TÉCNICO LISBOA

Computer Electronics

Lecture 9: Digital Circuits and Verilog IV

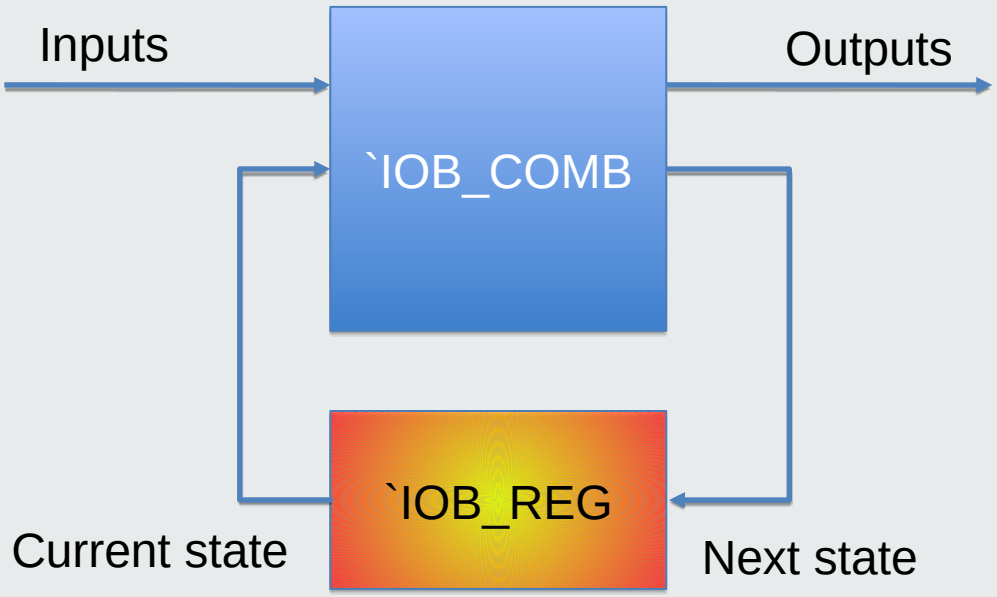
Finite State Machines

Create circuits with iob-lib

- By combining iob-lib modules and macros, one can create many basic digital circuits
- Using the `IOB_VAR to compute signal values in `IOB_COMB, gives the user the true meaning of the Verilog ‘reg’ type and the ‘always @*’ statement
  - ‘reg’ is a data type to hold the result a of computation; it does not necessarily represent a physical wire and it NEVER represents a register despite its unfortunate name.

Finite State Machines

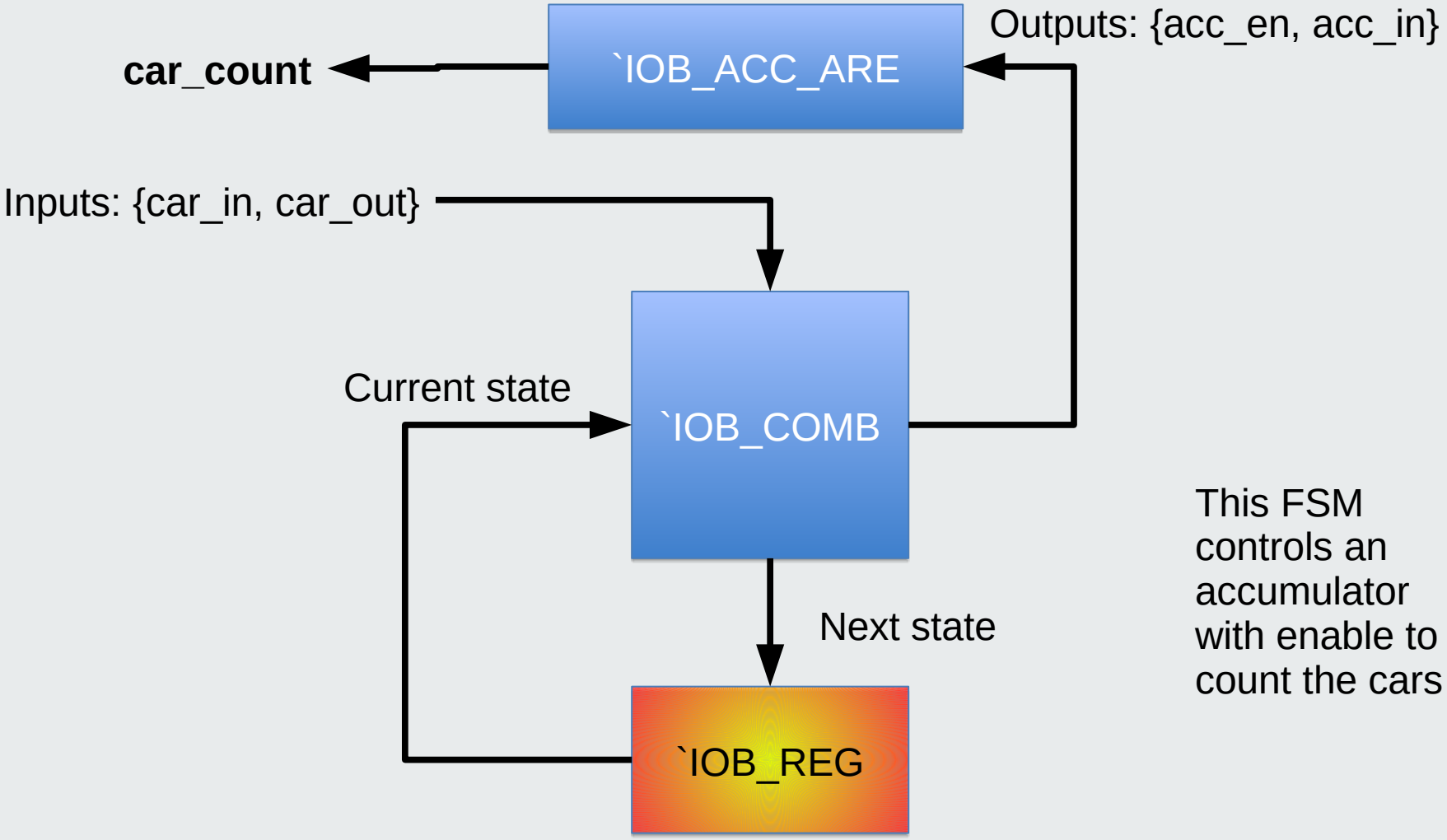
- Finite State Machines (FSMs) should be created when it is not obvious how to make the circuit with the basic iob-lib components
- An FSM is basically an `IOB_COMB block connected to an `IOB_REG block



FSM Types: Moore and Mealy

- Moore: Outputs = F(Current State)
  - Advantages:
    - Critical path does not include inputs so one has better control on timing closure
    - One state / one output is sometimes easier to design
  - Disadvantages
    - May need a few state transitions (1 clock cycle each) to produce the right output: slower reaction
    - More states = more logic: more registers, more state decoding combinatorial circuits
- Mealy: Outputs = F(Inputs, Current State)
  - Advantages:
    - Fewer states = less logic: less registers, less state decoding combinatorial circuits
    - Needs less state transitions (1 clock cycle each) to produce the right output: faster reaction
  - Disadvantages
    - critical path includes inputs so one has worse control on timing closure
    - One state multiple possible outputs is sometimes more difficult to design

Example: counting cars in a parking lot



This FSM controls an accumulator with enable to count the cars

Moore: conclusions

- Needs 3 states
- State encoding done manually and conveniently
- Output decode logic avoided as outputs can be derived from state bits using trivial logic
- Next state decoding needs *logic synthesis* using:
  - Boolean algebra
  - Quine-McCluskey method
  - Karnaugh maps, etc
- Logic synthesis tools can do it very well from the Verilog code

Moore: Verilog

```
module moore (
  `IOB_INPUT(car_in, 1),        `IOB_INPUT( car_out, 1),
  `IOB_OUTPUT( acc_en, 1), `IOB_OUTPUT(acc_in, 2),
);
        `IOB_VAR(state, 2)
        `IOB_VAR(state_nxt, 2)
        localparam idle=0, incr=1, decr=3;
        `IOB_COMB begin
                state_nxt = state; //default assignment
                case(state)
                        idle:
                                if(car_in & ~car_out)
                                        state_nxt = incr;
                                else if(~car_in & car_out)
                                        state_nxt = decr
                        incr:
                                if(~car_in & car_out)
                                        state_nxt = decr
                                else if(car_in ^car_out)
                                state_nxt = idle;
                        decr:
                                if(car_in & ~car_out)
                                        state_nxt = incr
                                else if(car_in ^ car_out)
                                        state_nxt = idle;
                        default: //in principle this state is never reach
                                state_nxt = idle;
                endcase
        assign acc_en = (state == incr) || (state == decr);
        assign acc_in[0] =state[0];
        assign acc_in[1] = state[1];
endmodule
```

Mealy: conclusions

- For this example, the Mealy formulation leads to zero states because outputs can be a function of the inputs only
- In general, Mealy machines require less states than Moore's, not always zero states as in this example
- In general, the output decode logic is more complex than Moore's because the logic functions have more inputs; in this example, the output decode logic is trivial but this is not always the case
- Like Moore's, the next state and output decoding need *logic synthesis* using:
  - Boolean algebra
  - Quinn-mcluskey method
  - Karnough maps, etc
- Logic synthesis tools can do it very well from the Verilog code

Mealy: Verilog implementation

```
//Mealy formulation requires zero states in this case

module mealy (
  input car_in,
  input car_out,
  output acc_en,
  output [1:0] acc_in
);

assign acc_en = car_in^car_out;

assign acc_in[0] =acc_en;

assign acc_in[1] = acc_en & car_out;

endmodule
```