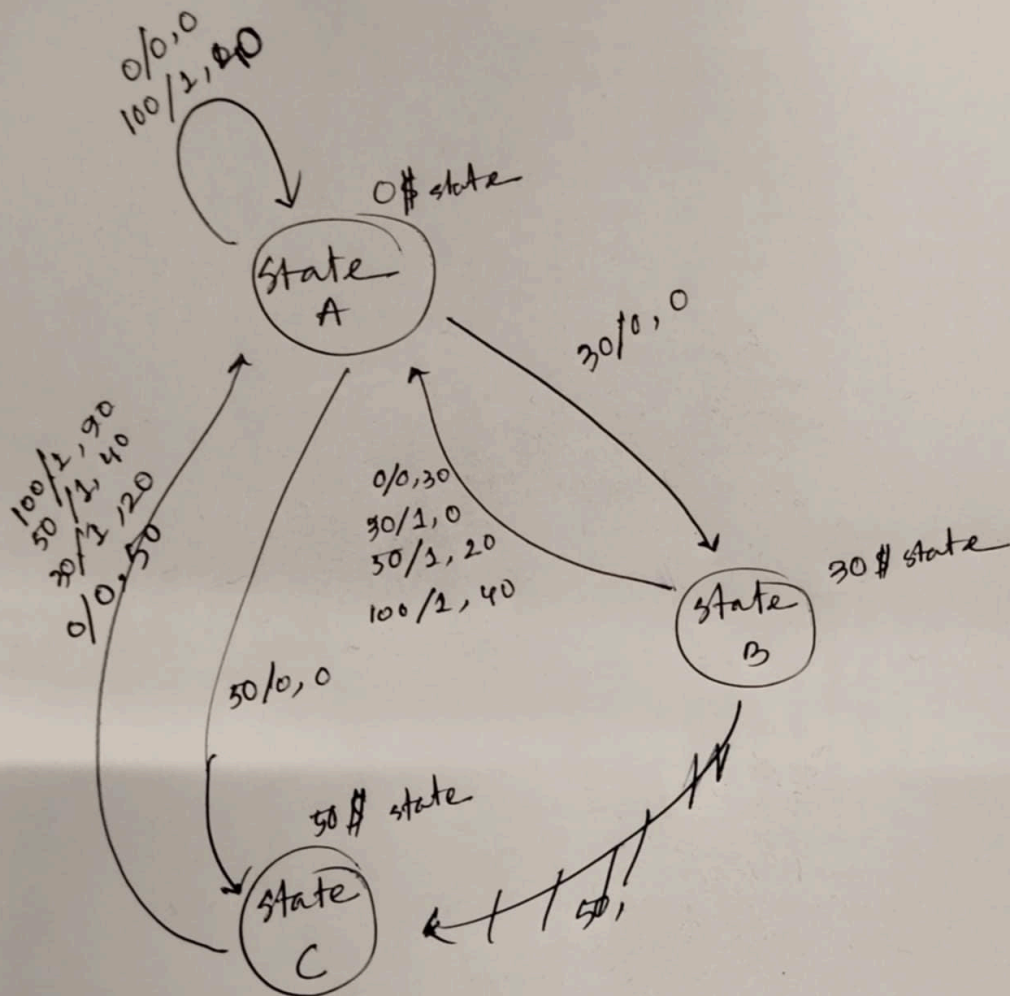
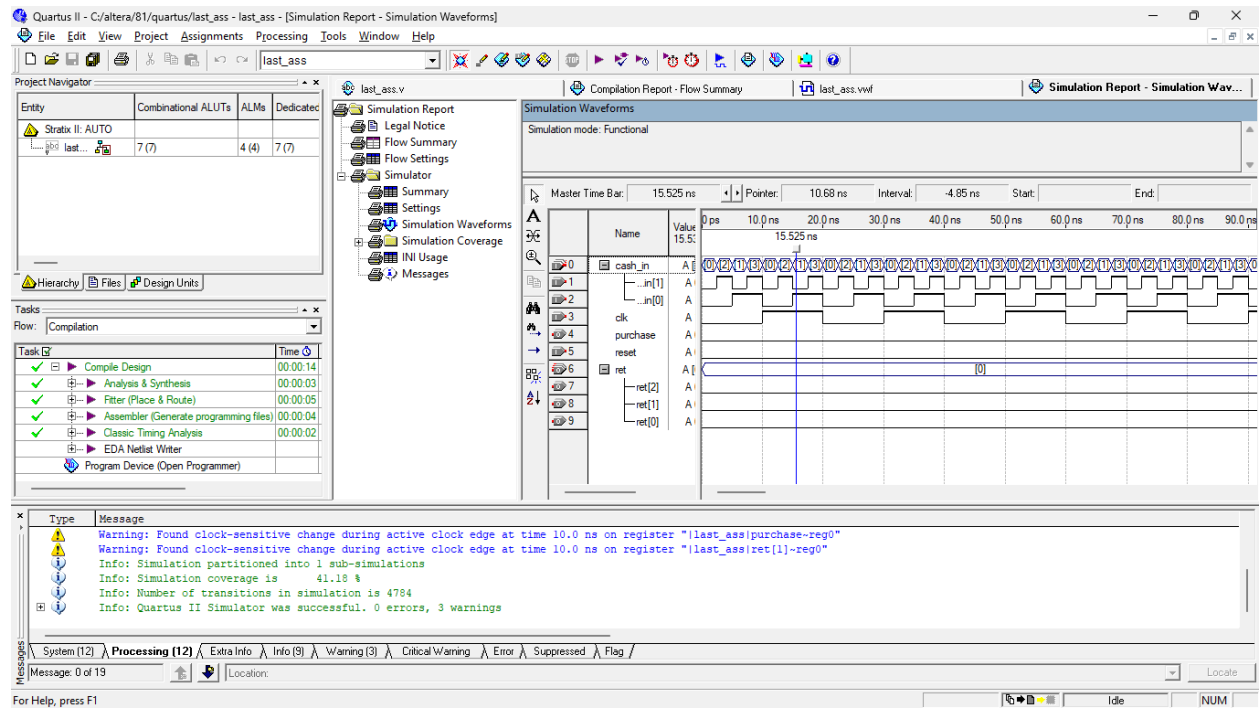


60 take product





Code

```
module last_ass(purchase, ret, cash_in, clk, reset);
```

```
    input clk, reset;
    input [1:0] cash_in;
    output reg purchase;
    output reg [2:0] ret;

    reg [1:0] state;
    parameter IDLE = 2'b00,
               WAIT_30 = 2'b01,
               WAIT_50 = 2'b10,
               DISPENSE = 2'b11,
               R0 = 3'b000,
               R20 = 3'b001,
               R30 = 3'b010,
               R40 = 3'b011,
               R50 = 3'b100,
               R70 = 3'b101,
               R90 = 3'b110;
```

```
    always@(posedge clk or posedge reset)
```

```

begin
  if (reset)
    begin
      state <= IDLE;
      purchase <= 0;
      ret <= R0;
    end
  else
    begin
      case(state)
        IDLE:
          case(cash_in)
            2'b00: begin
              state <= IDLE;
              purchase <= 0;
              ret <= R0;
            end
            2'b01: begin
              state <= WAIT_30;
              purchase <= 0;
              ret <= R0;
            end
            2'b10: begin
              state <= WAIT_50;
              purchase <= 0;
              ret <= R0;
            end
            2'b11: begin
              state <= IDLE;
              purchase <= 1;
              ret <= R40;
            end
            default: begin end
          endcase
        endcase

      WAIT_30:
        case(cash_in)
          2'b00: begin
            state <= IDLE;
            purchase <= 0;
            ret <= R30;
          end
          2'b01: begin
            state <= IDLE;

```

```

        purchase <= 1;
        ret <= R0;
    end
    2'b10: begin
        state <= IDLE;
        purchase <= 1;
        ret <= R20;
    end
    2'b11: begin
        state <= IDLE;
        purchase <= 1;
        ret <= R70;
    end
    default: begin end
endcase

```

```

WAIT_50:
case(cash_in)
    2'b00: begin
        state <= IDLE;
        purchase <= 0;
        ret <= R50;
    end
    2'b01: begin
        state <= IDLE;
        purchase <= 1;
        ret <= R20;
    end
    2'b10: begin
        state <= IDLE;
        purchase <= 1;
        ret <= R40;
    end
    2'b11: begin
        state <= IDLE;
        purchase <= 1;
        ret <= R90;
    end
    default: begin end
endcase

```

```

default: begin end

```

```
        endcase  
    end  
end  
endmodule
```