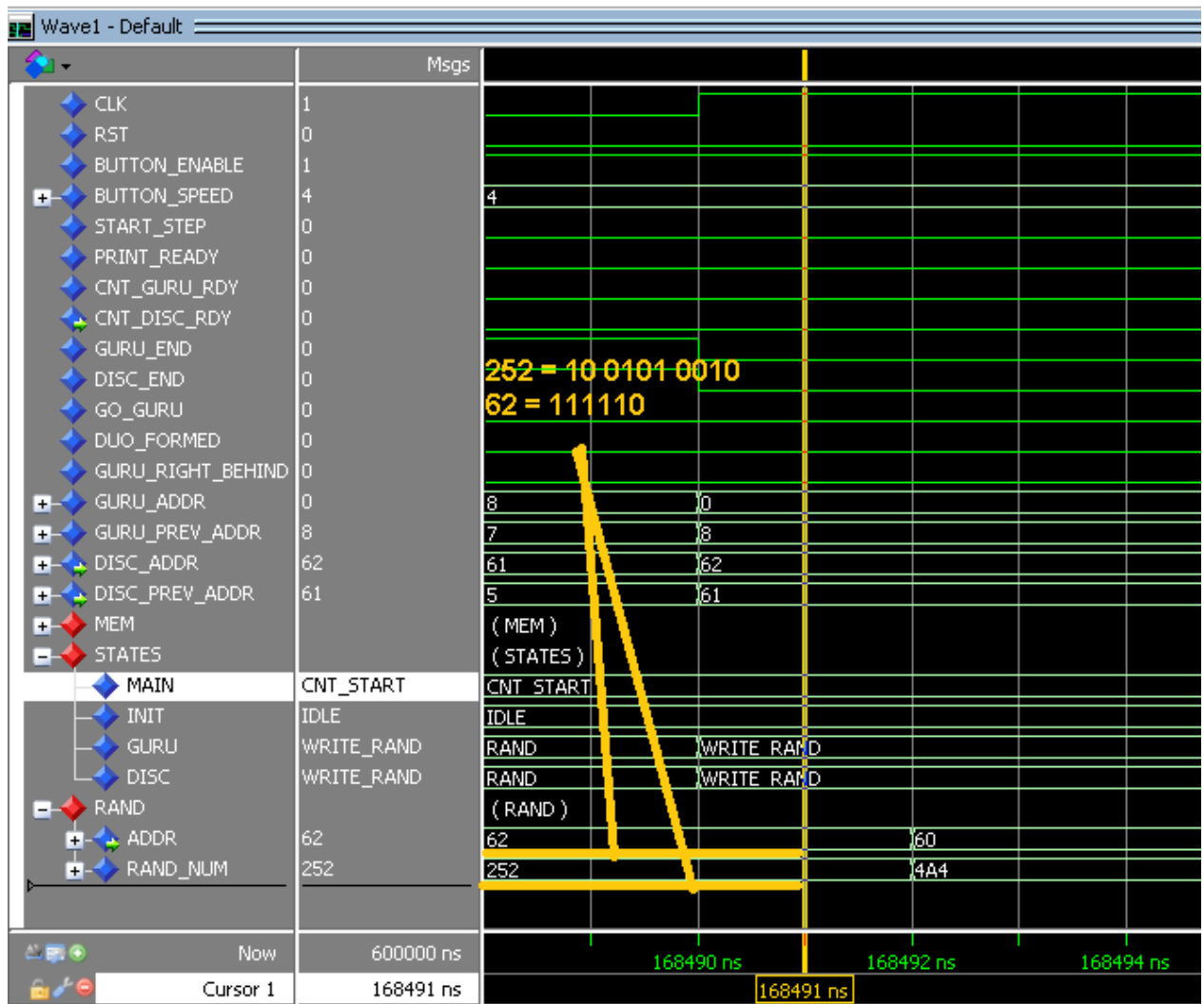
e) Faça uma explicação/justificativa detalhada de como os tempos da curva do item d) se correlacionam com o controle de tempo do trecho de código do item b) na obtenção do cenário "guru e discípulo terminam sem interação".

Como o discípulo está mais rápido, é gerada uma imagem analisando a posição dos dois a cada intervalo de tempo em que se dá o passo do discípulo, até que o discípulo ultrapasse a borda, depois disso é gerada uma imagem a cada intervalo de tempo em que se dá o passo do guru.

f) Faça uma explicação/justificativa detalhada de como os sinais da curva do item d) evidenciam a condição pretendida.

Podemos observar que `button_enable = 1` e `velocidade = 4` (diferente do item anterior), isso faz com que o discípulo caminhe junto ao guru dessa vez. Podemos observar que o discípulo termina a rodada antes do guru, assim não o encontra no tabuleiro (então `duo_formed` se mantém em 0) e também não está passando quando o guru está chegando (então `guru_behind` se mantém em 0), assim podemos ver que ambos andam e não se relacionam durante essa rodada.

g) Copie a seguir imagem(ns) do ModelSim, evidenciando com anotações, o valor aleatório pela saída do módulo `rand_num` (para a posição inicial do discípulo), o qual deve ser compatível com o mapa de posições.



6. Resultados das simulações do VHDL final (Wisdom completo) e do seu testbench no ModelSim para a condição de "guru e discípulo se encontram".

a) Identifique o cenário de acordo com a anotação do arquivo de mapas do item 3b.

-- -- Rodada com o botao pressionado e velocidade X2 . Deve-se observar a movimentacao do GURU e DISCIPULO formando duo

* NEW ROUND *

-----NEXT STEP-----

|..>....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....^..|

-----NEXT STEP-----

|..>....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....^..|

|.....|

-----NEXT STEP-----

|...>....|

|.....|

|.....|

|.....|

|.....|

|.....^..|

|.....|

|.....|

-----NEXT STEP-----

|...>...|

|.....|

|.....|

|.....|

|...^...|

|.....|

|.....|

|.....|

-----NEXT STEP-----

|....>...|

|.....|

|.....|

|....^...|

|.....|

|.....|

|.....|

|.....|

-----NEXT STEP-----

|....>...|

|.....|

|...[^]..|

|.....|

|.....|

|.....|

|.....|

|.....|

-----NEXT STEP-----

|...>..|

|...[^]..|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

-----NEXT STEP-----

|...O..|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

-----NEXT STEP-----

|.....O|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

-----NEXT STEP-----

|.....O|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

b) Inclua abaixo o trecho da arquitetura do módulo Stimuli_wisdom modificado com os comandos para se chegar ao "guru e discípulo se encontram".

```
wait for 9500*CLK_PERIOD;

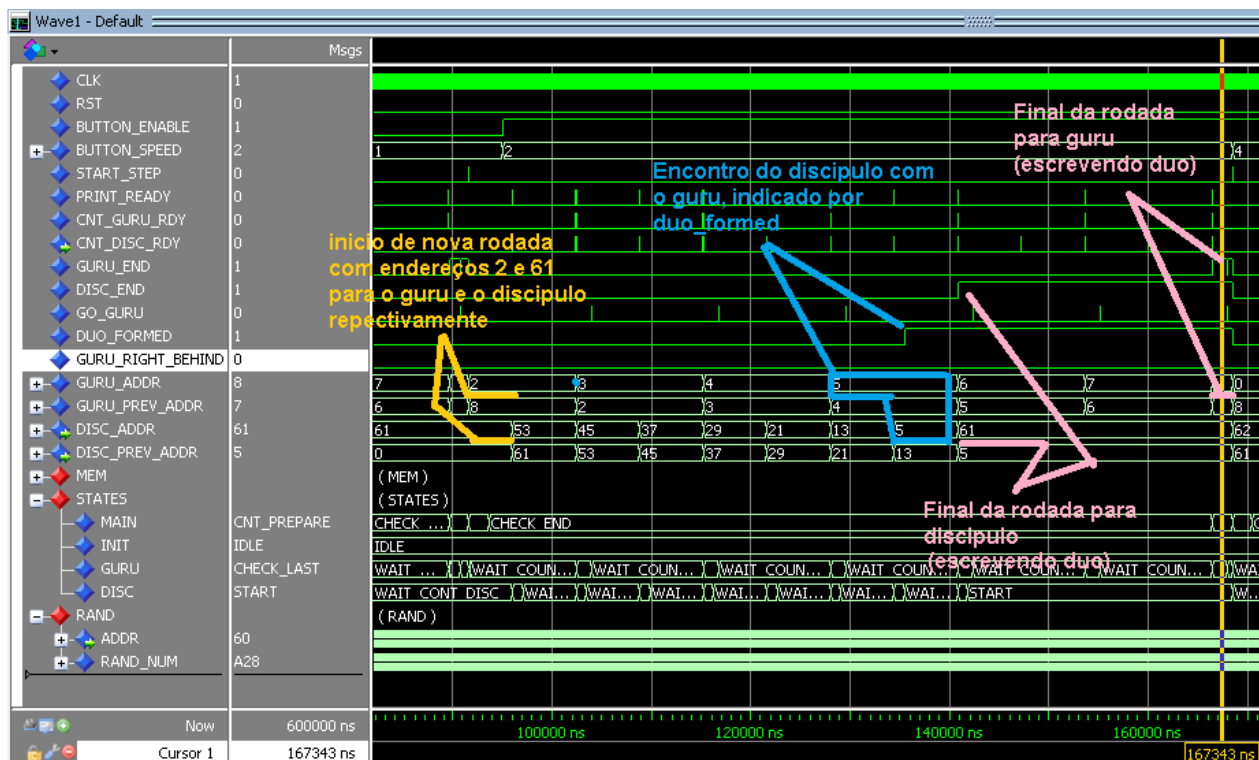
-- -- Rodada com o botao pressionado e velocidade X2 . Deve-se observar a movimentacao do GURU e DISCIPULO formando duo
checkWC('1',2);

wait for 7346*CLK_PERIOD;
```

c) Descreva abaixo como os comandos do Stimuli_wisdom correlacionam-se com os resultados do mapa do item a).

Fazemos checkWC('1',2); assim definimos enable= '1' e velocidade = 2, dessa forma o discípulo vai caminhar, pois enable = '1' permite o movimento do guru indicando que o jogador interage. Como a velocidade definida foi de 2, veremos o discípulo e o guru caminharem, mas o discípulo caminhar com velocidade 2x a do guru. Essa velocidade foi disparada no tempo certo para que o discípulo encontre o guru e forme duo.

d) Copie a seguir imagem(ns) do ModelSim, evidenciando com anotações, os sinais envolvidos na condição "guru e discípulo se encontram" sendo atingida (enable , end_of_guru/endo_of_disc e endereços do guru/disciple, etc.).



ATENÇÃO: Todos os sinais citados acima devem ser **RESSALTADOS** na(s) própria(s) imagem(s). O arquivo run_sim.txt, já fornecido junto aos arquivos de teste, pode ter sido modificado para possibilitar a visualização acima.

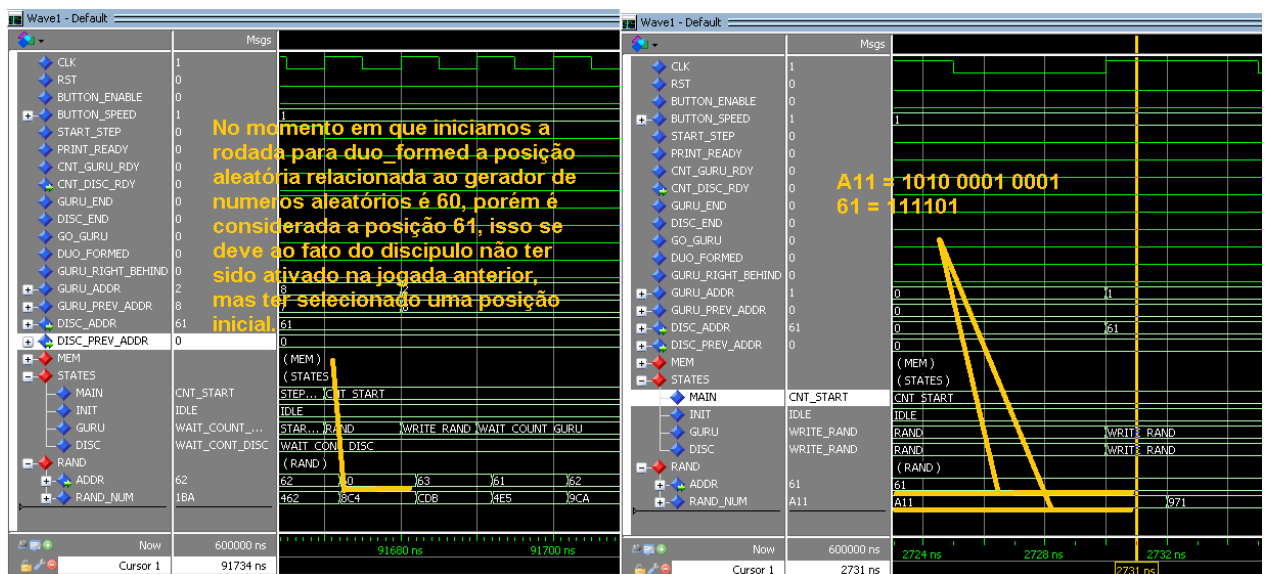
e) Faça uma explicação/justificativa detalhada de como os tempos da curva do item d) se correlacionam com o controle de tempo do trecho de código do item b) na obtenção do cenário "guru e discípulo se encontram".

Como o discípulo está mais rápido, é gerada uma imagem analisando a posição dos dois a cada intervalo de tempo em que se dá o passo do discípulo, até que o discípulo encontre o guru, depois disso é gerada uma imagem a cada intervalo de tempo em que se dá o passo do guru (representando com duo).

f) Faça uma explicação/justificativa detalhada de como os sinais da curva do item d) evidenciam a condição pretendida.

Podemos observar que `button_enable = 1` e `velocidade = 2` (diferente dos itens anteriores), isso faz com que o discípulo caminhe junto ao guru. Podemos observar que o discípulo encontra o guru na primeira linha do tabuleiro na posição 5, assim `duo_formed` é setado para indicar o encontro. O discípulo, então, incrementa para fora e indica final da rodada para ele com `end_disc = 1`, mas não apaga a posição anterior que foi escrita com `duo`, pois o `duo` seguirá no tabuleiro de acordo com o passo do guru. Quando o guru (com `duo`) passa pelo final do tabuleiro, guru indica final da rodada com `end_guru = 1`.

g) Copie a seguir imagem(ns) do ModelSim, evidenciando com anotações, o valor aleatório pela saída do módulo `rand num` (para a posição inicial do discípulo), o qual deve ser compatível com o mapa de posições.



7. Resultados das simulações do VHDL final (Wisdom completo) e do seu testbench no ModelSim para a condição de "o discípulo termina com o guru *right_behind*".

a) Identifique o cenário de acordo com a anotação do arquivo de mapas do item 3b.

-- Rodada com o botao pressionado e velocidade X8 . Deve-se observar a movimentacao do GURU e DISCIPULO com guru_behind = 1

* NEW ROUND *

-----NEXT STEP-----

|..>....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....^|

-----NEXT STEP-----

|...>....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....^|

-----NEXT STEP-----

|...>...|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....^|

-----NEXT STEP-----

|...>..|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....^|

-----NEXT STEP-----

|.....>.|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....^|

-----NEXT STEP-----

|.....>.|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....^|

|.....|

-----NEXT STEP-----

|.....>.|

|.....|

|.....|

|.....|

|.....|

|.....^|

|.....|

|.....|

-----NEXT STEP-----

|.....>.|

|.....|

|.....|

|.....|

|.....^|

|.....|

|.....|

|.....|

-----NEXT STEP-----

|.....>.|

|.....|

|.....|

|.....^|

|.....|

|.....|

|.....|

|.....|

-----NEXT STEP-----

|.....>.|

|.....|

|.....^|

|.....|

|.....|

|.....|

|.....|

|.....|

-----NEXT STEP-----

|.....>.|

|.....^|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

-----NEXT STEP-----

|.....>^|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

-----NEXT STEP-----

|.....>^|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

-----NEXT STEP-----

|.....>|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

|.....|

b) Inclua abaixo o trecho da arquitetura do módulo Stimuli_wisdom modificado com os comandos para se chegar ao "o discípulo termina com o guru *right_behind*".

```
wait for 11521*CLK_PERIOD;

checkWC('0',4); --para o discipulo não sair quando guru começar
wait for 4915*CLK_PERIOD;

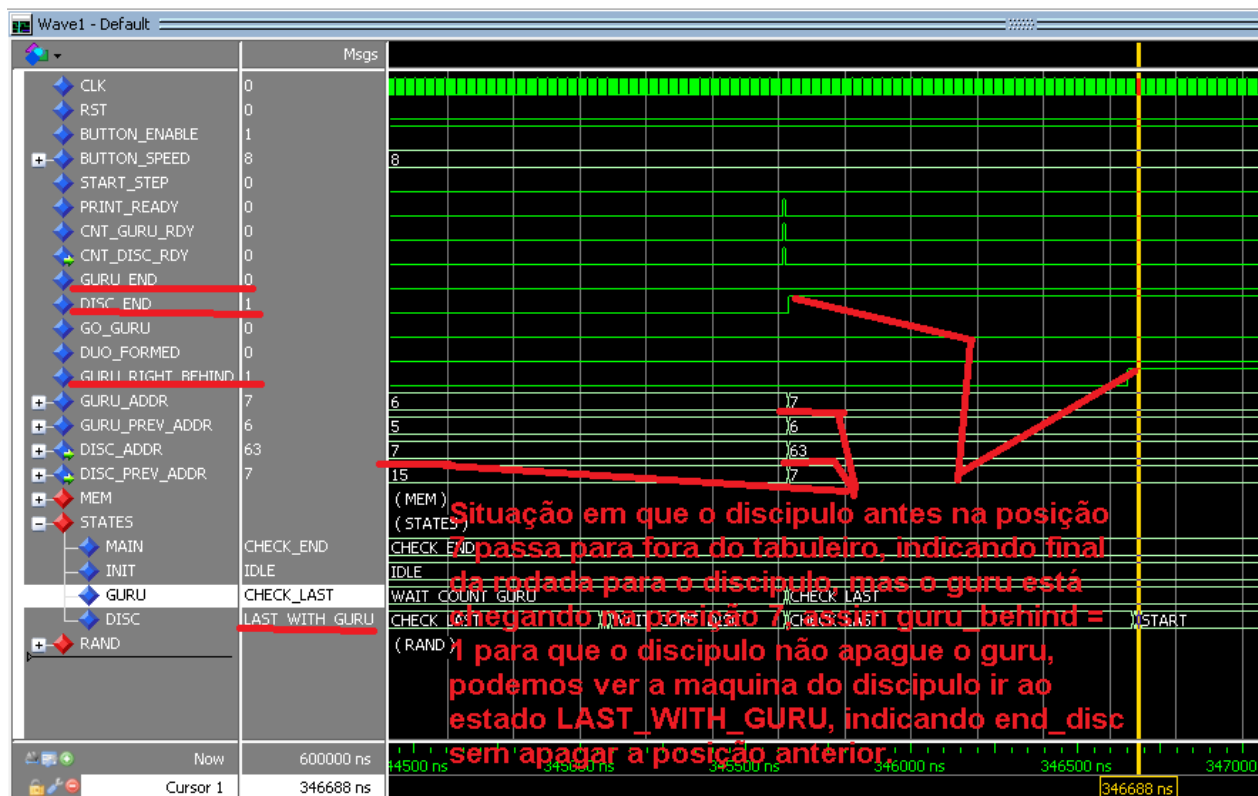
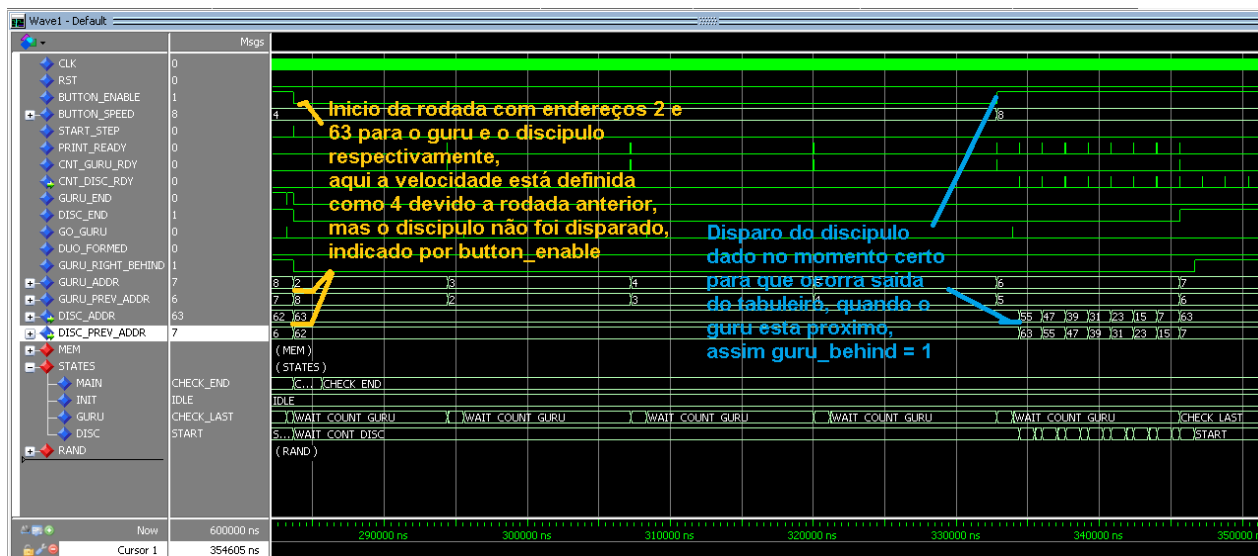
-- Rodada com o botao pressionado e velocidade X8 . Deve-se observar a movimentacao do GURU e DISCIPULO com guru_behind = 1
checkWC('1',8); -- para discipulo sair no momento esperado para guru_behind = 1

wait for 20000*CLK_PERIOD;           -- tempo suficiente para várias rodadas do jogo
```

c) Descreva abaixo como os comandos do Stimuli_wisdom correlacionam-se com os resultados do mapa do item a).

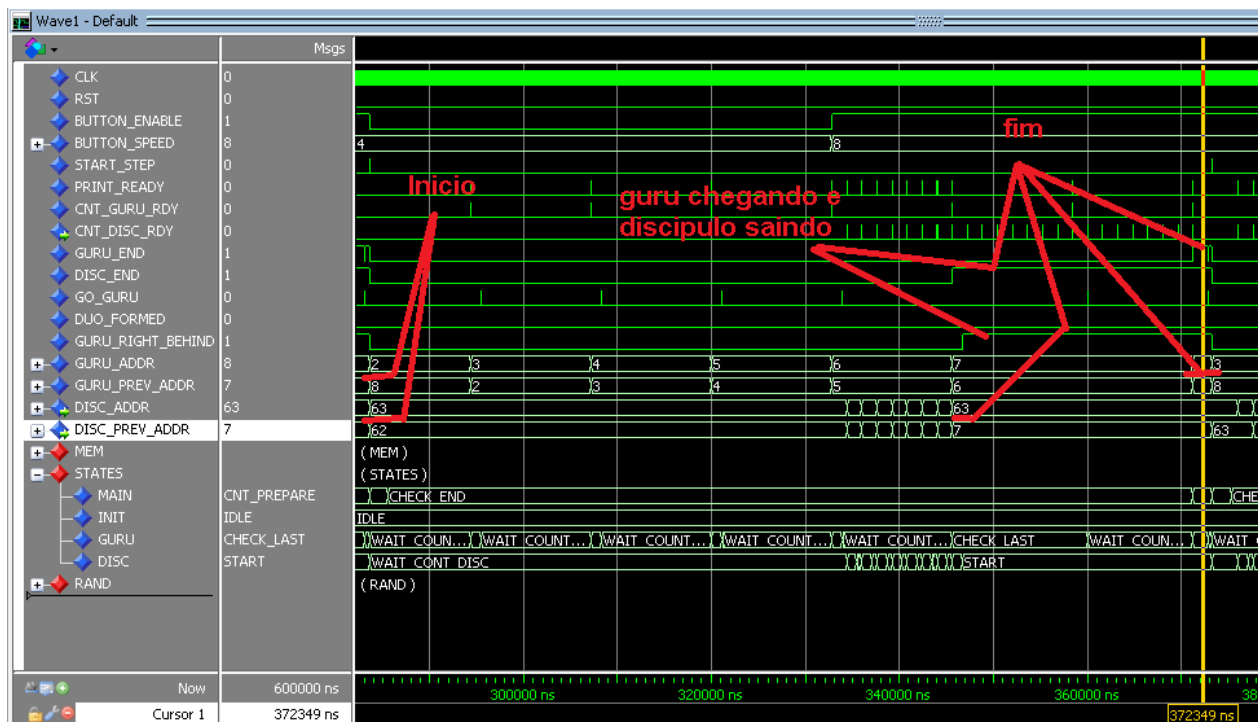
Fazemos checkWC('1',8); assim definimos enable= '1' e velocidade = 8, dessa forma o discípulo vai caminhar, pois enable = '1' permite o movimento do guru indicando que o jogador interage. Como a velocidade definida foi de 8, veremos o discípulo e o guru caminharem, mas o discípulo caminhar com velocidade 8x a do guru. Essa velocidade foi disparada no tempo certo para que o discípulo não encontre o guru, mas passe na casa da primeira linha quando o guru está na casa anterior, para que guru_behind = 1.

d) Copie a seguir imagem(ns) do ModelSim, evidenciando com anotações, os sinais envolvidos na condição "o discípulo termina com o guru *right_behind*" sendo atingida (enable , end_of_guru/endo_of_disc e endereços do guru/disciple, etc.).



Texto da imagem:

Situação em que o discipulo antes na posição 7 passa para fora do tabuleiro, indicando final da rodada para o discipulo, mas o guru está chegando na posição 7, assim guru_behind = 1 para que o discipulo não apague o guru, podemos ver a maquina do discipulo ir ao estado LAST_WITH_GURU, indicando end_disc sem apagar a posição anterior.



ATENÇÃO: Todos os sinais citados acima devem ser **RESSALTADOS** na(s) própria(s) imagen(s). O arquivo run_sim.txt, já fornecido junto aos arquivos de teste, pode ter sido modificado para possibilitar a visualização acima.

e) Faça uma explicação/justificativa detalhada de como os tempos da curva do item d) se correlacionam com o controle de tempo do trecho de código do item b) na obtenção do cenário "o discípulo termina com o guru *right_behind*".

Inicialmente o discípulo não está ativo, sendo assim é gerada uma imagem a cada intervalo de tempo do passo do guru, até que o discípulo seja ativo. Com o discípulo ativo, como ele está mais rápido, é gerada uma imagem analisando a posição dos dois a cada intervalo de tempo em que se dá o passo do discípulo, até que o discípulo ultrapasse a borda do tabuleiro, depois disso é gerada uma imagem a cada intervalo de tempo em que se dá o passo do guru novamente.

f) Faça uma explicação/justificativa detalhada de como os sinais da curva do item d) evidencia a condição pretendida.

Podemos observar que button_enable = 0 e velocidade = 4 inicialmente, isso faz com que o guru caminhe sozinho por um trecho, até que faz-se button_enable = 1 e velocidade = 8, isso faz com que o discípulo caminhe 8x mais rápido que o guru,

assim o discípulo ultrapassa o guru exatamente na casa da frente e sai do tabuleiro quando o guru está atrás dele, isso faz com que guru_behind seja setado para indicar que o guru está chegando na posição de saída do discípulo, em seguida o discípulo incrementa para fora e indica final da rodada para ele com end_disc = 1, mas não apaga a posição anterior que foi escrita com guru..Quando o guru passa pelo final do tabuleiro, guru indica final da rodada com end_guru = 1.

g) Copie a seguir imagem(ns) do ModelSim, evidenciando com anotações, o valor aleatório pela saída do módulo rand_num (para a posição inicial do discípulo), o qual deve ser compatível com o mapa de posições.

