2007 Paper 6 Question 1

ECAD

The following Verilog code describes a one-element FIFO.

```
module FIFO_one(clock, reset, dataInsert, insert, insertComplete,
                dataExtract, extract, extractComplete, full);

parameter nb = 7;  // num. bits: 7..0 bits (i.e. 8 bits) of data will be used
input  clock, reset;       // clock and reset signals
input  [nb:0] dataInsert;  // data to be inserted into the FIFO
input  insert;             // control: high means perform the insert
output insertComplete;     // control: high indicates data has been inserted
output [nb:0] dataExtract; // data extracted from the FIFO
input  extract;            // control: high means please perform an extract
output extractComplete;    // control: high indicates that dataExtract is valid
output full;               // control: high when the FIFO is full
reg    full;
reg    [nb:0] dataStore;
reg    insertComplete, extractComplete;

always @(posedge clock or posedge reset)
  if(reset) begin                          // comment A
    full <= 0;
    insertComplete <= 0;
    extractComplete <= 0;
    dataStore <= 8'bxxxxxxxx;
  end else begin
    full <= (insert || full) && !extract;  // comment B

    if(insert) begin                       // comment C
      if(!full) dataStore <= dataInsert;
      insertComplete <= !full;
    end
    else insertComplete <= 0;

    if(extract) begin                      // comment D
      extractComplete <= full || insert;
    end
    else extractComplete <= 0;
  end

assign dataExtract = dataStore;
endmodule
```

(*a*) What would be suitable comments on the behaviour of the code at points "comment A" to "comment D"? [4 marks]

(*b*) In the synthesised implementation, how will the `reset` and `clock` signals be connected to the D flip-flops that are used to hold the state inside the `always` block? [2 marks]

(CONTINUED ON NEXT PAGE)

(*c*) In Verilog a wire can transmit not only Boolean values `0` and `1`, but also the values `x` and `z`. How is `x` used in simulation and what will it be converted to when synthesised to real hardware? Illustrate your answer with reference to assignments to `dataStore` in `FIFO_one`. [3 marks]

(*d*) What is the state diagram describing the empty/full status of `FIFO_one`? Include the inputs `(insert, extract)` and outputs `(insertComplete, extractComplete)` on the arcs of the state diagram and ignore the data path. [4 marks]

(*e*) Is it possible to insert and extract data on the same clock cycle? [1 mark]

(*f*) How could two instances of `FIFO_one` be joined to produce a two-element FIFO? [4 marks]

(*g*) For your design in part (*f*), how many clock cycles of latency would there be from input to output if data were always extracted as quickly as possible? [2 marks]