این سوال را به شکل زیر پیاده‌سازی می‌کنیم.

```
module key_checker(clock, reset, start, in, key, ready, valid, out);
    parameter N = 32;
    parameter W = 32;
    parameter M = N * W;

    input clock, reset, start;
    input [W-1:0] in;
    input [M-1:0] key;

    output reg ready, valid;
    output reg [M-1:0] out;

    initial begin
        ready = 1'b0;
        valid = 1'b0;
        out   = 1'b0;
    end

    always @(posedge reset, posedge start) begin
        ready = 1'b0;
        valid = 1'b0;
    end

    always @(posedge clock) begin
        if (start == 1'b1 && reset == 1'b0) begin
            out = (out << W) + in;
        end
    end

    always @(negedge start) begin
        if (reset == 1'b0) begin
           valid = (out == key);
           ready = 1'b1;
        end
    end
endmodule
```

در حالت اولیه، همه مقادیر خروجی صفر خواهند بود. توجه کنید که صفر یک بیتی زمانی که به out نسبت داده میشود، از سمت چپ با صفر پر می‌شود.

هر زمان که start و یا reset برابر یک شود، مقدار ready و valid برابر صفر میشود تا با توجه به داده ورودی پردازش‌های جدید انجام شود.

همگام با لبه بالارونده کلاک، ورودی in دریافت میشود و به سمت راست چیزی که تا به حال دریافت شده، اضافه میشود. توجه بفرمایید که عرض ورودی W است. برای اضافه کردن in به سمت راست out، out را W واحد به سمت چپ شیفت میدهیم و آن را با in جمع میکنیم. البته با اپراتور concatenation نیز این کار میتوانستیم انجام دهیم. دقت بفرمایید که شرط انجام عملیات این است که start فعال و reset غیر فعال باشد. در زمان لبه پایین رونده startمی توانیم مقدار مورد نظر را محاسبه کرده و در نهایت گزارش کنیم.

برای تست، این ماژول را با N و W کوچکتر می‌سازیم و تست می‌کنیم.