LSU EE 4755 **Homework 6** Solution **Due: 13 November 2017**

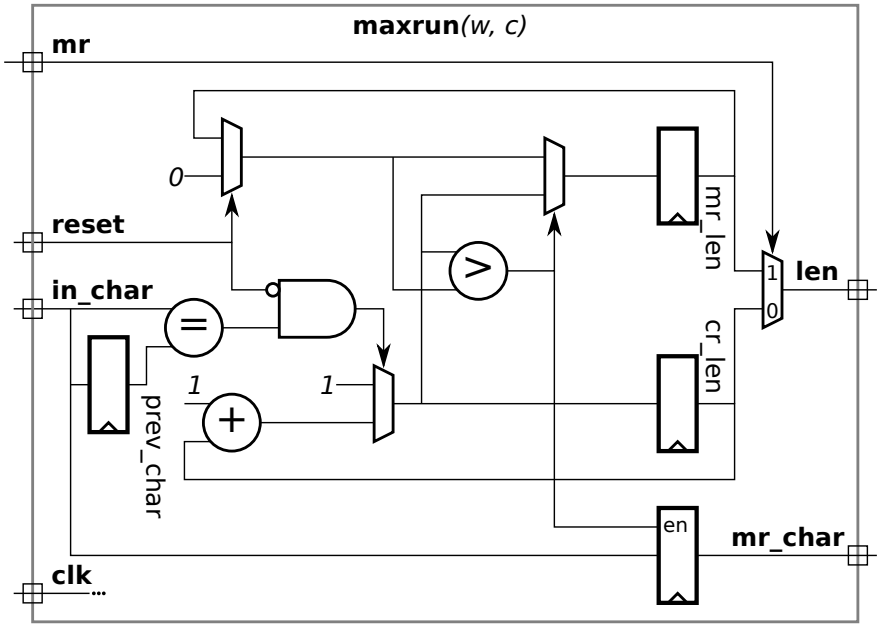
Problem 1: The solution to Homework 4, `http://www.ece.lsu.edu/koppel/v/2017/hw04-sol.v.html`, includes two modules, `maxrun` and `maxrun_opt`.

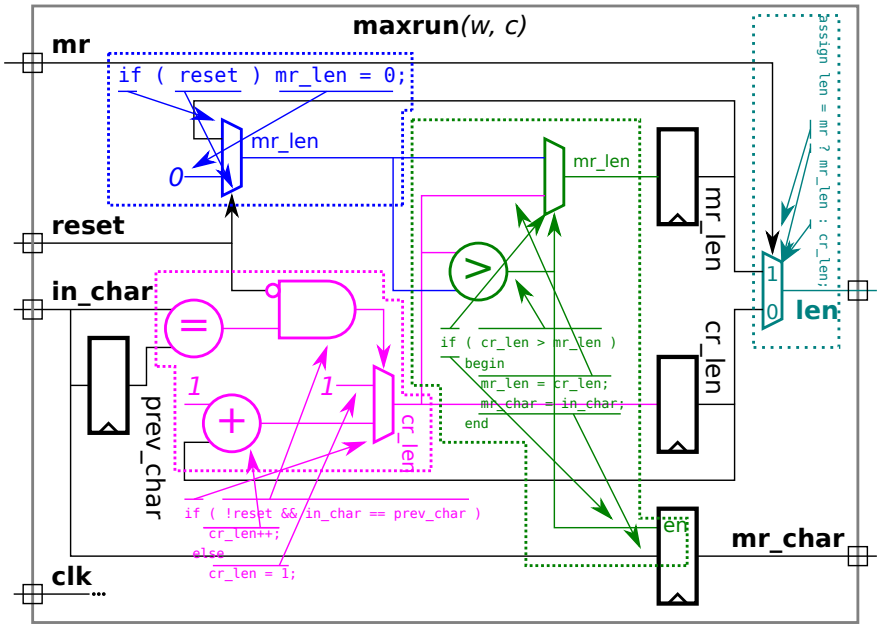
(*a*) Show the hardware inferred for `maxrun`. The Verilog code appears below.

```
module maxrun #( int w = 2, int c = 4 )
   ( output uwire [w-1:0] len,       output logic [c-1:0] mr_char,
     input uwire clk, reset, mr,     input uwire [c-1:0] in_char );
   logic [w-1:0] cr_len, mr_len;
   logic [c-1:0] prev_char;
   assign len = mr ? mr_len : cr_len;

   always_ff @( posedge clk ) begin

      if ( reset ) mr_len = 0;

      if ( !reset && in_char == prev_char )
        cr_len++;
      else
        cr_len = 1;

      if ( cr_len > mr_len )
        begin
           mr_len = cr_len;
           mr_char = in_char;
        end

      prev_char = in_char;
   end
endmodule
```

The solution appears below.

A common difficulty was properly accounting for order of assignments to `mr_len` and `cr_len`. The last assignment in the `always` block creates the value that is written to a register. The first illustration below shows the inferred hardware, the one below it shows the inferred hardware labeled with the Verilog code from which it was inferred.

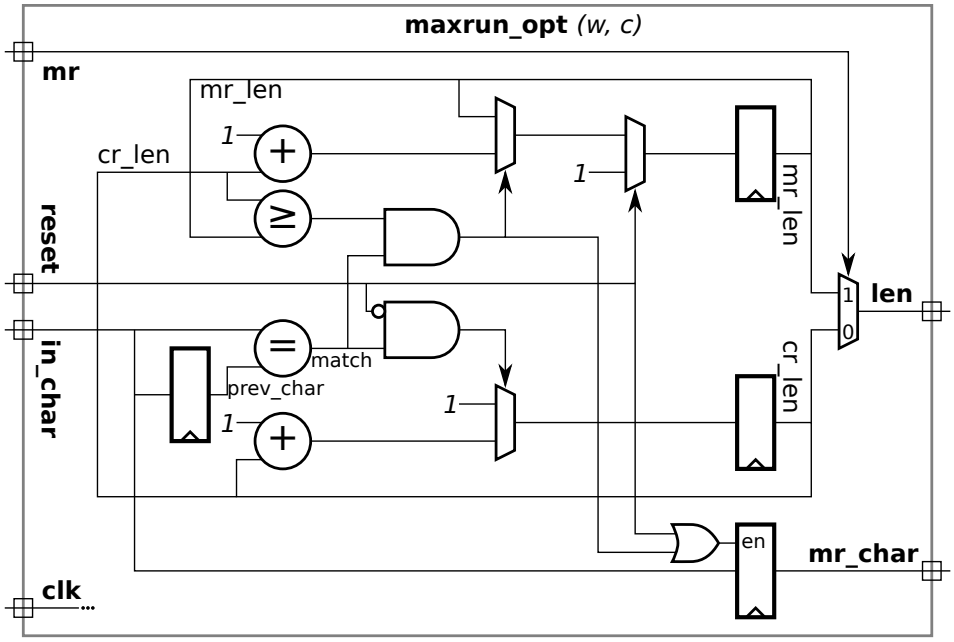




(*b*) Show the hardware inferred for `maxrun_opt`.

The solution appears below.

Note that there is no register for `match`. That is because it is not a live-out variable. That's obvious in this case because it is declared within the block.



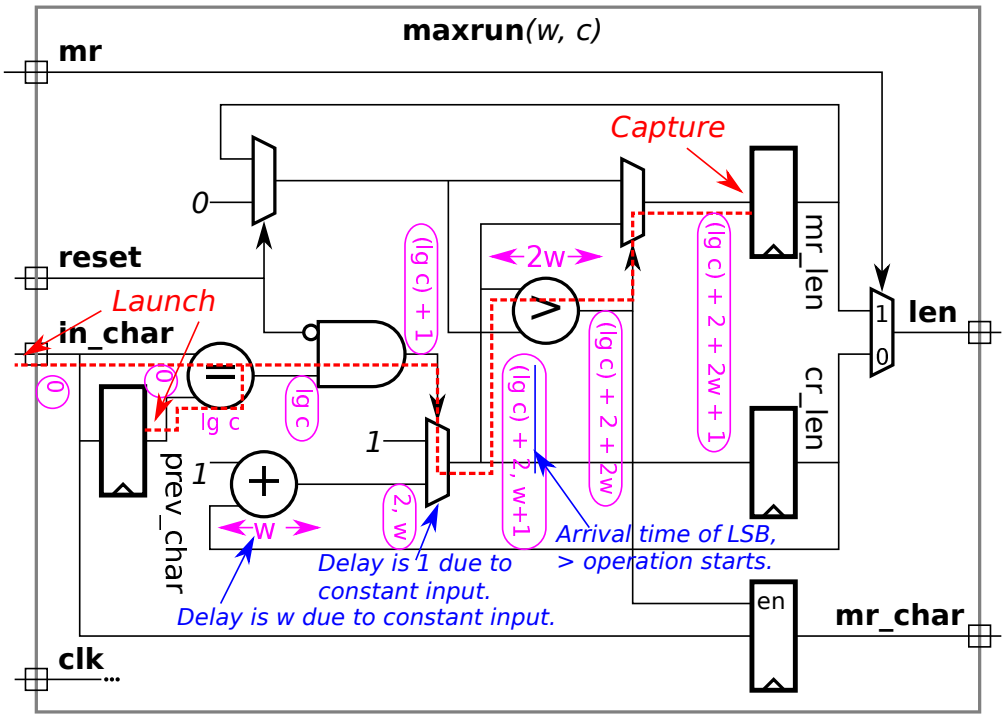
Problem 2: Compute the critical path for the `maxrun` and `maxrun_opt` modules using the simple model. The launch points (path starts) are at module inputs and register outputs, and the capture points (path ends) are at module outputs and register inputs. Note that with these definitions the critical path does not include the register itself. Show the critical path in terms of w, the number of bits in the `len` output and c, the number of bits in a character.

Short Answer: The critical path length is $(\lg c) + 2w + 3$ and its route is marked with a red dashed line in the illustration below. *Grading Note: In too many submissions the critical path was not marked on the diagram, instead relying on a prose description or just hints such as the modules the path passes through. Please show the path in the diagram.*

Explanation: The critical path starts at the `in_char` input and `prev_char` register output and follows the course shown. An interesting part of the critical path is the first mux on the path. The LSB of the lower data input arrives at $t = 2$ and the MSB arrives at $t = w$, which is later than the select signal, which arrives at $(\lg c) + 1$. Normally that would mean the lower data input is on the critical path. However, because the comparison unit can start when the LSB is ready the LSB arrival time determines criticality, and since $2 < (\lg c) + 1$ the select signal, not the data input, is on the critical path.

The `maxrun` module is slowed because the $>$ comparison must wait for the equality test.

Also note that multiplexors with constant inputs have a delay of 1 and that a w-bit ripple adder with a constant input has a delay of about w.



Short Answer: Assuming that $2w > \lg c$ the critical path length is $2w + 4$. The route of the path is shown with a red dashed line in the diagram.

Explanation: The critical path starts at the outputs of `mr_len` and `cr_len` and ends at the `mr_len` input. Unlike `maxrun`, the magnitude comparison and the equality test both start at $t = 0$, reducing the critical path.