## UNIVERSIDADE FEDERAL DA BAHIA – UFBA ESCOLA POLITÉCNICA / DEPARTAMENTO DE ENGENHARIA ELÉTRICA 2017.1 – ENGG52 – LABORATÓRIO INTEGRADO I-A – PROF. PAULO CÉSAR

# ATA DE REUNIÃO – 12/06/2017

Em 12/06, restando uma semana para a entrega do jogo, os estudantes da turma do professor Paulo Cesar Farias reuniram-se para verificar e dar prosseguimento à implementação Verilog. Estiveram presentes:

- 1. Cristian Araújo (Secretário de Mesa)
- 2. Diego Raian (Coordenador)
- 3. Douglas Wiliam Fernandes de Souza
- 4. Felipe Mateus Boaventura
- 5. Lucas Menezes Pereira (Secretário de Quadro)

#### \* METAS:

- a. Prosseguimento da descrição da FSM Posicionando Peças;
- b. Criação da lógica para verificação dos limites de mapa e sobreposição de embarcações;
- c. Verificação das waveforms de simulação dos módulos;
- d. Usar clock como elemento sincronizador dos eventos do sistema: transição de Estado Futuro para Estado Atual.

Como o tempo não foi hábil para finalizar todas as tarefas, algumas pendências foram resolvidas durante a semana por encontros online, além da reunião presencial no sábado (17/06).

O detalhamento de cada item acima consta a seguir:

a. O módulo Posicionando Peças é a FSM que começa, posicionando 1 a 1, todos os barcos do primeiro jogador, através da escolha da posição inicial (posição base) para cada embarcação. Durante a escolha das coordenadas é verificado se o barco transgredirá algum limite (borda) do mapa ou se irá sobrepor alguma parte de embarcação posicionada antes. Tanto a transgressão dos limites como a sobreposição de barcos são operações inválidas, daí, idealmente, volta-se a pedir outras duas coordenadas-base para o barco.

b. Não é possível posicionar uma embarcação cuja extensão ultrapasse os limites do mapa. Mesmo que o mapa seja virtual (não precisa ser construído uma matriz-mapa 8x8), é necessário saber se o barco cabe naquele espaço, para ser inserido. Para tanto, separou-se os modelos das embarcações e usou-se o exemplo de verificação a seguir:

# b.1. CHECANDO LIMITES DO MAPA (para o porta-aviões, sendo a parte alaranjada externa ao mapa):

- Condição válida para inserção: x1<3 e y1<3, sendo x1 e y1 as coordenadas do começo do portaaviões

y							
7							
6		P	P	P	P	P	
5							
4					Е	Е	Е
3							

2									
1									
0									
	0	1	2	3	4	5	6	7	X

Fig 1: "P" indica um dos conjuntos de posições válidas para o porta-aviões, seguindo x1<3 e y1<3. "E" indica posições inválidas para ele; aqui x1=6, y1=4.

## **b.2. PARA CHECAR SOBREPOSIÇÃO EM EMBARCAÇÕES:**

Como cada embarcação é um vetor, é exigência de que um par de coordenadas quaisquer (x, y) esteja somente em um dentre todos os vetores de um mesmo player. Quando se deseja verificar a qual barco pertence essa posição, todos os vetores são "varridos" para identificar qual é a peça. Dessa forma, durante a execução do módulo Posicionando Peças, a cada escolha de posição definida pelo player, varrem-se os todos os vetores criados até aquele momento para verificar se as coordenadas escolhidas para o novo barco coincidem com algum outro já posicionado. De acordo com o diagrama abaixo, caso haja conflito de posição, o sistema volta a solicitar novo local para a inserção da peça.

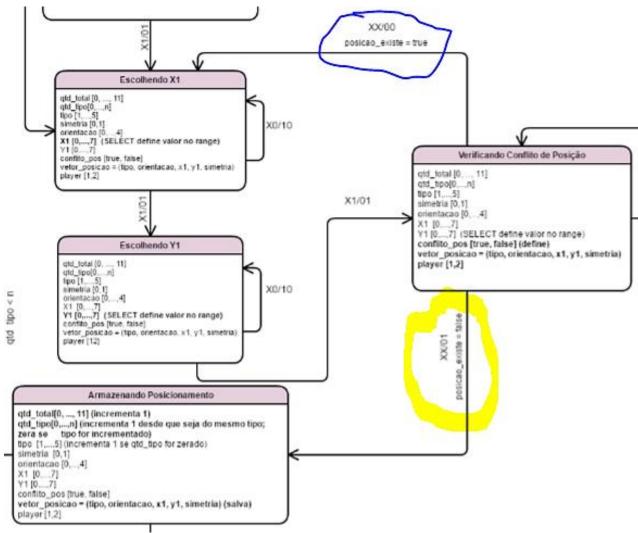


Fig. 2: Trecho do diagrama de Posicionando Peças que impede a sobreposição das (partes de) embarcações. Em azul, indica-se que se a posição já estiver ocupada, retorna à escolha de novas coordenadas. Em amarelo, destaca-se que a posição estava livre, então pode-se armazenar os valores de x e y no vetor da embarcação e assim, inserir a peça no "mapa".

\* OBS: Uma vez que se trabalha em baixo nível (nível de hardware), cada vetor é um grande conjunto de registradores, indicado pelo nome "reg" nos códigos Verilog implementados.

- c. As waveforms são uma ferramenta de simulação presente no Quartus Prime e ModelSIM (softwares utilizados pela equipe). Eles pegam diferentes atribuições das entradas e calcula a saída, exibindo-a em forma de onda. Foi necessário simular as saídas de todos os módulos, para corrigir possíveis erros.
- d. Iniciamente, pensou-se na transição entre os estados das FSMs ser independente de clock (não teria clock), pois o progresso do jogo ficaria sujeito unicamente ao controle do jogador através dos botões SELECT e ENTER. Posteriormente, foi observado que tal decisão afetaria o andamento do jogo. Como se tratava de uma Máquina de Mealy (orientada a transições), era necessário um elemento sincronizador que salvasse o estado futuro na memória (conjunto de Flip-flops), para que depois passasse a ser o estado atual. A ilustração desse tipo de máquina está abaixo:

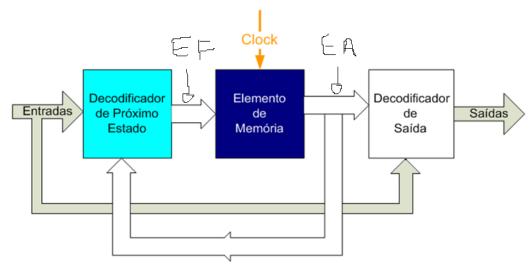


Fig. 3: Esquema da Máquina de Mealy. Enquanto executa uma tarefa, o sistema precisa segurar o estado futuro para que a ação atual não seja interrompida. Para reter temporariamente a ação do estado futuro é usada a memória (FFs).

Pendências posteriores foram gradativamente corrigidas na reunião presencial, no sábado 17/06 e online, no domingo, 18/06.