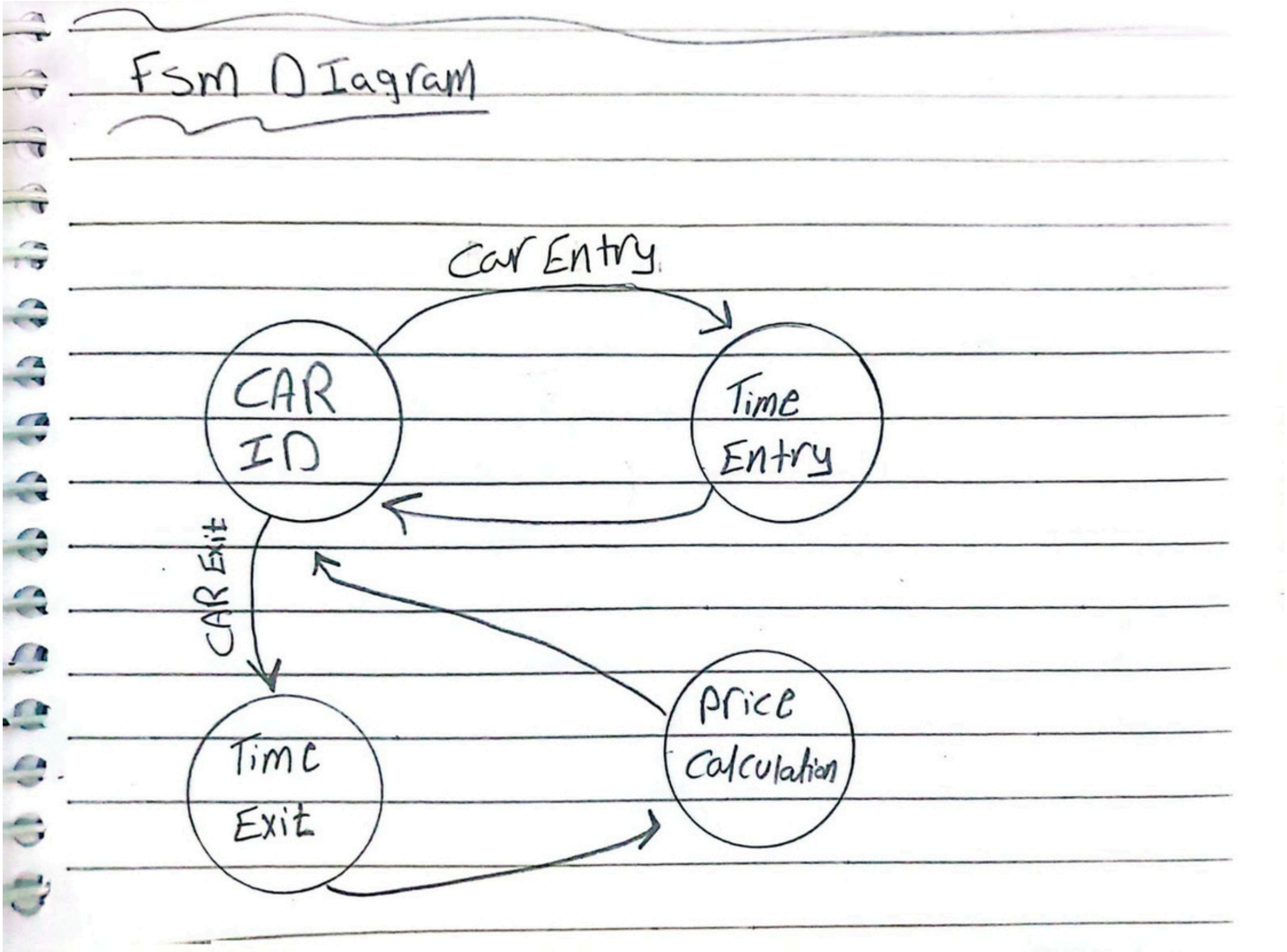
TABELS:

CARS	CAR ID	PRICE CALCULATION
CAR A	00	1000101(69$)
CAR B	01	101001(41$)
CARC	10	110111(55$)

HINT: PRICE CALCULATION= (TIME EXIT-TIME ENTERY)*1

HINT:PRICE CALCULATION BY 32 BIT

DIAGRAM:



CODE COMMENT:

- □ **module of alu :**

this module can make and gate, or gate and make add , subract, multibly,greater than less than and equal

- □ **module of cost calculation:**

This module appears to be part of a parking or toll system where:

- entry_time represents when a vehicle entered
- exit_time represents when the vehicle exited
- The cost is calculated as the simple time difference

- □ **module of car counter:**

What it does: Counts how many cars are in the parking lot

How it works:

  - ○ Starts at 0 cars when turned on (reset)
  - ○ Adds 1 when a car enters (up to maximum 3 cars)
  - ○ Subtracts 1 when a car leaves (won't go below 0)
  - ○ Has two lights:
    - ▪ "Full" light turns on when 3 cars are inside
    - ▪ "Empty" light turns on when no cars are inside

Timer Counter (Incomplete Part)

- This part seems to be for measuring how long cars stay
- But the counting mechanism isn't shown in the code

Real-World Comparison

Imagine a parking lot with:

- A gate that counts cars going in/out
- A display showing "0/3", "1/3", up to "3/3 FULL"
- The system prevents more than 3 cars from entering

- □ **module of car buffer**

How It Works:

1. **3 Memory Slots** (for 3 cars: Car 0, Car 1, Car 2)
2. **When a Car Enters:**
   - ○ You press "record" (store_entry=1)
   - ○ Choose which car (0, 1, or 2) using car_id
   - ○ The system saves the current time (entry_time) for that car
3. **When You Need the Time:**
   - ○ Ask "When did Car X arrive?"
   - ○ The system shows you the stored time (exit_entry_time)
4. **Reset Button:**
   - ○ Press rst to erase all saved times (sets everything to 0)

□ **module of divider**

1. **Input:** A super-fast clock signal (like a hummingbird's wings flapping)
2. **Counts 6,250,000 ticks** of the fast clock
3. **Flips the output** (ON→OFF or OFF→ON) every time it reaches that number
4. **Result:** A slow "blinking" signal that changes every **¼ second** (250ms)

□ **module of parking system**

1. Three Main Parts

- **Clock** - Tracks time (like a stopwatch)
- **Car Counter** - Counts cars (0 to 3 max)
- **Memory** - Remembers when each car arrived

2. What It Checks

If a car enters:

- Counter goes up (+1)
- Saves the arrival time

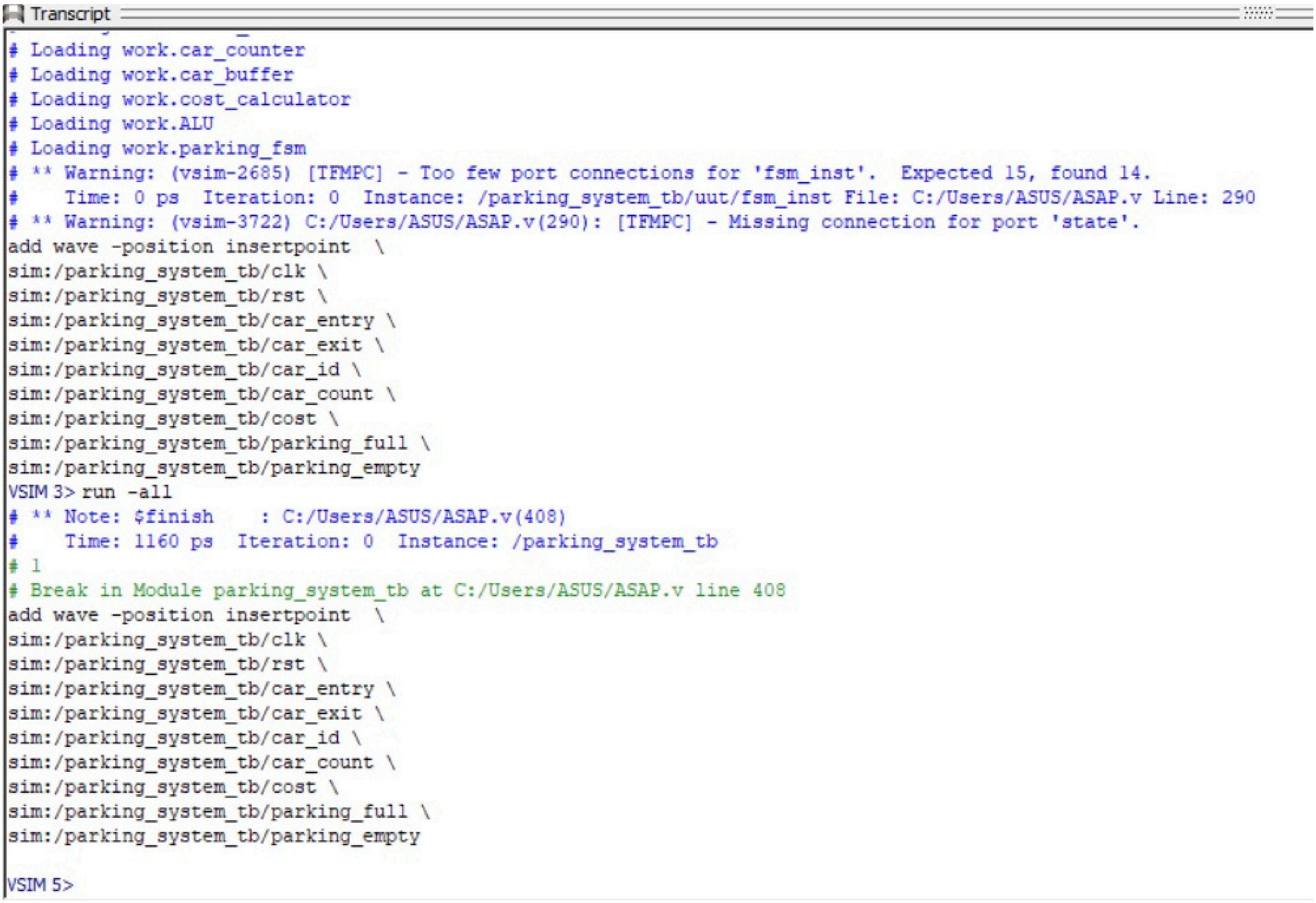
If a car leaves:

- Counter goes down (-1)
- (Later, it could calculate parking fees)

Shows "FULL" when 3 cars are inside
Shows "EMPTY" when no cars

RESULT:

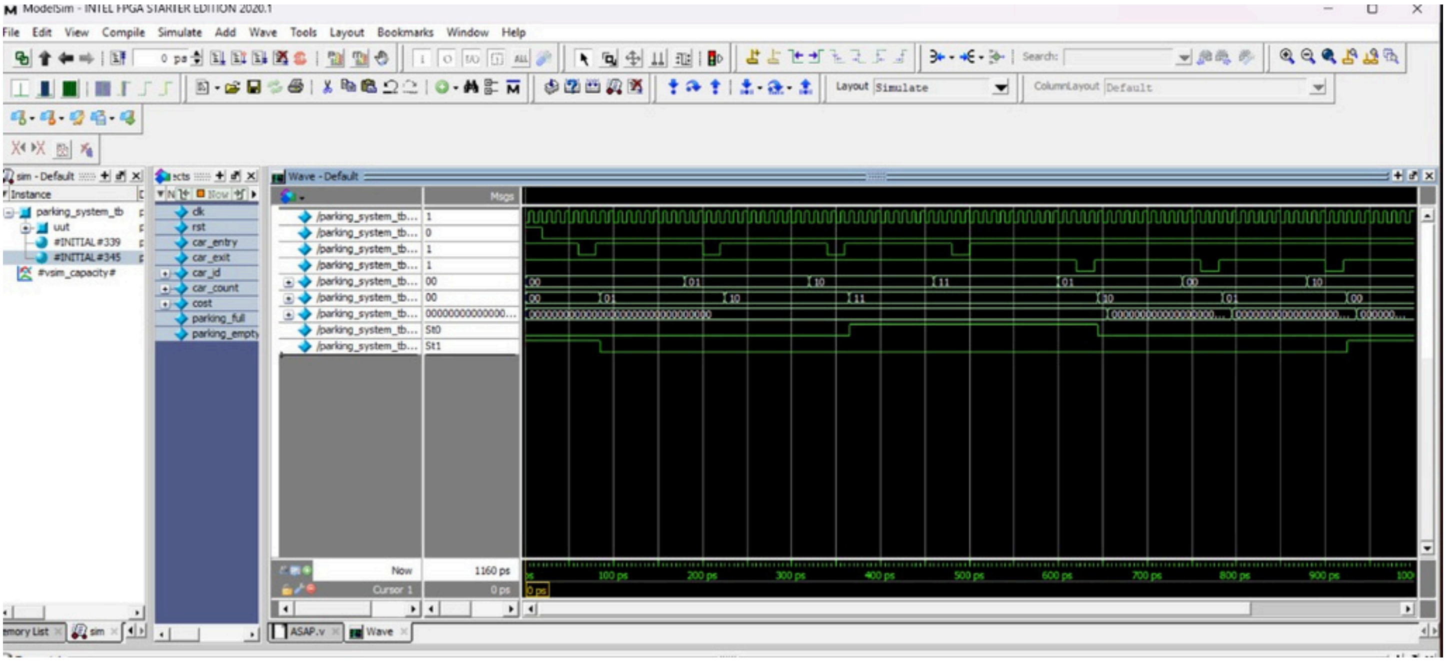
code compile successfully



```
# Loading work.car_counter
# Loading work.car_buffer
# Loading work.cost_calculator
# Loading work.ALU
# Loading work.parking_fsm
# ** Warning: (vsim-2685) [TFMPC] - Too few port connections for 'fsm_inst'.  Expected 15, found 14.
#    Time: 0 ps  Iteration: 0  Instance: /parking_system_tb/uut/fsm_inst File: C:/Users/ASUS/ASAP.v Line: 290
# ** Warning: (vsim-3722) C:/Users/ASUS/ASAP.v(290): [TFMPC] - Missing connection for port 'state'.
add wave -position insertpoint  \
sim:/parking_system_tb/clk \
sim:/parking_system_tb/rst \
sim:/parking_system_tb/car_entry \
sim:/parking_system_tb/car_exit \
sim:/parking_system_tb/car_id \
sim:/parking_system_tb/car_count \
sim:/parking_system_tb/cost \
sim:/parking_system_tb/parking_full \
sim:/parking_system_tb/parking_empty
VSIM 3> run -all
# ** Note: $finish    : C:/Users/ASUS/ASAP.v(408)
#    Time: 1160 ps  Iteration: 0  Instance: /parking_system_tb
# 1
# Break in Module parking_system_tb at C:/Users/ASUS/ASAP.v line 408
add wave -position insertpoint  \
sim:/parking_system_tb/clk \
sim:/parking_system_tb/rst \
sim:/parking_system_tb/car_entry \
sim:/parking_system_tb/car_exit \
sim:/parking_system_tb/car_id \
sim:/parking_system_tb/car_count \
sim:/parking_system_tb/cost \
sim:/parking_system_tb/parking_full \
sim:/parking_system_tb/parking_empty

VSIM 5>
```

The wave form of the code



ModelSim - INTEL FPGA STARTER EDITION 2020.1
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
Layout Simulate
ColumnLayout Default
sim - Default
Instance
parking_system_tb
uut
#INITIAL#339
#INITIAL#345
#vsim_capacity#
dk
rst
car_entry
car_exit
car_id
car_count
cost
parking_full
parking_empty
Wave - Default
Msgs
/parking_system_tb... 1
/parking_system_tb... 0
/parking_system_tb... 1
/parking_system_tb... 1
/parking_system_tb... 00
/parking_system_tb... 00
/parking_system_tb... 00000000000000...
/parking_system_tb... St0
/parking_system_tb... St1
Now 1160 ps
Cursor 1 0 ps
100 ps
200 ps
300 ps
400 ps
500 ps
600 ps
700 ps
800 ps
900 ps
ASAP.v
Wave