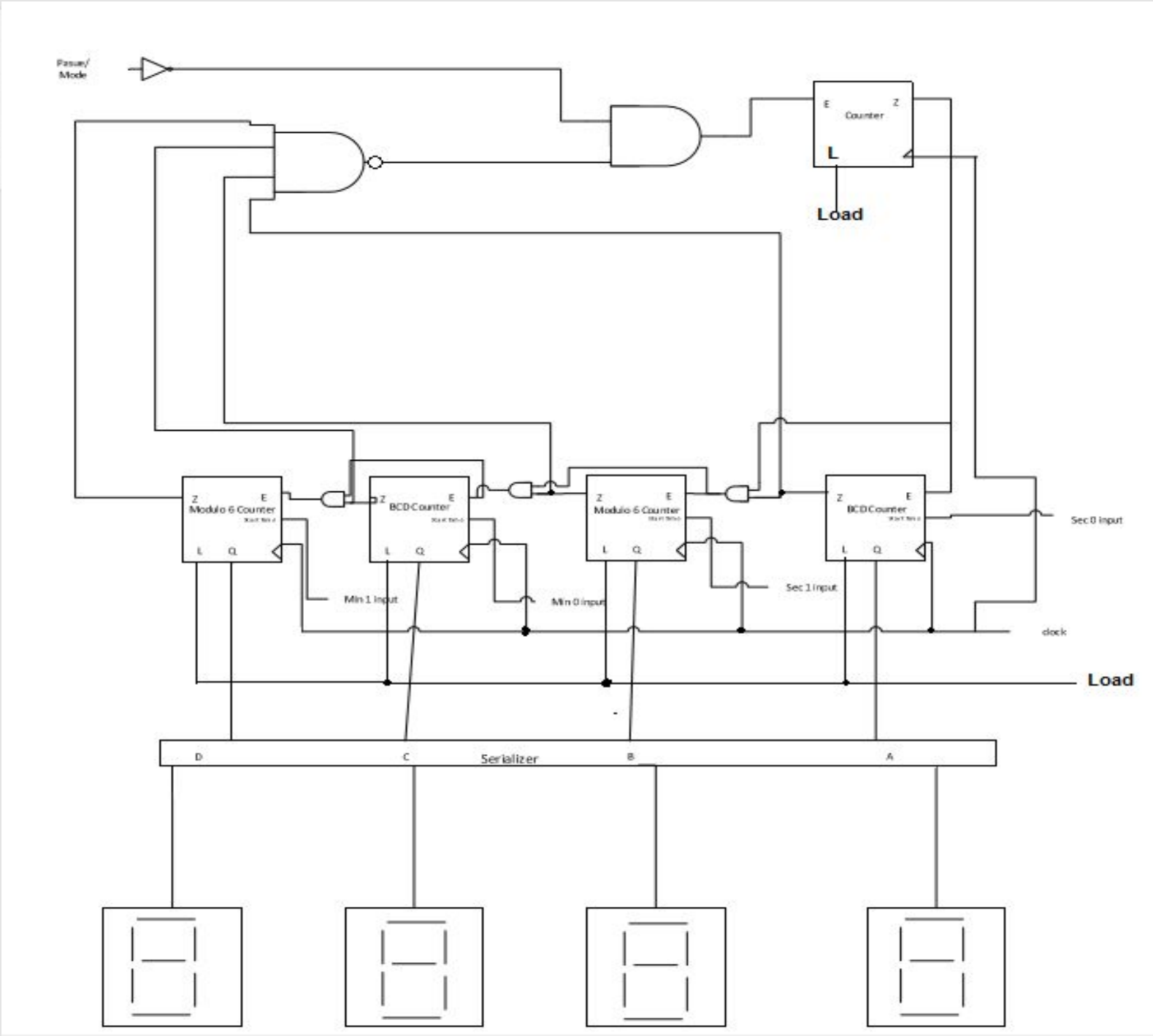
Project: Count Down Timer

Design Engineers:
Syed Hussain
Matthew Buffa
Tala Fakhouri
Rohit Timmagi

Project: Count Down Timer

- Why Timer?
- The need for Count Down Timer is very essential in our life. We can use it to cook, study, test our self, and has many other use cases.
- Functionalities
- Set any time from 59 min, 59 sec to 0.
- Pause the time.
- Reset time.
- led light shows to indicate time has ended.

Schematic


Pasuay/
Mode
E
Z
Counter
L
Load
Z
E
Modulo 6 Counter
L
Q
Z
E
BCD Counter
L
Q
Z
E
Modulo 6 Counter
L
Q
Z
E
BCD Counter
L
Q
Sec 0 input
Min 1 input
Min 0 input
Sec 1 input
clock
Load
D
C
Serializer
B
A

BCD and Modulo-6 Counters

Modulo-6 Counter

```
6     Port ( clock : in  STD_LOGIC;
7            resetn : in  STD_LOGIC;
8            enable : in  STD_LOGIC;
9            Q : out  STD_LOGIC_VECTOR (2 downto 0);
10           z : out  STD_LOGIC;
11           startTime: in STD_LOGIC_VECTOR (2 downto 0);
12           load: in STD_LOGIC);
13 end modulo6counter;
14
15 architecture Behavioral of modulo6counter is
16 signal Qt : std_logic_vector (2 downto 0);
17 begin
18    process (resetn,clock, enable, startTime, load, Qt)
19       begin
20       if resetn = '0' then
21       Qt <= "101";
22       elsif (clock'event and clock = '1') then
23       if load = '1' then
24                Qt <= startTime;
25             elsif enable = '1' then
26                if Qt = "000" then --if it's enabled and it reaches zero
27                   Qt <= "101";--start back at 5
28                else --otherwise if it's not at 0
29                   Qt <= Qt - "001";--decrease count by 1
30                end if;
31             end if;
32       end if;
33    end process;
34
35 z <= '1' when Qt = "000" else '0';
36 Q <= Qt;
37
38 end Behavioral;
39
```

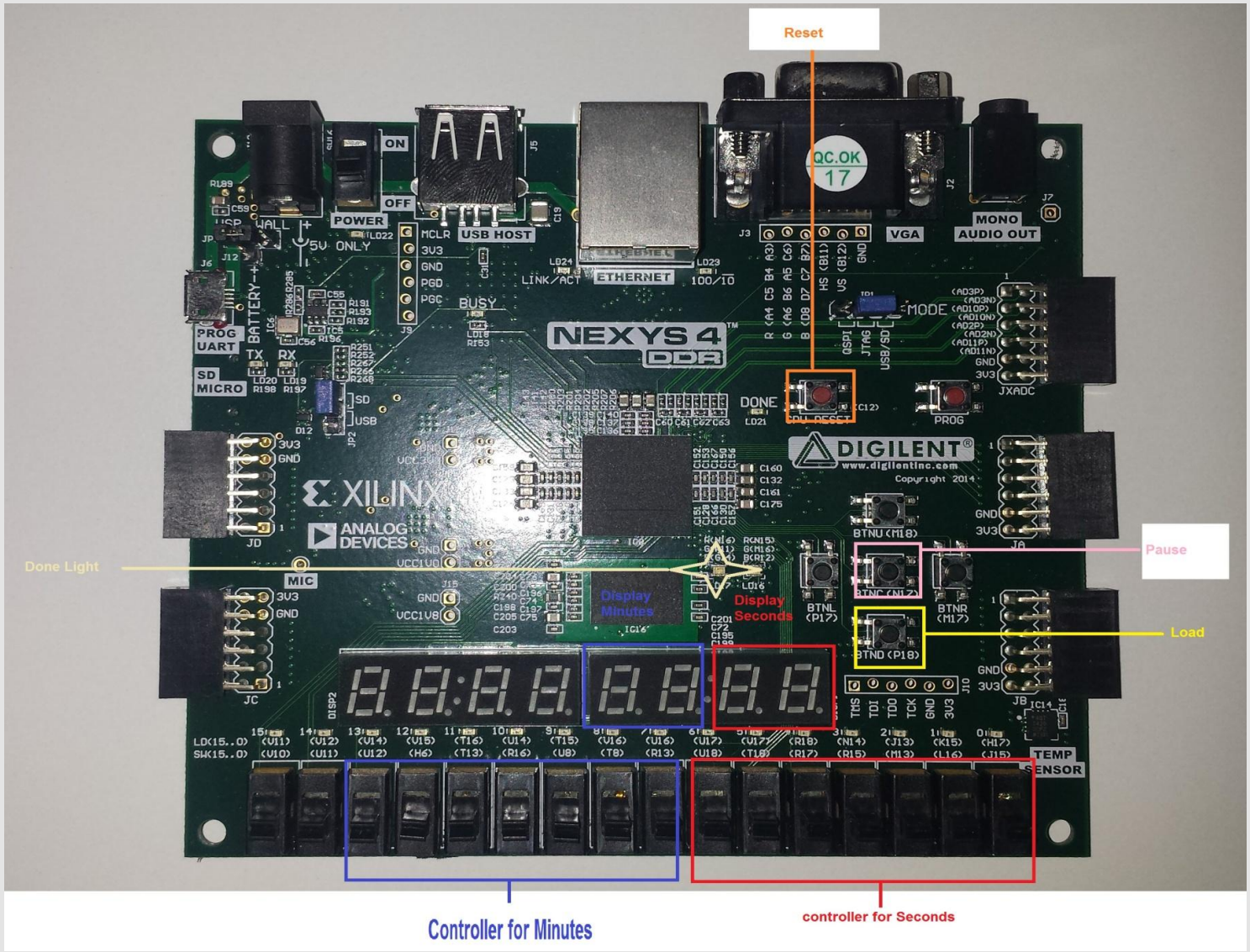
BCD Counter

```
4
5  entity bcdCounter is
6      Port ( clock : in  STD_LOGIC;
7             resetn : in  STD_LOGIC;
8             enable : in  STD_LOGIC;
9             Q : out  STD_LOGIC_VECTOR (3 downto 0);
10            z : out  STD_LOGIC;
11            startTime: in STD_LOGIC_VECTOR (3 downto 0);
12            load: in STD_LOGIC);
13 end bcdCounter;
14
15 architecture Behavioral of bcdCounter is
16 signal Qt : std_logic_vector (3 downto 0);
17 begin
18    process (resetn,clock, enable, startTime, load)
19       begin
20       if resetn = '0' then
21          Qt <= "1001";
22       elsif (clock'event and clock = '1') then
23             if load = '1' then
24                Qt <= startTime;
25             elsif enable = '1' then
26                if Qt = "0000" then
27                   Qt <= "1001";
28                else
29                   Qt <= Qt - "0001";
30                end if;
31             end if;
32       end if;
33    end process;
34
35 z <= '1' when Qt = "0000" else '0';
36 Q <= Qt;
37
38 end Behavioral;
```

Modified Counters to

- Load user input
- Decrement their value when enabled

Board Layout

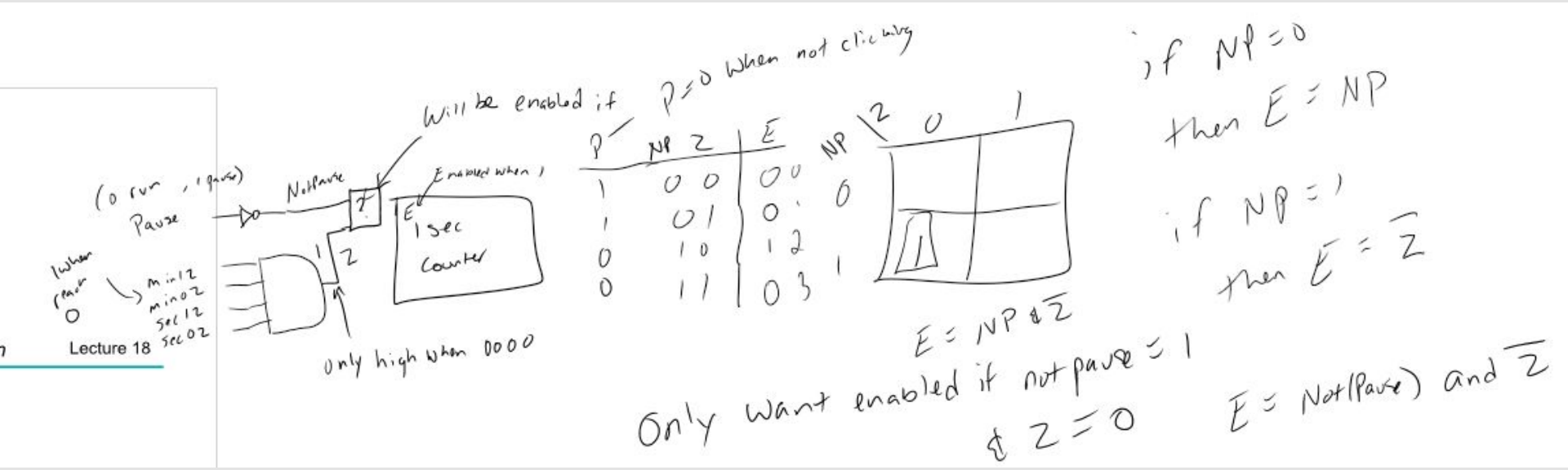


Overcoming Design Drawback

- Setting timer → Used Switches
- Getting the time to countdown
- Incorporated load as input to BCD, and modulo-6 counters
- Redefined our port map properly in order for the timer to countdown to fix warning.
- Having Time Pause at 00:00

→ Used K-map to derive Boolean equation that used inputs from BCD and mod-6 counters to disable the generic counter at 00:00

Overcoming Design Drawback: Pausing at 00:00



Conclusion

- Through the use of VHDL software and DDR4 hardware, we were able to design a Count Down Timer by using a generic counter, 2 BCD counters, 2 modulo-6 counters, and a serializer that had component of another genpause and hex 7 segment display.
- We port mapped all of them in our top level file along with using many different signals. Finally, we used constraint file to implement our design.

Thank You

- Time for Demonstration