```
`timescale 1ns / 1ps
/*********************************************************************************
 *
 * Author:   Jesus Luciano & Rosswell Tiongco
 * Filename: IO.v
 * Date:     3/21/2019
 * Version:  1.0
 *
 * Notes:1.0 Module sends out interrupt request and keeps signal high until
 *           interrupt acknowledge input is set high by external source. Will
 *           need to add memory unit as well
 *
 *       1.1 1K x 32 Memory module added
 *
 *       1.2 Memory module control signal io_cs, io_rd, and io_wr added for
 *           synchronous write and asynchronous read control
 *
 ********************************************************************************/
module IO(clk, intr, inta, io_address, io_d_in, io_out, io_rd, io_wr, io_cs);
    //interrupt inputs/outputs
    input clk, inta, io_rd, io_wr, io_cs;
    output intr;

    //memory inputs/outputs
    input  [31:0] io_address, io_d_in;
    output [31:0] io_out;

    //IO Memory 1k x 32
    reg [31:0] M [1023:0];

    reg intr;
    integer i;

    //IO Memory Write - control by io signal
    always @(posedge clk)
        if(io_wr & io_cs)
            M[io_address[9:0]] = io_d_in; //write to memory every clock pulse
    //IO Memory Read - 9 LSB's
    assign io_out = (io_rd & io_cs) ? M[io_address[9:0]] : 32'hz;//asynchronous read

    //interrupt generator
    initial //generate interrupt at random time
      begin
        //Display time in nanoseconds
        $timeformat(-9, 1, " ps", 9);
        #200
        intr = 1'b1;
        @(posedge inta);//wait for interrupt acknowledge
        intr = 1'b0;
      end



endmodule

```