

# Laboratórium 1 felkészülési feladat

Hallgató:

Mérés sorszáma: 10

---

Készítsen egy olyan Verilog modult, ami Z kimenetén 1-el jelzi, ha az egyetlen X bemenetére ciklikusan érkező 4 bites számok az alább közölt feltételnek megfelelnek! A közbülső ütemekben a kimenet 0 legyen! A legnagyobb helyiértékű bit (MSB) jön elsőnek.

A vizsgált feltétel: **A bemenet bináris súlya (1 bitek száma) 2**

A mérés kezdetén be kell mutatni a működő Verilog kódot és a működést igazoló szimulációt. A modulnak legyen órajel engedélyező bemenete is! (Ez a tesztelhetőség szempontjából fontos.)

Példa: (A feladatot a fent megadott feltételre oldja meg, ez csak egy értelmezést segítő példa!)

Feltétel: a bemenet osztható 7-tel

X: 0010 0111 1100 1110 0000 0110...

Z: 0000 0001 0000 0001 0001 0000...

**Kötelező kritériumok a beadandó anyagra:** Ezek be nem tartása esetén a feladatot nem fogadjuk el!

Hozza el a teljes működő ISE projektet! Készítsen egy word (vagy pdf) dokumentumot, amiben szövegesen összefoglalja a kód felépítését, működését! Nem elég csupán a forráskódot bemásolni!

E dokumentumnak kötelezően tartalmaznia kell a működést igazoló szimulációs hullámformát és annak szöveges értelmezését. Az értelmezés nélküli képeket nem értékeljük, nem fogadjuk el.

A fájl kinyomtatása nem szükséges, ennél a mérésnél elektronikus dokumentációt kérünk.

A beadás tudnivalói:

- **Az önállóan kidolgozott feladatot a következő mérési gyakorlat elején a mérésvezetőnek kell bemutatni, - a mérési útmutatóban előírtak szerint - írott vagy elektronikus formában.**
- A felkészülési feladat utólag már nem adható be. Pótlására a szorgalmi időszak végén egy alkalommal, az adott mérési gyakorlat pótlásával egy időben van lehetőség.

A feladatokat önállóan, meg nem engedett segítség igénybevétele nélkül oldottam meg:

.....  
aláírás

hazi.v

```
`timescale 1ns / 1ps

module hazi(
    input clk,
    input rst,
    input ce,
        input in, //X bemenet
        output out // Z kimenet
    );

reg [3:0] data;
reg [1:0] counter;

reg outreg;

assign out = outreg;

always @ (posedge clk)
begin
    if (rst)
        begin
            counter <= 0;
            data <= 0;
            outreg <= 0;
        end

    else if (ce)
        begin
            data <= {data [2:0],in}; //bekérjük az adatot, balra shiftelünk
            if (counter==0)
                begin
                    if (data==4'b0011) outreg<=1;
                    if (data==4'b0101) outreg<=1;
                    if (data==4'b0110) outreg<=1;
                    if (data==4'b1001) outreg<=1;
                    if (data==4'b1010) outreg<=1;
                    if (data==4'b1100) outreg<=1;
                end
            else
                begin
                    outreg <= 0;
                end
            counter <= counter+1;
        end
    end
end

endmodule
```

```
`timescale 1ns / 1ps

module hazitest;

    // Inputs
    reg clk;
    reg rst;
    reg ce;
    reg in;

    // Outputs
    wire out;

    // Instantiate the Unit Under Test (UUT)
    hazi uut (
        .clk(clk),
        .rst(rst),
        .ce(ce),
        .in(in),
        .out(out)
    );

    initial begin
        // Initialize Inputs
        clk = 0;
        rst = 1;
        ce = 0;
        in = 0;

        #100 rst =0;
        #1 ce=1;
        #2 in = 1;
        #2 in = 0;
        #2 in = 0;
        #2 in = 1;
        #2 in = 0;
        #2 in = 1;
        #2 in = 1;
        #2 in = 1;
        #2 in = 1;
        #2 in = 0;
        #2 in = 0;
        #2 in = 1;
        #2 in = 1;
        #2 in = 0;
        #2 in = 1;
        #2 in = 0;

        end

always #1 clk<=~clk;
endmodule
```

## Feladatmegoldás:

Miután megadom a modul ki és bemeneteit, definiálom a registereket.

Egy darab 4 bites registert használok a bemeneti jelek tárolására, egy darab kettő biteset számlálóként, és egy darab egy bites registert a kimeneti jel tárolására.

A feladatban először megvizsgálom a reset jelet, amennyiben értéke 1, úgy alaphelyzetbe állítom az értékeket.

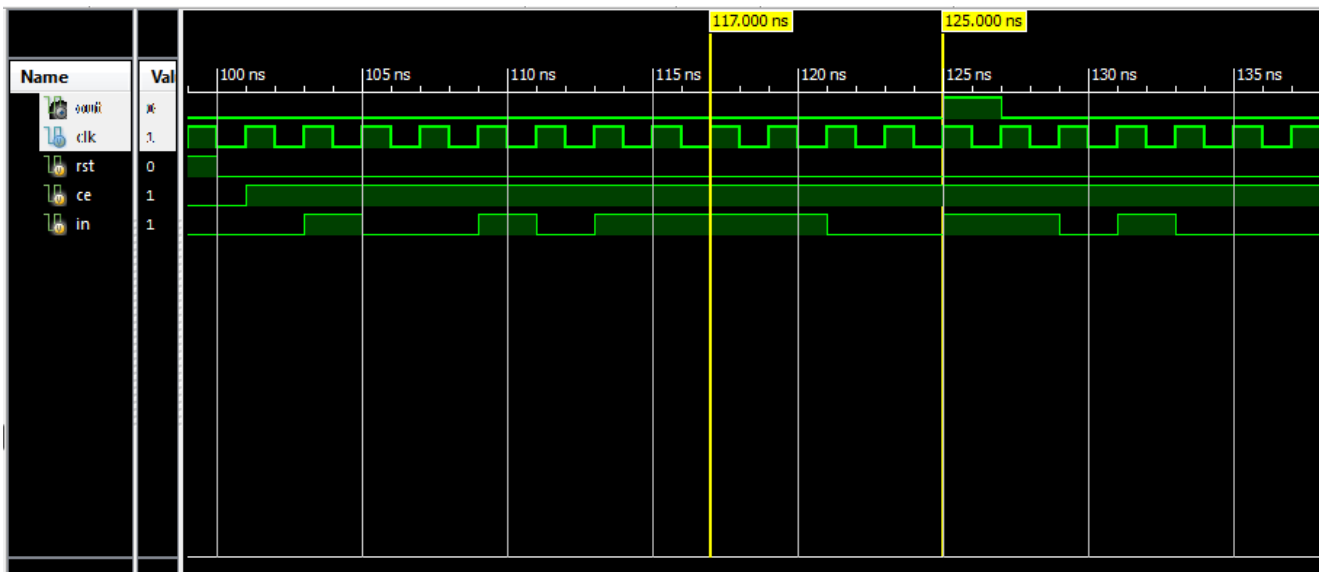
Egyébként, ha az órajel engedélyező jel (ce) értéke 1, akkor a bejövő adatot a registerbe (data) helyezem. (balra shift-eléssel).

Ezután megvizsgálom, hogy a counter nevű számlálóm értéke nulla –e, amennyiben igen, úgy megvizsgálom, hogy a 4 bites számom súlya kettő-e.

Ha igen a kimenetre egy-et adok, egyébként pedig nullát.

Ezenfelül a számláló értékét mindig növelem eggyel.

*(Feladatomban az X bemenetnek az in, a Z kimenetnek az out felel meg, a jobb átláthatóság kedvéért ezt a megoldást választottam.)*



A képen láthatjuk a működést igazoló szimulációs hullámformát,

Látható, hogy a (reset) rst magas jel 100 ns-ig tart, a clk órajel 2 ns periódusidejű, a ce engedélyező bement 101 ns-tól magas értékű.

Szimulációt az in bemeneten megjelenített változó véletlen jellel vizsgáltam.

A képen megfigyelhető hogy csak egyetlen bitsorozat felel meg a feltételnek, amit sárga határvonalakkal illusztráltam.

A feladatkiírásnak megfelelő sorozat után megjelenik a kimeneten egy órajel periódus ideig az 1 (magas) szint.