Portland State
UNIVERSITY

ECE 593 Spring 2022 Assignment -1 Report

Group 16

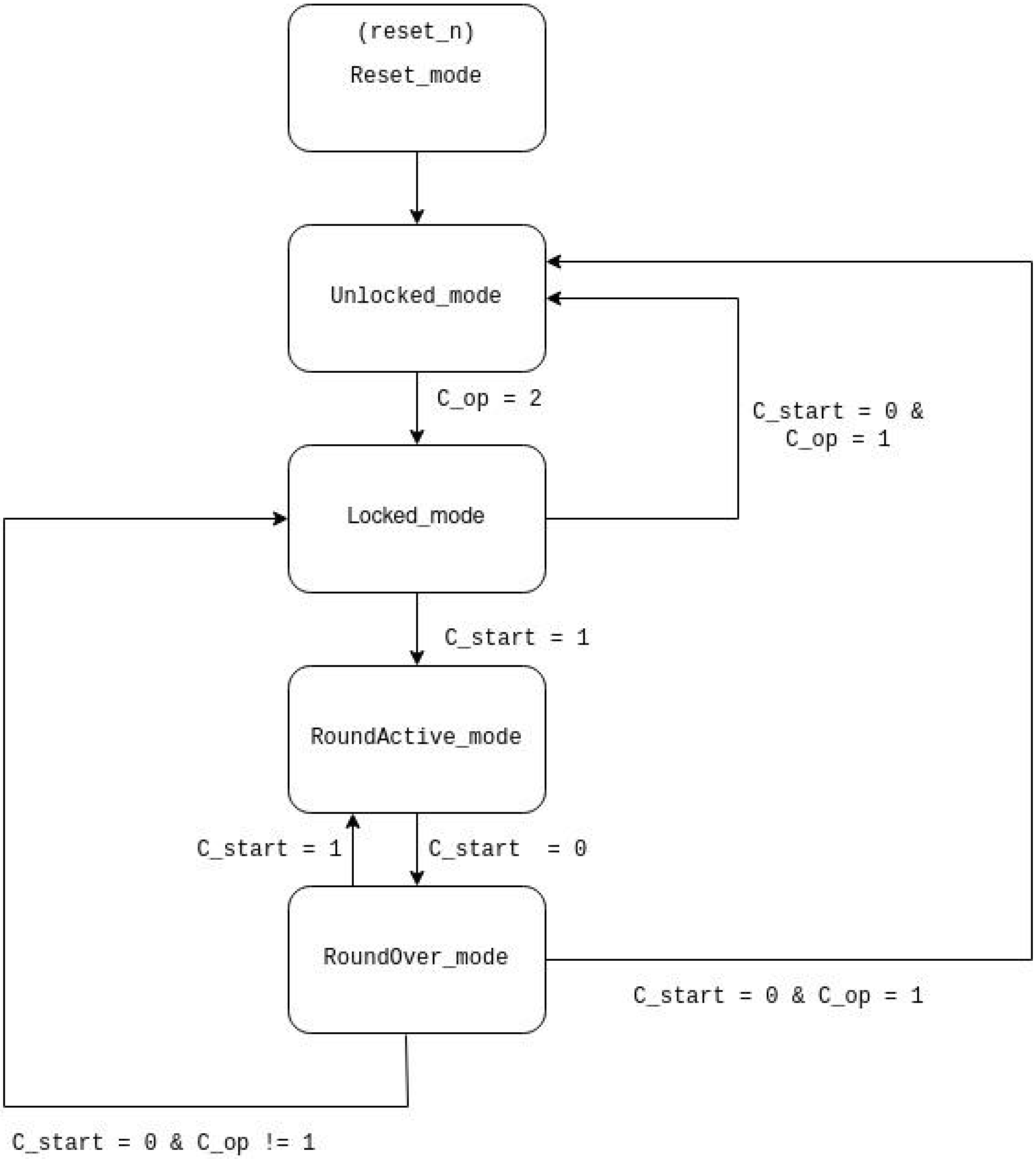
BIDS22

Sreeja Boyina
PSU ID: 912891363
sreeja@pdx.edu

Supreet Gulavani
PSU ID: 949311031
sg7@pdx.edu

Implementation

State Diagram:



- With reference to the above state diagram, the state when the reset_n is asserted, the code enters the Reset_mode
- In Reset_mode, all the output variables are set to 0 and the registers are set as follows:
    - mask = 3'b111
    - timer = 4'b1111
    - key = 0
    - bid_cost = 1
    - X_value = 0
    - Y_value = 0
    - Z_value = 0
- The next state is Unlocked_mode. In this state, we can update all the register values based on the opcodes i.e X_value, Y_value, Z_value, mask, timer, bid_cost
- From Unlocked_mode, we can only transition to Locked_mode if the opcode C_op is 2
- C_start cannot be asserted when in Unlocked_mode, err (error) would be asserted if done so
- Once in Locked_mode, if C_start is asserted, the rounds would begin by transitioning to RoundActive_mode else key would be checked, if validated would go to Unlocked_mode else would stay in Locked_mode
- If RoundActive_mode is entered, the bid signals if asserted for every player, retract signal not asserted can the player bid
- Participation of the player can only be determined if the mask bits are set likewise i.e mask[2] would mean X can participate, mask[1] Y, mask[0] Z
- _ack signals would go high, meaning no err (error) and otherwise if mask bits not set for that particular player
- Once the _bidAmt (bid amounts) are accepted, and C_start is deasserted, transition to RoundOver_mode would take place.
- In RoundOver state, the maxBid would be calculated, the winner would be declared, _balances would be updated for every player, and roundOver would go high meaning the round is over
- If C_start is asserted, the transition to RoundActive_mode would take place for another round else C_op would be checked for unlock and key would be validated to go to the Unlocked_mode , if bad would go to Locked_mode

Test Cases

Following are few scenarios that could be tested to check the functionality of the bid controller

- Every state is reached at least once
- Loading of internal value registers, timer, bid cost, mask registers by providing correct `C_op`
- Incorrect inputs i.e bad key, trying to unlock while being unlocked, asserting C_start when unlocked, invalid operation
- `_err(`X, Y, Z) : checking for inactive rounds, insufficient balance and invalid requests
- Equal bids from the players