**Collision:**

The collision entity detects when a bullet fired from one tank intersects with the other tank. Two collision entities are instantiated, one for each tank. The inputs are tank\_x and tank\_y, (the top-left x and y coordinates of a tank), bullet\_x and bullet\_y (the center of the bullet), clk, reset, and pulse. The one output is a one-bit flag is\_hit. The bullet is much smaller than the tank and collision is detected just on a single point.

entity collision is

port (

tank\_x : in std\_logic\_vector(9 downto 0);

tank\_y : in std\_logic\_vector (8 downto 0);

bullet\_x : in std\_logic\_vector(9 downto 0);

bullet\_y : in std\_logic\_vector(8 downto 0);

clk : in std\_logic;

reset : in std\_logic;

pulse : in std\_logic;

is\_hit : out std\_logic

);

end entity;

Calculating collision was implemented with a finite state machine that consisted of two states: idle and waitOnPulseLow. The idle state does nothing unless a pulse is detected at which point it takes in the current tank and bullet positions and checks for a collision. A collision is detected simply by comparing the x-positions of the bullet and tank to see if the bullet’s x-position is within the tank’s x-position. Both sides of the tank must be checked so the constant TANK\_WIDTH from the tank package is used. If the bullet currently is within the left and right edge of the tank, the same comparison is done for the y-positions (using the constant TANK\_HEIGHT). If there is overlap in the y-positions, then the is\_hit flag is set (the signal is clocked, so only set on a new clock edge) and the state changes to waitOnPulseLow. If no collision is detected, the is\_hit flag is set to 0 and the state does not change. The waitOnPulseLow is a state that holds the is\_hit signal for the entire pulse width so that is available to other components (mainly score).