Alarm Clock

By:

Grant Parker, James Fazecas, Jacob Monks, Nate Dumitrescu

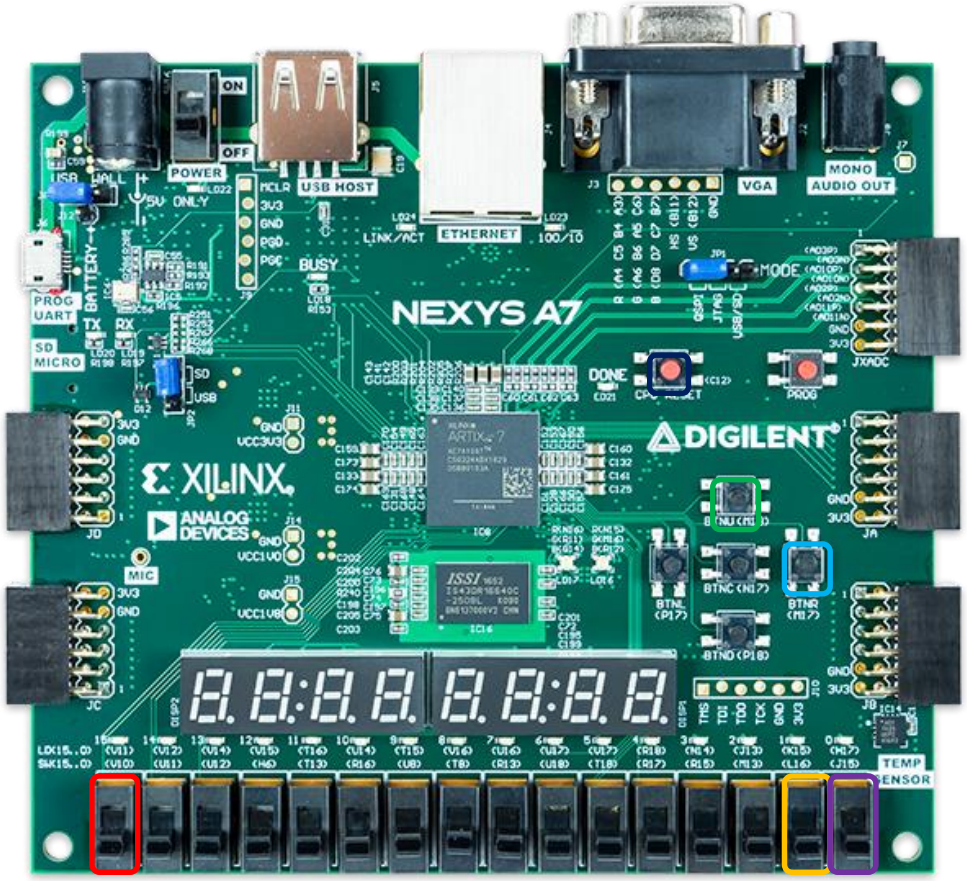
Video Demonstration

- https://www.youtube.com/watch?v=oZfep3RcyMA

Overview

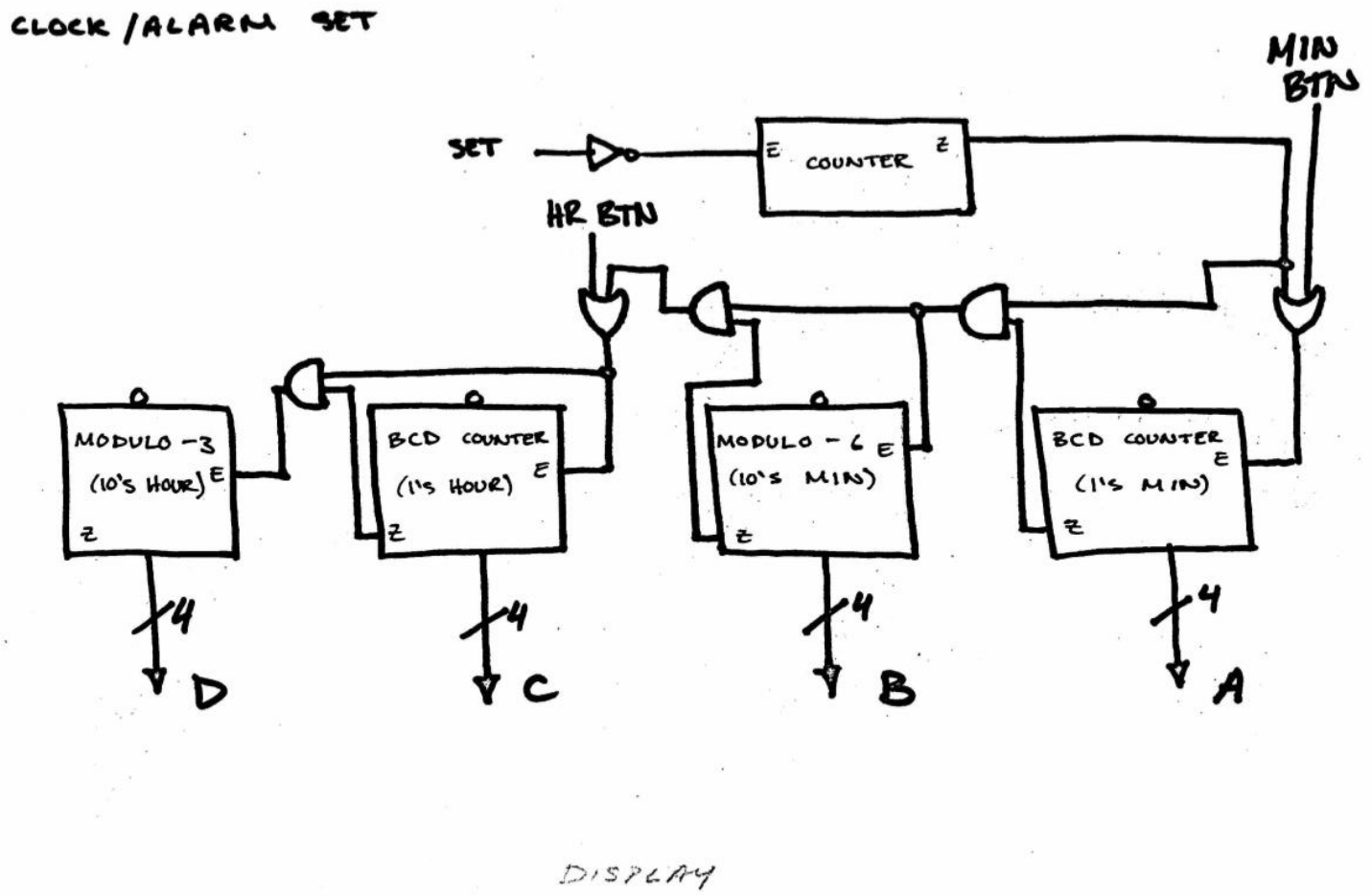
- The end goal is to set a time for a buzzer to go off, indicating that the requested time input has passed.
- Use of VDHL code for implementation
- 7Segment Display shows the time in Military Form (24 hour)
- Button CPU Reset (C12) used for reset. Switch 1 (J15) is used to turn on the display, and Switch 2 (L16) is used to start the clock. Switch 15 (V10) is used to set an alarm time, and the upper (M18) and right (M17) buttons are used for toggling hours and minutes, respectively.

Hardware

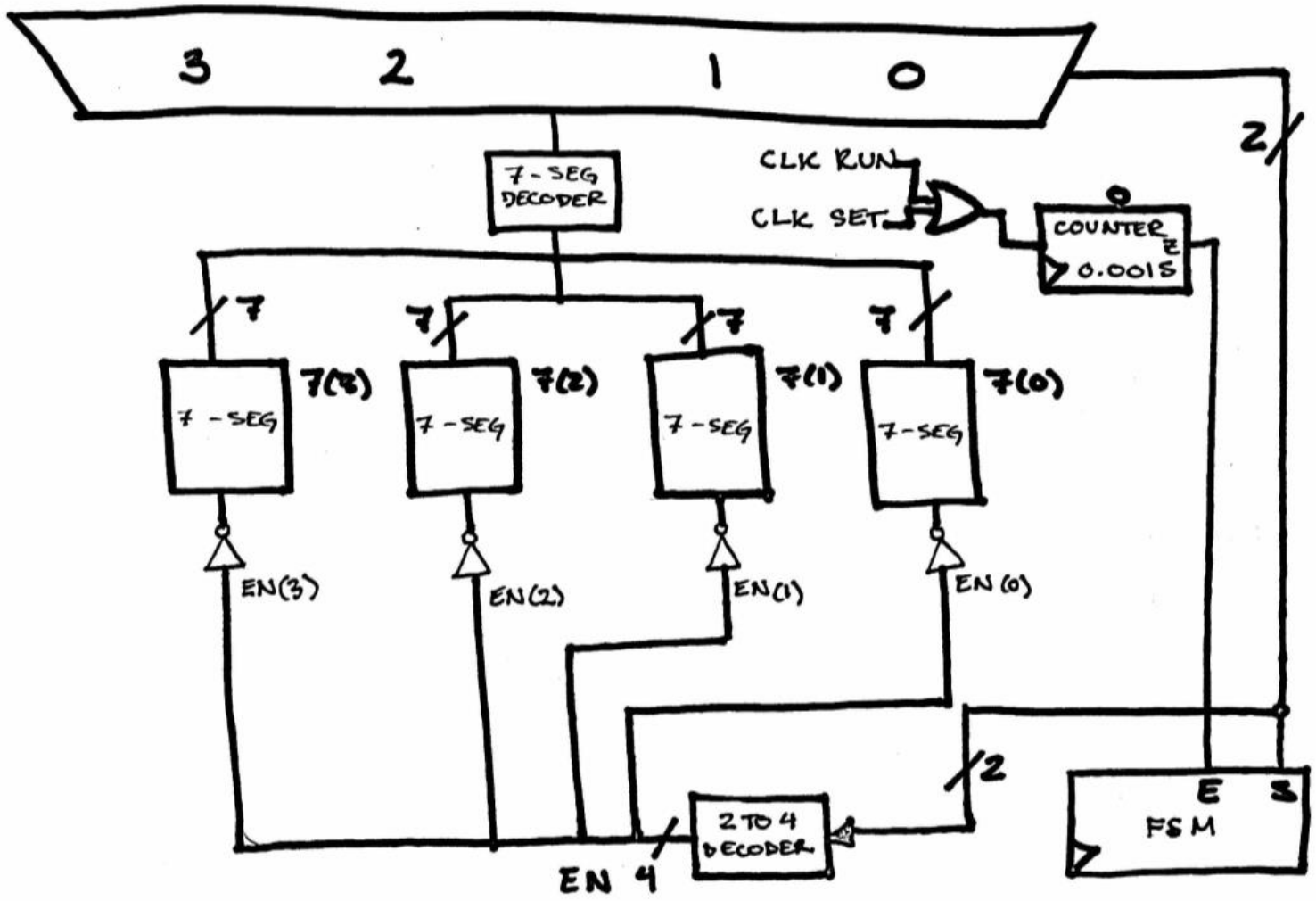


- Alarm Set
- Turns Clock On
- Turns Display On
- Minute Value Selector
- Hour Value Selector
- Reset

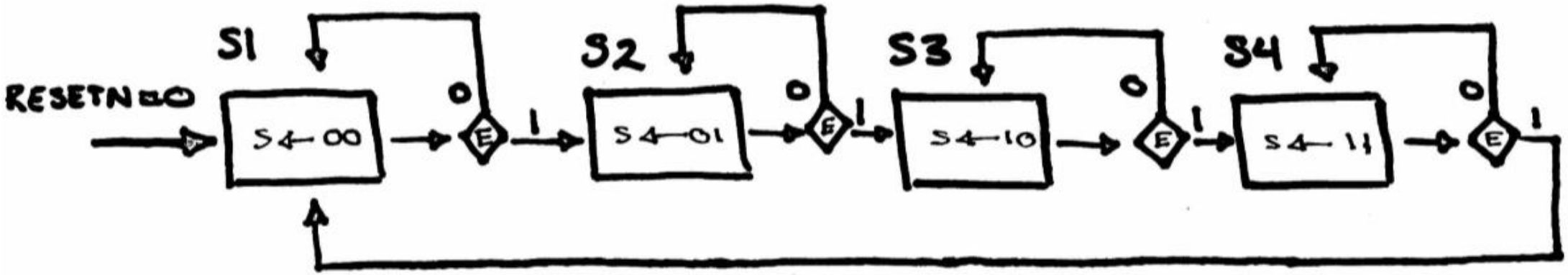
Clockset


CLOCK / ALARM SET
MIN BTN
SET
E COUNTER Z
HR BTN
MODULO -3 (10's HOUR) E Z
BCD COUNTER (1's HOUR) E Z
MODULO - 6 (10's MIN) E Z
BCD COUNTER (1's MIN) E Z
4
4
4
4
D
C
B
A
DISPLAY

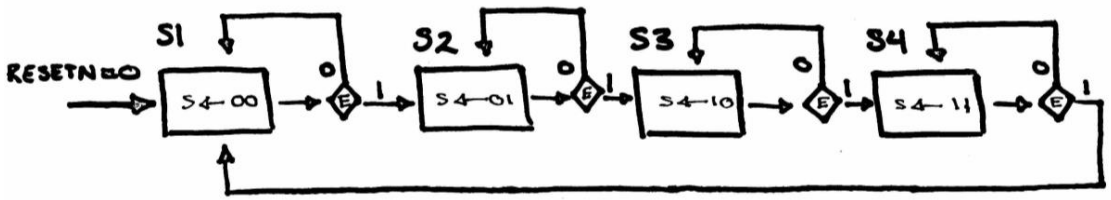
Display

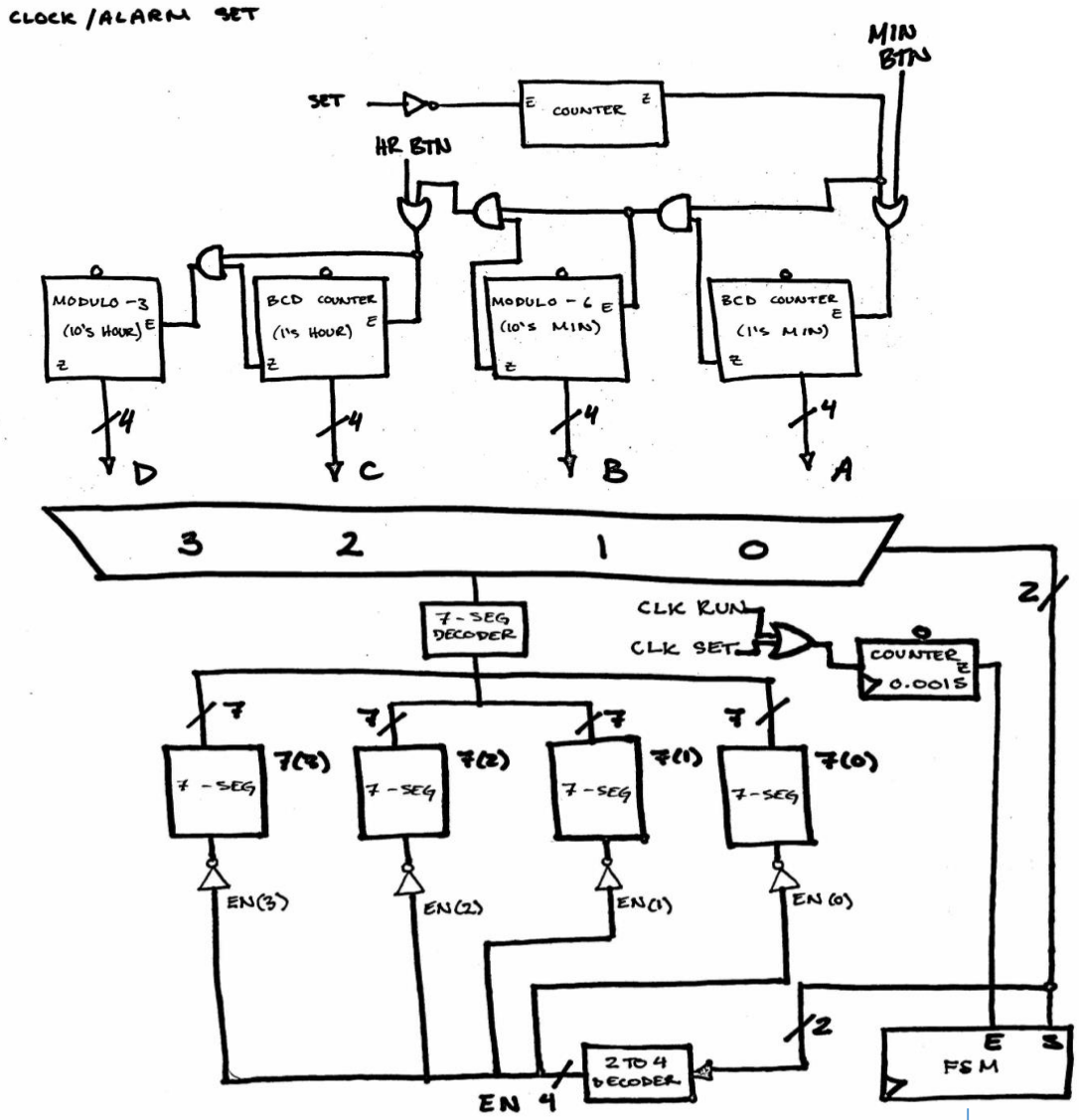

3
2
1
0
7-SEG DECODER
CLK RUN
CLK SET
COUNTER
0.001S
2
7
7
7
7
7-SEG
7-SEG
7-SEG
7-SEG
7(3)
7(2)
7(1)
7(0)
EN(3)
EN(2)
EN(1)
EN(0)
2TO4 DECODER
2
EN 4
E
S
FSM

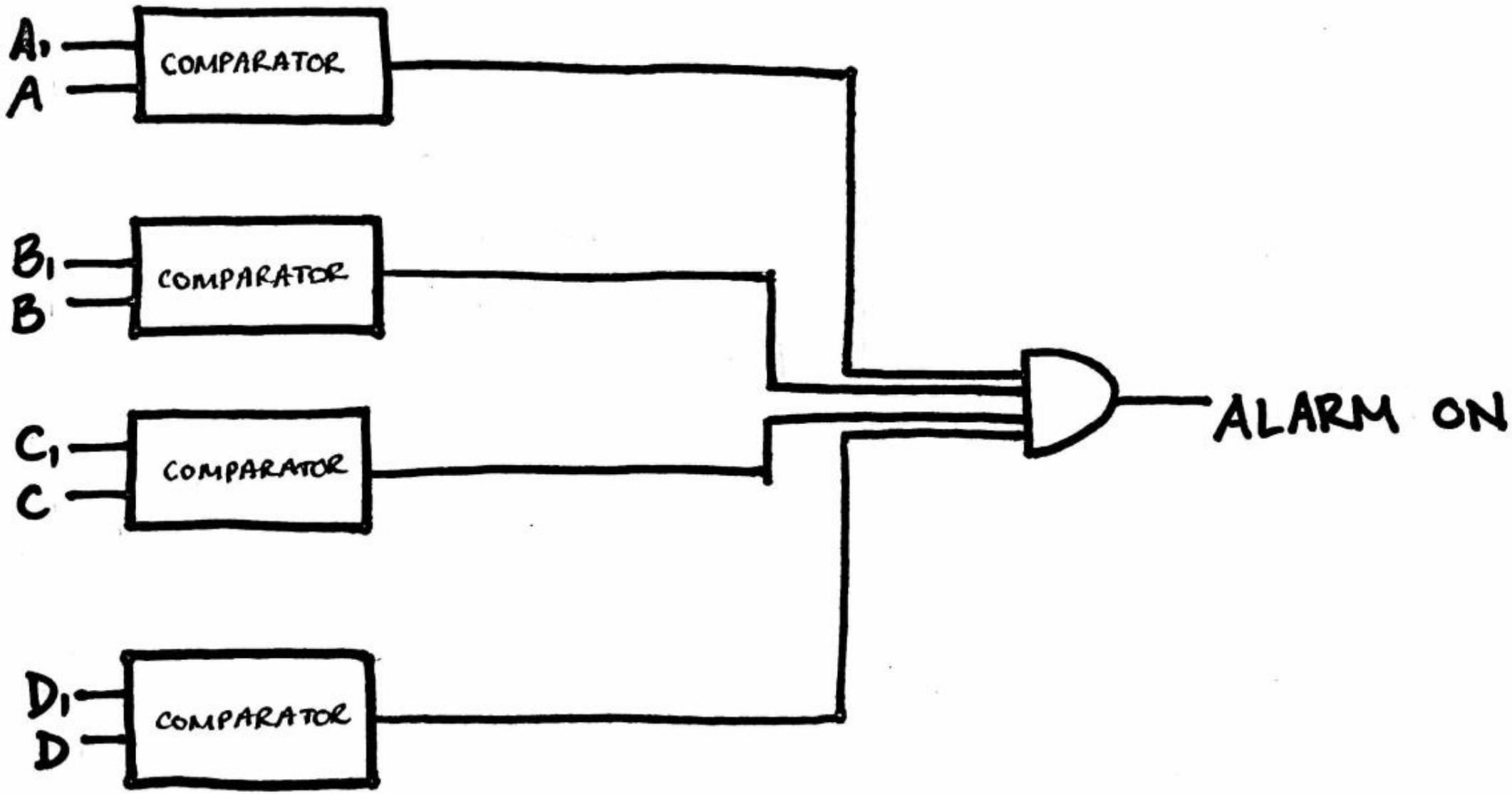
FSM



Overall Diagram Here






A1
A
COMPARATOR
B1
B
COMPARATOR
C1
C
COMPARATOR
D1
D
COMPARATOR
ALARM ON

Issues/Debugging

- The buttons had to be debounced in order to change the time. Button debouncing was problematic. Instead we implemented a clock pulse circuit.
- The Clock and Alarm were both being changed by button presses. This was fixed with a simple “if” statement in the TopFile.

Why we chose the alarm clock?

- The alarm clock is an everyday tool that is crucial to modern day business and tasks. This experiment helped us gain an understanding of how valuable the counters in VHDL really are, and we no longer take the alarm clock for granted.


Questions