Faculty of Science, Engineering and Technology

Computer Systems

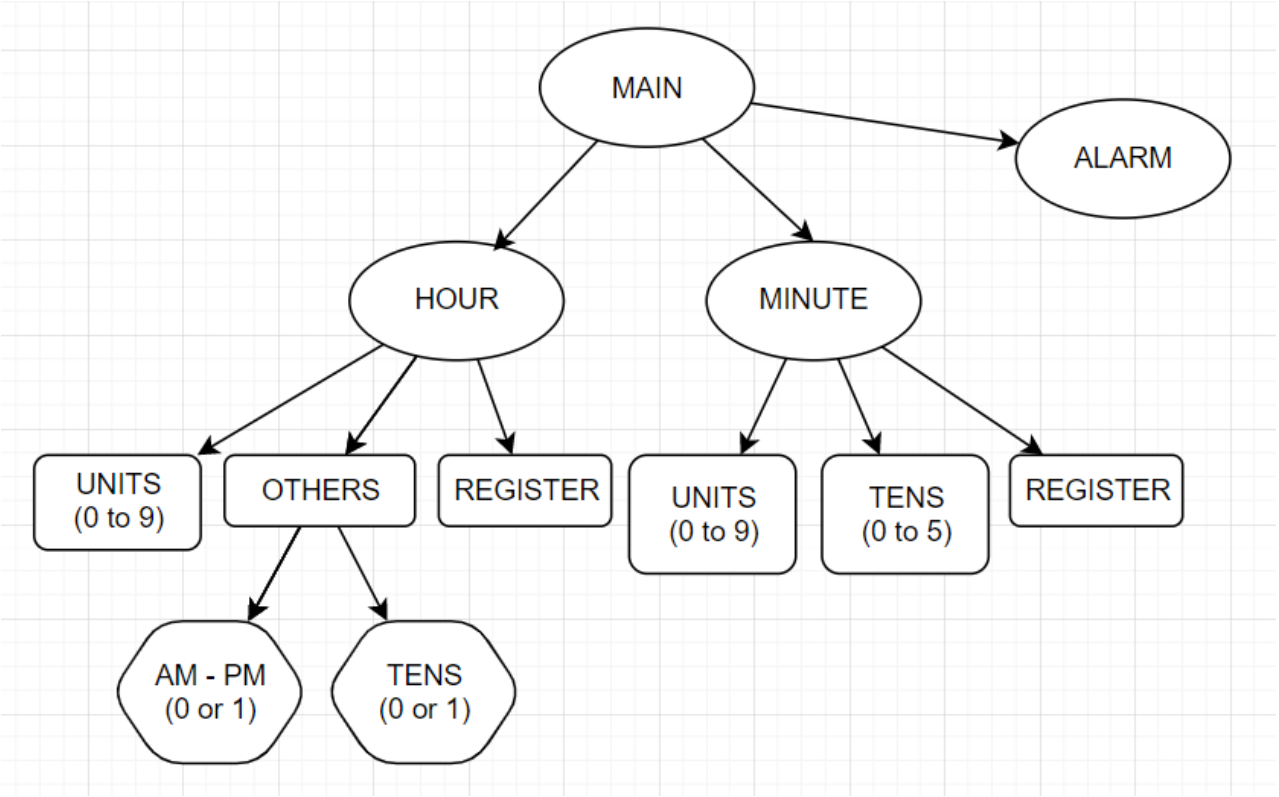
Name: Le Quang Hai

Student ID: 104175779

Unit Code: COS10004

Lab Session: Assignment 1

Hierarchical structure: (screenshot in step 4)



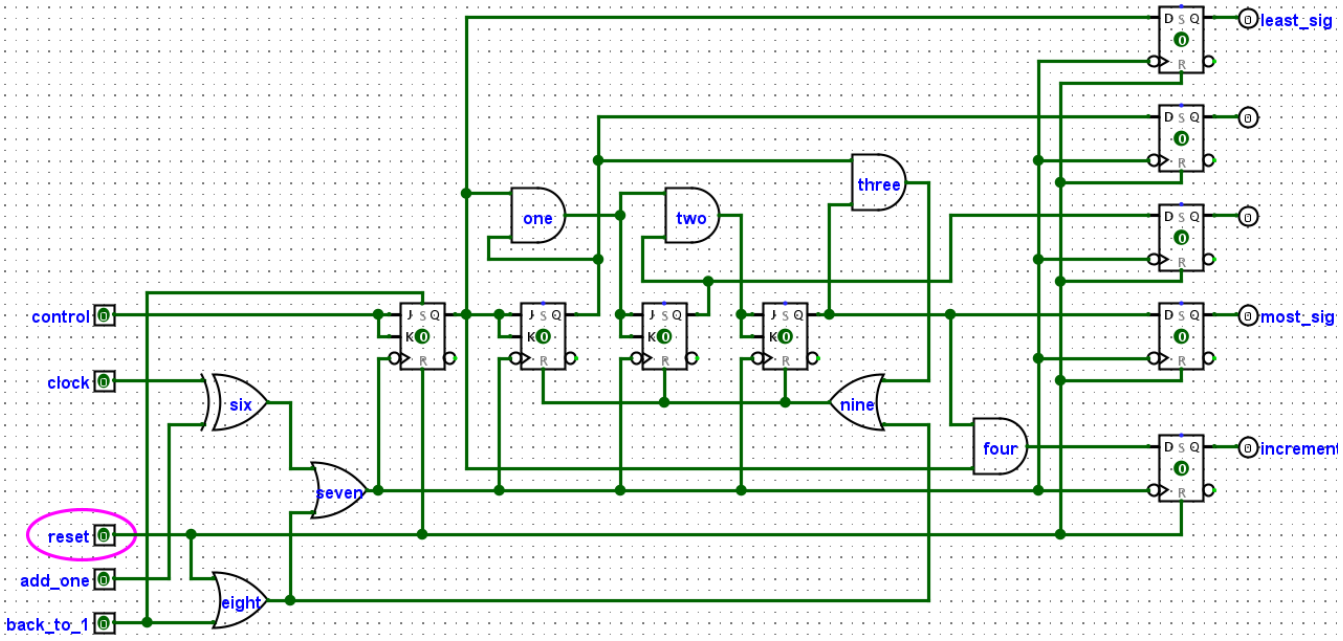
Note: In this assignment's circuit:

- All flip flops have been configured to trigger on FALLING EDGE.
- The sub-circuit screenshots may contains redundant inputs which will be necessary in the upcoming steps
- I call a 'period of a clock' a 'turn'

Stage 1: Implement the minutes counter and display.

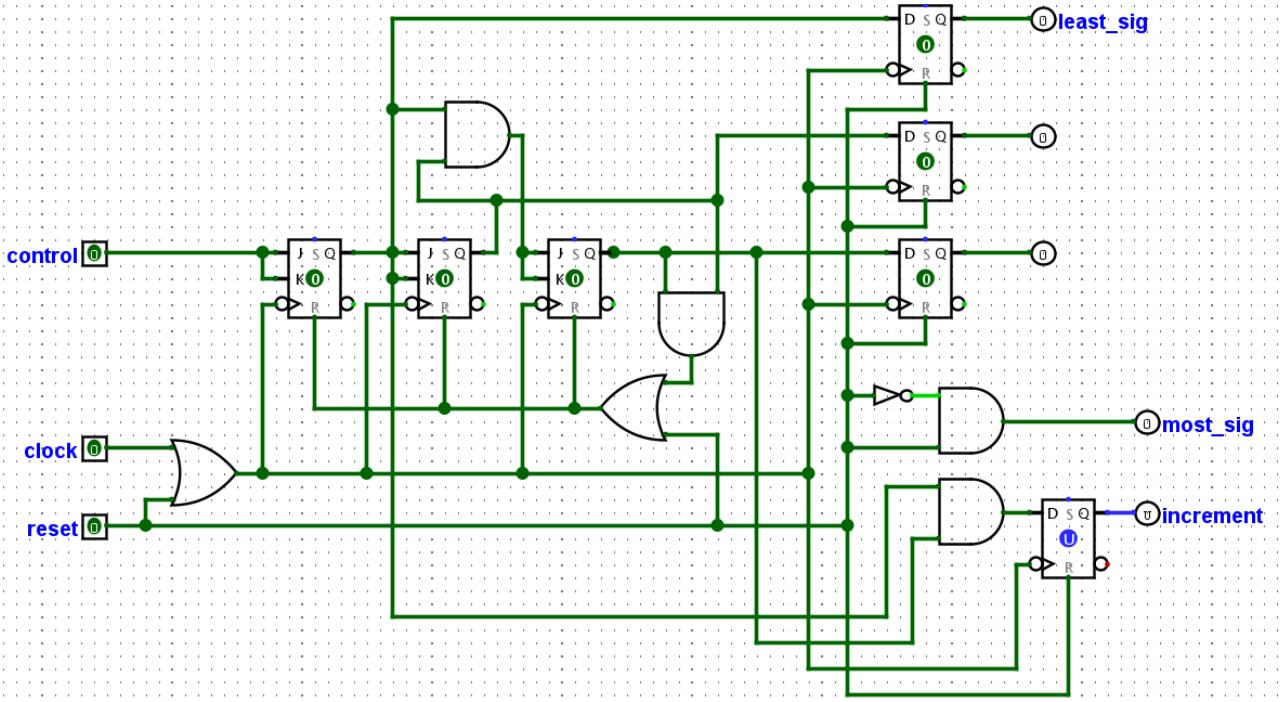
On this stage, a modulo ten counter for counting the units column and a modulo six counter are created following the lab done earlier on.

In this modulo-10-counter:



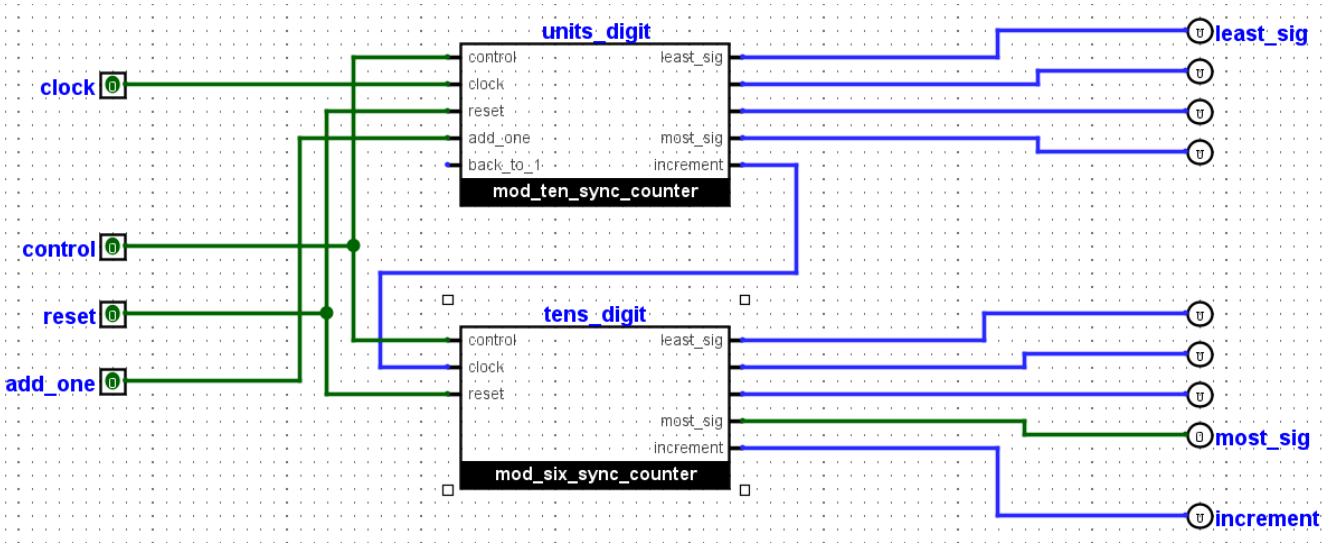
- The ‘control’ input when on makes the whole run as designed, if it is 0, the circuit will not run as designed.
- The clock input indicates the clock and is connected to all clock gates of flip flops. However, some other events are also designed to trigger 1 period of a clock so several OR, XOR gates are added.
- The ‘reset’ input is connected to the reset gate of all flip flops to reset the timer to 0 when needed.
- The 4 D flip flops play a role as a buffer.
- The 4 JK flip flops represent 4 bits (0-15). When ‘control’ is 1, the leftmost JK flip flop (which represents the least significant bit) flip flop between 0 and 1 every turn.
- When all JK flip flops to the left are 1, the next JK flip flop turns to 1 and turns all JK flip flops to the left back to 0. We can achieve this because:
    - AND gate ‘one’, ‘two’, turn the next flip flop to 1
    - When the flip flop to the left are 1, the next turn it changes (from 1 to 0)
- The ‘three’ OR gate is 1 when the JKs reach 10 (1010) resetting JKs to 0 (0000).
- The ‘four’ AND gate is 1 when the JKs reach 9 (1001). On the next turn, the ‘increment’ output receives 1 and back to 0 at the following turn. This acts as a rising-falling in a clock. The ‘increment’ output is connected to the clock of the higher digit circuit (tens in this case)
- There is an extra input ‘add_one’, this is used to increment output by 1. Like ‘reset’ input, when changes 1 to 0, it acts like a falling edge in a clock, accelerating the JKs by 1 turn.
- In the upcoming stages, there is a specific part that requires the output to be reset to 1, instead of 0, after reaching 9. ‘Back to one’ input triggers the ‘nine’ OR gate to reset the 3 most important bits while setting the least significant bit to 1 (from 1010 to 0001)

In this modulo-6-counter:



- The JKs, Ds, clock, reset, control is the same as in the modulo-10-counter.
- Representing 0-5 only requires 3 bit so 3 pairs of JK-D flip flop is used
- The most significant bit is always 0

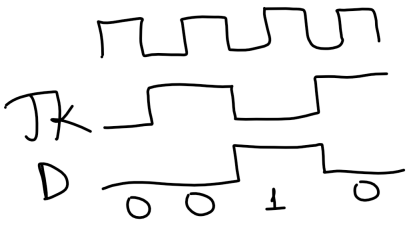
Build-up 00-59 counter combining the 2 counters:



- The clock of the ‘units’ digit connects to the clock of the whole minute circuit.
- The clock of the ‘tens’ digit connects to the ‘increment’ output of the ‘units’ circuit. When the ‘units’ digit reset (from 9 to 0), the ‘increment’ output of the ‘units’ circuit generates a falling edge, triggering a turn in the ‘tens’ circuit.

Assumptions, problems & solutions:

- At the beginning, all flip flops are set to 0 in the first turn. In the second turn the D flip flops get the 0 again from the JK flip flops in the first turn. Therefore we get an unexpected outcome when it counts 0-0-1-2-3…(0 twice!).

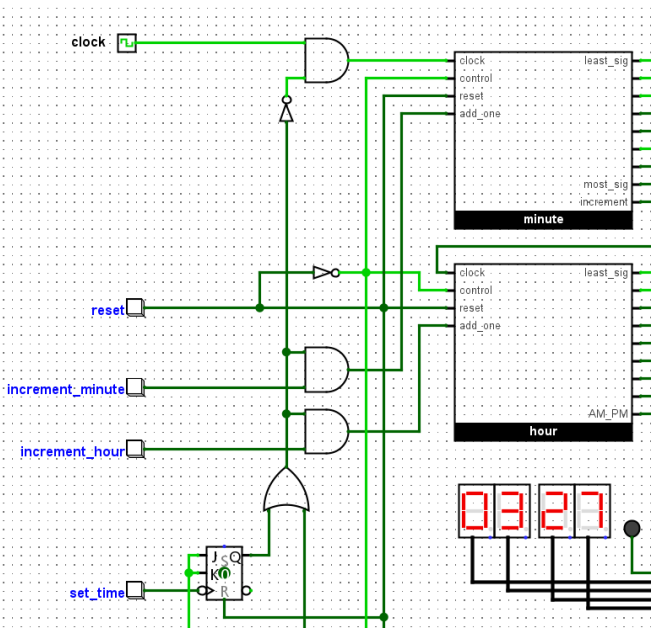


- To solve that, I let the ‘reset’ input and ‘clock input go through an OR gate so that after resetting. When reset changes 1 to 0, it acts like a falling edge in a clock, the JK flip flops are set from 0 (0000) to 1 (0001).
- When the counters reset counting, there is a moment the number count reaches out of the range. Exp: the modulo-ten-counter counts from 0 to 9, then 10 - triggering the ‘three’ AND gate to immediately set back to 0, that toggle may allow a ‘A’ (10 in hexadecimal) to be displayed. Therefore, the D flip flops are added to avoid the behavior.
- For those bits that is always 0, I used a construction that could never be true
    - P AND (NOT P)

Stage 2: Start implementing the Set Time modality.

The minute circuit is the 00-59 counter from step 1.

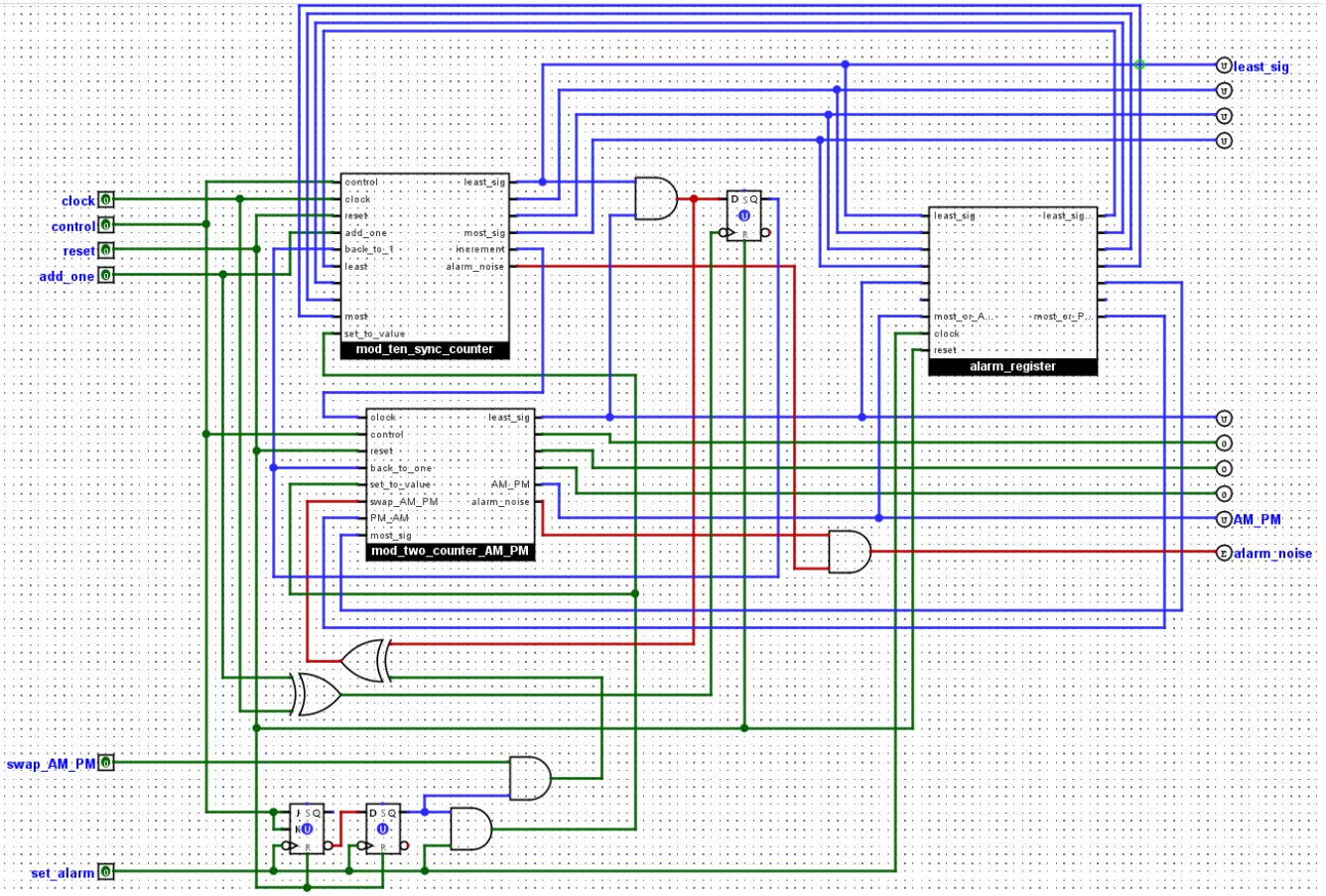
- I notice that when the counter is running, ‘control’ is 1 and ‘reset’ is 0. When ‘reset’ is 1 the value of ‘control’ does not affect anything therefore I came to the conclusion that ‘reset’ can be NOT(control)
- The clock goes through an AND gate, also called a ‘controlled switch’ in week 1, this makes signal from the clock to be blocked when the ‘set_time’ is 1 as instructed.
- The ‘increment_minute’ button is connected to the ‘add_one’ input of the modulo-ten-counter. As mentioned in stage 1, the ‘add_one’ input and the ‘clock’ input in the modulo-ten-counter are connected to an OR gate, whose output is connected to all flip flops’ clock input. When the button is pressed, ‘add_one’ turns 1 then 0 - just like a falling edge and the ‘minute’ display is incremented.



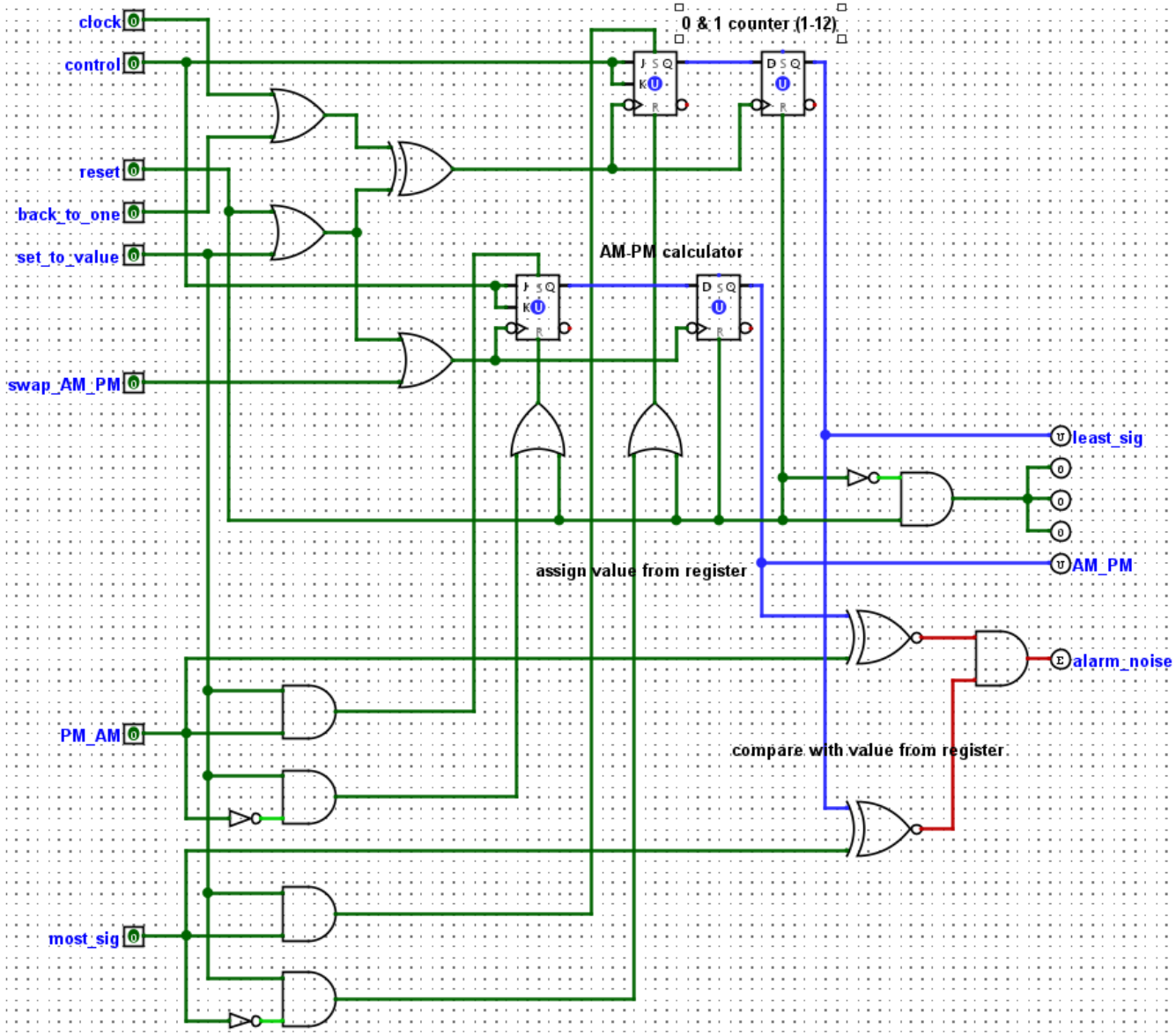
- The ‘increment_minute’ should only take effect when ‘set_time’ is on so an AND gate is added
- ‘set_time’ is a button connecting to the clock of a JK flip flop, the state of the JK flip flop represents the set_time state. When pressed the set time state changes 0 to 1 or 1 to 0.

Problem: If the set_time input changes from off to on when the clock is in ‘1’ state, it accidentally causes an effect similar to a ‘falling edge’, incrementing the ‘minute’ by 1 before pausing. However this is inconsiderable if the buttons/pins are only used as in design.

Stage 3: Implement the hours display and complete the Set Time functionality.

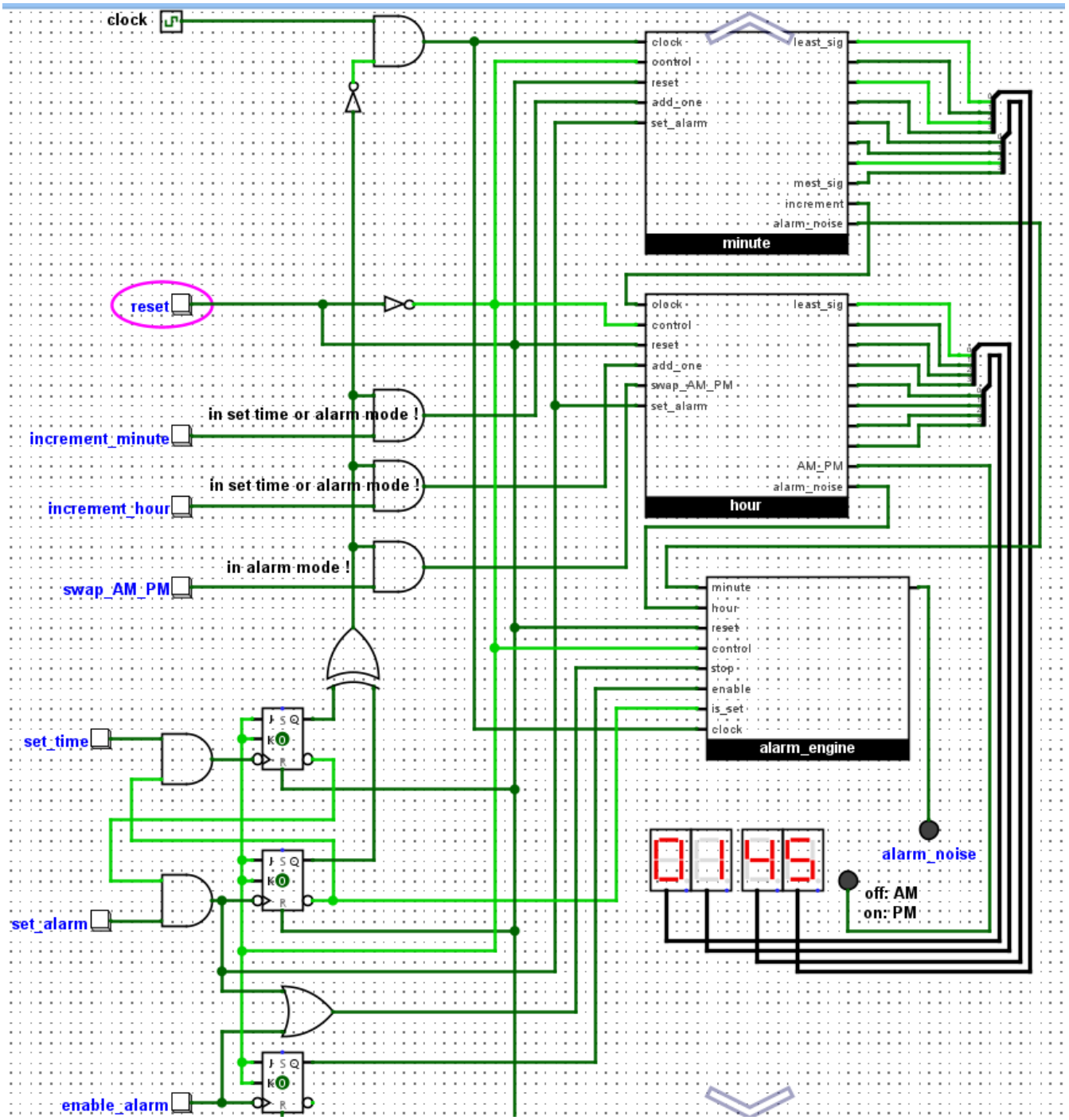


And inside the mod-2-counter + AM-PM calculator

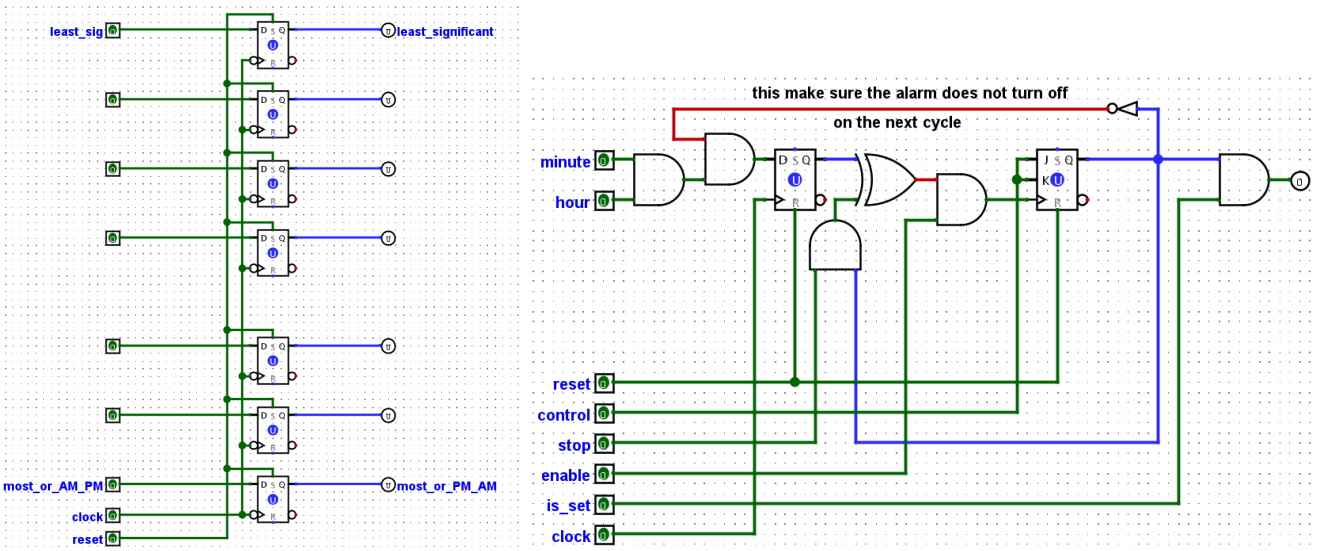


- A mod-ten-counter is used to count ‘units’ of the hour
- The ‘tens’ digit value of the hour is only 0 or 1 - similar to that of a JK flip flop
- The am pm is the same, we can set 0 for am and 1 for pm.
- → 2 sets of a D and a JK are used.
- 3 most significant bit always = P AND (NOT P) = 0
- **Problem:** The JK flip flop of ‘tens’ digit only changes when the ‘unit’ digit reaches 9. We want it to do the same when the hour reaches 12. However, due to the encapsulation we cannot connect to the JK flip flop inside the mod-ten-counter.
- **Solution:** use an AND gate to identify if the final output reaches 11 and a D flip flop as buffer to synchronize the changes so the reset is activated when the clock reaches 12. This has to also trigger the clock of the ‘tens’ digit to trigger the D flip flop to change (because the ‘tens’ digit only changes when the ‘units’ reaches 9, not 2.

Stage 4: Integrate the hours and minutes of the clock display so that hours and minutes tick with the system clock

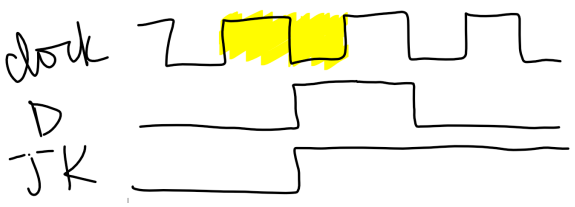


Stage 5: Set alarm functionality.



Inside the alarm register and alarm engine

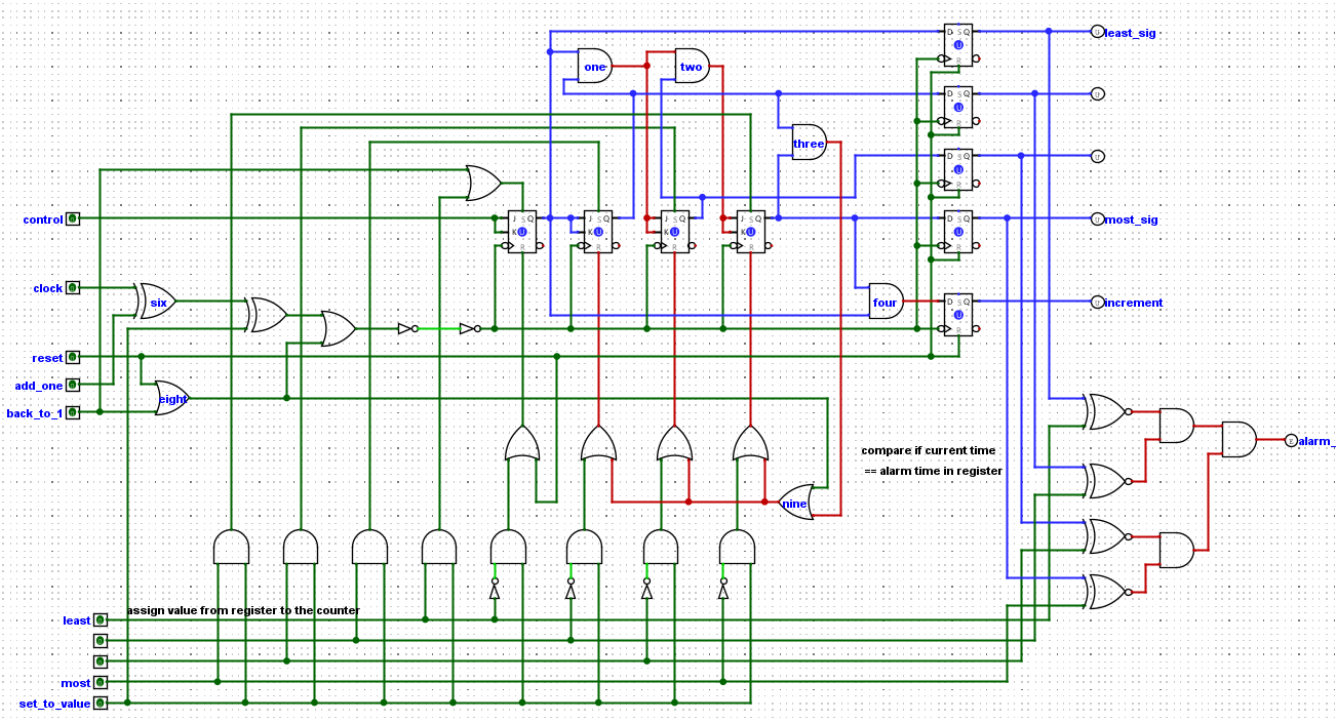
- A 7 bit register is added to store the current time value in set_alarm mode and to store the alarming time value when the clock is running
- An 'enable' input is added to control the alarm signal (using an AND gate)
- Problems:
    - When set time and set alarm are both on, they eliminate each other function
    - Unexpected behavior: example when alarm is set to 18:00. The clock is at 18:59. It increment the minute slightly faster that the hour and cause the alarm to be triggered because there is no synchronizing between the minute and the hour engine: 18:59 → **18:00** → 19:00
    - Initially the register has the value 0000000 corresponding with 00:00 AM which triggers the alarm message immediately when the clock starts running
    - When first reaches the alarm time the 'alarm_noise' output turns on (0 to 1) and remains unchanged until the second time it reaches the alarm time (1 to 0). This is expected since the alarm massage has to stay on until it is turned off manually.
    - The stop button turns the alarm message from 1 back to 0 but also turns 0 to 1 when the alarm output is 0.
- Solutions:
    - A structure using AND gates are used to prevent set_time or set_alarm to be turned on when the other is not off.
    - Use a D flip flop as a buffer but configure it to be triggered on 'rising edge' because we do not want the alarm output to turn 1 **after a delay of 1 minute.** Though this still makes the alarm message turn on a bit late (half a clock period), however



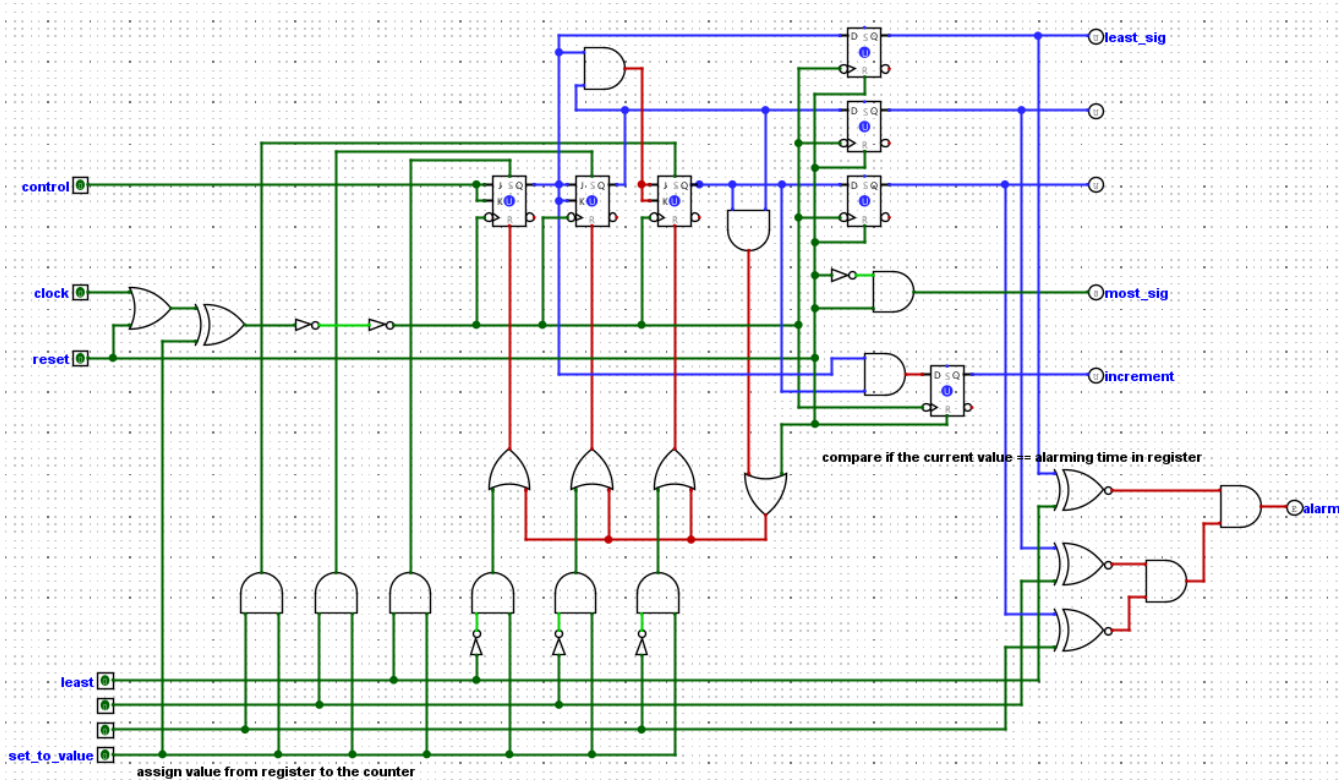
this is inconsiderable since the alarm still turns on the same turn (with a bit delay).

- Initially set the register to 1111111, this is an invalid time value so the alarm could never be triggered when alarm time is not set.
- An AND gate is added with one input being the output state, this indicates that once the alarm message is on, the next cycle collision will not affect the output 'on' state.
- The 'stop_alarm' input connects to an AND gate with the alarm output. The stop button now works only when the alarm output is on.

- All counters are equipped with a construction to assign the value from register to current counter value and another to compare between the alarm value stored in register and the current time
  - NXOR gates for comparing
  - AND gates as controlled switch
  - NOT gates for activating RESET when the register value is 0



Mod_10_counter



Mod_6_counter