

VE2570 Quiz 9 7.30 Candy - Candies

Convert the following C-like code into an HLSM. (30 points)

Then develop datapath to support the HLSM. (20 points)

Inputs: byte grade[100], byte bar, bit go

Outputs: byte pass, byte fail, bit done

PASSING_NUMBER:

```
while (1) {
    while (!go);
    done = 0; pass = 0; fail = 0; i = 0;
    while (i < 100) {
        if (grade[i] >= bar) pass = pass + 1;
        else fail = fail + 1;
        i = i + 1;
    }
    done = 1;
}
```

HLSM

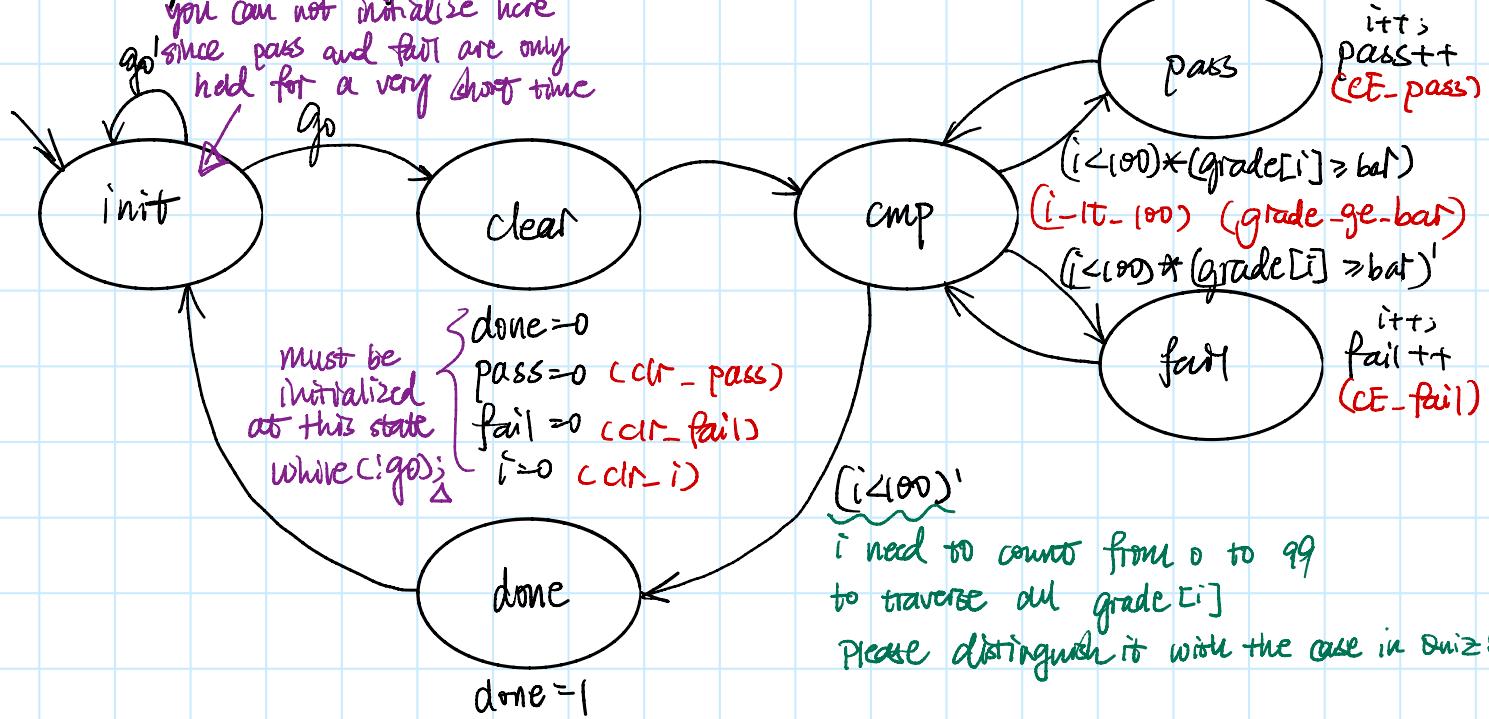
Remember to write done Inputs/Output/Local Registers

Inputs: byte grade[100], byte bar, bit go.

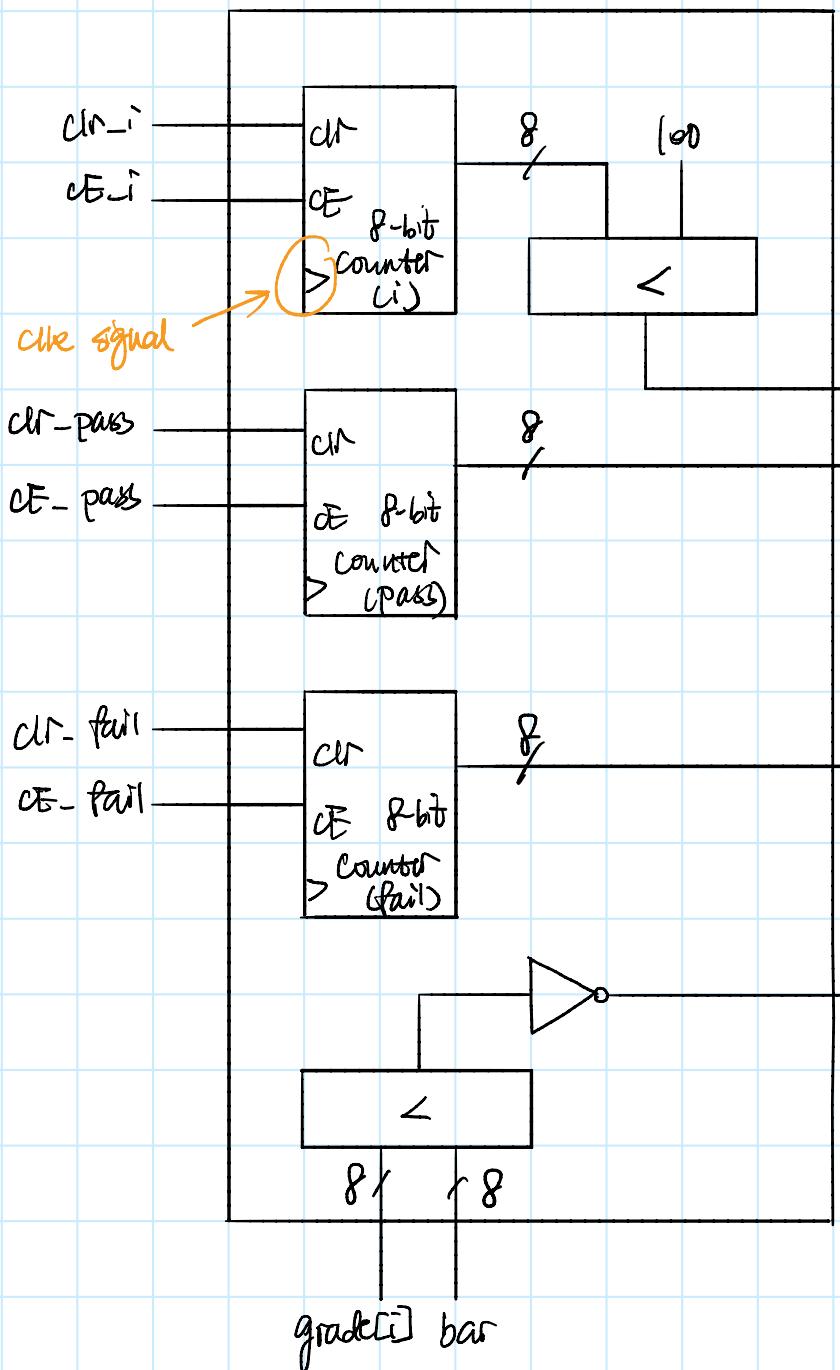
Outputs: byte pass, byte fail, bit done.

Local register : byte i;

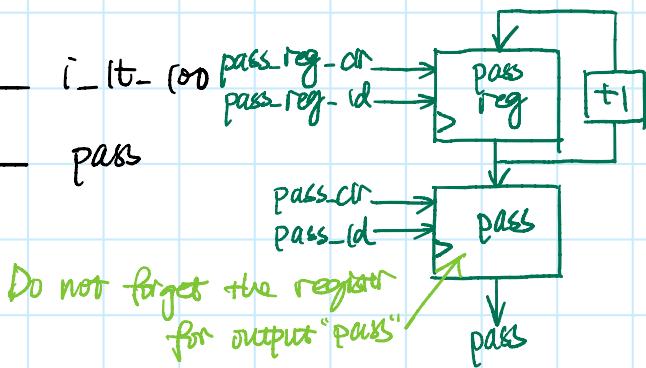
you can not initialize here
since pass and fail are only
had for a very short time



Datapath:



* There is no need to define two more local registers "pass-reg" and "fail-reg" because "pass" and "fail" are defined as outputs. For the similar problem in HW 9 "a-reg" and "b-reg" are need because "a" and "b" are inputs. For those who define "pass-reg" and "fail-reg", your registers for pass/fail should look like this



"pass-reg", "fail-reg", "pass", "fail" should all be initialized.

Please do not tell me that it's functionally correct to use a mux. You are deducted because I want you to get rid of this bad habit. =)

DO NOT use you compare result between grade and bar and a mux to load your pass and fail registers!

① As I have said before, do not struggle to find out a way to let datapath complete the whole system. It is really a bad habit because it makes the circuit larger and requires more resources.

② According to the design of HLSM, there is no need for you to use a mux. The component of your datapath depends on the operations and judgement condition of your HLSM. For example, $i < 100$ requires a comparison and it requires a counter or a register. Also, you can set "Pass" and "fail" state give control sign to load the two registers.

No need for a "done" register. you can output "done=1" in the controller