Ultrasonic Sensing

Measuring distance and displaying it on a seven-segment display

Nguyen Phan

Ayoub Kakish

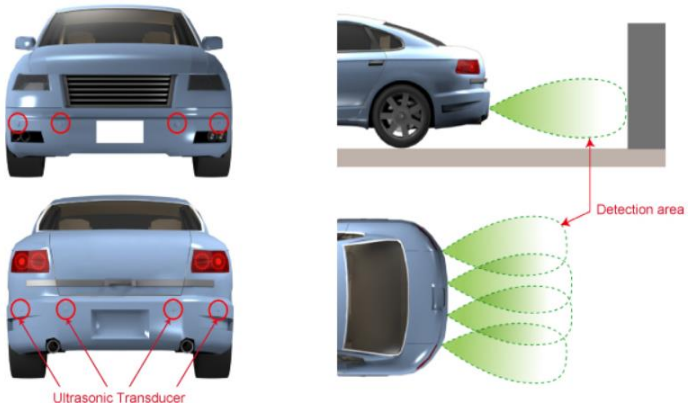
Nick Yang

Alex Alnajjar

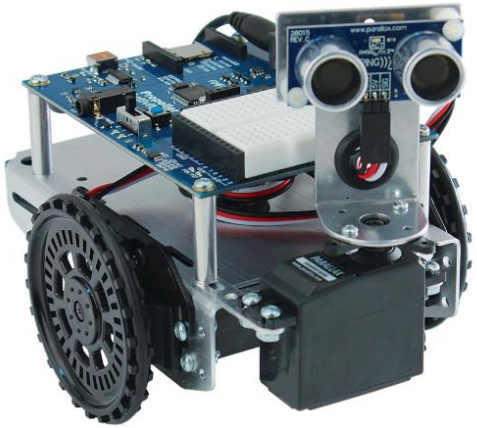
Objective

1. Ultrasonic technology can be used to measure distances
2. We can use an objects distance to control a system
3. Use ultrasonic sensor to measure distance and display in centimeters on display

<Hella oil level sensor>



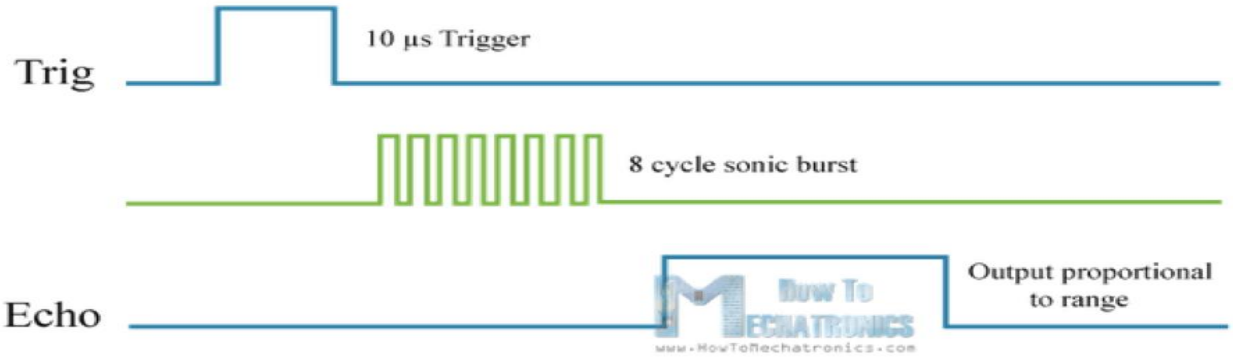
<Parking sensors>



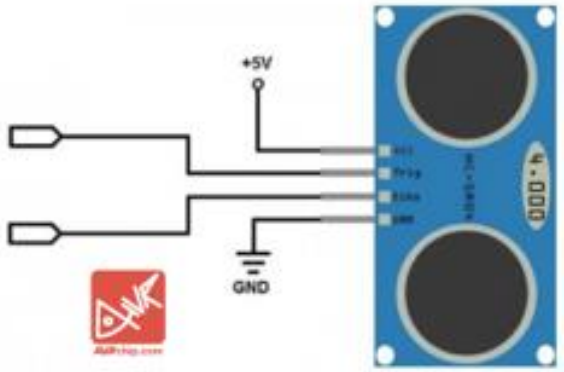
<Autonomous Driving>

Proximity Sensor: HC-SR04

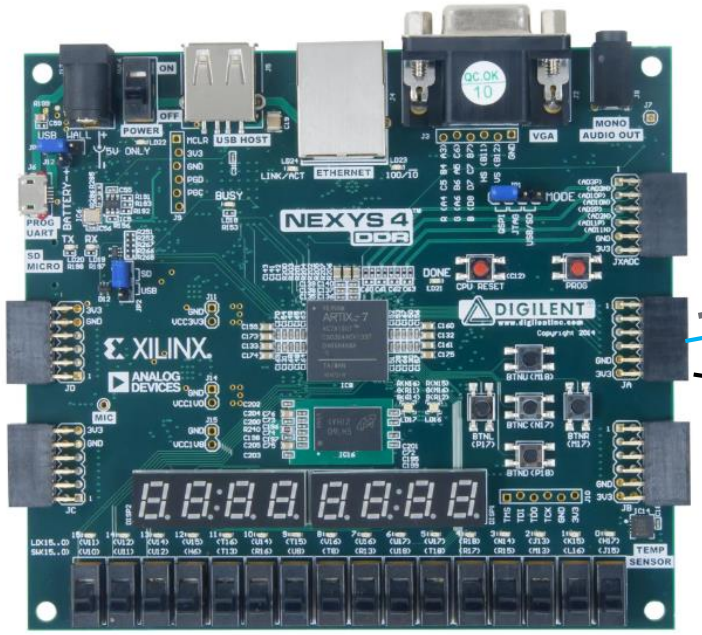
•The proximity sensor emits an ultrasound at 40 kHz. When there is a type of object in front of the sensor, the ultrasound bounces back to the proximity sensor. The distance of the object can be calculated when considering the travel time and the speed of sound.



•The Proximity Sensor has 4 pins: Ground, Vcc, Trig and Echo.



System



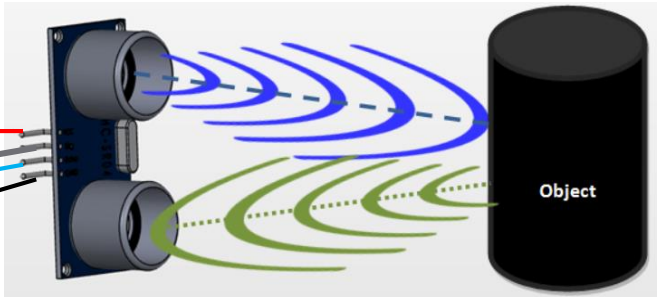
Trigger

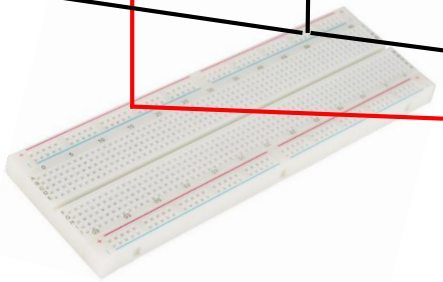
Echo

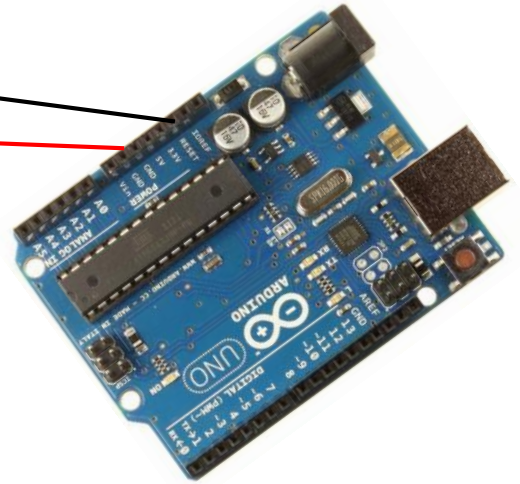
GND

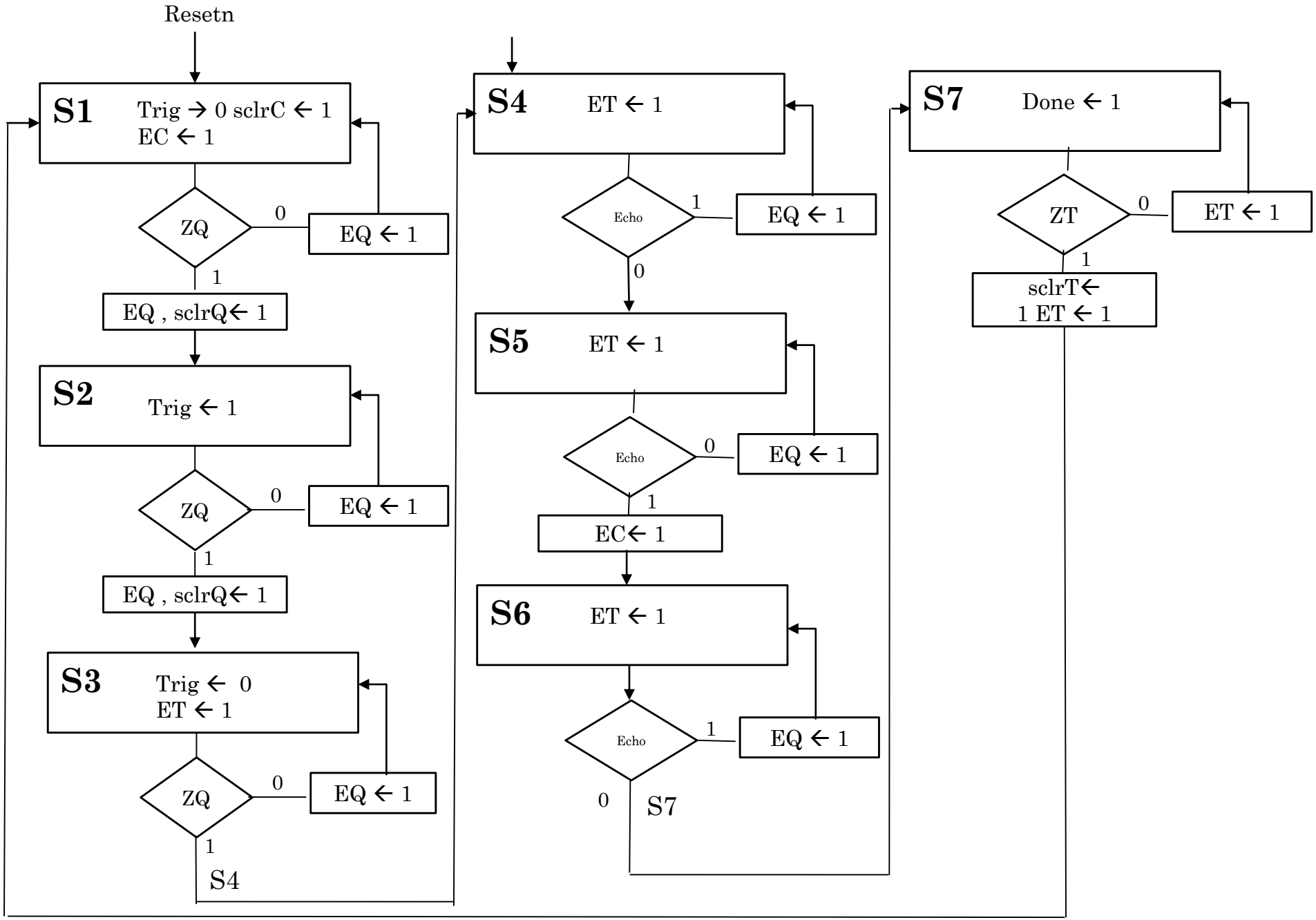
Vcc

GND

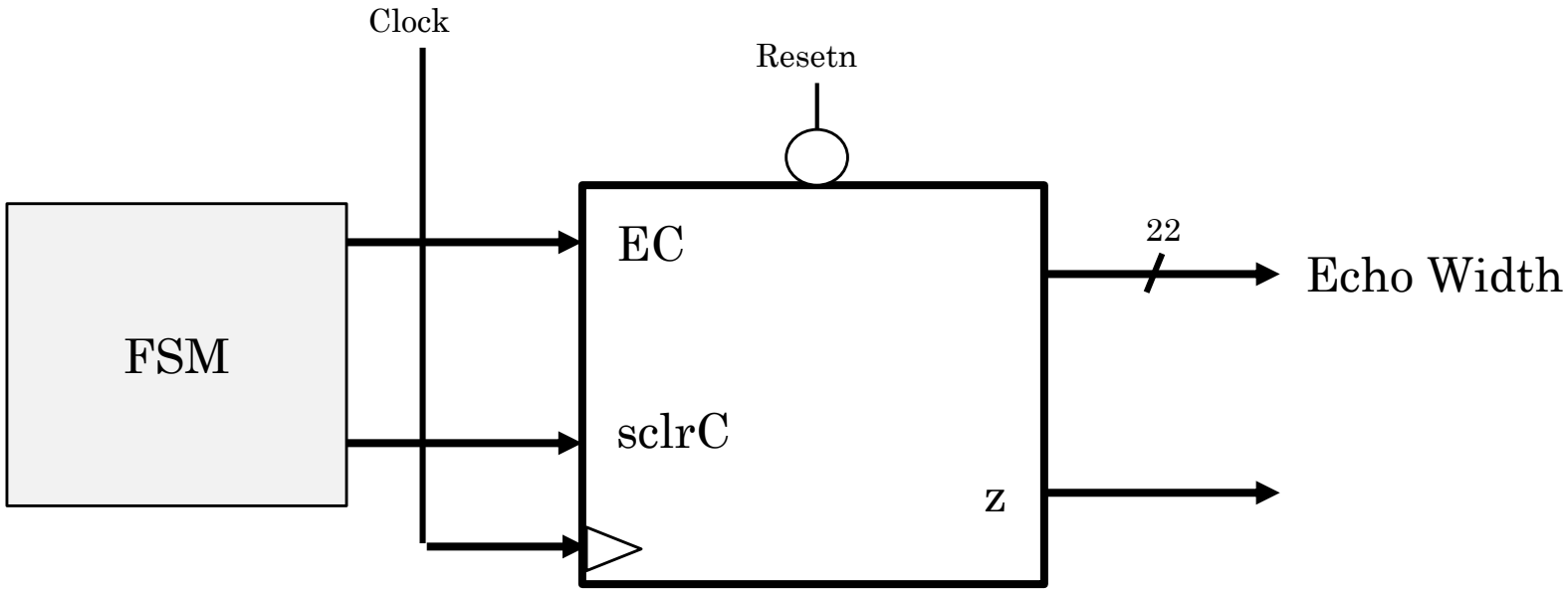


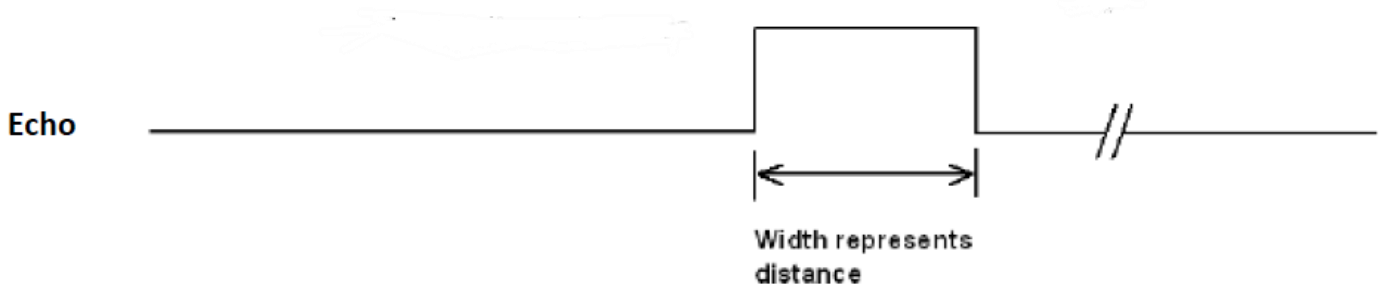





Resetn
S1
Trig → 0 sclrC ← 1
EC ← 1
ZQ
0
EQ ← 1
1
EQ , sclrQ← 1
S2
Trig ← 1
ZQ
0
EQ ← 1
1
EQ , sclrQ← 1
S3
Trig ← 0
ET ← 1
ZQ
0
EQ ← 1
1
S4
S4
ET ← 1
Echo
1
EQ ← 1
0
S5
ET ← 1
Echo
0
EQ ← 1
1
EC← 1
S6
ET ← 1
Echo
1
EQ ← 1
0
S7
S7
Done ← 1
ZT
0
ET ← 1
1
sclrT←
1 ET ← 1

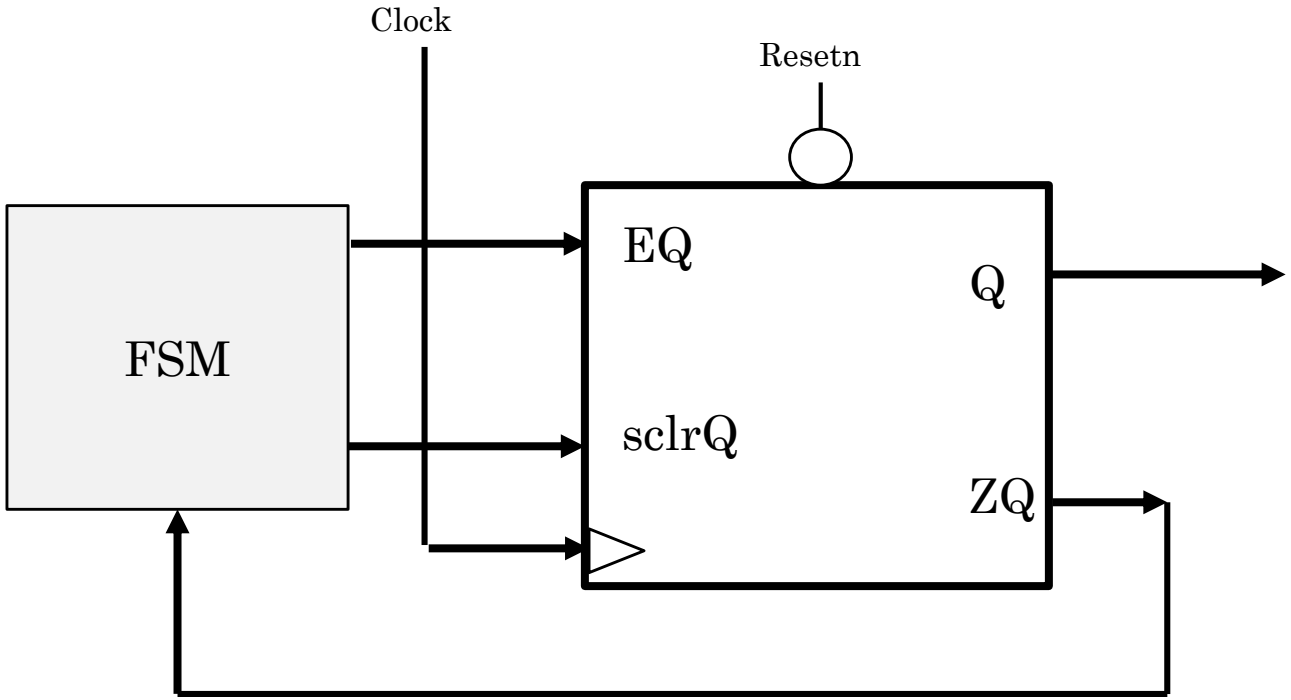
Modulo-Counter

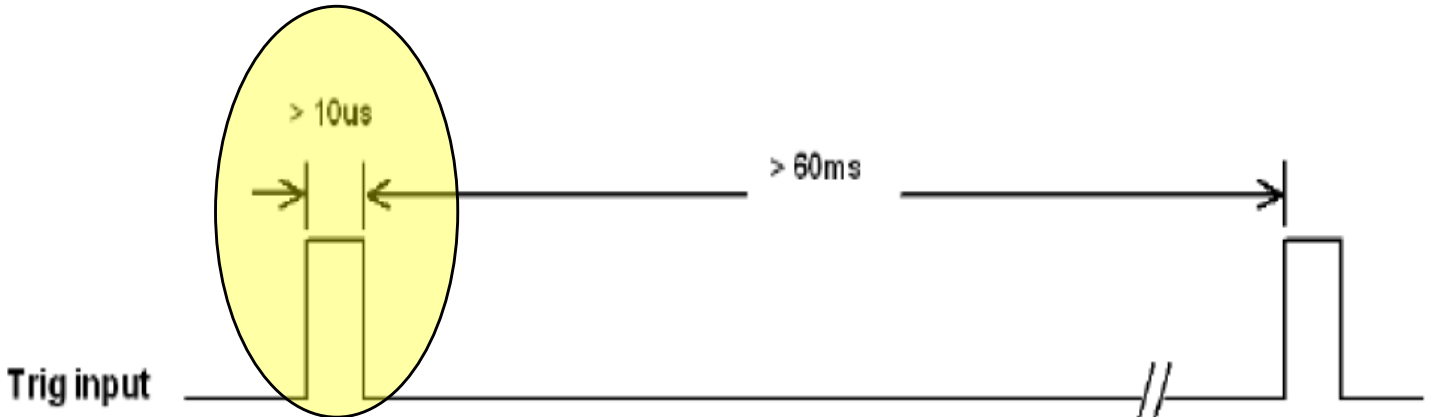




- Modulo-Counter is a counter that can output a number of different states
- The output of the counter is used to calculate the distance

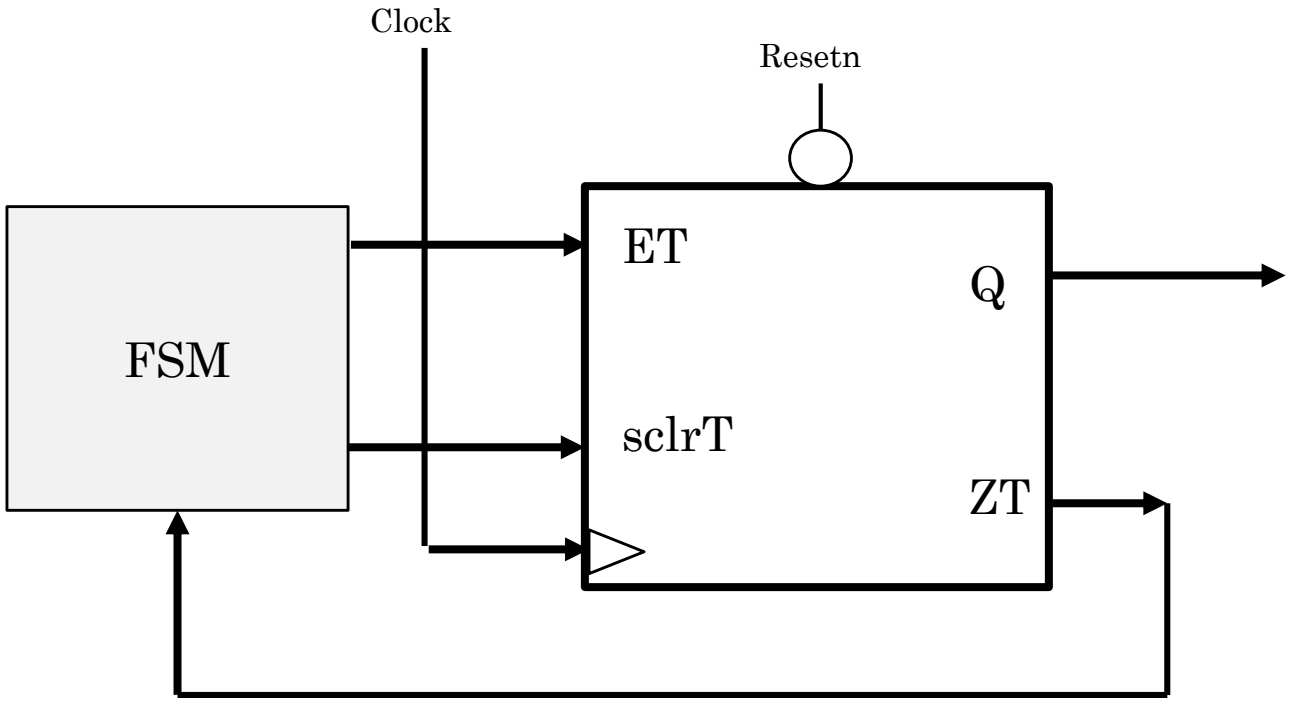
Counter (10µs)

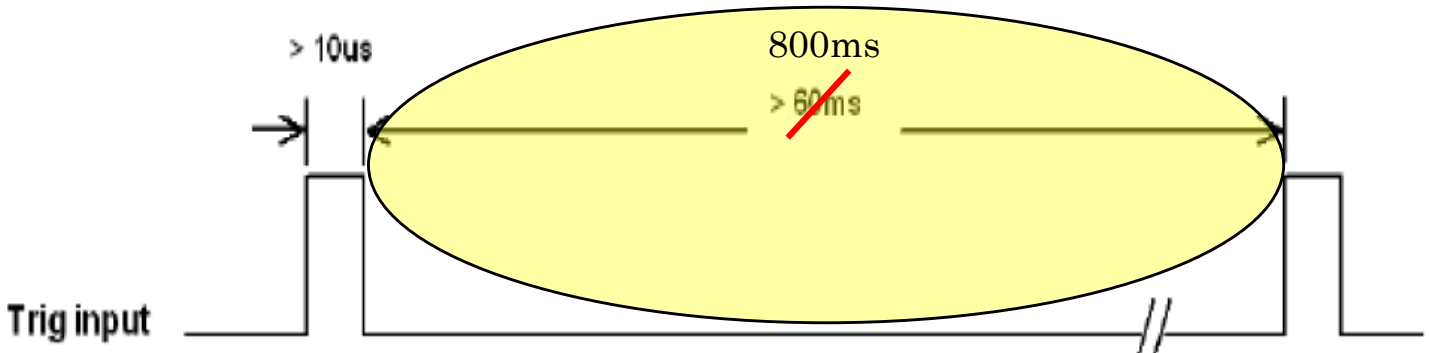




- This counter is responsible for the trigger input
- Every 800ms the trigger input goes to 1
- This is the pulse that is generated from the FSM to the sensor
- This pulse is output from the trigger of the sensor searching for the object in front

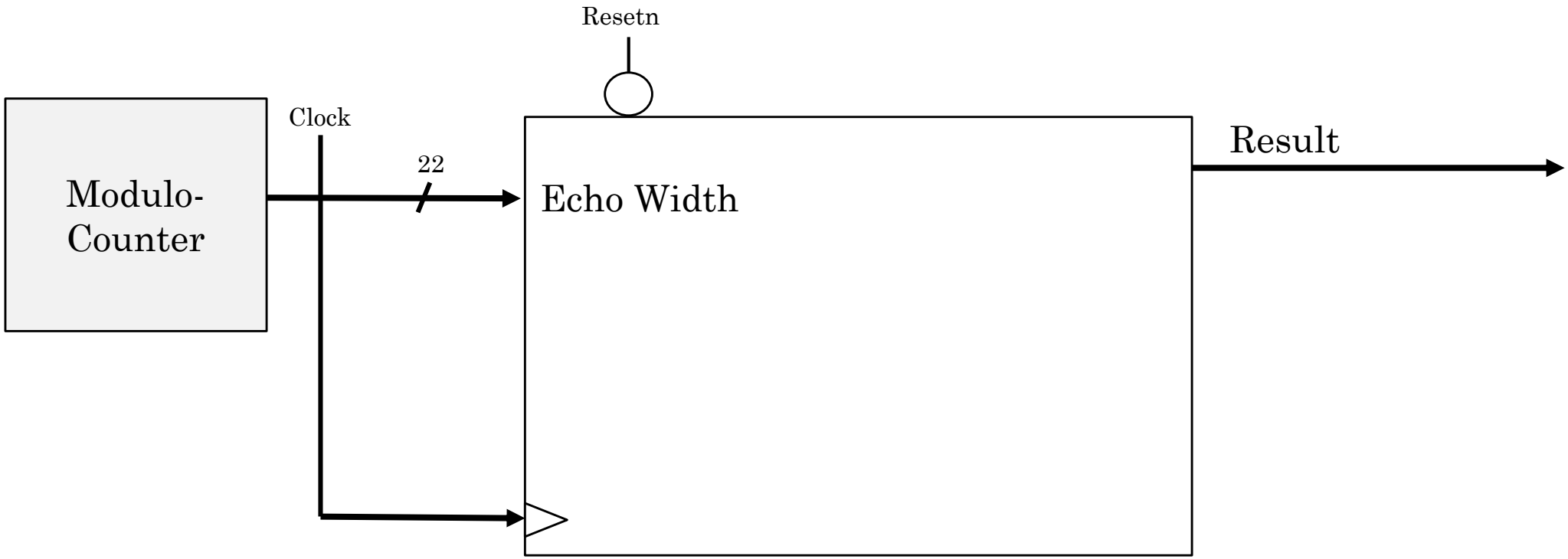
Counter (800ms)





- This clock is responsible for the delay in between the trigger
- Initially 60ms was used, but that was too fast
- It was set to 800ms to make the display clear and transition of numbers smoother

Distance Calculator



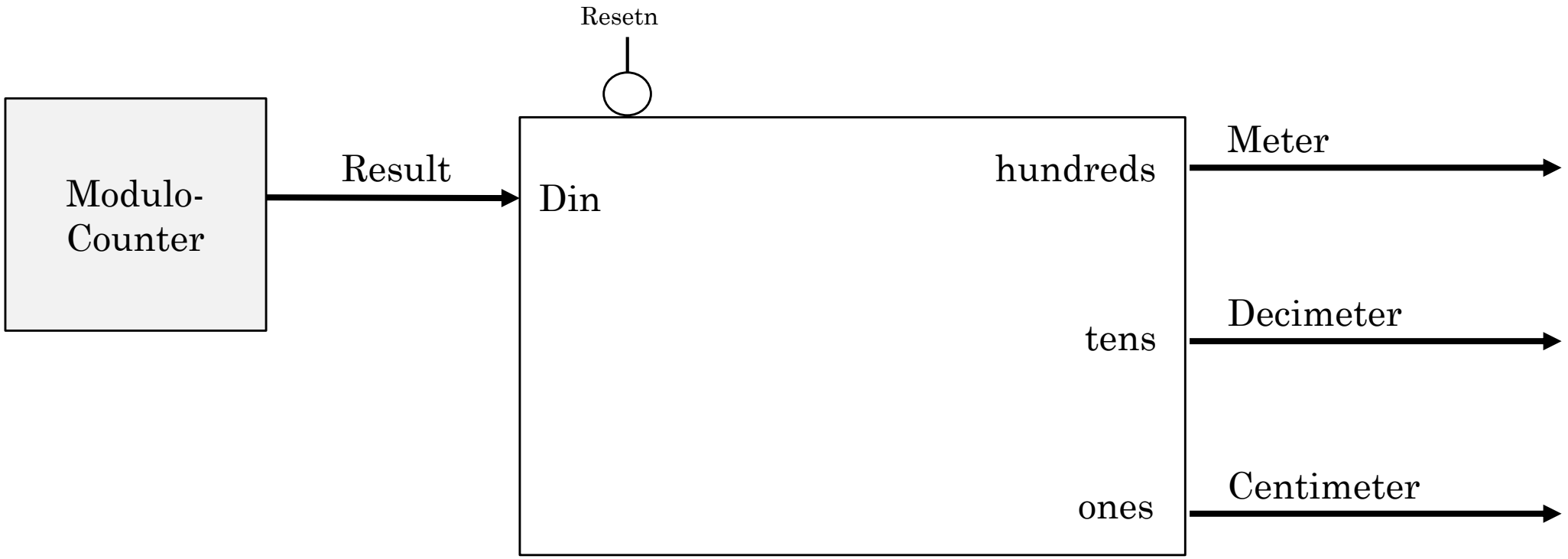
- Needed to convert received echo width from nanoseconds to microseconds so that the equation was correct.
- To convert echo width to microseconds → $\frac{Echowidth}{1000}$
- Distance(cm) = $\frac{\frac{Echowidth}{1000}}{58} = \frac{echowidth}{58000}$ → 22 bit * 20 bit = 42 bit
- In VHDL we needed to multiply by $\frac{1}{58000}$ → $1.724*10^{-4}$

000000000000000000.000000000000000000

- Only the 9 bits before the decimal were used

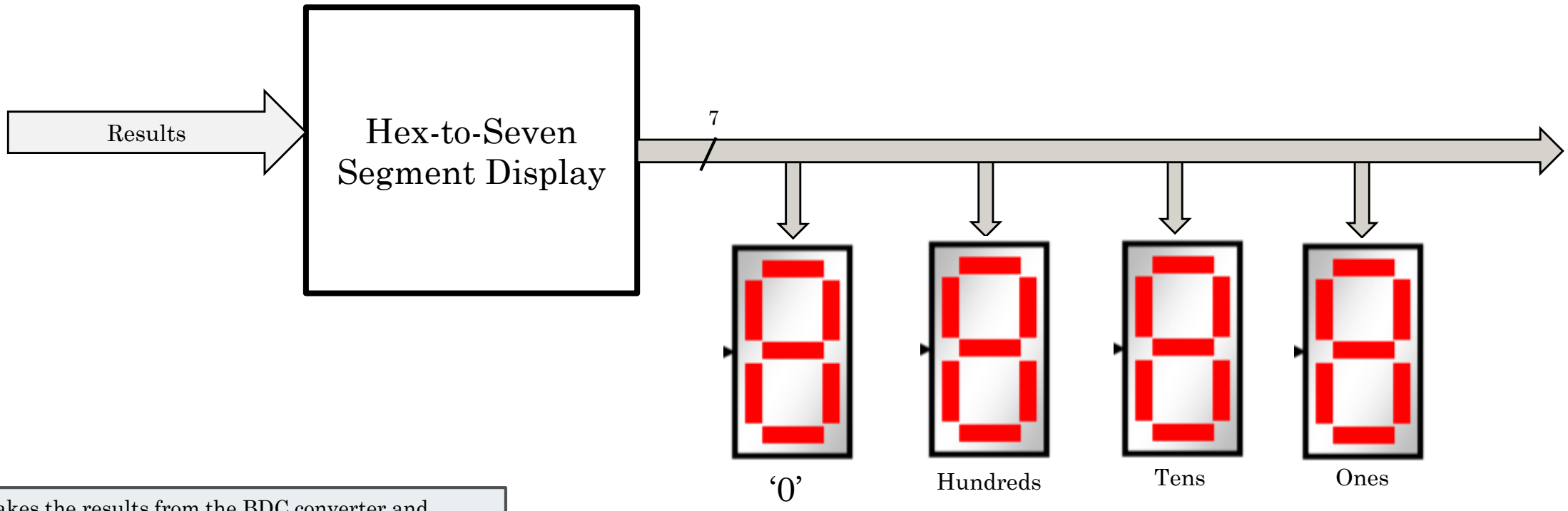
$$\boldsymbol{Distance(cm) = \frac{echowidth(\mu s)}{58}}$$

BCD-Converter



- Here we took numbers calculated from the Distance Calculator and converted them into BCD binary
- These numbers were then sorted into hundreds, tens, and ones decimal places so that on the display the correct number was displayed
- The output of the BCD converter then were inputted into the Hex-to-Seven-Segment Display

Hex-to-Seven Segment Display

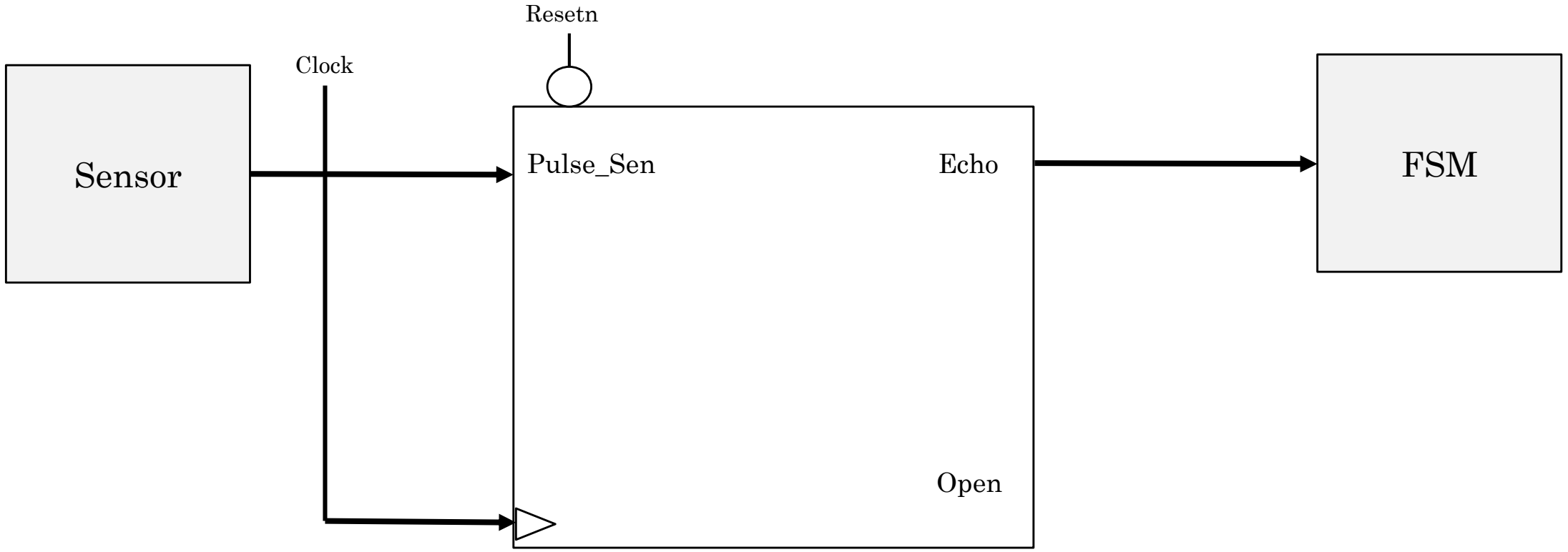


- Takes the results from the BDC converter and displays the distance in cm
- 4 displays will be on

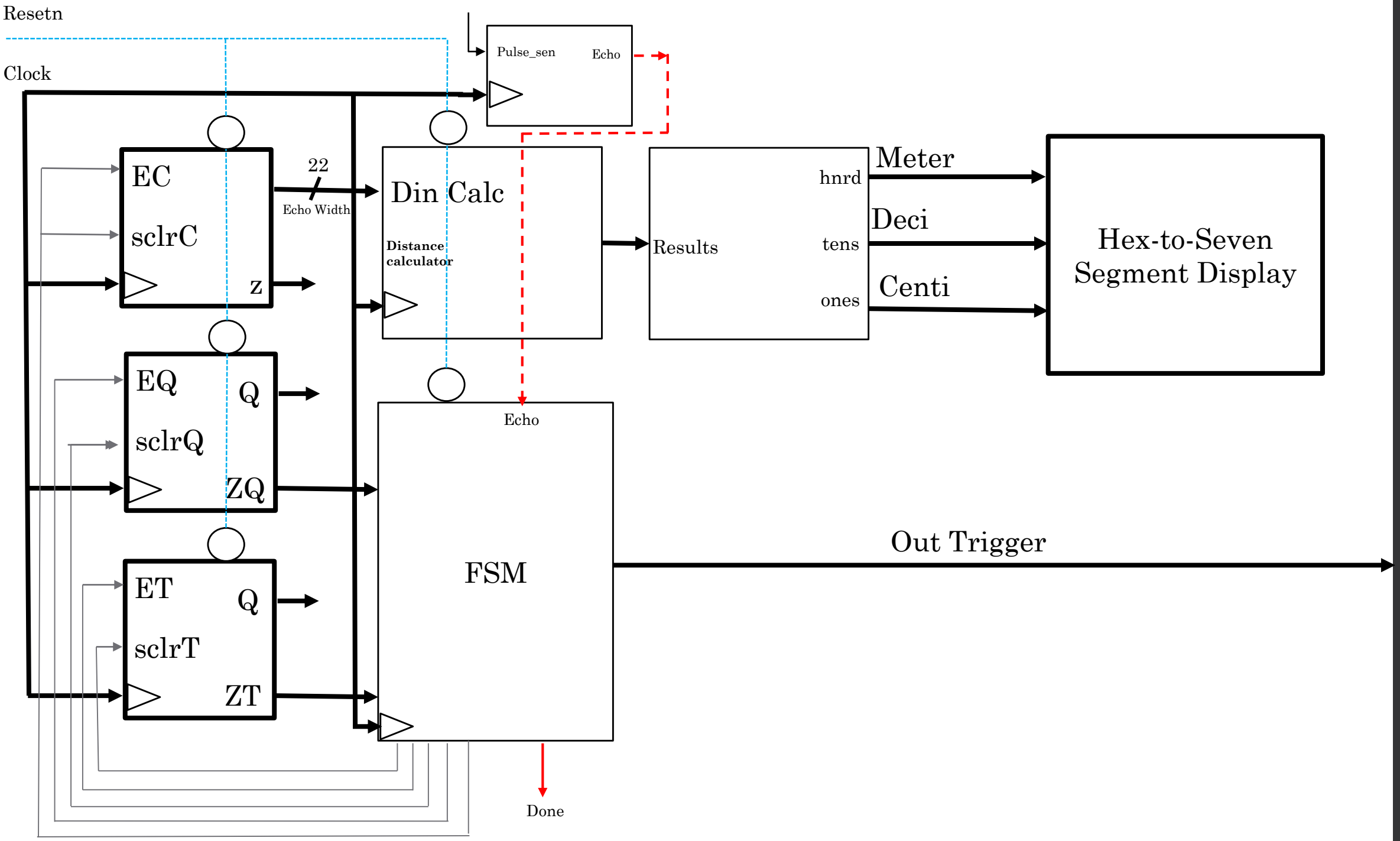
→ 1 staying at zero
→ Hundreds
→ Tens
→ Ones

- As the object moves closer and further away, the display should change seamlessly

Synchronous D-Flip-Flop



- This component was added later in our project to make the system work smoother
- Initially, the display would work fine for about a minute and then read only “0000”
- This was added to make sure the sequence continued for longer durations which worked successfully


Resetn
Clock
Pulse_sen
Echo
EC
sclrC
z
22
Echo Width
Din Calc
Distance calculator
Results
hnrd
tens
ones
Meter
Deci
Centi
Hex-to-Seven Segment Display
EQ
sclrQ
Q
ZQ
Echo
FSM
Out Trigger
ET
sclrT
Q
ZT
Done