

Név:	Index szám:	Pontszám: (max. 50)
------	-------------	------------------------

1. Rajzolja le a két négybites szám összeadására alkalmas logikai kapcsolás tömbvázlatát megfelelő számú egybites teljes összeadó segítségével! Mutassa be az összeadás menetét az $a=1011_2$ és a $b=0111_2$ számok összeadásán keresztül, megadva az egyes bemenetek és kimenetek értékeit!

(10 pont)

2. D flip-flop-ok és tetszőlegesen megválasztott logikai kapuk segítségével tervezzen egy olyan Moore-féle hálózatot, amely $A=0$ bemenet esetén tartja az elért állapotot, $A=1$ esetén 01,10,11 kódban számlál! Az 11 állapotot követő órajel ciklusban az új állapot $A=0$ esetén 01 legyen, $A=1$ esetén viszont 10! Az Y kimenet az 11 állapotban van magas logikai szinten.

(15 pont)

3. A megadott Verilog HDL leírásban keresse meg és javítsa ki az összes hibát!

(10 pont)

```

module Exam (in, clock, reset, out);
  parameter idle = 2'b00;
  parameter read = 2'b01;
  parameter write = 2'b11;
  parameter wait = 2'b10;
  input clock, reset, in;
  output out;
  reg [1:0] current_state, next_state;
  always @ (posedge clock)
  begin
    if (Reset == 1'b1)
      current_state <= 2'b00;
    else
      current_state <= next-state;
    end
  always @ (current_state or in)
  begin
    out = 2'b01;
    next_state = current_state;
    case (current_state)
    idle:
      out = 1'b0;
      if (in)
        next_state = 2'b11;
    write:
      out = 1'b0;
      if (in == 1'b0)
        next_state = 2'b10;
    read:
      if (in == 1'b1)
        next_state = 2'b01;
    waits:
      if (in == 1'b1)
        next_state = 2'b00;
    endcase
  endmodule

```

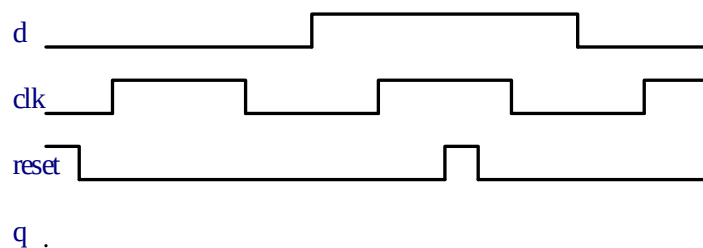
4. Rajzolja meg a q jel idődiagramját, hogy a viselkedés megfeleljen az alábbi hardver-leírásnak!

(15 pont)

```

module Exam (q, d, clk, reset);
  input d, clk, reset;
  output q;
  reg q;
  always @ (posedge clk
  or posedge reset)
  if (reset)
    q <= 1'b0;
  else
    q <= d;
  endmodule

```



Ime:	Broj indeksa:	Broj poena: (max. 50)
------	---------------	--------------------------

1. Nacrtajte blok šemu četvero-bitnog sabirača pomoću potrebnog broja jednobitnih potpunih sabirača! Prikažite proces sabiranja na primeru sabiranja brojeva $a=1011_2$ i $b=0111_2$ uz navođenje vrednosti svih ulaza i izlaza!

(10 poena)

2. Pomoću D flip-flop-ova i proizvoljno odabranih logičkih kola projektujte Moore-ovu mrežu koja pri ulazu $A=0$ održava postignuto stanje, pri ulazu $A=1$ broji u kodu 01,10,11! U takt intervalu posle stanja 11 pri $A=0$ novo stanje treba da je 01 a za $A=1$ novo stanje je 10! Izlaz Y je na visokom logičkom nivou u stanju 11.

(15 poena)

3. U datom Verilog HDL opisu pronađite i korigujte sve greške!

(10 poena)

```

module Exam (in, clock, reset, out);
  parameter idle = 2'b00;
  parameter read = 2'b01;
  parameter write = 2'b11;
  parameter wait = 2'b10;
  input clock, reset, in;
  output out;
  reg [1:0] current_state, next_state;
  always @ (posedge clock)
  begin
    if (Reset == 1'b1)
      current_state <= 2'b00;
    else
      current_state <= next-state;
    end
    always @ (current_state or in)
    begin
      out = 2'b01;
      next_state = current_state;
      case (current_state)
        idle:
          out = 1'b0;
          if (in)
            next_state = 2'b11;
        write:
          out = 1'b0;
          if (in == 1'b0)
            next_state = 2'b10;
        read:
          if (in == 1'b1)
            next_state = 2'b01;
        waits:
          if (in == 1'b1)
            next_state = 2'b00;
      endcase
    end
  end
endmodule

```

4. Docrtati vremenski dijagram za signal q tako da ponašanje hardvera odgovara datom opisu!

(15 poena)

```

module Exam (q, d, clk, reset);
  input d, clk, reset;
  output q;
  reg q;
  always @ (posedge clk
  or posedge reset)
  if (reset)
    q <= 1'b0;
  else
    q <= d;
  endmodule

```

