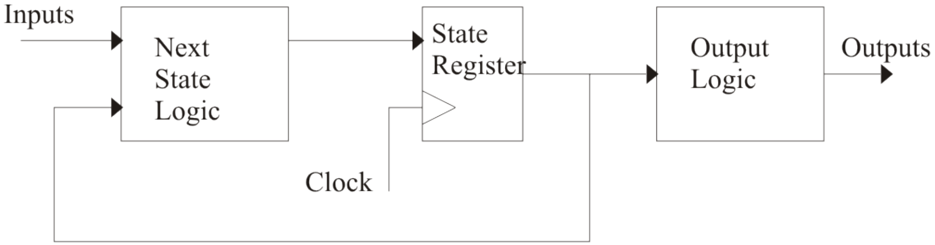
Electronics and
Computer Science

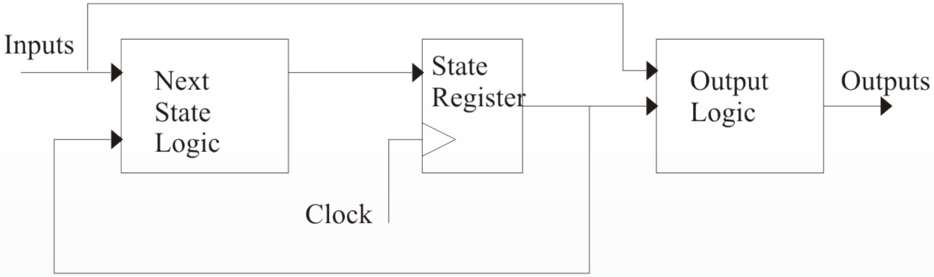
UNIVERSITY OF
Southampton

6.2 More State Machines

Other models



Moore Machine



Mealy Machine

- Separate always_comb for Next State and Output
- One always_ff for everything (but outputs will be flip-flops!)

Version 2 of Traffic Light Controller

```
enum {G, R} state;
always_ff @(posedge clock,
            negedge n_reset)
  begin: SEQ
  if (!n_reset)
    state <= G;
  else
    unique case (state)
        G:  if (car)
              state <= R;
        R:  if (timed)
              state <= G;
        endcase
  end
```

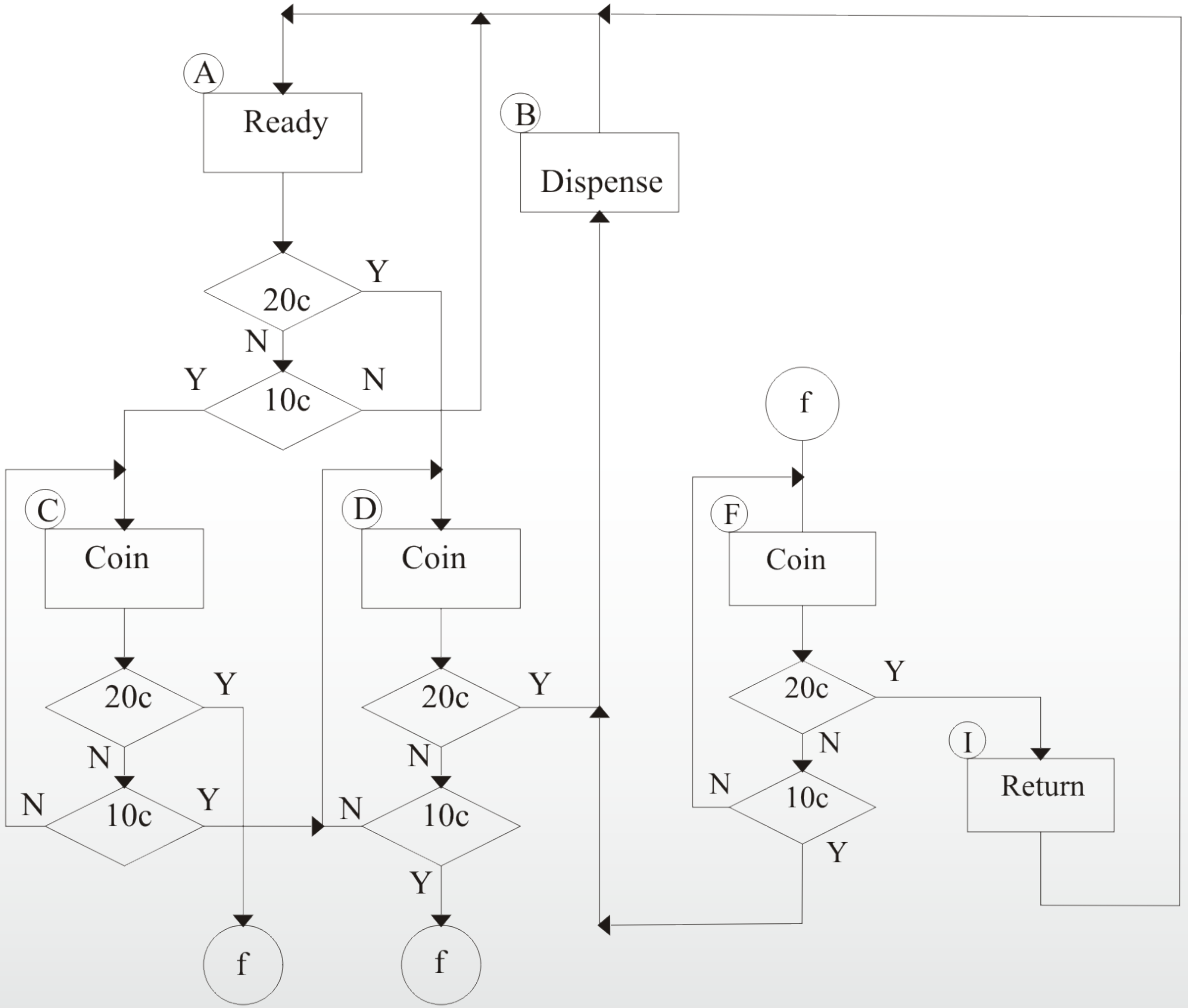
```
  always_comb
    begin: OP
    start_timer = '0;
    minor_green = '0;
    major_green = '0;
    unique case (state)
      G: begin
         major_green = '1;
         if (car)
           start_timer = '1;
         end
      R:
         minor_green = '1;
    endcase
  end
endmodule
```

Comments on version 2

- Next state logic is included with state registers
  - (Not Mealy or Moore)
- Works on all synthesis tools
- Some duplication of structure in the code

Vending Machine

- A drink costs 40c. (Euro cents.) The machine accepts 20c and 10c coins (all other coins are rejected by the mechanics of the system).
- Once 40c have been inserted, the drink is dispensed. If more than 40c are inserted, all coins are returned.
- The machine has two lights: one to show that it is ready for the next transaction, and one to show that further coins need to be inserted.


A
Ready
B
Dispense
20c
Y
N
10c
Y
N
f
C
Coin
D
Coin
F
Coin
20c
Y
N
10c
N
Y
20c
Y
N
10c
N
Y
20c
Y
N
10c
N
Y
I
Return
f
f

```
module vending(output logic ready,dispense,ret,coin
               input logic clock,n_reset,twenty,ten);

  enum {A, B, C, D, F, I} present_state, next_state;

always_ff @(posedge clock, negedge n_reset)
  begin: SEQ
  if (~n_reset)
    present_state <= A;
  else
    present_state <= next_state;
  end
```

```
always_comb
  begin: COM
  ready = '0;
  dispense = '0;
  ret = '0;
  coin = '0;
  unique case (present_state)
    A : begin
        ready = '1;
        if (twenty)
          next_state = D;
        else if (ten)
          next_state = C;
        else
          next_state = A;
        end
    B : begin
        dispense = '1;
        next_state = A;
        end
    C : begin
        coin = '1;
        if (twenty)
          next_state = F;
        else if (ten)
          next_state = D;
        else
          next_state = C;
        end
    D : begin
        coin = '1;
        if (twenty)
          next_state = B;
        else if (ten)
          next_state = F;
        else
          next_state = D;
        end
    F : begin
        coin = '1;
        if (twenty)
          next_state = I;
        else if (ten)
          next_state = B;
        else
          next_state = F;
        end
    I : begin
        ret = '1;
        next_state = A;
        end
  endcase
  end
endmodule
```

For you to do

- Rewrite the traffic light controller to use a single always_ff block (and no always_comb blocks).
  - Why does this behave a little differently to the other versions?