

Thinking Strategy for Verilog-Based AI Control System

1. Understand the System as a Set of Agents

Each ship is an independent agent with rules:

- They can accelerate, fire, shield, or cloak.
- They have limited energy.
- They see visible enemies and know who is cloaked or destroyed.

Design each ship as an FSM reacting to the state.

2. Break Down Inputs and Outputs

Inputs per ship:

- Own position, energy
- Enemy positions (if visible), enemy cloaked flags

Outputs per ship:

- Fire (directional)
- Accelerate
- Shield
- Cloak

Think: inputs -> decision -> outputs

3. Define Behaviors and Prioritize Decisions

Strategy layers:

Offense: Fire if enemy visible and aligned

Defense: Cloak/shield if low energy or under threat

Mobility: Move to engage or escape

Use priority logic:

if (can fire) fire;

else if (should shield) shield;

else if (should cloak) cloak;

else move;

4. Model in Verilog: Loop Over Ships

Use a loop over 3 ships:

- Read data for each ship
- Compare with enemies
- Apply decision logic based on rules

Helps in code reuse and structured logic.

5. Simplify Actions into Conditions

Examples:

Firing:

if (enemy_visible && aligned && energy > 10) fire;

Shield:

if (enemy_near && energy > 5) shield;

Cloak:

if (energy < 5) cloak;

Accelerate:

move_towards(enemy) or avoid edges.

6. Energy Is Your Resource Manager

Every action costs energy. Add energy checks:

if (energy >= FIRE_COST) fire;

else if (energy >= SHIELD_COST) shield;

Helps avoid draining power and getting destroyed.

7. Simulation is Key

Use test inputs for:

- Enemy positions

- Energy levels
- Cloaked flags

Simulate output behavior to validate decisions.

Mindset Checklist

- Each ship is an FSM agent
- Inputs = world state
- Outputs = actions
- Energy = constraint
- Strategy = conditional logic
- Loop through ships, apply rules
- Test through simulation