VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

CO3091

LOGIC DESIGN PROJECT

Report

Digital Chess Clock Circuit

Advisor(s):	Pham Hoang Anh	
Student(s):	Dinh Le Minh Quan	2152262
	Pham Nguyen Quang Nhan	2152206
	Le Huy	2152584

HO CHI MINH CITY, JANUARY 2024

Contents

1 Abstract

1.1 What is a digital chess clock?

A digital chess clock is a device used in keeping track of each player's left time in a chess game. This includes a clock and a button for players to press when they have finished a move. When a player presses their button, their clock stops and their opponent's clock starts counting down. If the time runs out for one player, that player will lose the game.

Digital chess clocks are used by chess players of all levels, from beginners to grandmasters. They are also used in official chess tournaments and competitions.

Players can adjust the time control for their games using the numerous settings and modes that digital chess clocks offer. Common modes with predetermined time parameters include bullet (3 mins or less), blitz(5-30 mins), rapid (30-60 mins) , and classical (120 mins). Additionally, settings like delay and increment, which add extra time to a player's clock after a specific number of moves or a specific amount of delay time, may frequently be changed by players.

1.2 Clock's Principle

Digital chess clocks operate on the principle of counting down the allotted time for each player's moves. On a digital display, these clocks often have two independent timers — one for each participant. When a player makes a move, they press a button, which is frequently found on top of the clock, which delays their own timer and immediately begins the timer of their opponent. This step guarantees that only one player's clock is active at any given time, fostering a competitive race against time.

1.3 Why is it useful?

In general, digital chess clocks offer a precise, adaptable, and user-friendly method of controlling time during chess matches, boosting the strategic aspect of the game.

The clock ensures that both players have an equal amount of time to think for their moves, and reduce game times for players and watchers.

2 Introduction

2.1 Project scope and purpose

This is a project for **Logic Design Project** (Subject Code: **CO3091**).

The scope of this project is to logic design and implement a circuit which has identical functions of a digital chess clock in a bullet chess mode (**2 minutes per match plus 3 additional seconds per move**). The circuit must meet the following requirements:

- Be able to count down time for each player independently.
- Be able to display the remaining time for each player.
- Be able to add additional time per move.

2.2 Methodology

2.2.1 Research

The first step is to research chess clock circuits. The information to be collected includes:

- The basic components of a chess clock circuit (buttons, time counter, additional time adder, etc.).
- The operating principles of a chess clock circuit.
- Existing chess clock circuit designs.

2.2.2 Requirement Analysis

After the research is complete, the project requirements must be analyzed. The aspects to be analyzed include:

- The functions that a chess clock have.
- The technical requirements of a chess clock circuit.

2.2.3 Design & Implementation

The next step is to design, simulate and implement (optional) the chess clock circuit. This step include:

- A logic circuit design on Logism
- A circuit simulation on Proteus
- A demo circuit on bread board

3 Requirement Analysis

3.1 Non-functional Requirement

A chess clock is an essential tool for competitive chess matches, ensuring that both players have a fair amount of time to make their moves. It consists of two interconnected clocks that alternate between counting down each player's time. When a player completes their move, they press their clock button, stopping their timer while simultaneously starting their opponent's clock. This mechanism prevents players from stalling the game or taking an excessive amount of time to make their decisions.

A nicely fundamental digital chess clock circuit should perform as a typical chess clock. Therefore, we come up with several Non-functional requirement:

- Easy-to-read displays: Time should be clearly display and understand.
- Accurate timekeeping: Reliable time counting-down throughout the match.
- Portable and lightweight: Easy to bring and set up anywhere.
- Distinctive signals: Alarms or LED blinking for game end.

Those requirements leads to some main basic functions of our clock circuit:

- **Time keeping**: Accurately counting down and tracking the time remaining for each player throughout the game.
- **Move indication**: Indicating which player's turn it is by keeping their clock active and the other player's clock paused.
- **Time control**: Which is how much time a player has to make all the moves in a game. Most time controls for casual or online games are consisting of a base time and an optional increment of bonus time to be added per move. This is usually written as [base time] | [increment]. For instance, a five-minute blitz game with no bonus would be 5|0, and a two-minute game with a three-second bonus would be 2|3.

3.2 Functional Requirement

For this project, our team planned to design a chess clock for classical mode chess games so the functions above will be specified as:

- For **the operation of the clock**, there is a system of buttons to **play, reset the timer and complete players' turn**.
- For **time keeping function**, there will be **a separate digital clock for each player which shows the time left** of that player so there are two separated clocks for both players, **the clocks will count down the time each player have and indicate it through 7-Segment Anode LEDs**.
- For **move indication funtion**, When a player completes his/her turn and hits the complete button, another player's clock will active and count down while his/her clock pause.
- For **time control** function, as the project scope is for a bullet chess game, so time control will be 2|3, which means that **the clock initialy start at 2:00 and each player get a bonus 3 seconds whenever he/she completes a turn**.

3.3 Components & Modules Requirement

3.3.1 Buttons

First, we have a **Button Modudle** to operate the clock which includes:

- 1 button start/stop.
- 1 buttons for change turn (complete turn).
- 1 button for reset time of both players.

3.3.2 Digital LED Clock

Then, we will come to two separate **Digital Clock Components** which each for 1 player includeing:

- A 1Hz clock generator (circuit using crystal or IC NE555).
- 3 7-Segment LED to indicate the time (1 for minute and 2 for second) and the BCD to 7-Segment-Pin Converter associated with each LED.
- 1 MOD6 time counters and 2 MOD9 time counters for the time (processed from the DECADE UP/DOWN COUNTER 74LS192).

3.3.3 Player Turn Module

Moreover, there is an indispensable **Player Turn Module** to determine which player's clock is running at any time (while the other is pausing):

- A logic component using T flip flop (in this case we use JK flip flop with J pin and K pin connected in Proteus due to incapability to find T flip flop while simulating) and NPN transitors to accept or prevent 1Hz clock get to Clk pin in IC counter. Each transistor is trigger by "comp" signal whenever player press complete button.

3.3.4 End Game Indicator Module

Besides, there are two **End Indicator Module** when any **Digital Clock Components** counts to 0:00 which, in other words, means that one player's time is out and he/she lose the game immediately, this module includes:

- A logic design circuit to check if each LED value is all 0.
- A LED that indicates timeout (LED BLUE means Player 2 win, LED RED means Player 1 win).
- A Pin to trigger to lock the clock pulse generator.

3.3.5 Time Control Module

Last but not least, so as to conduct the time control function, add a bonus time (in this case is 3 seconds) after each move, we design a **MOD10 Adding Operator** which is:

- A logic component in Logisim that add 3 to the second digit(sec in the simulation) and update second digit and 10-second digit(min in the simulation, if necessary) with input a, b, c and output q1, q2, q3, carry
- Or similarly a module using IC 74LS283(4-bit full adder), 40175(Hex/Quad D flip flop), 4063(4-bit Magnitude Comparator), 74HC157(2-input multiplexer) in Proteus.

4 Circut Design And Simulation

4.1 Block diagram of the circuit

Below is the block diagram of the circuit:

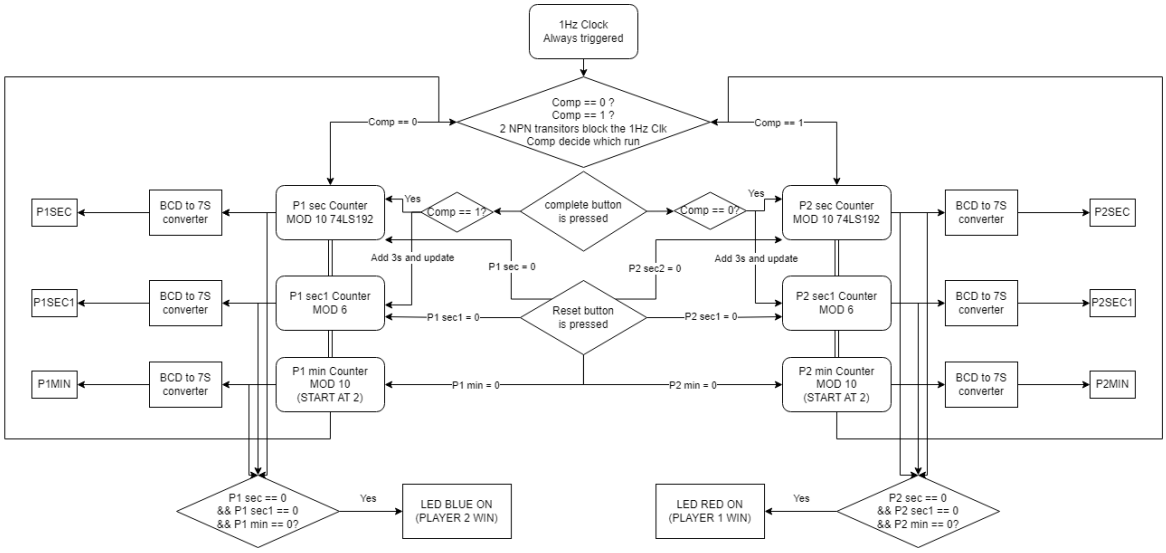


Figure 4.1: Chess clock's Block Diagram

Respectively from the Block diagram, below is the schematic design in Proteus and Logism:

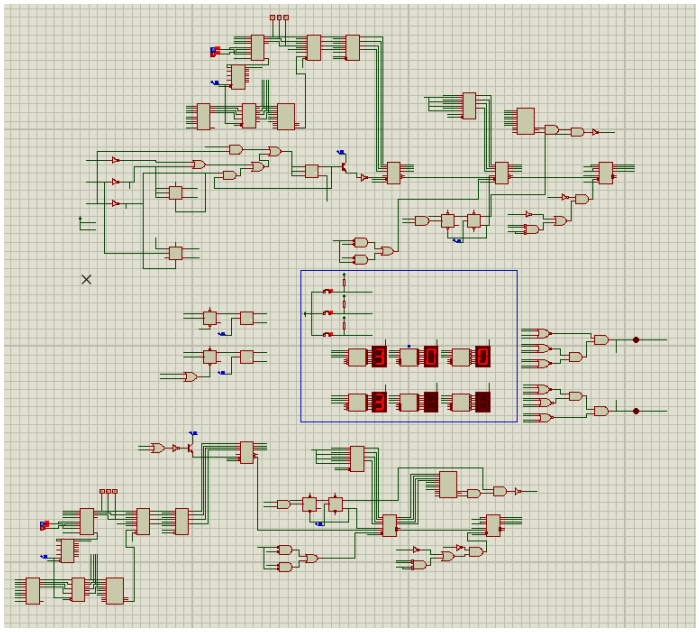


Figure 4.2: Proteus Design Overview

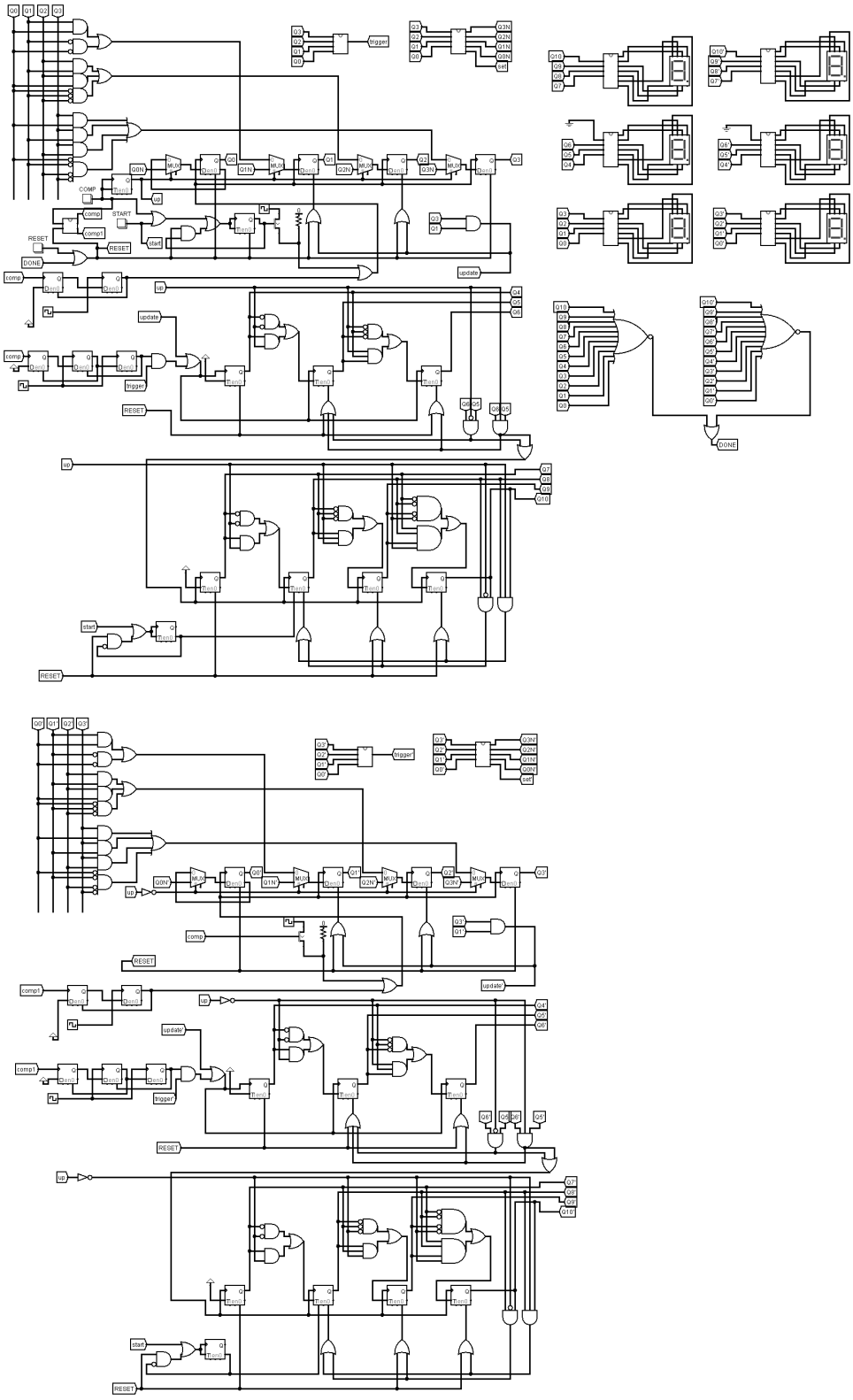


Figure 4.3: Logism Design Overview

4.2 Detailed description of each components design

4.2.1 Buttons

In this step we are using three button, which is processed by T flip flops to triggered Start, Complete turn and Reset respectively as in schematics below:

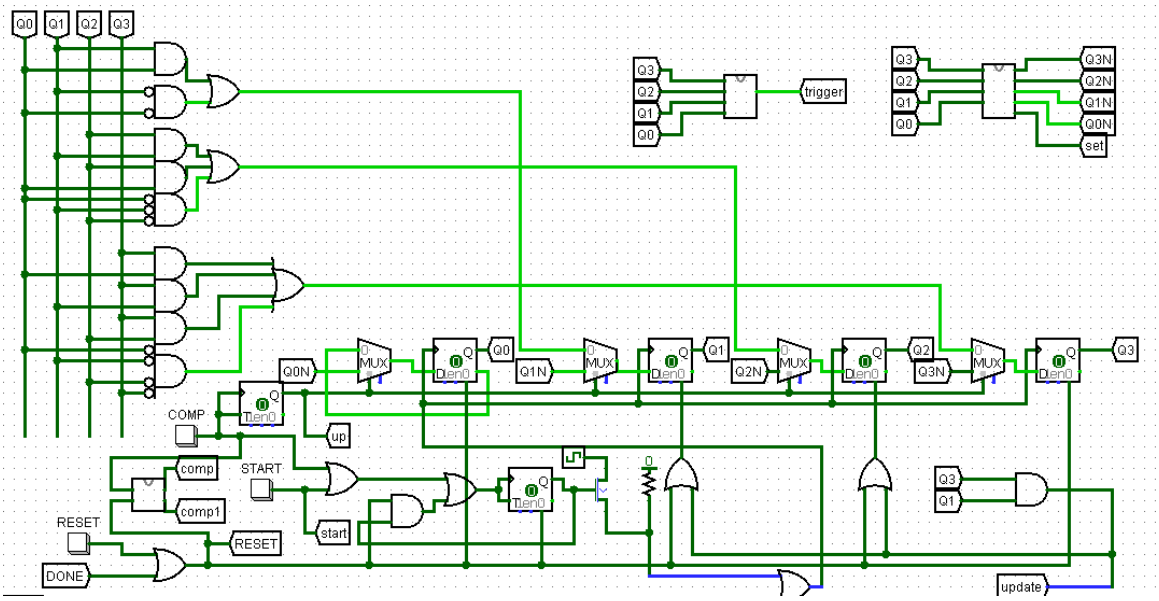


Figure 4.4: Logism's Buttons Module

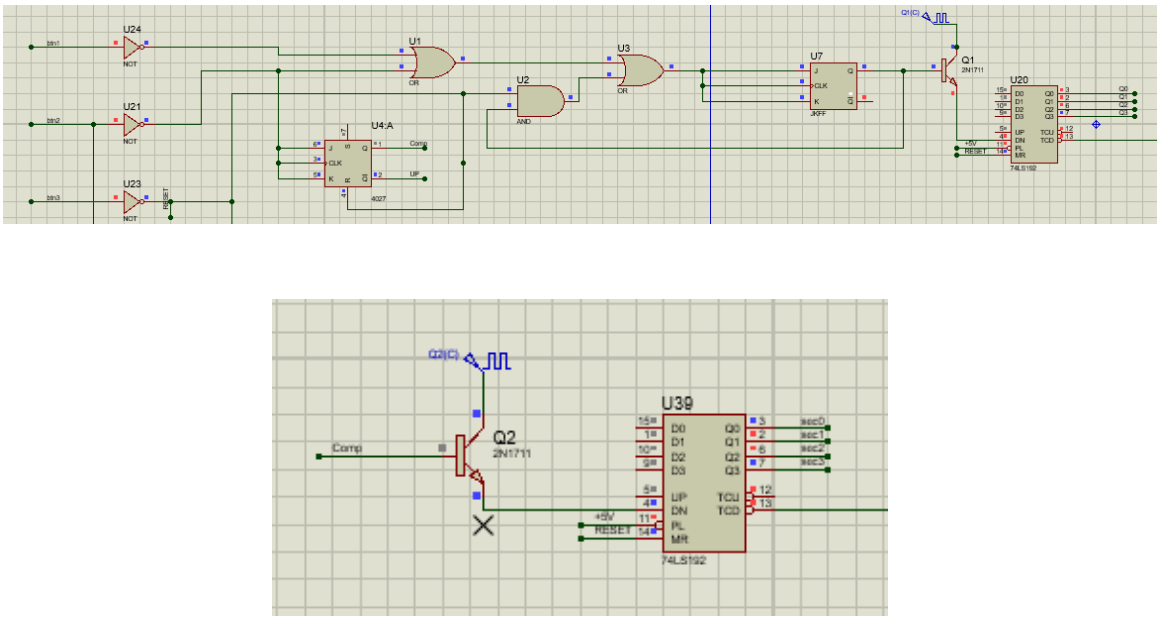


Figure 4.5: Proteus's Buttons Module

- **Start button**: When player pressed the start button, one of the Base source of two NPN transistor accept 1Hz clock pulse through and the clock start counting.
- **Reset button**: There is a signal named "comp" in the simulations which is to decide the current player turn. When (comp == 0), it's player 1's turn and when (comp == 1), it's player 2's turn. The principle of this operation is that the "comp" signal is connect to the Base source of two NPN transistors whose Collector is connect to 1Hz clock generator while the Emmiters are connected to Clk pin of IC 74LS192, therefore when "comp" signal is triggered HIGH to one in two Base it will enable clock pulse of IC 74LS192 to count down that player's clock.
- **Complete button**: When player pressed the reset button, both two Base sources is locked then the clear pins of Counters' flip flops (in Logisim, or the pin MR, Asynchronous Master Reset Input (Active HIGH) in Proteus) are triggered to reset all IC 74LS192 to 0.

4.2.2 Digital Real Time LED Clock

1Hz Clock circuit

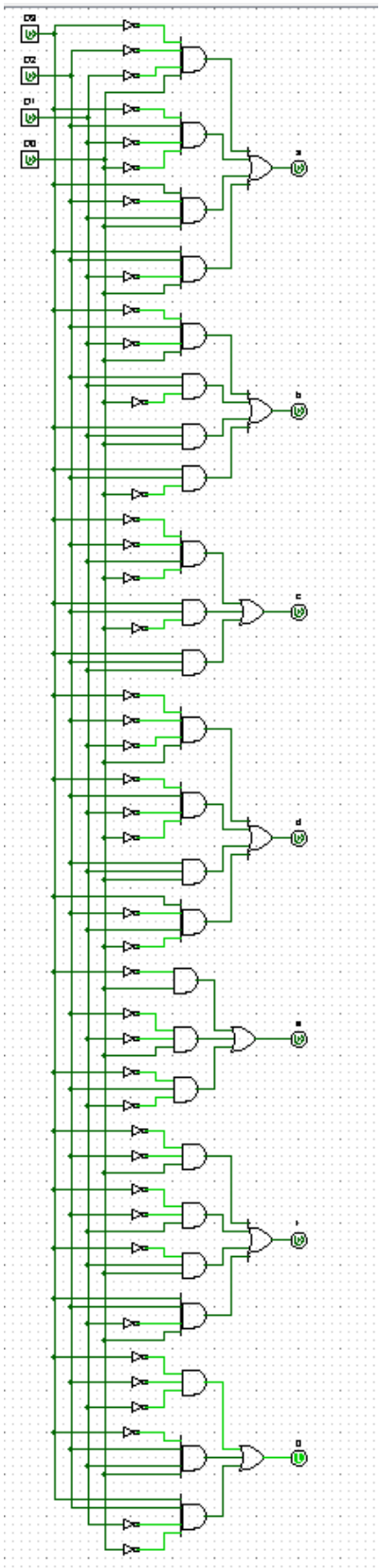
Team are still using clock generators Logisim and Proteus for stable testing. We are also implementing separated IC NE555 or crystal oscilator clock generator to add later.

BCD to 7-Segment-LED-pin converter

In this step, unlike in Proteus we can directly use IC 74247 (BCD to 7-Segment Decoder/Driver) to convert BCD signal from IC Counter 74LS192, in Logisim we need to design a converter circuit which have the truth table as below:

D3	D2	D1	D0	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0
0	0	1	1	0	0	0	0	1	1	0
0	1	0	0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0	1	0	0
0	1	1	0	0	1	0	0	0	0	0
0	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	1	0	0
1	0	1	0	0	0	0	1	0	0	0
1	0	1	1	1	1	0	0	0	0	0
1	1	0	0	0	1	1	0	0	0	1
1	1	0	1	1	0	0	0	0	1	0
1	1	1	0	0	1	1	0	0	0	0
1	1	1	1	0	1	1	1	0	0	0

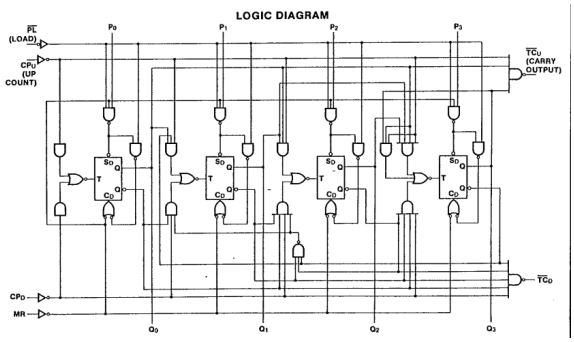
From the truth table, we can build the converter's logic circuit as below:



Building MOD6 Counter from DECADE Counter 74LS192

As can be seen from the block diagram, we need MOD6 Counters and MOD10 Counters to indicate sec1, and sec, min digit respectively.

The 74LS192 is a BCD decade counter IC that can be used to count from 0 to 9. It has four outputs (QA, QB, QC, and QD) that represent the count in binary coded decimal (BCD). Provided that the IC 74LS192 is already a MOD10 UP/DOWN counter itself, now we only need to implement the MOD6. First, we need to take a look at IC 74LS192's datasheet:



To build a MOD6 counter from a MOD10 74LS192, we have a truth table as below:

$f(ABCD) = \bar{A}B + \bar{A}C$

Truth Table

A	B	C	D	f(ABCD)
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0

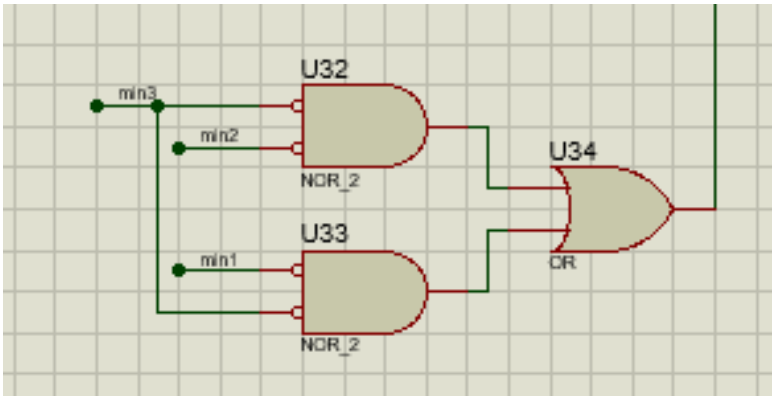
It is clearly stated that so as to disable value 6-9 to get 0-5 (MOD6) from a DECADE Counter, we only have to make value range 6-9 be "illegal" by a logic expression **f(ABCD) = !A!B + !A!C**. Luckily in this case the 74LS192 is able to do this by using the pin PL (Parallel Load) which is triggered like below:

MODE SELECT TABLE

MR	$\overline{PL}$	CP_U	CP_D	MODE
H	X	X	X	Reset (Asyn.)
L	L	X	X	Preset (Asyn.)
L	H	H	H	No Change
L	H	⌐	H	Count Up
L	H	H	⌐	Count Down

L = LOW Voltage Level
H = HIGH Voltage Level
X = Don't Care
⌐ = LOW-to-HIGH Clock Transition

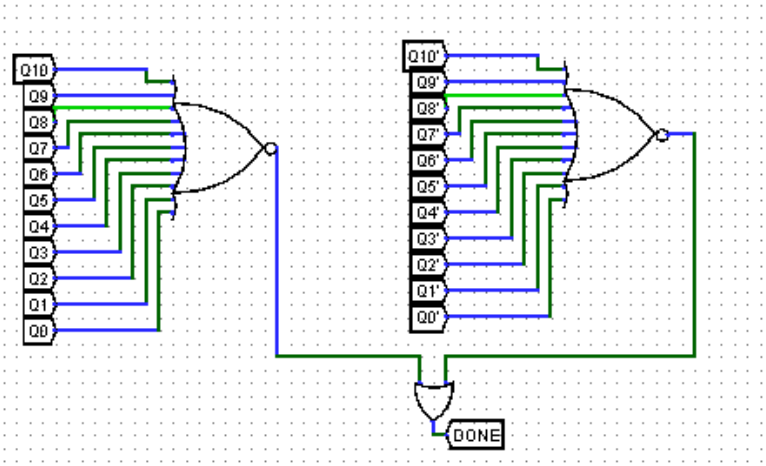
PL pin will reset the counter when it's triggered LOW, so as to make the MOD6 counter, we only need to add the **f(ABCD) = !A!B + !A!C** to PL Pin as we have done in Proteus:



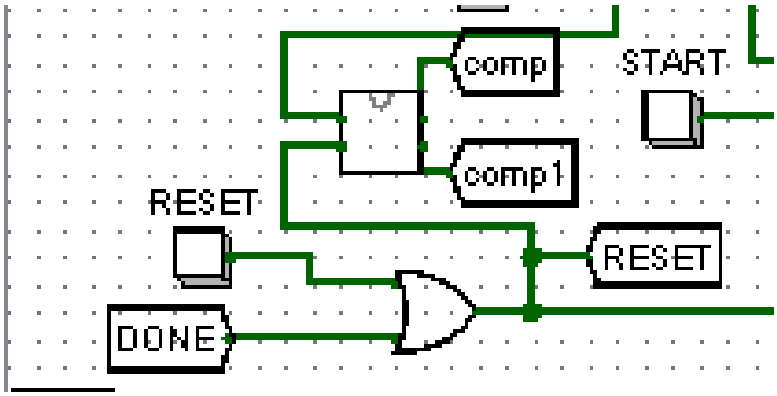
4.2.3 End Game Indicator Module

For the design of **End Indicator Module**, we designed logic gates on checking whether the bit digit is 0:00 first as below:

In Logism:

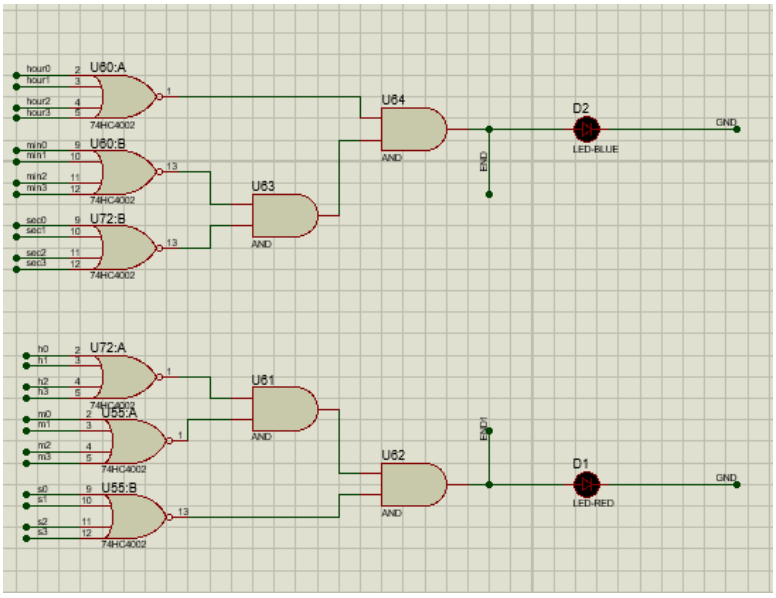


If one of two LED clock trigger 0:00, the OR Gate will trigger 1 for DONE pin, this Pin is treated as a RESET button triggered through this OR Gate below:

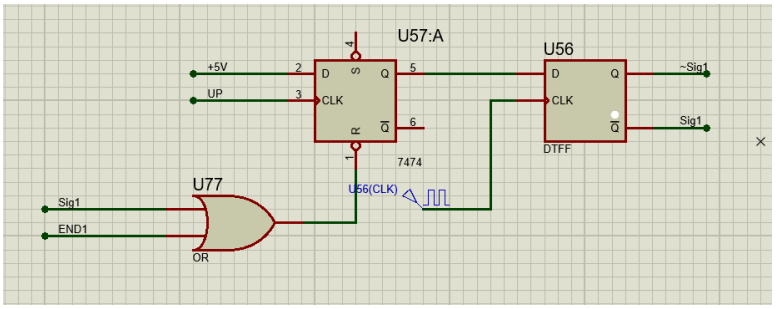


Afterwards the whole system will operate as a Reset method.

In Proteus:



In Proteus, if Player 1's LED clock is 0:00, the LED_BLUE will be toggled ON so as to show that the Player 2 is the winner because Player 1 is time out. On the contrary, if Player 2's LED clock is 0:00, the LED_RED will be toggled ON so as to show that the Player 1 is the winner because Player 2 is time out. Meanwhile, There are two Pin END and END1 which will trigger with the Pin Sig and Sig1 to stop Clock Pulse on both Clock.



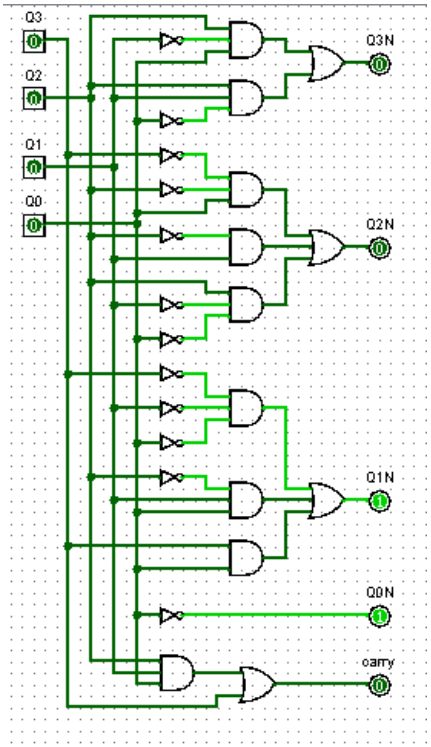
4.2.4 Time Control Module

Like we have mentioned in the previous sections, this Module is to conduct the time control function, add a bonus time (3s) after each move, we design a MOD10 Adding Operator and update the result.

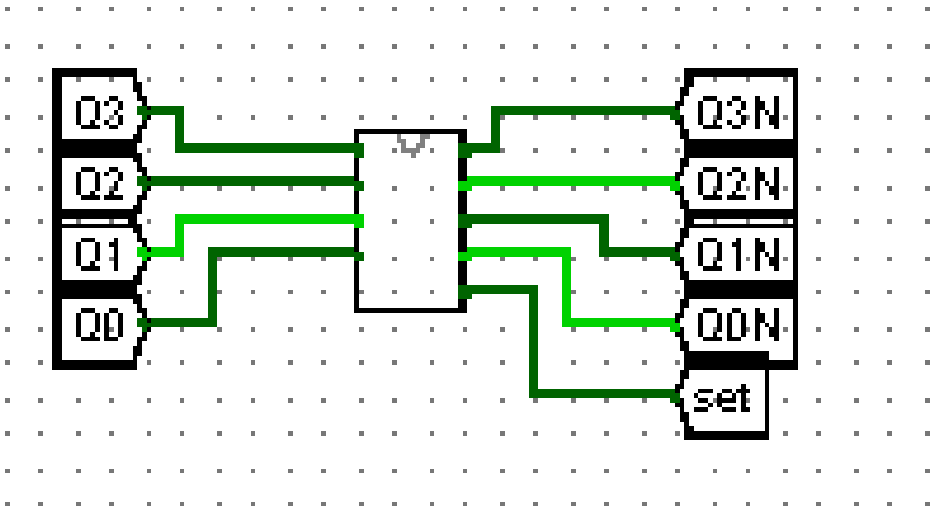
First, in as a base in Logisim, we design a logic operator to add 3 to the 4-bit input Q0, Q1, Q2, Q3 and return 4-bit result Q0N, Q1N, Q2N, Q3N within a carry or not. A truth table of this expression is showed as:

Q3	Q2	Q1	Q0	Q3N	Q2N	Q1N	Q0N	carry
0	0	0	0	0	0	1	1	0
0	0	0	1	0	1	0	0	0
0	0	1	0	0	1	0	1	0
0	0	1	1	0	1	1	0	0
0	1	0	0	0	1	1	1	0
0	1	0	1	1	0	0	0	0
0	1	1	0	1	0	0	1	0
0	1	1	1	0	0	0	0	1
1	0	0	0	0	0	0	1	1
1	0	0	1	0	0	1	0	1
1	0	1	0	0	1	0	1	1
1	0	1	1	0	1	1	0	1
1	1	0	0	0	1	0	1	1
1	1	0	1	1	0	1	0	1
1	1	1	0	1	0	0	1	1
1	1	1	1	0	0	1	0	1

From the truth table, we can design:

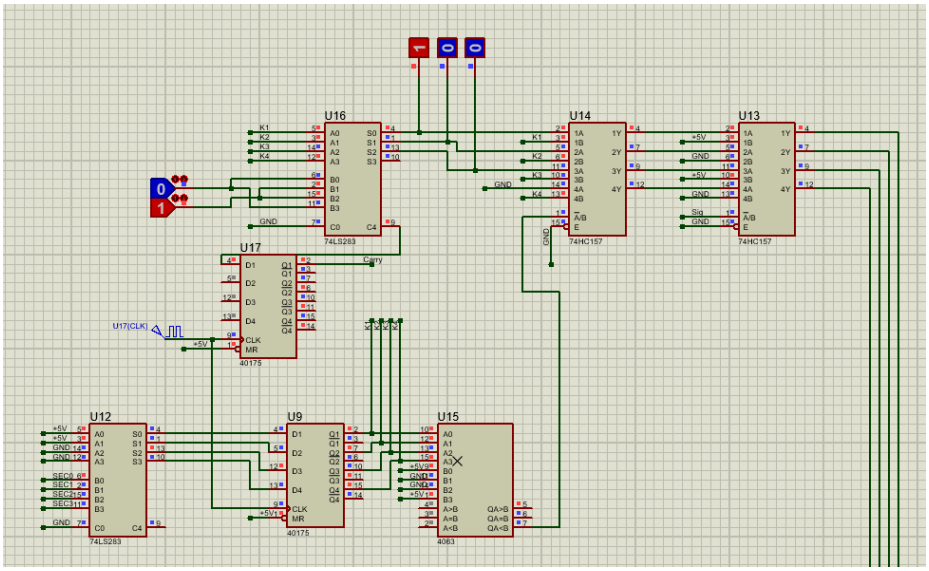


After finished design, we apply it into the circuit as below:



For example in this case, the operator take input Q0 Q1 Q2 Q3 which indicate the sec digit, the input is Q0=0, Q1=0, Q2=1, Q3=0 which is 2 in deximal so the output would be 2 + 3 = 5 so the output would be 5 without carry triggered which is Q0N = 1, Q1N = 0, Q2N = 1, Q3N = 0 and carry(set) = 0 like above.

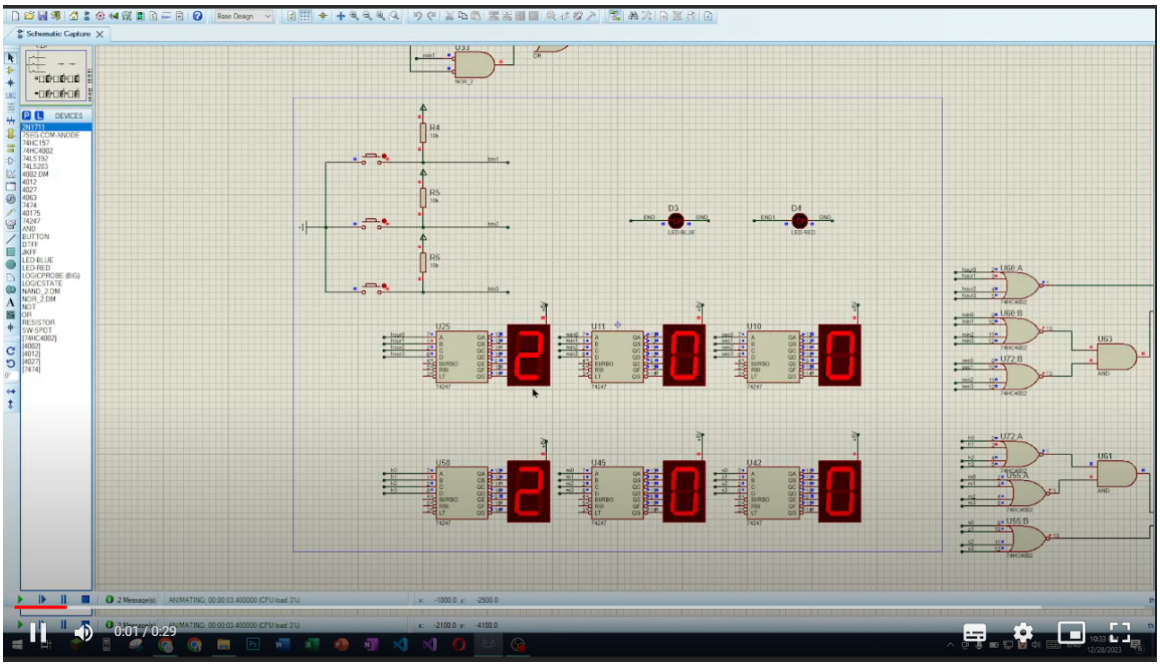
Secondly, we come to implement that function in Proteus, because Proteus is not allowed user to create a new IC or manipulate the components and flip flop inside an IC, we need to conduct the adding process by using multiple ICs (IC 74LS283(4-bit full adder), 40175(Hex/Quad D flip flop), 4063(4-bit Magnitude Comparator), 74HC157(2-input multiplexer)):



In this example case, first, the input B0,B1,B2,B3 for the bottom left IC 74LS283 are Q0=1,Q1=0,Q2=0,Q3=0 which indicate sec digit output, and the input A0,A1,A2,A3 is digit 3 (which is 0011 in binary). Then the output is S3=0,S2=1,S1=0,S0=0 which is digit 4 connected to the D flip flop 40175 to have K1, K2, K3, K4 (this flip flop is there to delay one cycle). After that, the result (4, 0100) is passed into IC 4063 to compare with 9(1001). If it's lower than 9, a HIGH signal is triggered to !A/B Pin of IC 74HC157(Multiplexer) to choose K1, K2, K3, K4 to be the result for sec1 digit. Meanwhile, the K1, K2, K3, K4 is passed to be the input of the top left IC 74LS283 which is used to subtract 10 digit from input, Next, the output is connect to the same 74HC157 above and be chosen as the result for sec digit.

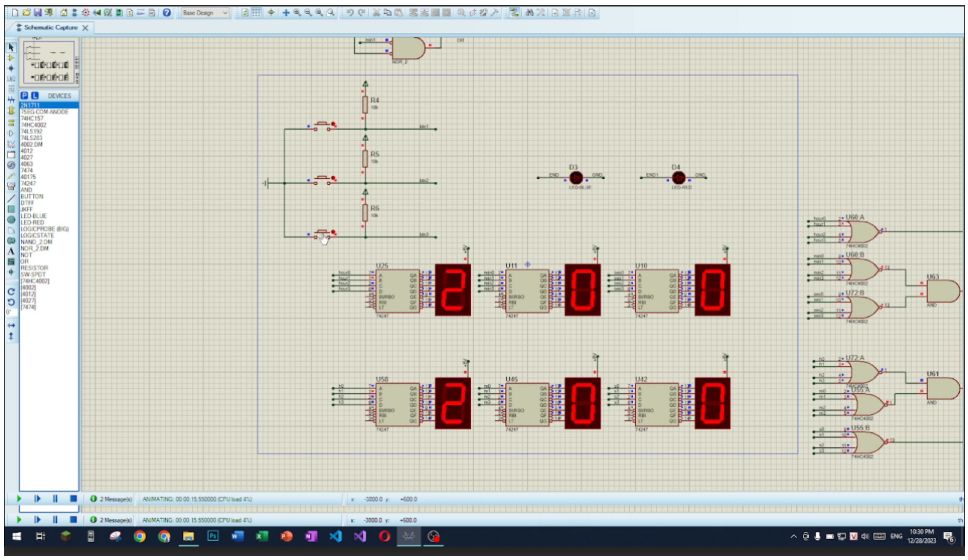
4.3 Simulation

Below is the simulation of Proteus schematic to run normally within pressing button 2 (Complete) to change Player's turn:



This is the link: Demo.

Then we will go through the end game indication module's testing:



This is the link: Demo.

5 Extension

5.1 Classical Mode Chess Clock

Similarly to the Bullet Mode Digital Chess Clock, team also managed to design and simulate a Classical Mode Digital Chess Clock with time control 120|30 as below:

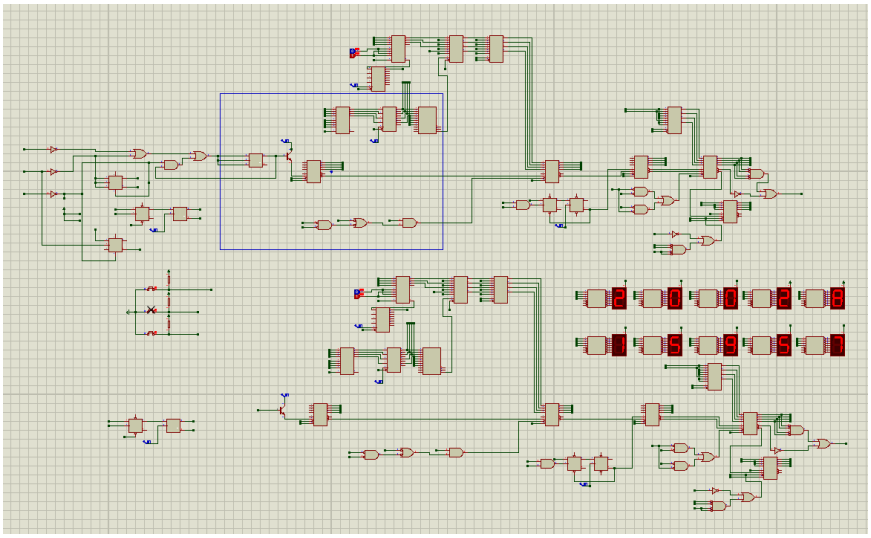


Figure 5.1: Simulation

6 Conclusion

This project successfully designed and implemented a digital chess clock circuit for bullet chess mode, fulfilling the requirements of the **Logic Design Project (CO3091)** subject. The circuit effectively replicates the expected functionalities of a digital chess clock, including independent timekeeping for each player, real-time remaining time display, and adding bonus time per move.

Key achievements:

- **Accurate timing:** The circuit accurately counts down time for both players based on the bullet chess rules (2 minutes per match + 3 seconds per move).
- **Clear display:** The remaining time for each player is clearly displayed on the **digital LED clock**, ensuring transparency and fair play during the game.
- **Precise control:** The additional **time adder module** precisely adds 3 seconds to the designated player's clock after each move, as specified in the bullet chess format.
- **Modular design:** The circuit is divided into distinct modules for each functionality (buttons, clock, player turn, game end, and time control), facilitating easier troubleshooting and potential future modifications.

Overall, this project above demonstrates a successful application of logic design principles to create a functional and practical digital chess clock circuit.

References

[1] Mark E Appel. The chess clock: A time-management technique for complex cases. *Dispute Resolution Journal; New York*, 2006.

[2] Mark S. Inocentes, Rowlen V. Notarte, and Jeanne Lee C. Puentespina. Automatic digital chess timer. 2012.

[3] Glen G. Jr. Langdon. *Logic Design: A Review Of Theory And Practice (ACM monograph series)*. Elsevier Science, 1974.

[4] Ronald J. Tocci Neal S. Widmer Gregory L. Moss. *Digital Systems: Principles and Applications, 12th Edition*. Pearson Education Limited, 2016.

[5] Charles K. Alexander Matthew N.O. Sadiku. *Fundamentals of Electric Circuits*. McGraw-Hill Companies, 2013.