

**UNIVERSIDADE DE SÃO PAULO  
ESCOLA POLITÉCNICA  
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E  
SISTEMAS DIGITAIS  
PCS3635 - LABORATÓRIO DIGITAL I**



**PLANEJAMENTO SEMANA 1 - POLI-ASTEROIDS**

Felipe Luis Korbes - NUSP: 13682893

João Felipe de Souza Melo - NUSP: 13682913

**Turma:** 5

**Professor:** Reginaldo Arakaki

**Data da experiência:** 20/03/2024

São Paulo

2024

**Sumário**

<b>1. Introdução e Objetivos.....</b>	<b>1</b>
<b>2. Progresso da Semana 2.....</b>	<b>1</b>
2.1 Fluxo de Dados.....	1
2.2 Unidades de Controle.....	5
2.3 Testes.....	13
<b>3. Planejamento da aula prática.....</b>	<b>14</b>
<b>4. Relatório.....</b>	<b>14</b>
<b>5. Cronograma.....</b>	<b>14</b>

## **1. Introdução e Objetivos**

O objetivo deste laboratório é o de desenvolver as atividades do projeto, especialmente na implementação dos requisitos funcionais previstos pelo cronograma para essa semana. Deve-se então continuar o projeto lógico do circuito, com o diagrama de blocos e o modelo da máquina de estados finita, e programar os arquivos de descrição de hardware em Verilog.

## **2. Progresso da Semana 2**

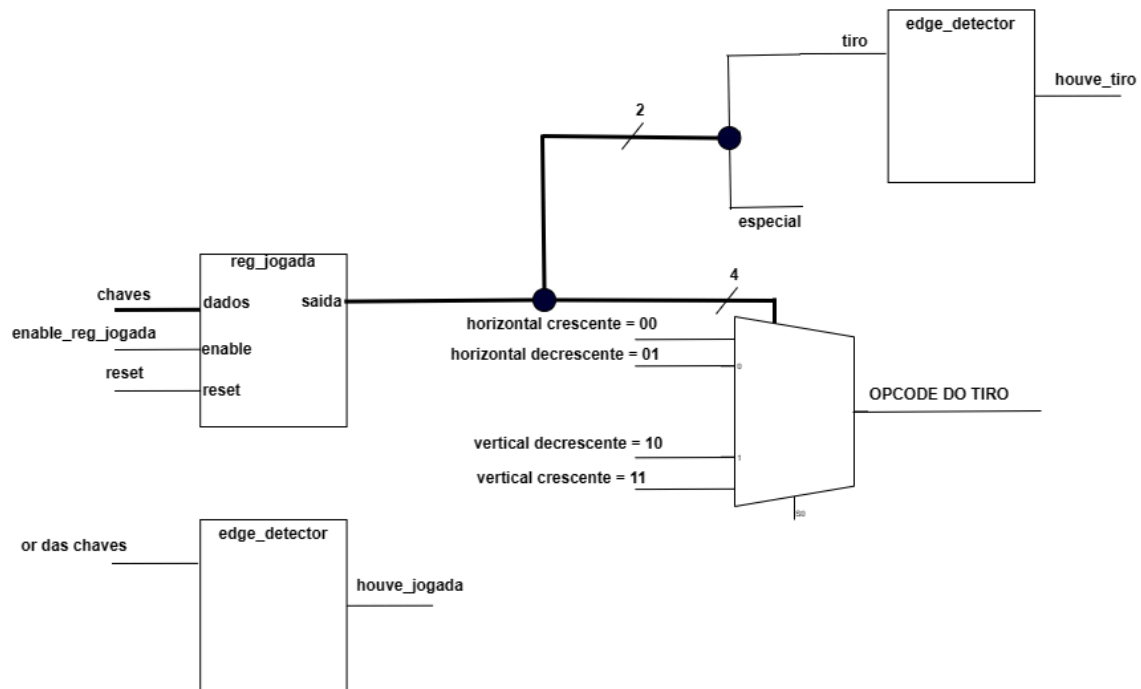
Na segunda semana do projeto, a complexidade do circuito foi incrementada significativamente em comparação com a semana anterior. Desenvolvemos várias máquinas de estado para garantir o funcionamento adequado do circuito, as quais serão apresentadas mais adiante. Como equipe, ficou evidente que lidar com essa maior complexidade trouxe grandes desafios na implementação do circuito completo em Verilog. Inicialmente foi desenvolvida uma lógica para o programa, cada objeto (tiro e asteroide e nave) tem coordenadas X e Y e essas coordenadas são utilizadas para comparar posições e renderiza-los.

### **2.1 Fluxo de Dados**

O fluxo de dados do projeto é composto por três módulos distintos: um para os asteroides, um para os tiros e outro para registrar as ações do jogador. O módulo dos asteroides, conforme ilustrado na Figura 2, opera por meio de um processo de seleção de asteroides de uma memória. Em seguida, o asteroide selecionado é tratado com base em sua posição inicial, sendo movido na direção x ou y correspondente. Após essa seleção e movimentação, a posição do asteroide é comparada com a posição da nave para detectar colisões. Caso ocorra alguma colisão com a nave, o asteroide é destruído e removido do jogo.

O módulo dos tiros opera de maneira similar ao dos asteroides, com a exceção de que adicionamos o tratamento dos tiros saírem da tela e deixarem de ser renderizados. O diagrama de bloco do fluxo de dados dos Tiros pode ser visto na Figura 3.

Para o fluxo de dados do Registra Jogada, temos algo mais simples, onde a jogada feita pelo jogador é tratada entre um tiro normal ou um tiro especial, e a direção que esse tiro toma. O fluxo de dados de Registra jogada pode ser visto na Figura 1.



*Figura 1 - Fluxo de dados de Registra Jogada*

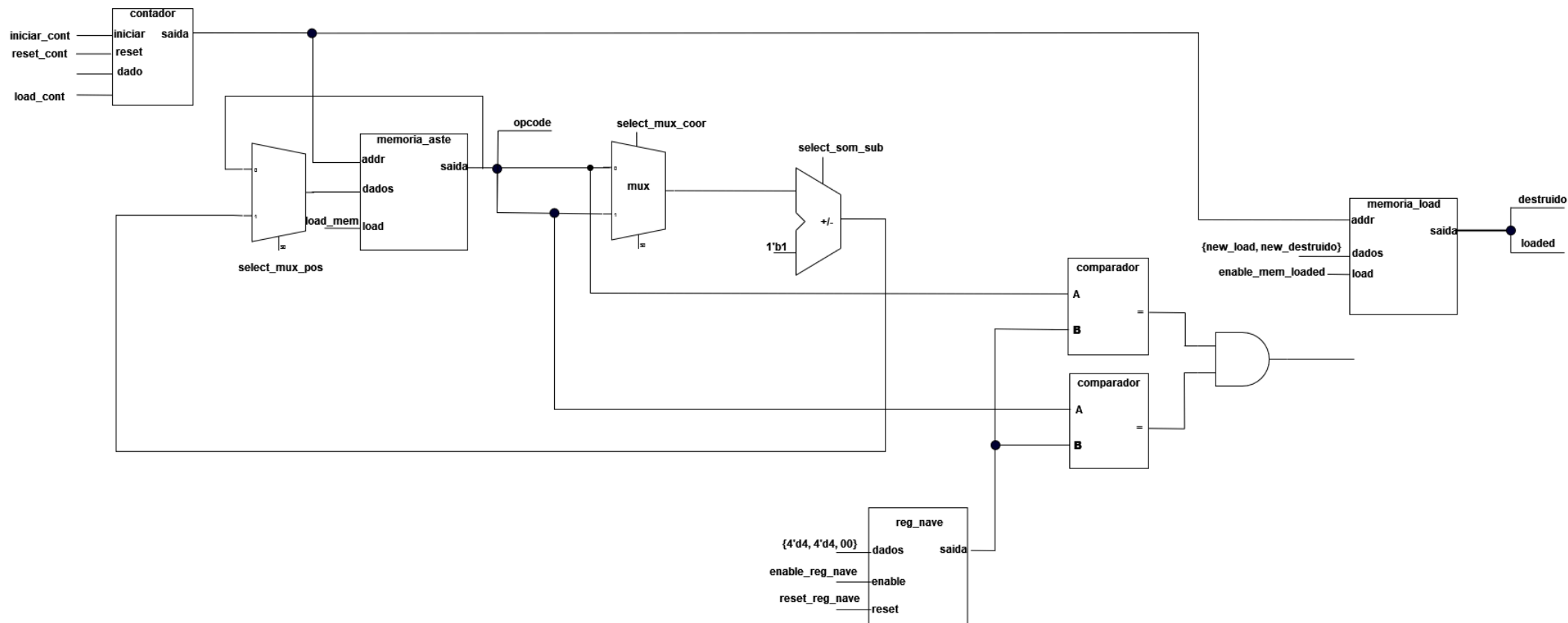


Figura 2 - Fluxo de dados do Asteroide

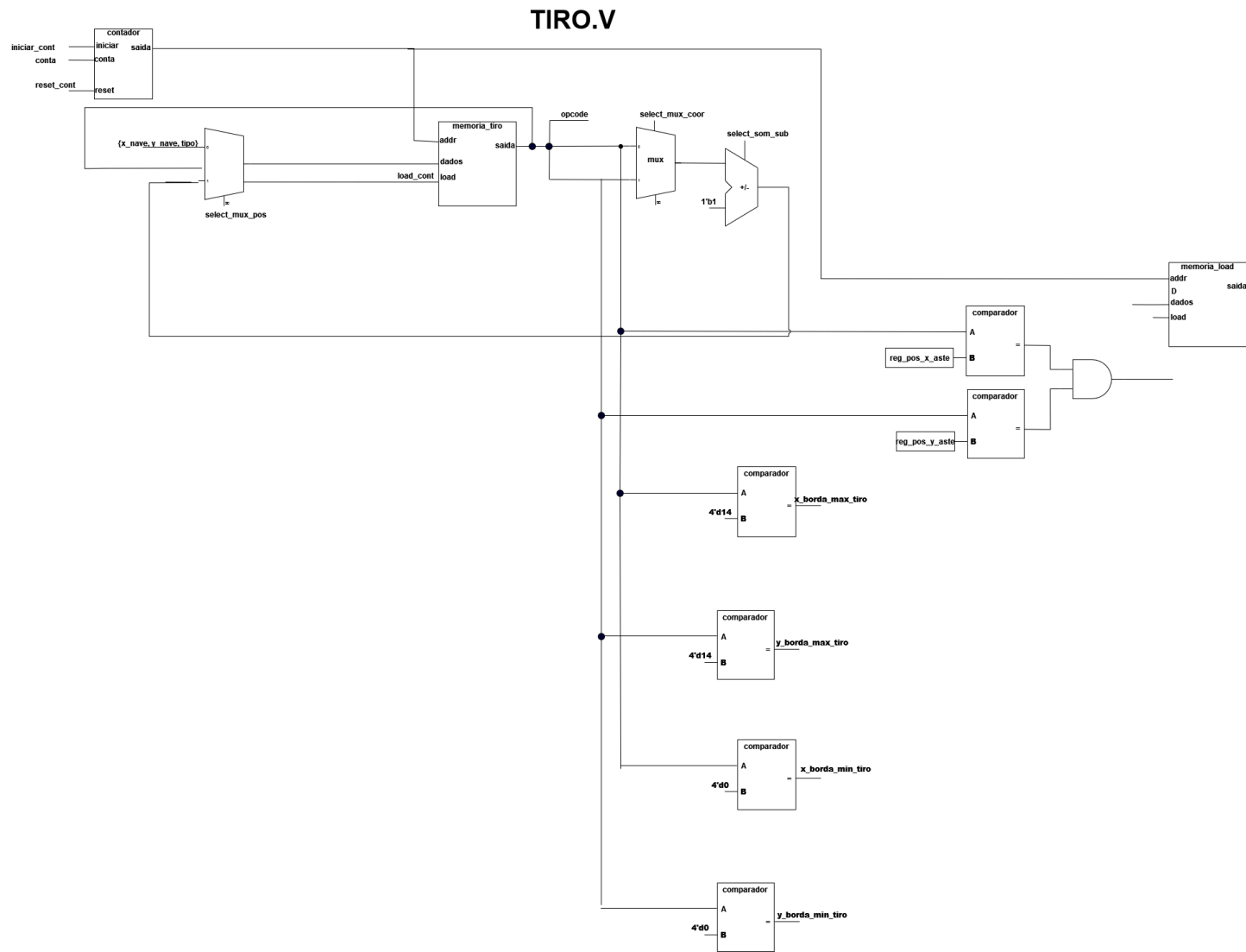


Figura 3 - Fluxo de dados do Tiro

## 2.2 Unidades de Controle

No total, foram desenvolvidas 7 unidades de controle para gerenciar todos os aspectos do jogo. Cada unidade de controle possui sua própria especialidade e é fundamental que trabalhem em sincronia para garantir o funcionamento adequado do jogo. Em várias partes da execução do código, uma unidade de controle pode chamar outra para executar uma função específica. Durante essa execução, a unidade de controle original entra em espera até que a outra unidade de controle conclua seu processo. Após a conclusão, a unidade de controle chamada permanece em espera até ser chamada novamente.

É importante destacar que, devido à grande complexidade das muitas unidades de controle, há muitas transições de estados em nossa máquina. No entanto, considerando que estaremos utilizando um clock de 1GHz ou superior em nosso projeto, espera-se que essas transições não sejam perceptíveis para o jogador.

A unidade de controle principal do jogo, representada na Figura 4, é bastante simples e opera verificando continuamente se o jogador realizou alguma jogada, disparou algum tiro e se o número de vidas do jogador é maior do que zero. Caso as vidas cheguem a zero, o jogo é encerrado. Além disso, ela é responsável por iniciar o movimento dos tiros e asteroides, enviando um sinal para outra unidade de controle. Também é incumbida de registrar os tiros efetuados pelo jogador.

A segunda unidade de controle, sendo a primeira a ser acionada pela unidade de controle principal, tem a responsabilidade de executar a movimentação dos asteroides e dos tiros, além de comparar as posições entre os tiros, asteroides e a nave para lidar com possíveis colisões. Esta unidade de controle chama outras 4 unidades para realizar todas essas tarefas. Seu diagrama de transição de estados pode ser visto na Figura 5.

Na Figura 6 temos a unidade responsável por fazer a comparação completa dos asteroides com a nave e dos tiros com a nave, sendo esse último feito por outra unidade de controle, vista da Figura 7, sendo ela responsável apenas pela comparação dos tiros com os asteroides.

Para a movimentação dos Tiros e Asteroides, temos a unidade de controle da Figura 8, responsável pela movimentação dos Tiros e a unidade da Figura 9, responsável pela movimentação dos Asteroides.

Por fim, a unidade de controle da Figura 10 registra os tiros feitos pelo jogador, completando nosso ciclo de máquinas de estado.

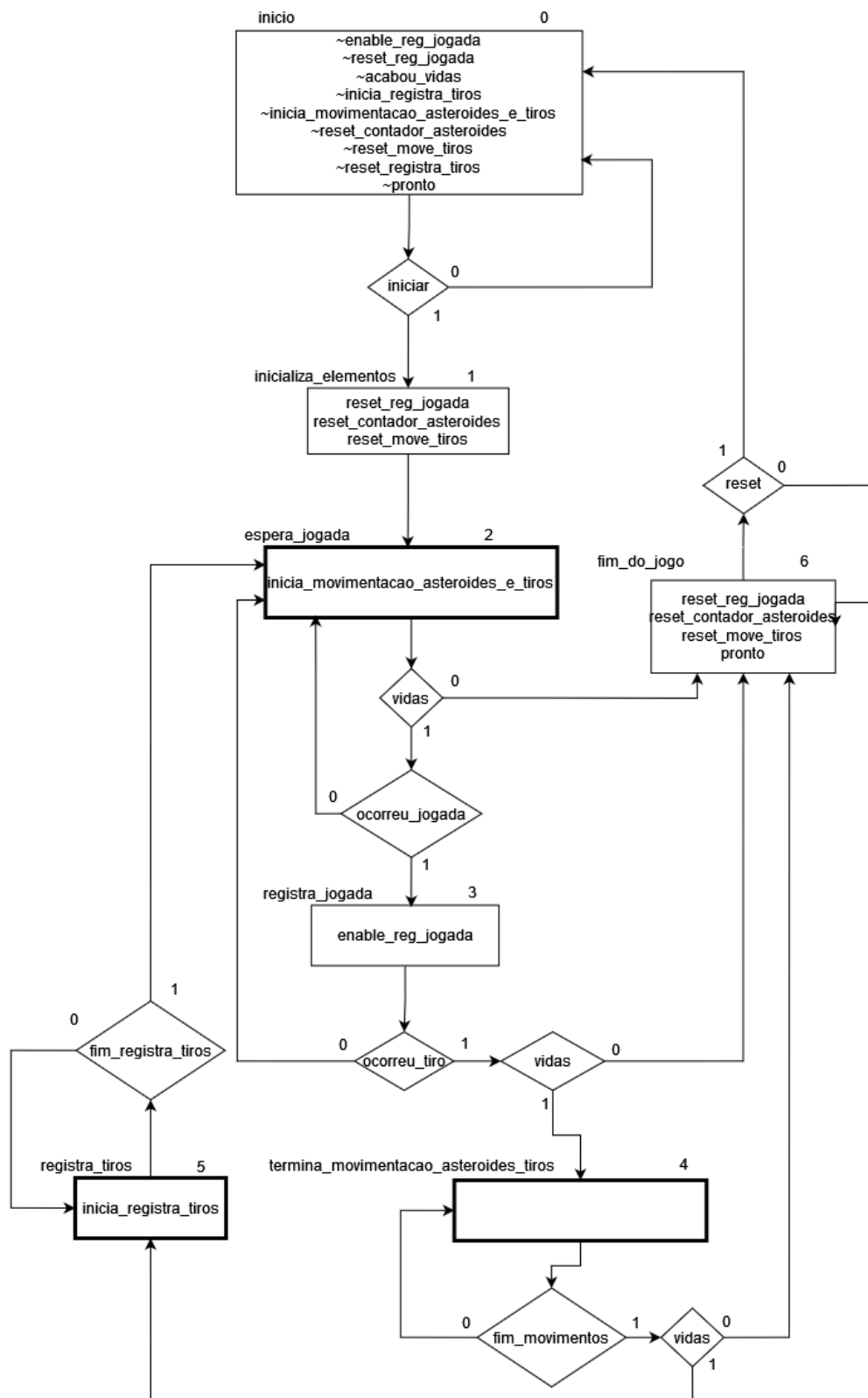


Figura 4 - Unidade de Controle do Jogo Principal



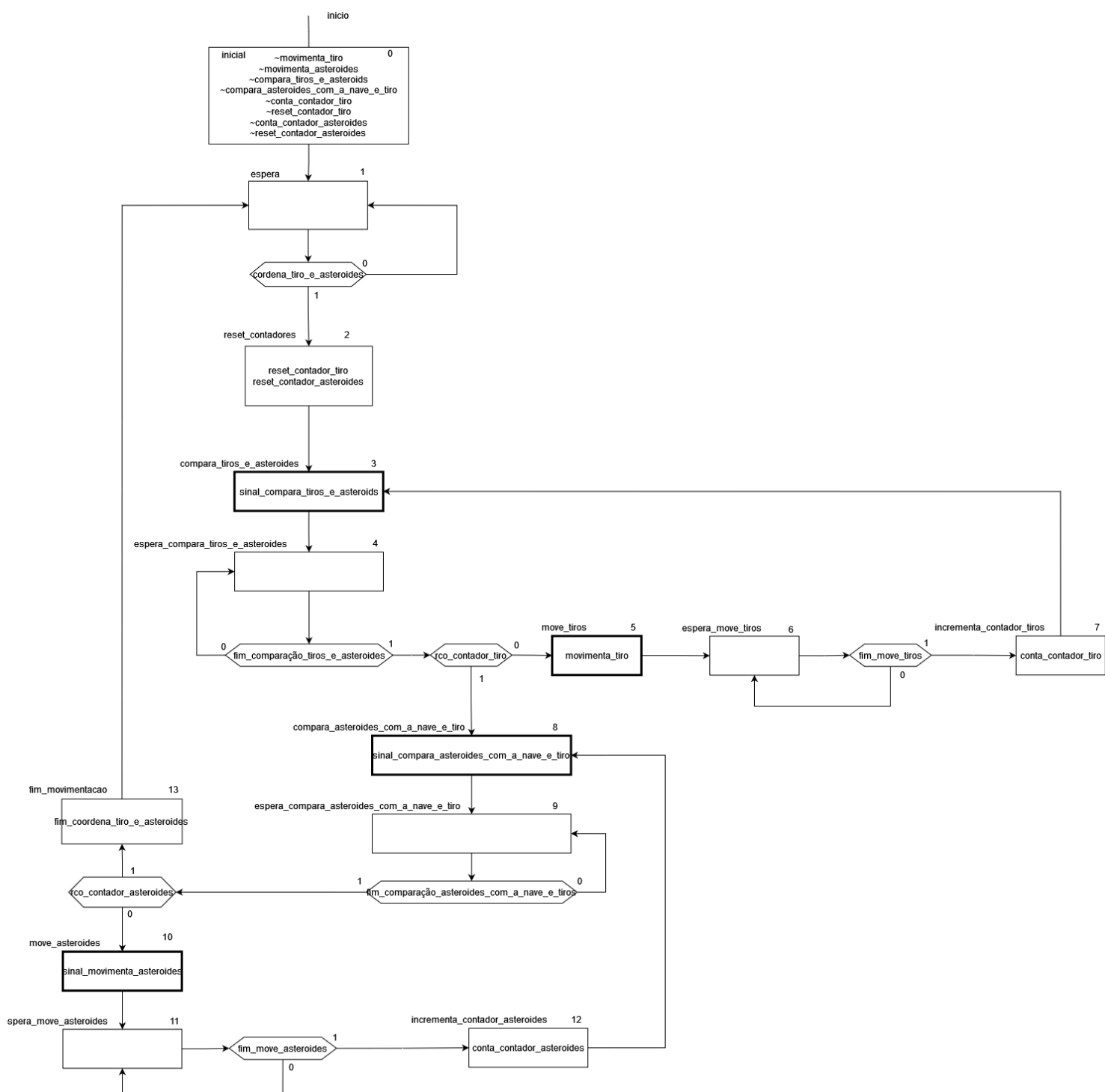


Figura 5 - Unidade de Controle que Coordena os Asteroides e Tiros

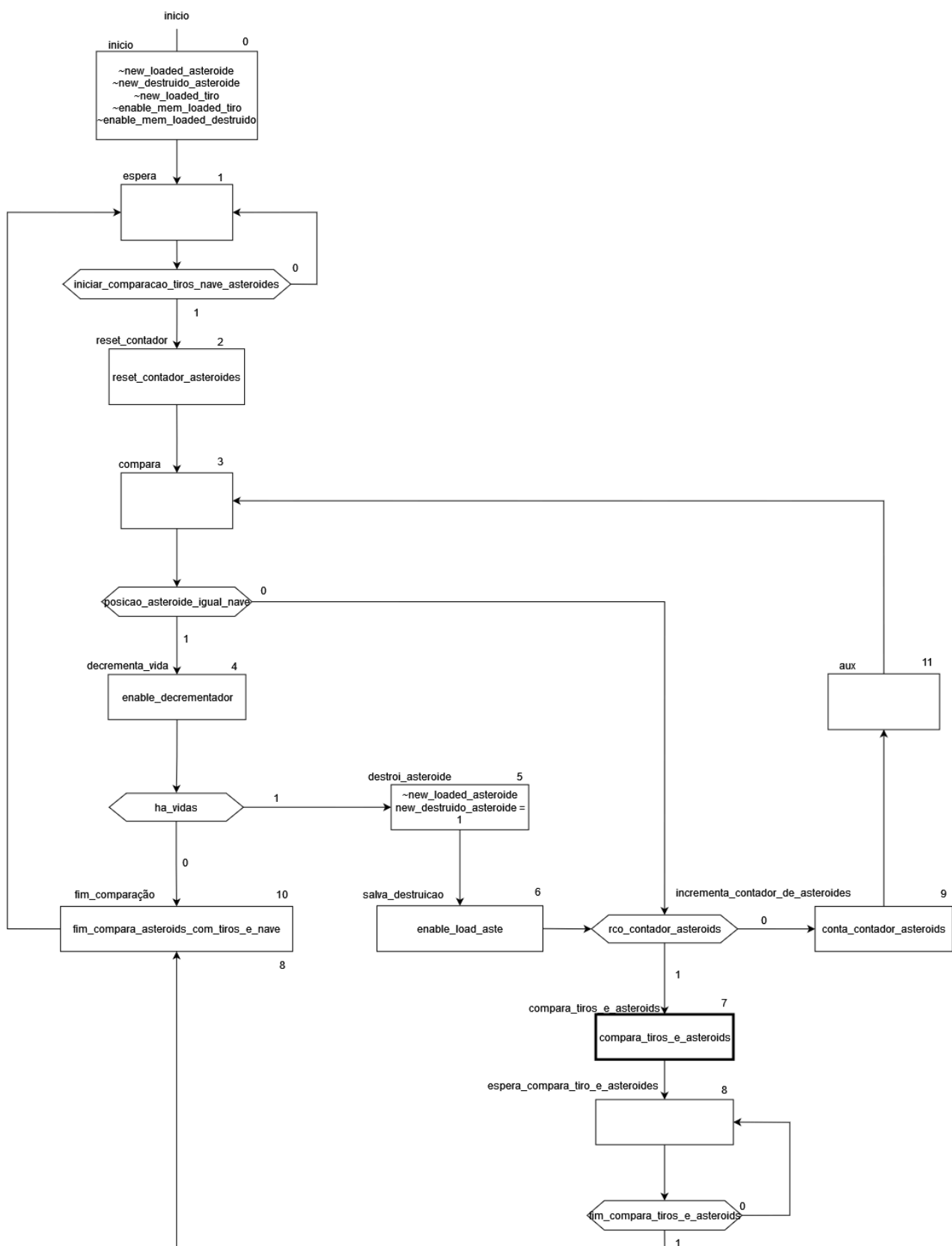
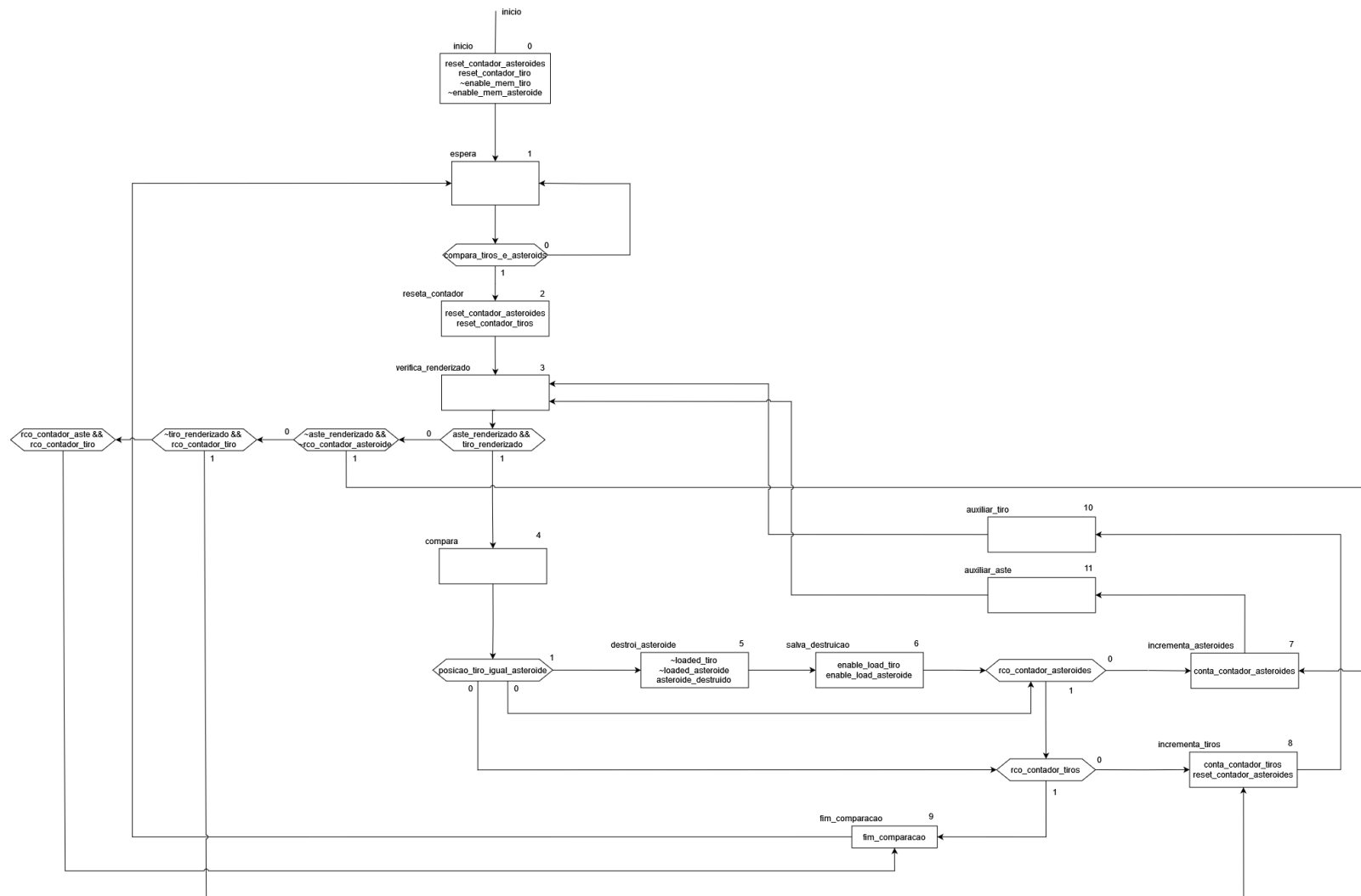


Figura 6 - Unidade de Controle que Coordena as Colisões





*Figura 8 - Unidade de Controle que Movimenta os Tiros*

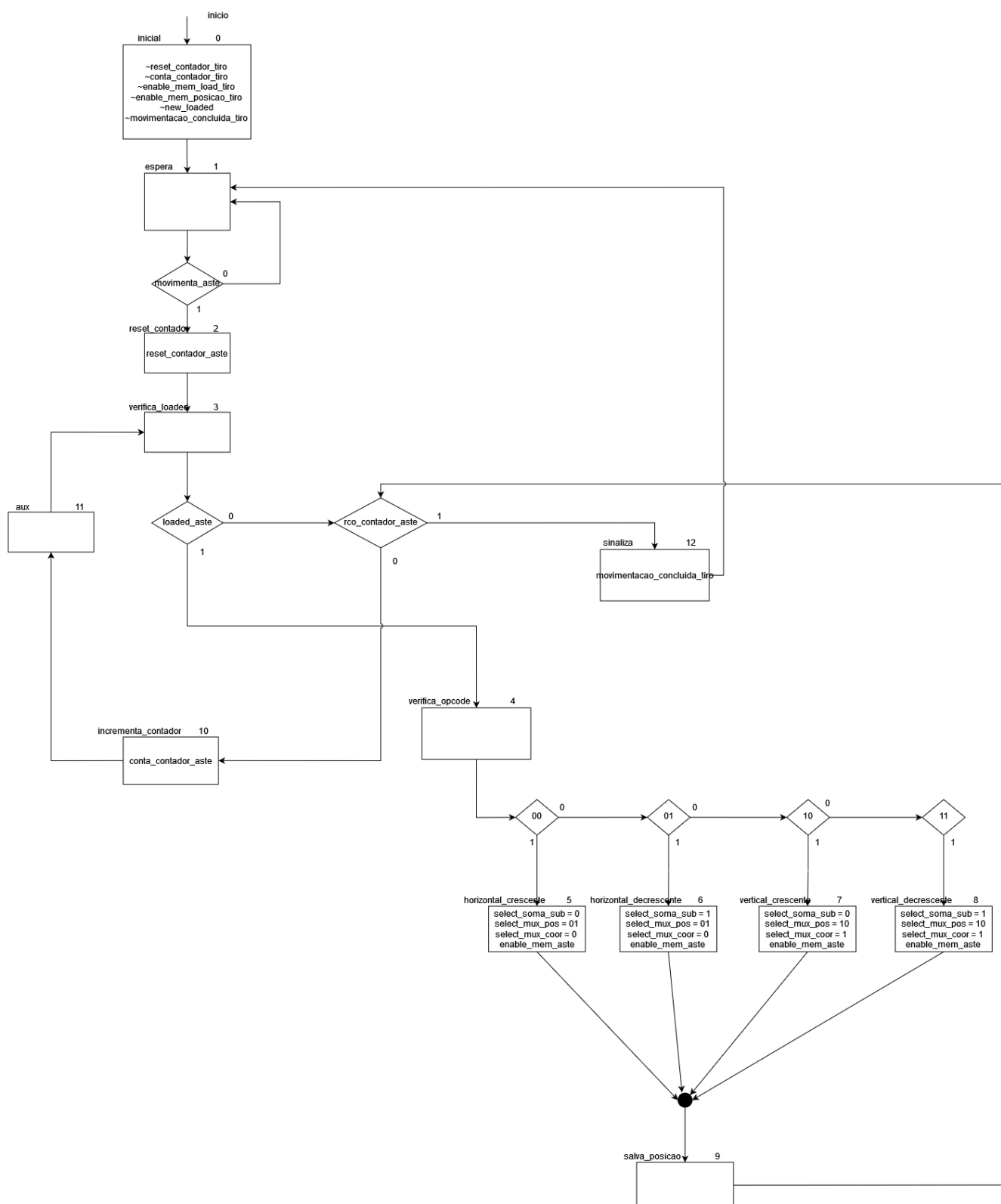


Figura 9 - Unidade de Controle que Movimenta os Asteroides

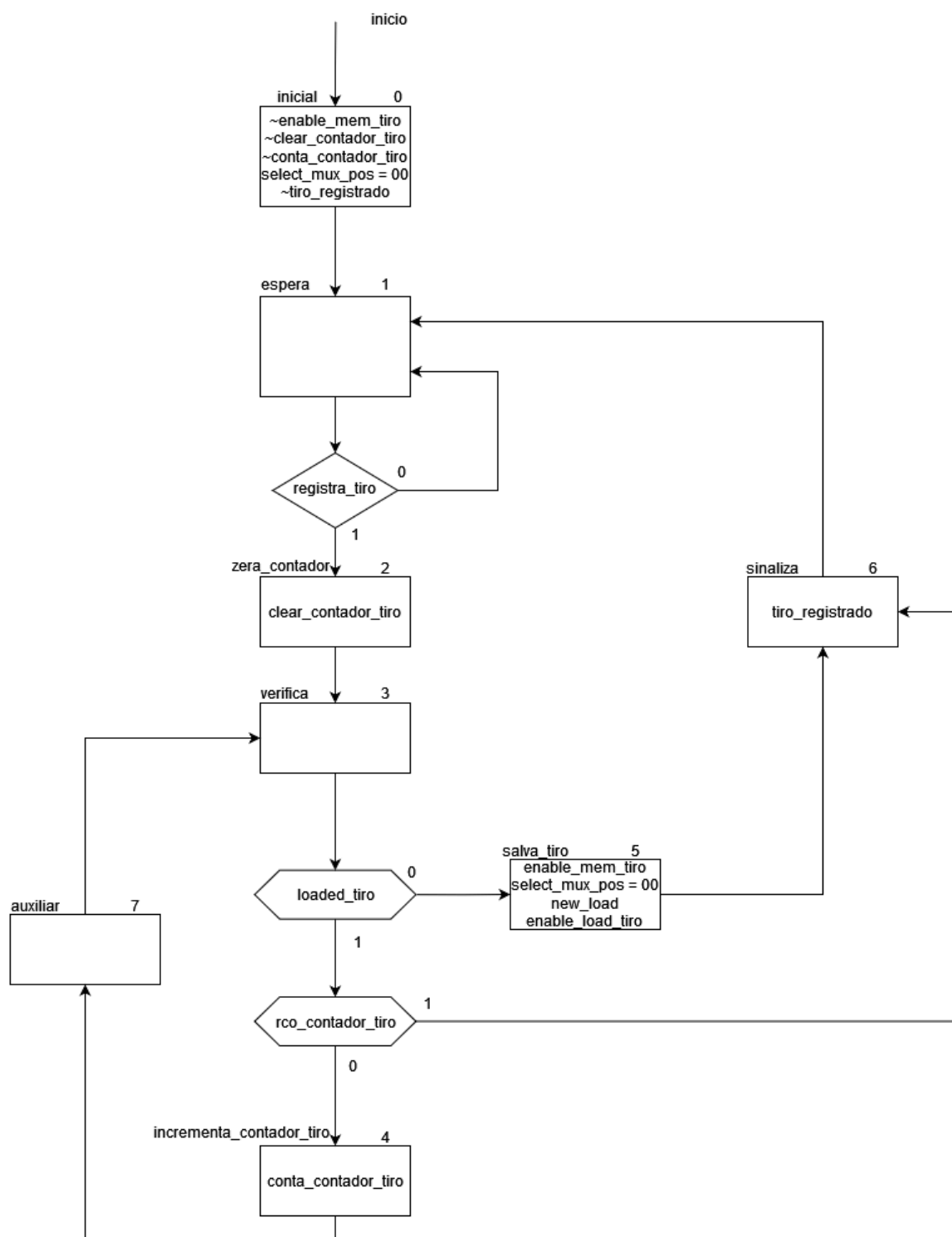
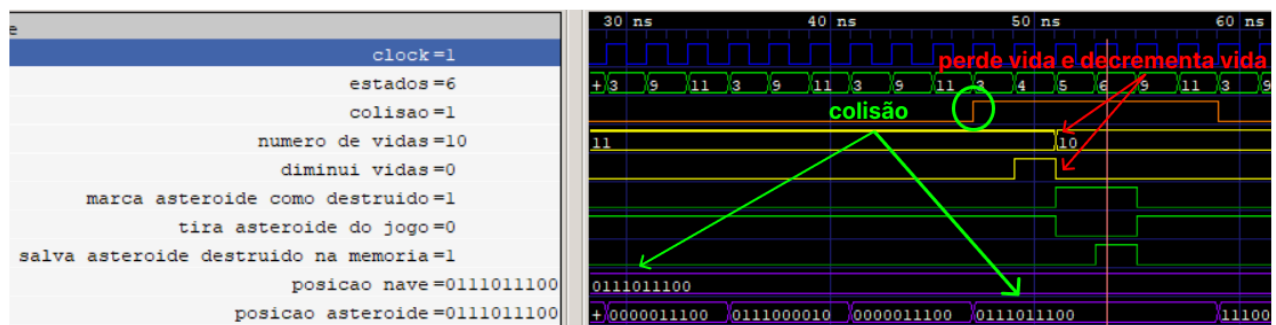


Figura 10 - Unidade de Controle que Registra os Tiros

## 2.3 Testes

Para verificar o comportamento do circuito, optamos por utilizar testbenches feitos em Verilog e, em seguida, visualizar suas formas de onda usando o GTKWave. Foram criados dois casos de teste para esta entrega: um para verificar a perda de vida e a destruição do asteroide, e outro para testar o tratamento do movimento do asteroide.



*Figura 11 - Teste de Perda de Vida e Destruição do Asteroide*

Primeiro foram zerados todos os sinais de entrada da unidade de controle e após isso foi resetou-se a mesma. Após isso iniciou-se o teste em que há colisão e há vidas, ou seja, é necessário decrementar vidas e destruir o asteroide e não renderizar mais o asteroide.

Para os testes da unidade de controle que é responsável por movimentar os tiros, inicialmente foi dado um valor inicial para todas as entradas e após isso foi reiniciada a máquina de estados. Foi iniciado a máquina de estados e é possível observar as formas de ondas na figura abaixo:



*Figura 12 - Teste de Movimentação dos tiros*

A semelhança do fluxo de dados do módulo asteroide e do módulo tiro torna a unidade de controle responsável por movimentar os asteroides muito semelhante à unidade de controle responsável por movimentar os tiros, por conta disso, seria redundante adicionar estes testes nesse planejamento. Os testes da movimentação dos asteroides também foram consistentes.

### 3. Planejamento da aula prática

### 4. Relatório

### 5. Cronograma

É evidente que, devido à alta complexidade assumida pelo circuito nesta última semana, pode parecer que pouco foi implementado. No entanto, foi dedicado bastante tempo ao design do projeto para garantir que ele funcionasse de acordo com os requisitos e para evitar a necessidade de readaptações futuras. Por esse motivo, dois requisitos não funcionais foram implementados por proxy, incluindo a escalabilidade do circuito e sua fácil manutenção, graças à alta modularidade implementada.

Além disso, foi concluído o sistema de geração de asteroides, que estava pendente da semana 1, e finalizamos o sistema de destruição de asteroides e execução de tiros, conforme planejado para a semana 2. O único item que ainda falta da semana 2 é o sistema de pontuação.



Não pretende-se modificar o cronograma para a semana 3, pois ainda há refinamentos adicionais a serem feitos no circuito atual. No entanto, espera-se que, uma vez feitos esses refinamentos, o resto dos requisitos seja de mais fácil implementação.

Cronograma do projeto Resumido	
Atividade	Semana
<div>Finalizado ▾ - RF1 (Implementação Física na placa FPGA)</div> <div>Finalizado ▾ - RF2 (Controle da Nave Espacial)</div> <div>Finalizado ▾ - RF5 (Sistema de Geração de Asteroides)</div> <div>Finalizado ▾ - RF3 (Sistema de Vidas)</div>	06/03 a 13/03
<div>Finalizado ▾ - RF7 (Tiro Realizado)</div> <div>Finalizado ▾ - RF6 (Destruição do Asteroide)</div> <div>Não iniciado ▾ - RF4 (Sistema de Pontuação)</div> <div>Finalizado ▾ - RNF3 (Escalabilidade)</div> <div>Finalizado ▾ - RNF4 (Manutenção)</div> <div>Finalizado ▾ - RF11 (Hitbox)</div>	13/03 a 20/03
<div>Não iniciado ▾ - RF8 (Registro de Pontuação dos Players na memória)</div> <div>Não iniciado ▾ - RF9 (Menu de Interação)</div> <div>Não iniciado ▾ - RF10 (Monitor)</div>	20/03 a 27/03
<div>Não iniciado ▾ - Correções de bugs e implementações finais</div> <div>Não iniciado ▾ - RNF5 (Som atrativo)</div> <div>Não iniciado ▾ - Montagem do protótipo para o produto final</div>	27/03 a 03/04