[Note that these are notes submitted in January 2003 but the question was shortened slightly. I have tried to take this into account by alterations but may have made mistakes. CHN 1/06]

(a)
```
module pwmcontroller (out, tycle, ton, go, clk);

    output out;
    input [15:0] tcycle;
    input [15:0] ton;
    input go;
    input clk;

    Parameter sReload = 0, sOn = 1, sOff = 2;

(i) reg [1:0] state;
    reg [15:0] oncount, offcount;

    always @(posedge clk) begin
        if (!go) state <= sReload;
(ii)    else case (state)
            sReload: begin
                oncount <= ton;
                offcount <= tycle - ton - 1;
                state <= sOn;
            end
            Son: if (oncount == 0) state <= sOff; else oncount <= oncount - 1;
            soff: if (offcount == 0) state <= sReload; else offcount <= offcount - 1;
                                                        (iii)
        endcase
    end
    assign out = state == Son;
                 (iv)
```

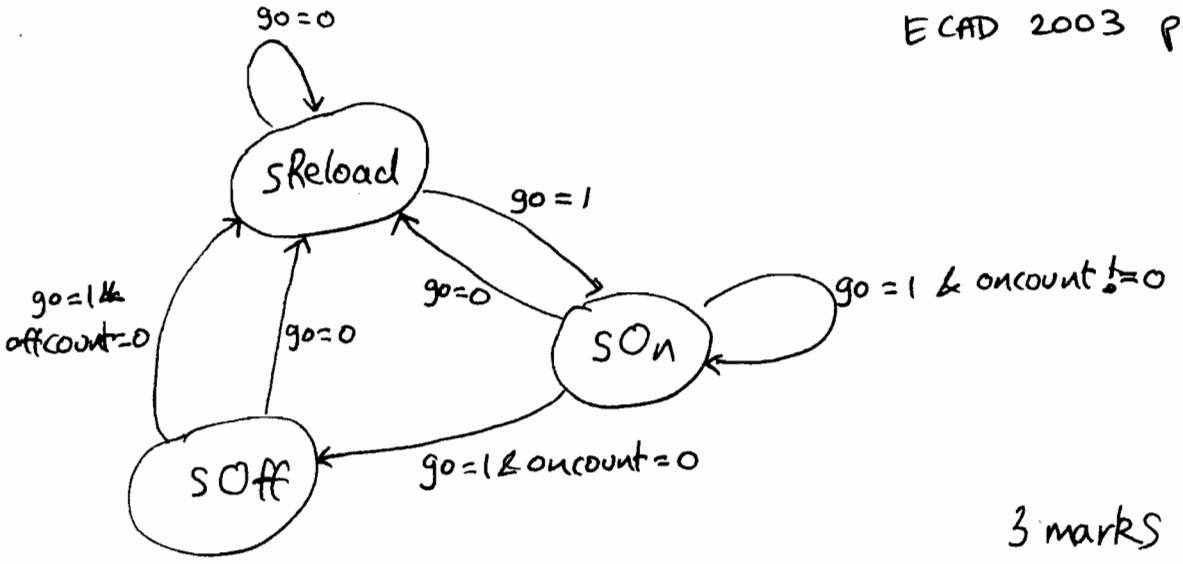
1 mark (i) oncount and offcount registers should be 16 bits wide, not just a single bit

2 marks (ii) If the 'else' keyword is missing the 'state' register can have two differing concurrent assignments – this is an error. Synthesis tools would object to this or produce an incorrect circuit.

2 marks (iii) If the next state after Soff is Son then the state machine will keep switching between sOn and sOff whilst the counters remain at zero. The intended specification is not met.

1 mark (iv) This is a continuous assignment and these can not be a concurrent assignment here.

(b)



3 marks

e.g.
1 mark for basic states and transitions
1 mark for correct transition labels
1 mark for remembering go=0 transitions

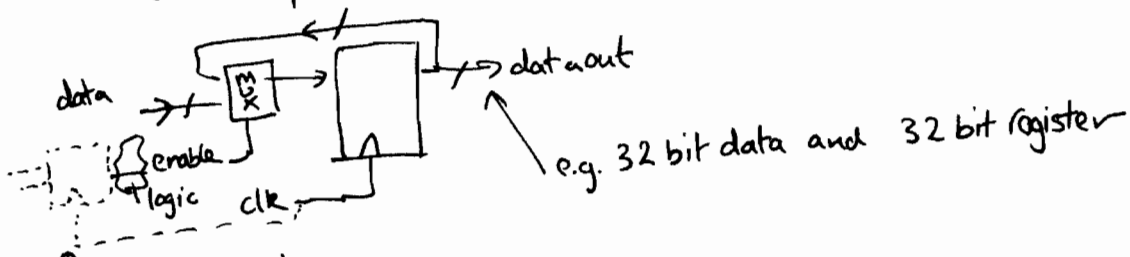
(c) An asynchronous reset is not needed. When go=0 a/c state in the circuit will become initialised before it is read. — 1 mark

1 mark

Clock gating involves gating (for example ANDing) the clock to a number of registers, typically a multi-bit register, with a control signal. This saves power as energy is not used to raise and lower the clock signal when the register does not need to be written.

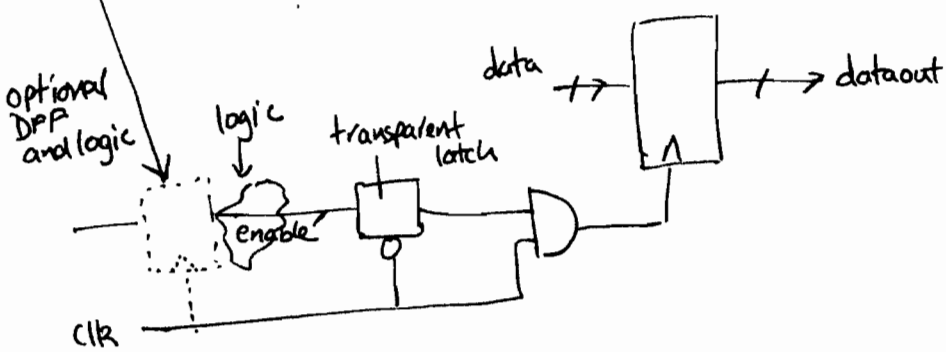
1 mark

For example



* diagrams 3 marks

Could become



(optional dotted bits not needed)

- the AND gate 'gates' the clock
- the transparent latch prevents glitches on the enable signal whilst the clock is high from spuriously clocking the DFF.

(include in 3 marks above) or give extra mark

(Such glitches can come from the combinational logic which generates the enable signal).

(Applying clock gating places timing requirements/constraints on the enable signal).

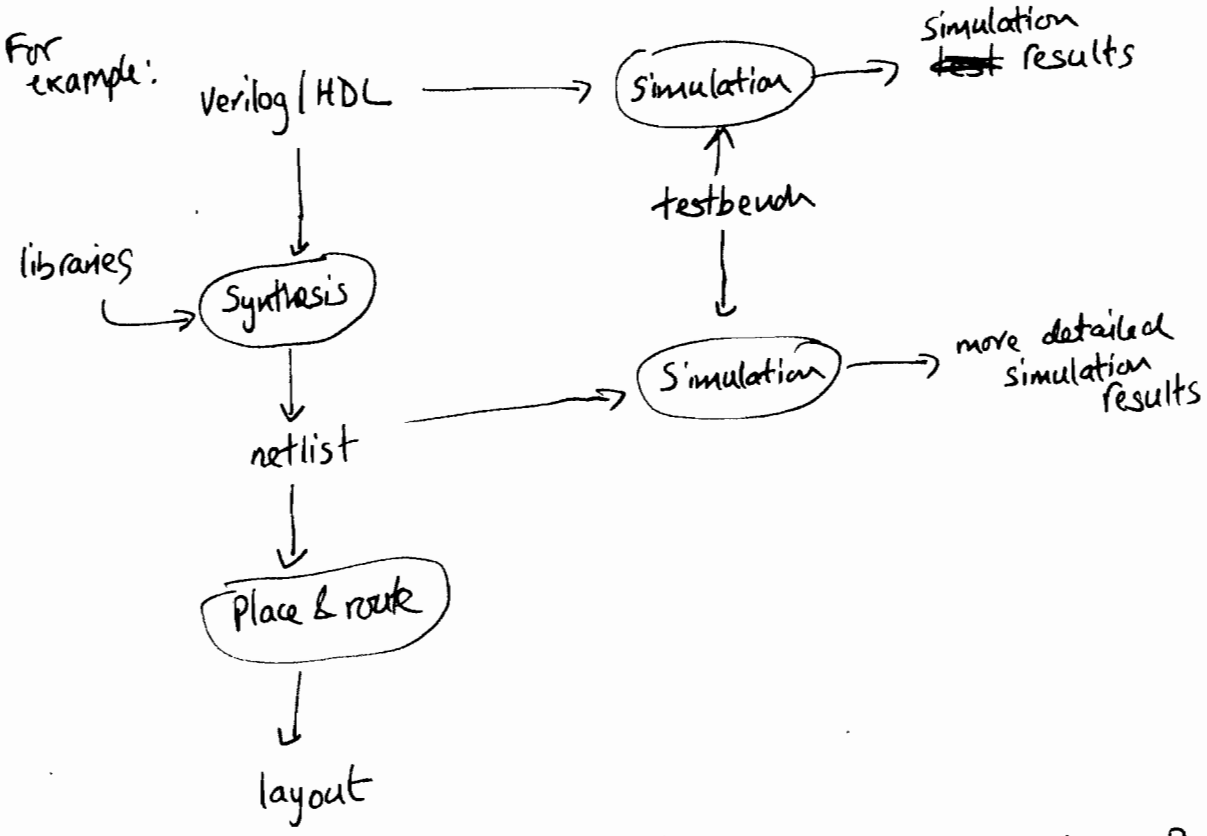
In the example circuit the clock to the oncount and offcount registers can be gated whilst the go signal is zero. (Note that oncount and offcount may well share the same register and decrementer).

1 mark

(d) The tycle_ton_1 calculation can easily be moved to the software. The tycle input then becomes a toff input. This removes a subtractor (adder) from the circuit. 1 mark

(Note the counters could not easily be moved to software – timing might be hard to maintain, and then there is no purpose of this module.)

(e) For example:



- There are many variations on this, for example add back annotation or physical verification / extraction. Sensible flows will get the marks!

2 marks

Production testing is to find manufacturing defects. In an ASIC these would be during device fabrication, and are part of normal manufacture – yield. Therefore all devices must be tested to discard faulty ones.

1 mark