G.S.M. Galadanci Int. Journal of Engineering Research and Applications www.ijera.com
ISSN : 2248-9622, Vol. 4, Issue 8(Version 1), August 2014, pp.34-56

RESEARCH ARTICLE OPEN ACCESS

Design, Implementation and Simulation of 12/24 Hours Digital Clock With Stop Watch and Date Indicator

G.S.M. Galadanci and S.M. Gana
Physics Department, Bayero University, Kano P.M.B. 3011, Kano Nigeria.

ABSTRACT
In this paper thedesign, implementation and simulation of a digital clock capable of displaying seconds, minutes and 12 or 24 hours timing, with a date indicator that display days ,months and years including a stop watch is presented. The architectural design was carried out using synchronous decade counters and logic gates. The basic clock frequency signal (in hertz) that drives the clock was generated using a clock voltage source of the simulator for frequency division of the desired clock pulse. The digital clock circuit was implemented and simulated using national instrument electronic work bench (multism) software version 13.0 on personal computer (PC). The result of the simulation showed that the designed clock and stop watch gives an approximate timing count, comparable to different existing digital clocks/stop watch with percentage error of 0.0033%.

Keywords: synchronous decade counter, logic gates, logic level, clock pulse,

I. INTRODUCTION

There are several existing clocks in this world; analogue and digital, which one may wonder what other uses the clock may have. A digital clock is a type of clock that displays the time digitally. Instead of the rotary mechanism of electromechanical clock, it uses digital counters that count second, minute and hours. Each sixty seconds make a minute and each sixty minutes an hour. After twenty four hours the clock resets and starts from initial condition. The functional unit of a digital clock is a counter that represents a second, minute or hour block. A counter [1] may be defined as a register i.e. a group of flip-flops that goes through a predetermined sequence of states upon the application of input pulses. The logic gates in a counter are connected in such a way as to produce a prescribed sequence of binary states in the register. Originally Timers were designed to fulfill a need in industry for a means of keeping time on certain devices. These timers were mechanical devices and used clockwork mechanisms as a means of keeping a regular time. With so many tasks that must be done on scheduled manner, keeping track of time and date has become so imperative that it is now a necessity [1, 2, 16]. It is a set of principles, practices, skills, tools and system that helps us use our time to accomplish what we want. Digital timers were finally invented to counter the complexity of mechanical timers and are light weight for easy excess. This research provides basic information and the fundamental use of digital circuitry and how to develop, design and implement a digital clock using synchronous decade counter and logic gate circuits [1, 2, 3]. These tools are used to design the different parts of the clock and then implemented to create a customized clock. The clock will provide and keep track of seconds, minutes, hours, days and months information digitally and will be operated using a basic clock frequency signal in hertz generated from the clock voltage source of the simulator, and fed into the clock counters synchronously. The second's and the minute's circuits are designed by cascaded arrangement of a divide-by-60 counter that will count from 00 to 59 and then recycle back to 00 for 60 seconds or 60 minutes count when a clock pulse is applied to their clock inputs [1]. The output is fed into the input of the minute's circuit so that for every 60 seconds the minutes counter will advance through its states from 00 to 59. Similarly, the Hours circuit section is formed by divide–by-10 and a truncated sequence divide-by-10 counter connected through the output of the minute's section such that for every 60 minutes the counter would advance through its states from 01-12 hours or 01-24 hours timing depending on the users choice of selection from a switch to affect the choice. For the day's circuit section two divide-by-10 counters with truncated sequence are used to keep track of the days in every month [4]. The input of the day's circuit section is connected to the output of the hour's circuit section through a single decade counter that will control the timing between 12 and 24 hours. If the user choice is 24 hours timing, the days circuit section is expected to advance through all its states for every 24 hours while if the choice is 12 hours the single decade count is expected to allow the days count to advance through its state if and only if it cycles twice to complete a 24 hours count [2]. The month circuit section consist of a divide-by-10 and a truncate sequence divide-by-10 counter display 01 to 12 to

indicate January to December whenever days circuit complete its cycle of days; at the same time control the number of days required for each month. This is accomplished by coupling the month circuit section back to the days circuit section such that whenever the clock is on April, June, September and November the days count should be 30, 28 for February alone and 31 day for the rest of the months. The year circuit section is design with four divide-by-10 synchronous decade counters that advance through there state by a HIGH logic level produce on the clock pulse that recycle the days circuit block from 31st days in the month of December (12) to 1st day of January (01). However the Gregorian calendar has only 365 days in a normal year, and 366 days in a leap year, hence a system is also design in the years circuit block to add a day to February (that is 29 days) every 4 years and every 400 years. The stop watch is made possible by connecting the second and minute's section of the digital clock separately from the main clock with another separate counter that can count up to 9 hours [5]. A switch is utilized to inhibit the enable input through all the counters of the stop watch when it is off, this allowed the desired timing be displayed throughout the counters since the clock pulse is still retain in them. The counter would be CLEAR when a push button is depressed to be ready for the next timing sequence of the stop watch. These different sections are connected together to form the required design of the digital clock and each information provided by them would be displayed using seven segment displays [1,2,3].

II. METHODOLOGY

To design a digital clock that has an approximate time precision with any workable clock, it is necessary to have knowledge on the basic ideas used for designing any digital system from a set of fully specified states tables or an equivalent representation of state diagram for sequential circuits or well defined truth table for combinational elements[1,2,3,4,5]. For this design fixed function integrated circuits are used that is a synchronous decade counters and some combinational elements are utilized for proper operation. The number of counters required is determined from the number of states needed in the circuit. The design process or procedures always involves transformation of the related problems faced in sequential-circuit design into a well-articulated combinational-circuit problem for better picture of the problems involved [2]. The steps involved for these design are outlined below;

1. The word description of the circuit behavior is stated, which is accompanied by obtaining a state diagram, a timing diagram, or other related information.
2. State table is obtained from the given information about the circuit.
3. The number of states may be reduced by any simple state-reduction method if the sequential circuit can be characterized with input-output relationships independent of the number of states.
4. Binary values are assigned letter symbols for each state from the state table obtained in step 2 or 3.
5. The number of counters needed is determined and letter symbol is assigned to each.
6. The type of combinational element required is identified.
7. From the state table, the circuit excitation, inputs and outputs characteristics are analyzed.
8. Simplification method is used to derive the circuit inputs and outputs functions where necessary.
9. Circuit diagram is obtained and implemented.

2.1 DESIGN PROCEDURES

For easier design, the general circuit of the digital clock is divided into eight sub sections namely;

1. Second counter circuit section.
2. Minute counter circuit section.
3. Hours counter circuit section.
4. Days counter Circuit section.
5. Months counter circuit section.
6. Years counter circuit section.
7. Initialization control Unit.
8. Stop watch circuit section.

The flow chart in Figure 1 gives step by step descriptions of the process involve in designing the digital clock from the second minute and hours circuit section. Figure 2 gives the continuation of the flow chart containing the days, month and years circuit section.

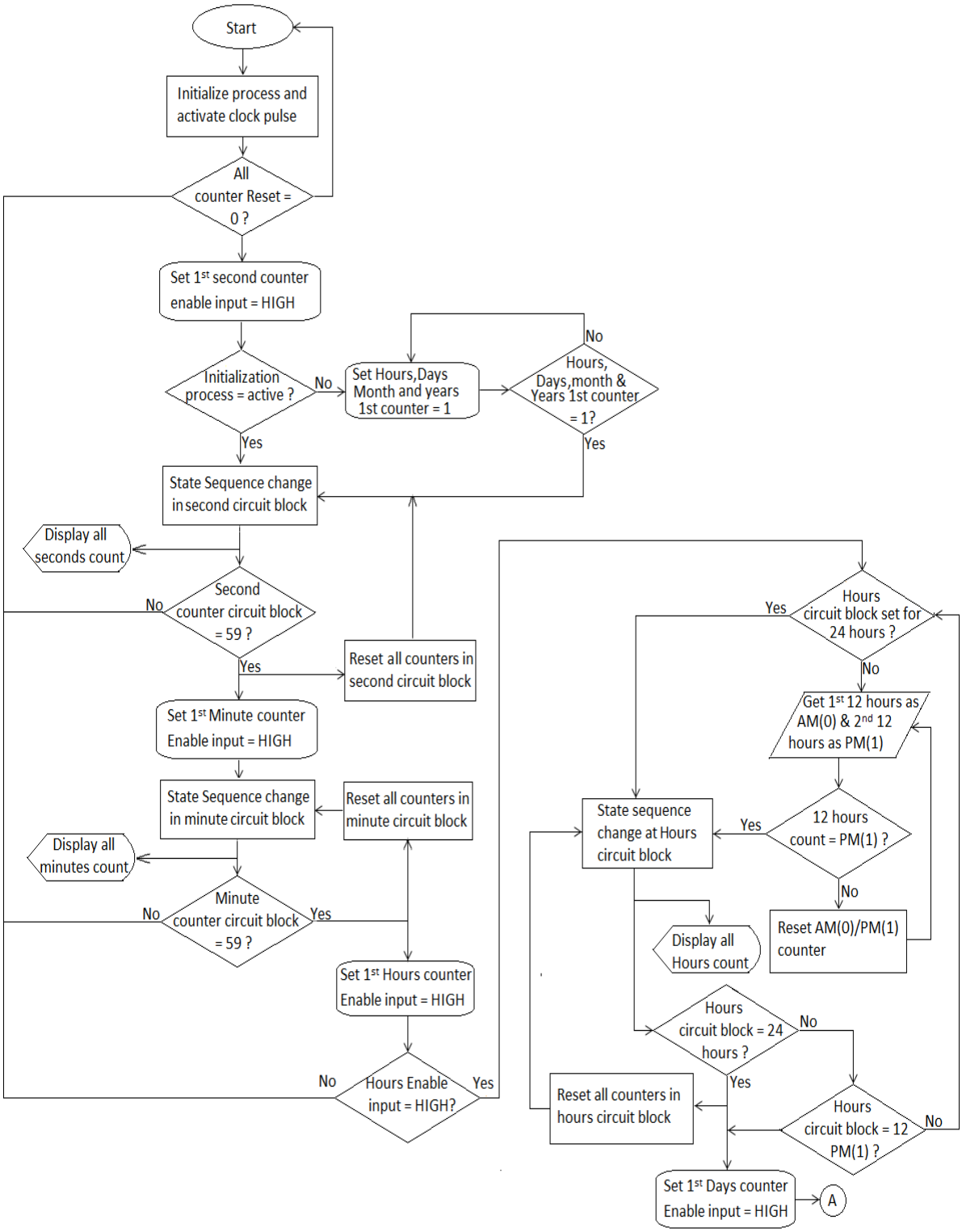

Figure 1: Flow chart of second, minutes and hours circuit section

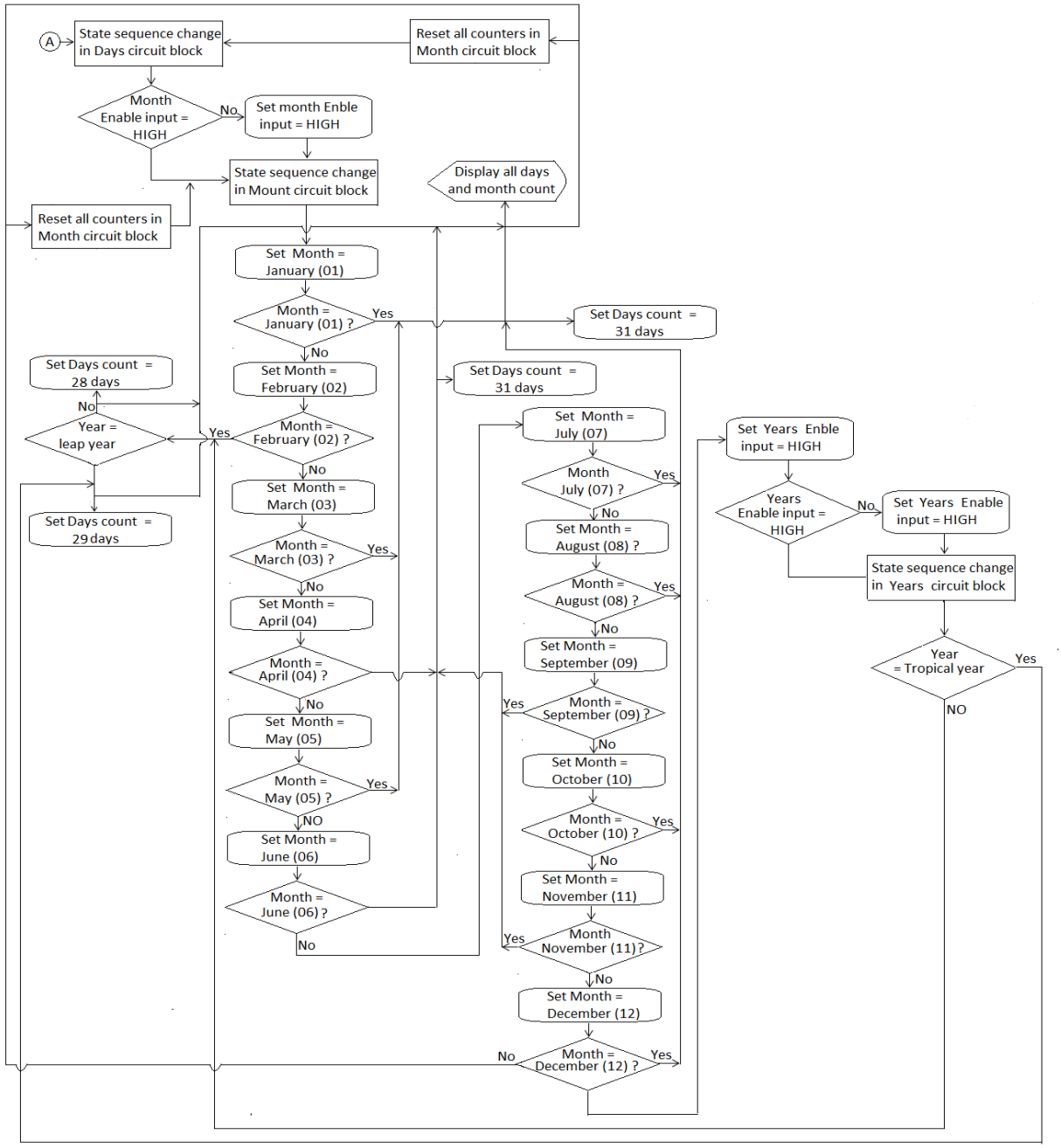

Figure 2: Flow chart of days, month and years circuit section.

These sections together form the digital clock, taking into consideration that the clock pulse that drives the design clock is used from the simulators clock voltage generator at a specific frequency. They can be implemented, tested and analyze quite easily if treated individually. Each of the sub-circuit is implemented by designing logic circuits that perform its duties. National instrument multisim version 13.0 is used to design the logic circuits in this research work. In general Table 1 below gives the name of component and series used for this design and the numbers marked on each of the so called component [8, 9].

Table 1: Components used and their series

Name	Series	Number marked
2- inputs AND gate	74HC108D	A
3- inputs AND gate	74HC11D	B
4-bit synchronous binary counter	74HC160D	C
2- inputs NAND gate	74HC01D	D
Quad 2- inputs exclusive OR gate	74HC86D	E
Quad 2- inputs exclusive NOR gate	74HC266D	F

2- inputs OR gate	74HC32D	G
Quad 2- inputs NOR gate	74HC02D	H
Hex inverter	74HC04D	J
Dual 4- inputs NAND	74HC20D	L

2.2 SECOND COUNTER CIRCUIT SECTION

The second counter circuit is design and implemented by cascaded arrangements of two 4-bit synchronous decade counters as shown in Figure 3.

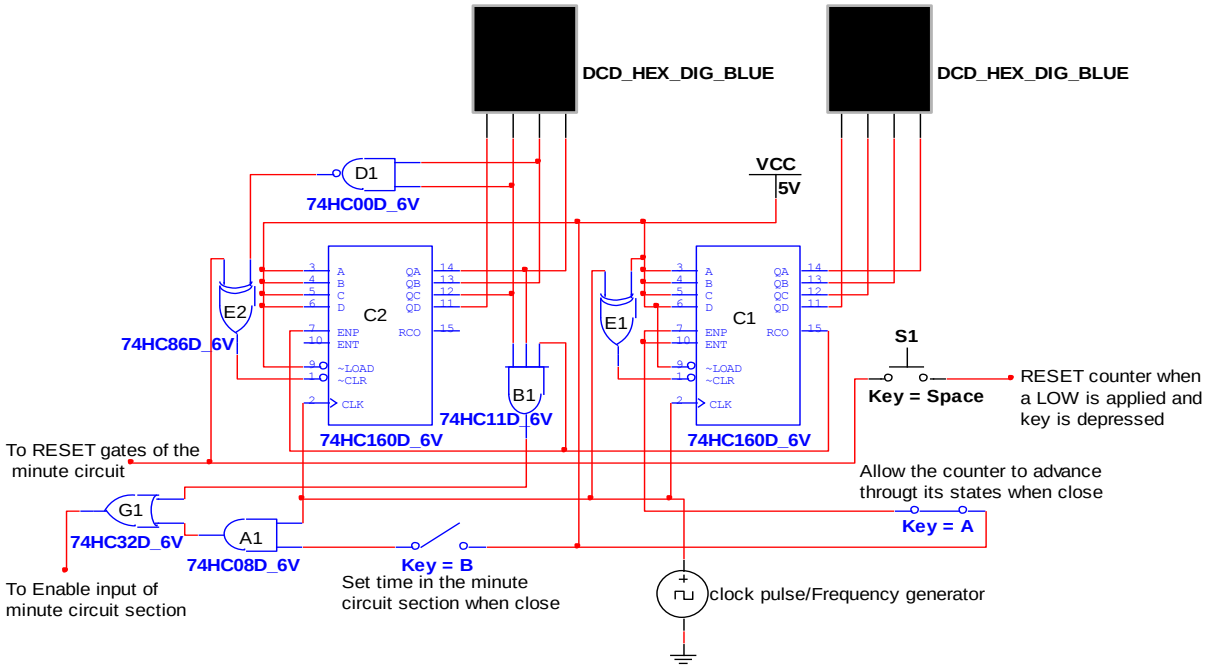

Figure 3: Second Counter Circuit.

The 1st counter (C1) is connected as a divide-by-10 counter that counts from 0 to 9 and then recycles back to 0, the 2nd counter (C2) is connected as a divide-by-6 portion of the counter that counts from 0 to 5 and then recycles back to 0. The ripple clock output (RCO) of the counter C1 is connected to the enable inputs (ENP and ENT) of C2 such that when C1 advances through all its states from 0 to 9, on the clock pulse that recycles it from 9 back to 0 also called the Terminal Count (TC), it output goes HIGH and hence activates the enable inputs (ENP and ENT) of C2 to illuminates a 1 on its display [1, 2, 3]. The total count is now 10 (the 1st counter is in the zero state and the 2nd counter is in the first state). The 2nd counter (C2) remains in the first state while the 1st counter (C1) keeps advancing through its states; 1 through to 9 for count 11, 12, up to 19 and on its Terminal count (TC) the output goes HIGH again and activates the C2 to illuminates a 2 on its display for a total count of 20 (the 1st counter is now in the zero state and the 2nd counter is in the second state). This process continues, for every Terminal count of the counter C1 (0 through 9) C_2 advance to the next state in its sequence until the total count is 59 before all the two counters recycles to 00 for count 60 [5, 6, 10]. In general, the 2nd Counter (C2) is inhibited by the LOW on its Enable inputs until the 1st counter (C1) reaches its last or terminal state before its terminal count output goes HIGH to activate it. In other words, C1 has to go through ten completes cycles from 0 to 9 before C2 advance through one cycle. Together these counters count from 00 for 00 second, 01 second up to 59 second and then recycle to 00 for 60 second count [4, 7]. The terminal count, 59 is also decoded through the gate B1to enable/allow the next Counter in the chain (minute circuit section) to sequence through it states by providing a HIGH logic level on its ENP and ENT inputs through gate G1. Similarly, switch B also allow the next Counter in the chain (minute circuit section) to advance through its state even before the terminal count (TC) of counter C1 and C2 is attain by providing a HIGH logic level to gate A1when close. This HIGH logic level enables the ENP and ENT inputs of the next Counter in the chain through gate G1to set the time in the minute's circuit section as desired [11]. However, it should be noted that the divide-by-6 portion of the 2nd counter (C2) is formed by decoding count 6 through the gate E1 such that when the counter is in the sixth state the output of the gate E1 becomes LOW, which in essence makes gate E2 LOW since one of its input connected to the RESET button is LOW (when the button is not depressed) [1, 2, 3]. Similarly, the Clear input (CLR)

of the 2nd counter (C2) has an Active-LOW input, hence, the LOW logic level on the output of gate E2 asynchronously clear the counters through the Clear input (CLR). Moreover, by pressing the RESET button gate E1and E2 can be used to asynchronously clear the counters C1and C2simultaneously. Furthermore, switch A allow the counter C1to advance through it state by constantly providing a HIGH logic level on its ENP and ENT inputs when close and inhibit the counter when open.

2.3 MINUTES CIRCUIT SECTION

The minute circuit section is similar to the second circuit section in its operation and the process of displaying its counts as shown in Figure 4. This operational process is resulted by utilizing the counters C1 and C2,andlogic gates as appropriate. However additional circuitries of logic gates are required to enable the divide-by-6 portion of the counter (C2) to have a perfect counting sequence and to enable/allow the next Counter in the chain (Hours circuit section) to sequence through it states[5, 12].

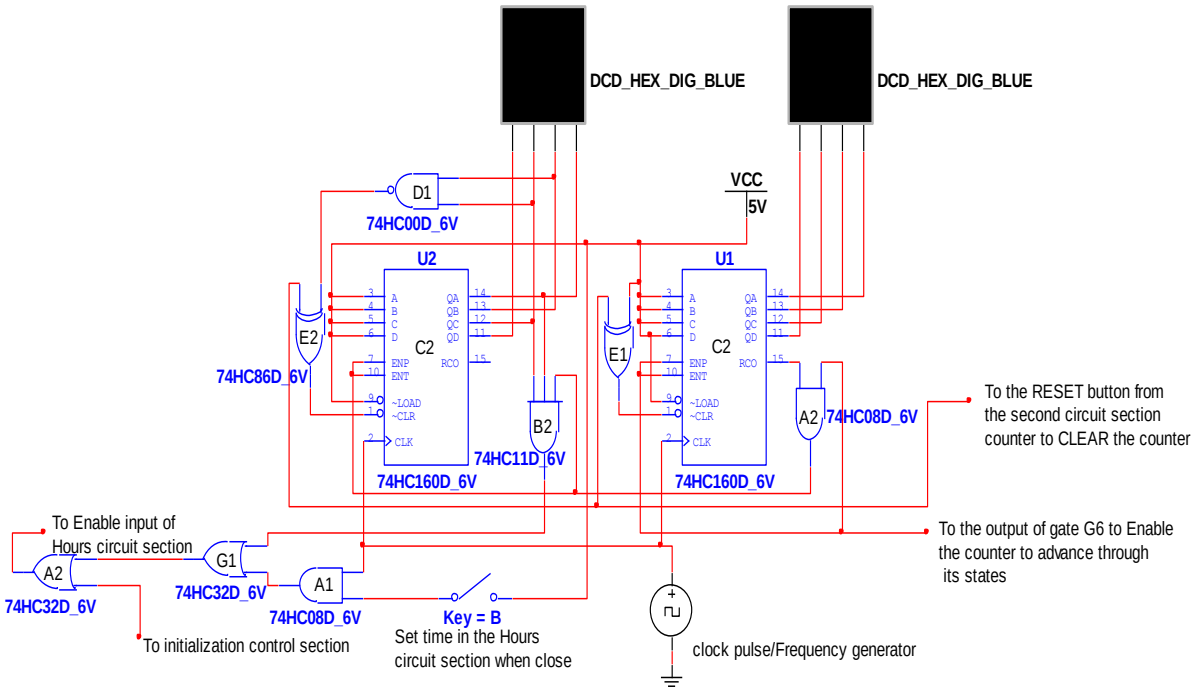

Figure 4: Minutes Circuit Section

2.4 HOURS COUNTER CIRCUIT SECTION

The hours counter circuit architectural design is carried out and produce by cascaded arrangements of two 4-bit synchronous decade counters with combination of some basic logics gates as required. The 1st and 2nd counter (C1and C2) are connected as a divide-by-10 counter that counts from 0 to 9, however the 2nd counter is design with additional features to exhibit different characteristic for specific timing. For easier implementation the design is sub divided into different unit with each unit performing its duty. These sub sections include:

1. Switch control unit
2. Timing control unit
3. Timing detection unit

These units connected together give the complete hours counter circuit.

2.4.1 Switch Control Unit

The switch control unit is design by taking into consideration that whenever the switch A is close for 24 hours timing the time counts exceed from 12 to 13, 14, 15, up to 24. Thus, if the switch is open for 12 hours timing while the count already exceeded the required count there is need for the clock to switch back to 12 hours timing by simply resetting both counters for fresh count. A truth table is written down for these counts and a standard Boolean expression is obtained by presenting all the possible value of the resulting outputs variables using karnaugh mapping. Table 2 gives the possible values in decimal and the corresponding outputs in their equivalent 4-bit binary coded decimal (BCD) group [6,7,12]. The timing from 13 to 19 is expressed separately likewise 20 to 24 so as to have each count with similar digit displayed from the 2nd counter C2 in one group for easier implementation and a 4-variable karnaugh mapping of each group is obtained individually. Each of the 4-bit groups are analyze keeping in mind that

the first 4-bit group is displayed from the 1st counter C1 and the second 4-bit is displayed from the 2nd counter C2 as shown in Table 2.

Table 2: 4-bit group displayed from the 1st counter and 2nd counters

Hours displayed from the 2nd counter C2 with 1 as the first digit	Hour displayed in decimal from both counters	Hours In 4-Bit BCD code displayed from the 2nd counter C2 through its Q output				Hours In 4-Bit BCD Groups displayed from the 1st counter C1 through its Q output			
		D	C	B	A	D	C	B	A
	13	0	0	0	1	0	0	1	1
	14	0	0	0	1	0	1	0	0
	15	0	0	0	1	0	1	0	1
	16	0	0	0	1	0	1	1	0
	17	0	0	0	1	0	1	1	1
	18	0	0	0	1	1	0	0	0
	19	0	0	0	1	1	0	0	1
Hours displayed from the 2nd counter C2 with 2 as the first digit	20	0	0	1	0	0	0	0	0
	21	0	0	1	0	0	0	0	1
	22	0	0	1	0	0	0	1	0
	23	0	0	1	0	0	0	1	1
	24	0	0	1	0	0	1	0	0

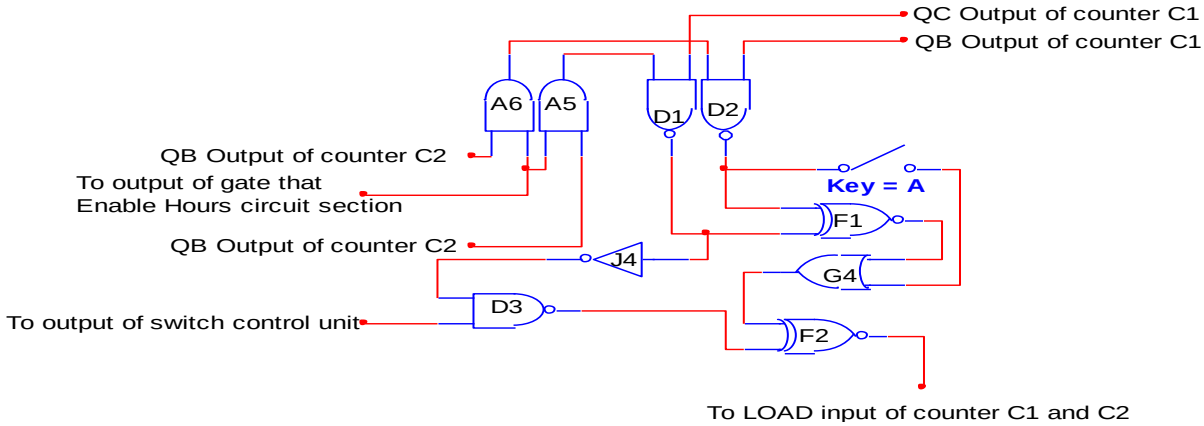

Figure 8: Timing control unit

2.4.3 Timing Detection Unit

The main essence of designing the timing detection unit is to keep record, detect, control and allow the days counter circuit section to advance through it state after every 24 hours. However if the timing selected by the user is 12 hours then there is need to detect and keep record of that timing until it recycle twice to complete the 24 hours cycle. Figure 9 shows the complete circuit of the timing detection unit.

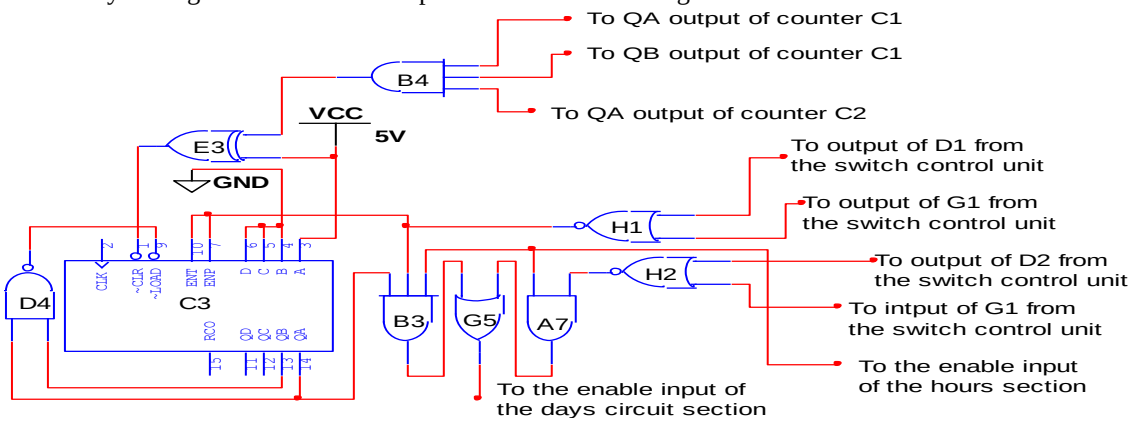

Figure 9: Timing detection unit

2.4.4 Main Circuit Section

The sub units are connected together for the complete design as indicated such that the enable input of the 1st counter is connected back to the output of the minute circuit section gate that allows or enable it to advance through it state after every 60 minutes. Figure 10 shows the complete circuit of the hour's circuit section.

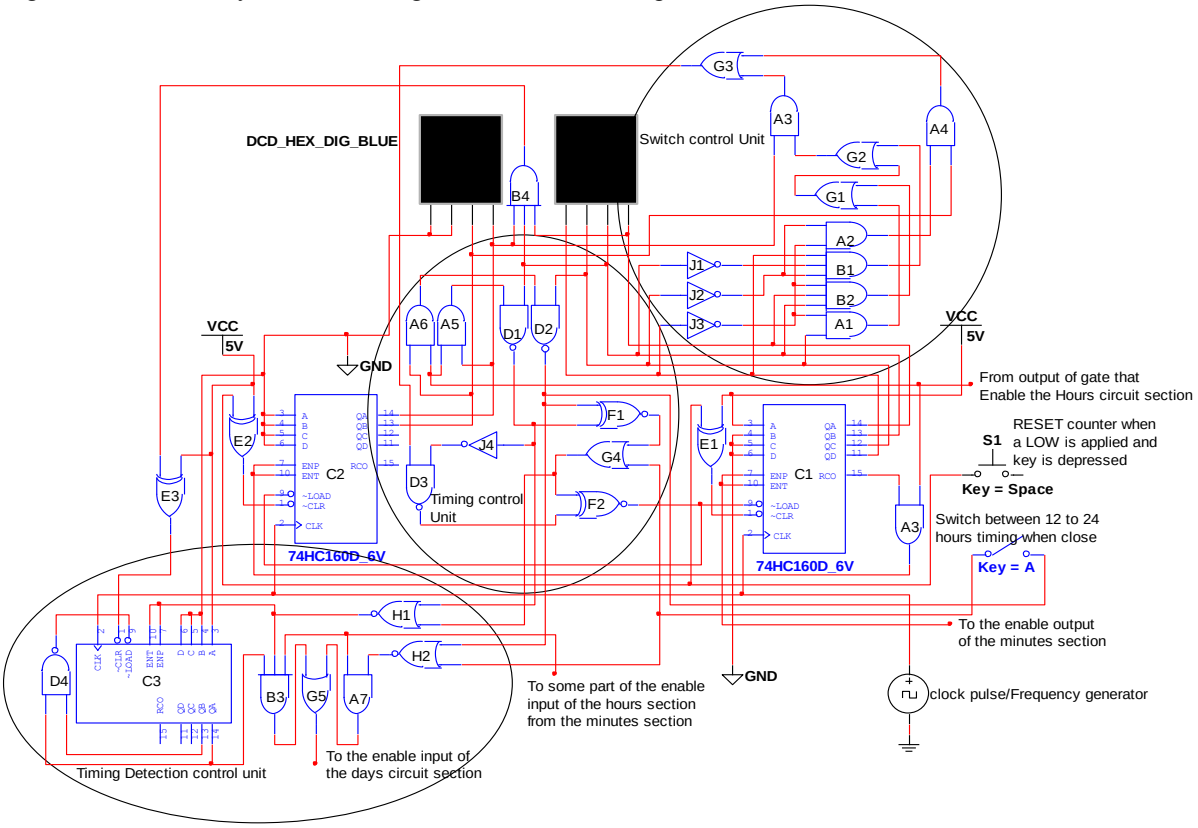

Figure 10: Hours Counter Circuit Section

2.5 DAYS COUNTER CIRCUIT SECTION

This circuit section is design by cascaded arrangement of two 4-bit synchronous decade counters with additional features using combinational elements. Each of the counters counts from 0 to 9 as usual. However, in a specific pattern this property is manipulated using the LOAD input of the counters to allow the counter to load the data input as required. In other to fully control how the days circuit section should count the days and its behavior; there is need to understand that each mount in a normal year or leap year has a specific days count, and this days count can either be 28, 30, or 31 and 29, 30 or 31 respectively depending upon the month and year. For easier implementation the days control unit is analyze separately, hence the circuit design is sub divided into two sections as follows;

1. Days control unit
2. Main circuit design

2.5.1 Days Control Unit

As previously stated, each month has a specific number of days and those numbers of days are displayed through the two counters (denoted C1 and C2) in decimal with a single number displayed through the individual counter. For example if the day is 15 (1111); 1(0001) is displayed through C2 and 5 (0101) through C1 instead of 15 completely as 1111. Hence, in other to critically detect those numbers they must be written down in their equivalent binary coded decimal (BCD) form of 4-bit groups each [3,5,9] . This group is then analyzes and decoded separately from each counter and later on put together to function as one. Table 5 gives the available days for a specific month and their equivalent BCD codes of 4-bit group with each group assign to a letter D, C, B, A corresponding to the QD, QC, QB and QA output of the counters respectively.

Table 5: Available days for a specific month and their equivalent BCD codes

Available days for a specific month in decimal	Equivalent BCD code displayed through the counter C2				Equivalent BCD code displayed through the counter C2			
	D	C	B	A	D	C	B	A
28	0	0	1	0	1	0	0	0
29	0	0	1	0	1	0	0	1
30	0	0	1	1	0	0	0	0
31	0	0	1	1	0	0	0	1

Taking the number 28 (28 days) separately to determine when a 0010 and 1000 occurs from the counter C2 and C1 respectively, we have to consider the first binary 0010. A simple AND gate (denoted A3) can be used as its basic decoding element by determining when the 3rd bit B is 1 (HIGH). This AND gate always produces HIGH output only when all inputs are HIGH, thus the HIGH logic level produce from the output that enable the days circuit section must be considered as the second input in other to fully control this binary detection (all other binary produce from C2 must utilize this Logic level). Similarly for 1000 produce through the counter C1 a NAND gate (denoted D2) is used as the decode element so as to determine when the least significant bit D is 1 (HIGH) and the AND gate output is (gate that detect 0010) HIGH to produces a LOW output as shown in Figure 11.

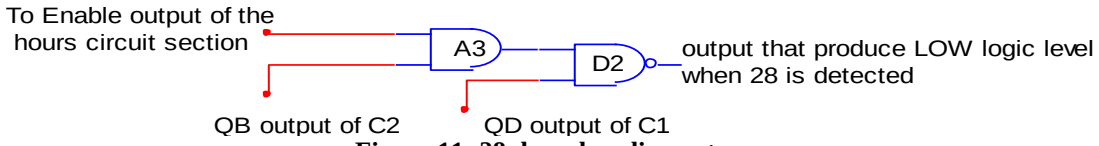

Figure 11: 28 days decoding gates.

At leap year 1 day is added to February making its days count to be 29 instead of 28. Hence for number 29 (29 days count) all detection factors through the gate A3 to determine when a 0010 occur at C2 is similar to that of 28 days. However, for 1001 produce through the counter C1 an AND gate (denoted A7) is used as the decoding element so as to determine when the most significant bit A and the least significant bit D are 1 (HIGH) and the AND gate (gate A3 that detect 0010) output is HIGH to produces a LOW output through the NAND gate D4 as shown in Figure 12.

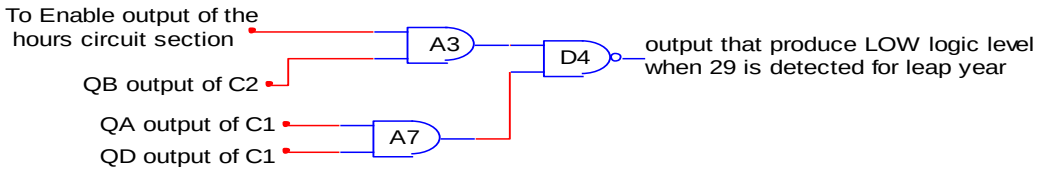

Figure 12: 29 days decoding gates.

For number 30 (30 days count), that is 0011 and 0000; similar AND gate (denoted A1) can be used to decode the 3rd and last bit (B and A) output of C2 to provide a HIGH level when the inputs (B and C) are but HIGH. This output is then connected to another AND gate (denoted A2) through one of its input and the other input is connected to the HIGH logic level produce to enable the hour's circuit section in other to fully control this binary detection and to provide a HIGH output. A 4-input NAND gate (denoted L1) can be used to decode all 4-bits 0000 to provide a HIGH level output. Together outputs of these gates are connected to a NAND gate to produce a LOW output whenever there are both HIGH as shown in Figure 13.

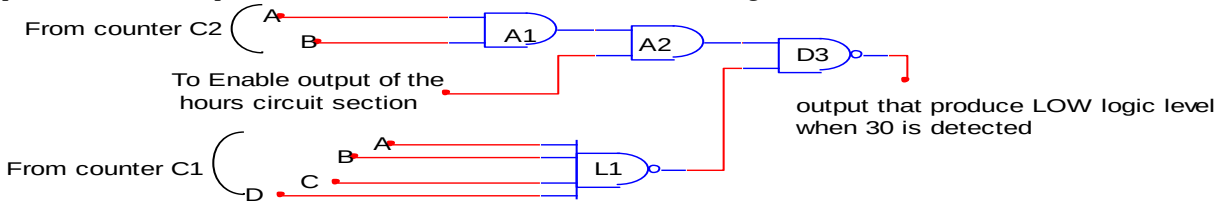

Figure 13: 30 days decoding gates.

Number 31 (31 days) has the 1st group as 0011 similar to number 30, hence the output of the same gates used for its decoding can be utilize. The last bit A (most significant bit) of 0001 can be decode by using a NAND gate (denoted D1) such that it other input will be connected to the output of the same gates used for decoding 30 (that is A2) as shown in Figure 14.

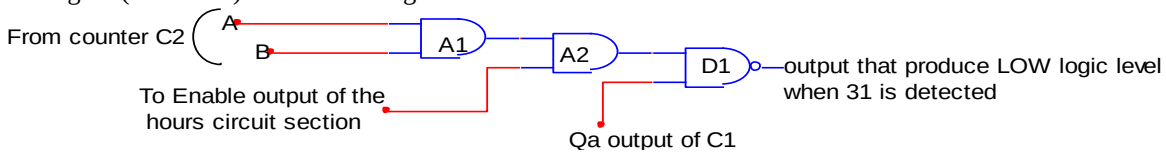

Figure 14: 31 days decoding gates.

Generally, to control these functions in the day's control unit other gates must be connected in the month's circuit section to be able to tell when a specific month has 28, 29(for leap year), 30 or 31 days count. Since all our decoders have a LOW level output when active, another control must be added to incorporate both the decoders output and the proposed days control unit from the month section by assuming them to have all their

output as similar to that of the decoders. Suppose we need to add these LOW levels from both unit the basic element needed is an OR gate connected to exhaust all units from both section. The 4-bit synchronous decade counter used for these design has only one Active-LOW LOAD input which will be utilize to exhibit these control function when the selected Data inputs are loaded through it, thus other basic element must be added that has the property of providing a LOW level output only if one of its inputs is LOW otherwise it will always be HIGH. The right choice is an exclusive OR gate and the design gates are shown in Figure 15.

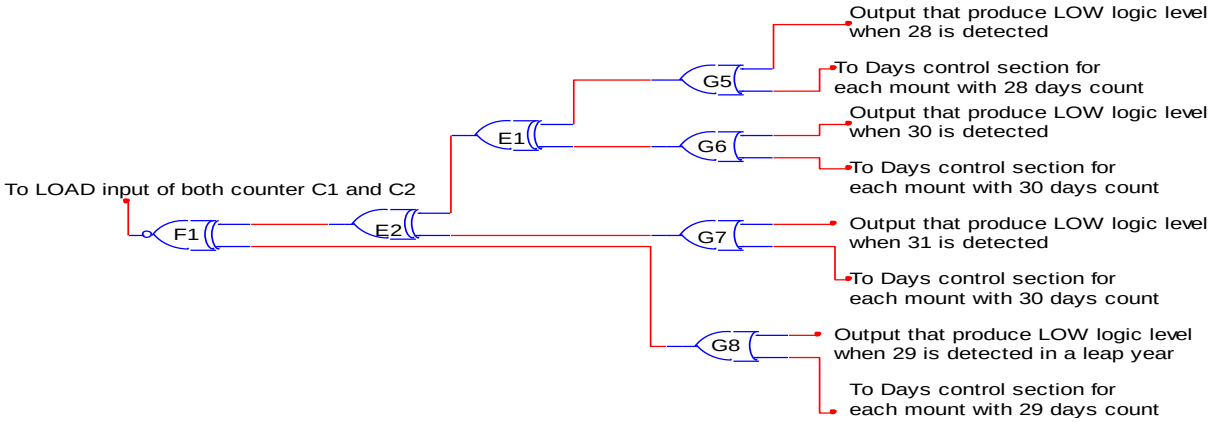

Figure 15: Days control unit.

2.5.2 Main Circuit Design

The architectural design of the main circuit is carried out and produce by cascaded arrangements of the two 4-bit synchronous decade counters as previously stated with the days control unit and some addition gates to accommodate other units. Figure 16 shows the complete days control circuit.

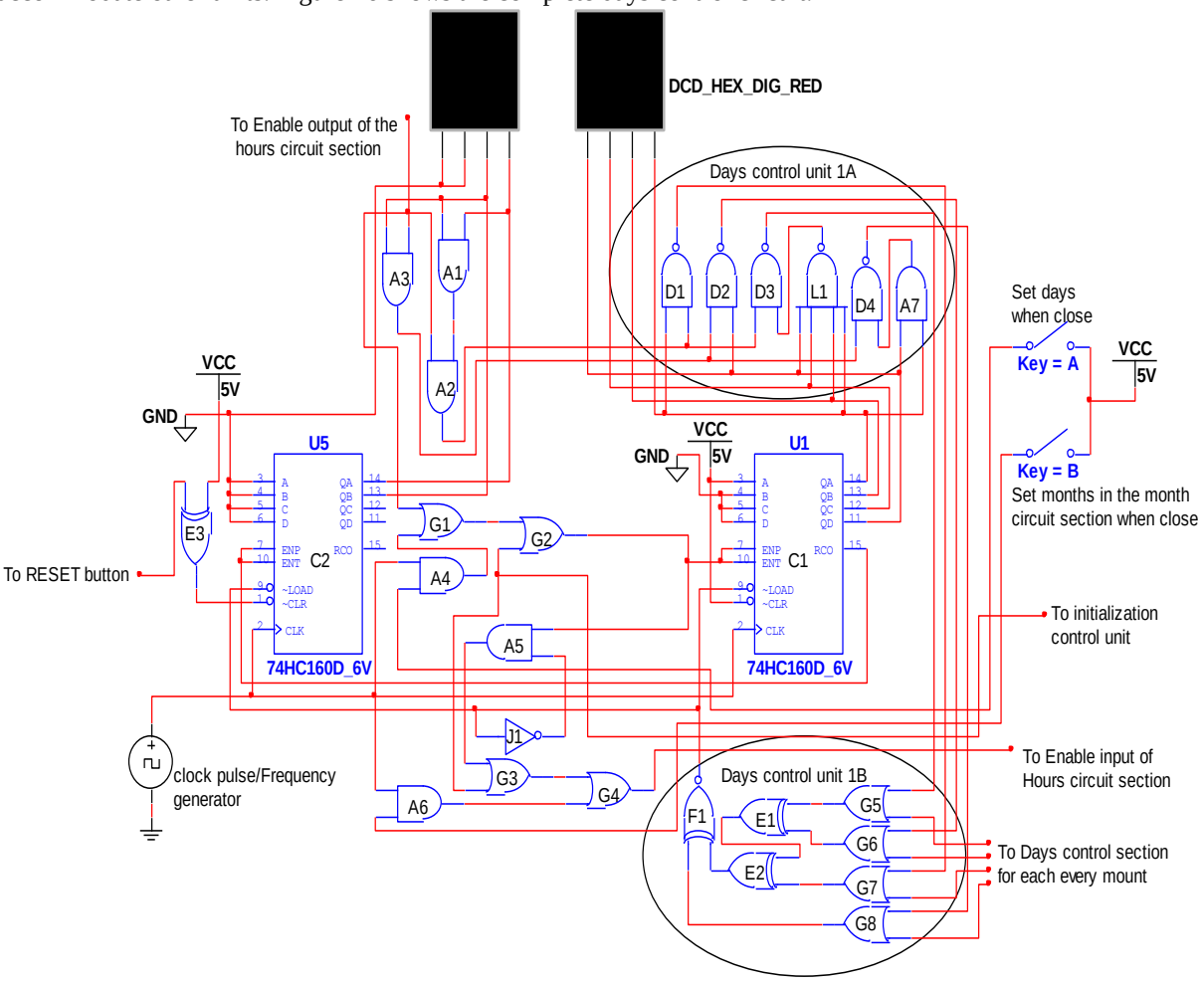

Figure 16: Days Control Circuit.

2.6 Months Circuit Section

The month's circuit section counter is implemented and produce by cascaded arrangements of two 4-bit synchronous decade (divide-by-10) counters with additional circuitries of combinational elements to control the days count for each respective month. Furthermore, in other to control the days count cycle for each respective month a combination of logic gates are implemented by writing down the decimal equivalent of each month base on the categories of the total day's count it has. That is 28 days (normal year) or 29 days (leap year) for February (02), 30 days for September (09), April (04), June (06) and November (11) where 31 days for the rest of the months. The decimal value assign for each respective month is then converted to binary coded decimal (BCD) equivalent of 4-bit code group each. For easier understanding the conversion and categories is writing down in Table 6.

Table 6: Available days in a specific month

Days	Months	Months In Decimal Digit	Months In 4-Bit BCD Groups
31 Days	January	01	0000 0001
	March	03	0000 0011
	May	05	0000 0101
	July	07	0000 0111
	August	08	0000 1001
	October	10	0001 1010
	December	12	0001 1100
30 Days	September	09	0000 1001
	April	04	0000 0100
	June	06	0000 0110
	November	11	0001 0001
28 or 29 Days	February	02	0000 0010

To design the control system each of the 4-bit group is analyze keeping in mind that the first 4-bit group is displayed from the 1st counter C1 and the second 4-bit is displayed from the 2nd counter C2. For example if January (01) is to be displayed the first 4-bit group 0000 is displayed in its decimal equivalent as 0 from the 1st counter C1 and the last 4-bit group 0001 is displayed its decimal equivalent as 0 from the 2nd counter C2 (together does counters displayed 01 for January) likewise all other months are displayed as such. For easier design day's count with minimum number of mouths are selected. As can be seen from table 6 for 30 days count September (09), April (04), and June (06) have their first 4-bit groups in BCD as 0000 while the last 4-bit groups are different hence a simplified Boolean expression can be archive using karnaugh mapping for the last 4-bit groups. Meanwhile November (11) has the 1st 4-bit group as 0001 and last 4-group as 0001; so both groups can be decode for the specific binary digit separately and later on put together. Each individual bit from the last 4-bit group of 04, 06 and 09 are taking from the 2nd counter that is Q_D, Q_C, Q_B and Q_A and assign variables D, C, B, A respectively and the minimum expression for the product terms are derive from the 4-variable Karnaugh map presented in Table 7 and Table 8.

Table 7: Last 4-bit group of 04, 06 and 09

MONTHS	LAST 4-BIT GROUP			
	D	C	B	A
April (04)	0	1	0	0
June (06)	0	1	1	0
September (09)	1	0	0	1

Table 8: Karnaugh map last 4-bit group of 04, 06 and 09

$$\frac{DCBA}{0110}\ \frac{DCBA}{1001}$$
$$\frac{0110}{\bar{D}C\bar{A}}\qquad \frac{1001}{A\bar{B}\bar{C}D}$$

a)

DC / BA	00	01	11	10
00		1		
01				1
11				
10		1		

b)

The resulting minimum sum of product expression is given as;

$$\bar{D}C\bar{A} + A\bar{B}\bar{C}D$$

And the expression is implemented to detect if the month is April (04), June (06) or September (09) as in Figure 17.However the last output is inverted to produce a LOW output for proper operation.

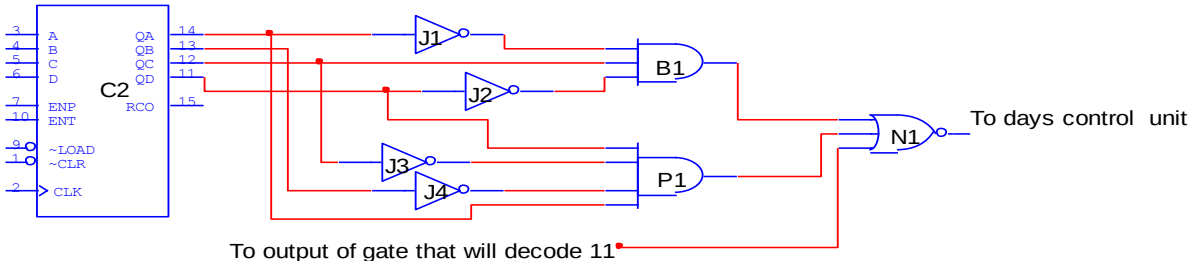

Figure 17: Implemented expression for April, June and September.

For November (11) the first and last 4-bit group 0001 and 0001 are decode from the 1st counter (C1) and 2nd counters (C2) Q_A output since both group has a 1 at that output and a 0 for the other outputs. Suitable gate that can detect a 1 at that output is identified and utilize to give a HIGH logic level at its output whenever the Q_A of both counters is HIGH (1). The remaining digits that complete the 4-bit group are unutilized as shown in Figure 18 and the output of the gate is connected to the input of the gate that detect if the month is April (04), June (06) or September (09).

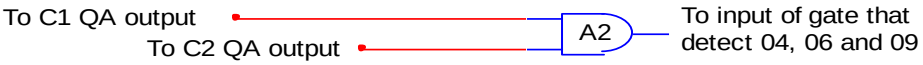

Figure 18: Decoding gate for November.

Similarly, for February (02) the last 4-bit group 0010 are analyze and the individual bits to be decoded are identify as the Q_A, Q_B and Q_c outputs of the 2nd counters (C2), since all binary digits before 0010 (that is 0000 and 0001) has a 0 at its Q_D output then the Q_D output is unutilized. Thus a suitable gate that will produce a HIGH logic level when 010 are detected from Q_A, Q_B and $Q_c$outputs is selected. Likewise the first 4-bit group 0000 resulted from the 1st counter (C1) is detected through its Q_A output and a proper gate is choosing to produce the HIGH logic level at its output. Together the gates that detect the first and last 4-bit group are connected to provide a LOW logic level at its output for proper days count control as shown in Figure 19. However, since February (02) can have both 28 days count for a normal year or 29 days count for leap year; a gate G1 is connected through the output of gate D2 to ensured that its output becomes LOW whenever 28 days is detected from the leap year control output. Similarly a gate D4 with an inverted input J6 is connected through the output of gate D2 to ensure that its output becomes LOW whenever 29 days is detected from the leap year control output.

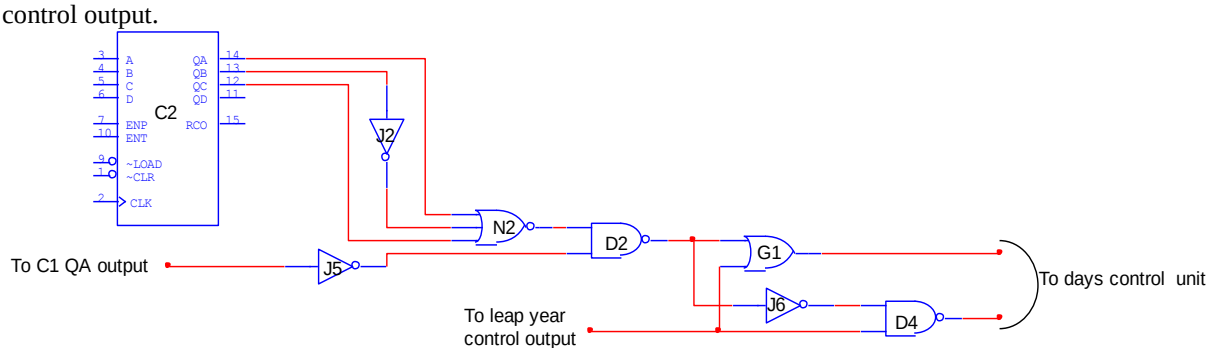

Figure 19: Decoding gate for February.

Moreover, since the days can either be 28 days, 30 days or 31 days count a gate is connected between the outputs of the gates that control 28 and 30 days count to provide a LOW logic level from its output when both or one or of its inputs is HIGH (that is if it is not 28 or 30 days count), this provide the decoding gate for 31 days count as indicated in Figure 20.

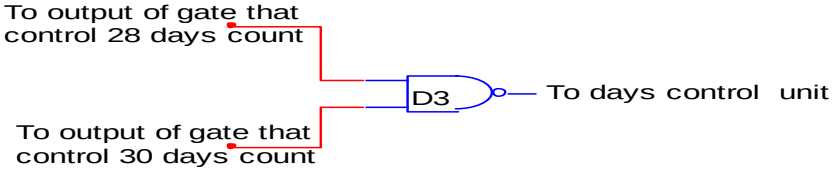

Figure 20: decoding gate for 31 days count.

Finally all the controls units are connected together to provide the days count control unit as shown in Figure 21.

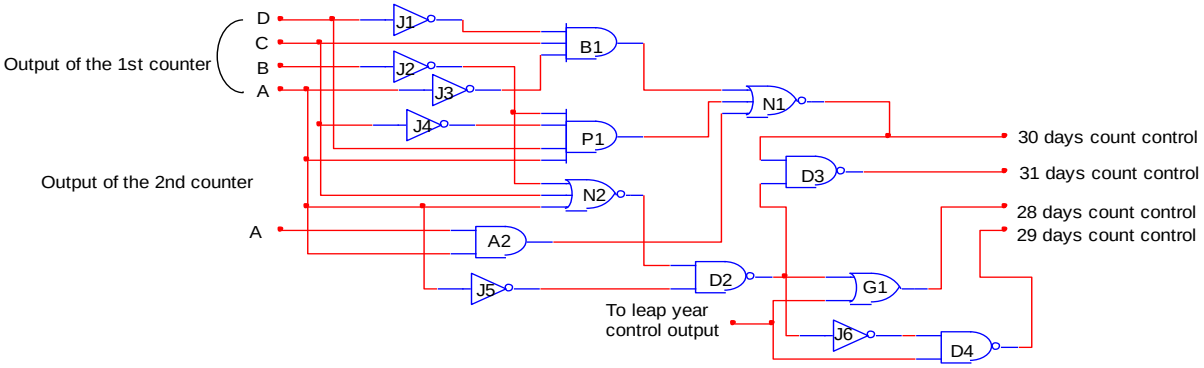

Figure 21: days count control unit for respective month.

This Design Units Together forms the month circuit as shown in Figure 22.

2.7 YEAR'S CIRCUIT SECTION

features using combinational elements. Each of the counters counts from 0 to 9 as usual. This counters display the years accordingly, and since it takes the earth approximately 365.242199 days or 365 days, 5 hours, 48 minutes and 47 second (tropical year) to cycle around the sun some additional features are added to the counter in a specific pattern to manipulate its property to get the appropriate years count [16]. However the Gregorian calendar has only 356 days in a normal year, thus adding a day to February (that is 29 days) every 4 years is necessary, if not we will loss almost six hours off our calendar every year and after every 100 years our calendar will loss 24 days. In other to fully control how a day is added to February to make up the leap year the counters are analyze separately for easier implementation [12]. Hence the circuit design is sub divided into two sections as follows;

1. Leap year detection unit
2. Main circuit design

2.7.1 Leap Year Control Unit

The fundamental essence of designing the leap year detection unit is to keep record, detect and control the years counter circuit section so as to keep our calendar in alignment with the Earth's revolutions around the sun by adding a day to February [13]. In other to fully control when February is 29 days the following criterions are consider; that In the Gregorian calendar the year is evenly divisible by 4; If the year can be evenly divided by 100, it is NOT a leap year, unless; The year is also evenly divisible by 400. Then it is a leap year. Table 9 gives the leap years from 1804 years to 2400 years and can be used to design the detection unit more clearly.

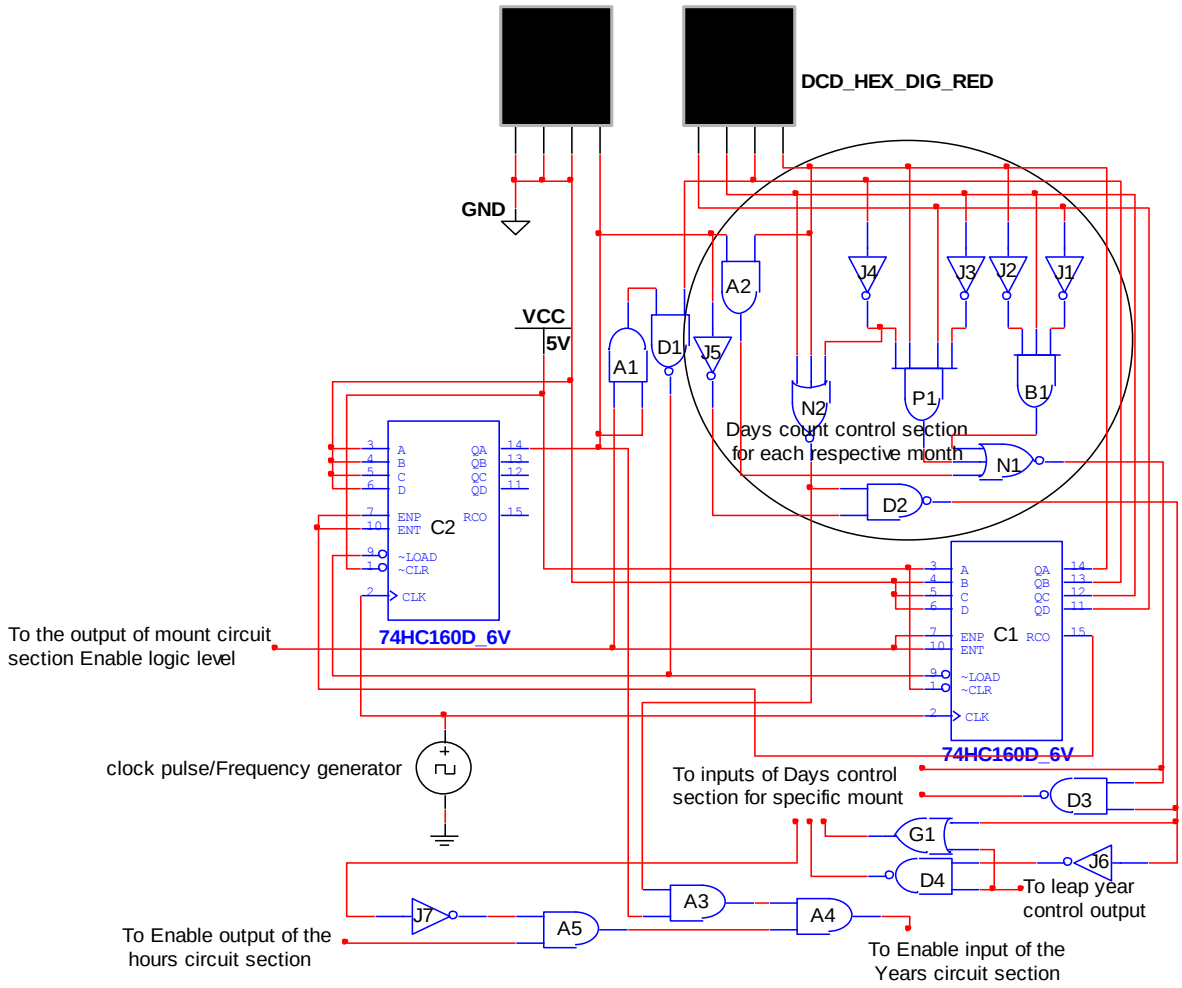

Figure 22: Months Circuit Section

Table 9: Leap years from 1804 years to 2400

ABCD	ABCD	ABCD	ABCD	ABCD	ABCD
1804	1904	2004	2104	2204	2304
1808	1908	2008	2108	2208	2308
1812	1912	2012	2112	2212	2312
1816	1916	2016	2116	2216	2316
1820	1920	2020	2120	2220	2320
1824	1924	2024	2124	2224	2324
1828	1928	2028	2128	2228	2328
1832	1932	2032	2132	2232	2332
1836	1936	2036	2136	2236	2336
1840	1940	2040	2140	2240	2340
1844	1944	2044	2144	2244	2344
1848	1948	2048	2148	2248	2348
1852	1952	2052	2152	2252	2352
1856	1956	2056	2156	2256	2356
1860	1960	2060	2160	2260	2360
1864	1964	2064	2164	2264	2364

1868	1968	2068	2168	2268	2368
1872	1972	2072	2172	2272	2372
1876	1976	2076	2176	2276	2376
1880	1980	2080	2180	2280	2380
1884	1984	2084	2184	2284	2384
1888	1988	2088	2188	2288	2388
1892	1992	2092	2192	2292	2392
1896	1996	2096	2196	2296	2396
	2000				2400

For designing the appropriate leap year control unit of Table 9 lets us consider that the years are in the decimal number system and thus, A is in thousand, B is in hundred, C is in tenth, D is in unit corresponding to the counters C4, C3, C2 and C1 respectively. If we consider the year 2000 to 2096, it can be seen that the number 0,4 and 8 occurs successively in the unit (D) part in correspondence with any of the numbers combination 0,2,4,6 and 8 from the tenth (C) part ; similarly the number 2 and 6 occurs successively in the unit (D) part in correspondence with any of the numbers combination 1,3,5,7 and 9 from the tenth (C) part [7, 9, 13]. This process and number combinations occurs repeatedly every four years and then every 400 years. However, in other to fully control the process, 400 years are detected separately by incorporating the previously stated unit and tenth combination of numbers with hundred (B) part of the combination. From the hundred (B) part it can be seen clearly that the number combination is consistent with the decimal combination 0 to 9 but every 400 years the combination change a little when the number combination 00 occurs from the tenth (C) and unit (D) part with number combination 0,4 and 8 repeatedly from the hundred (B) part[12,13]. A truth table is written down for the specific counts in 4 bit binary equivalent and a standard Boolean expression is obtained by presenting all the possible combinations from the hundred, tenth and unit as appropriate. The value of the resulting outputs variables is later obtained using karnaughmapping[1, 2, 3]. Table 10 gives the possible values in decimal of the unit (D) from the counter C1 separately from the corresponding outputs of the counter C1 in their equivalent 4-bit binary coded decimal (BCD) group.

Table 10: Possible decimal values of the unit (D) from the counter C1 in their equivalent 4-bit binary coded decimal (BCD) group

Decimal Combination	4-Bit Binary Coded Decimal (BCD) Group form Q output of 1st counter C1.			
UNIT PART (D) of 1ST COMBINATION	**D**	**C**	**B**	**A**
0	0	0	0	0
4	0	1	0	0
8	1	0	0	1
UNIT PART (D) of 2ND COMBINATION	**D**	**C**	**B**	**A**
2	0	0	1	0
6	0	1	1	0

The karnaugh map of the 4-bit group taking from the 1st counter Q output of the 1st combination 0,4 and 8 of the unit (D) part of the leap years is shown in Table 11.

Table 11: Karnaugh map of the 4-bit of 1st counter combination 0,4 and 8

$\frac{DCBA}{0000}$ $\frac{DCBA}{0000}$
$\frac{0100}{\bar{D}B\bar{A}}$ $\frac{1000}{C\bar{B}\bar{A}}$

a)

DC / BA	00	01	11	10
00	1	1		1
01				
11				
10				

b)

The karnaugh map of the 4-bit group taking from the 1st counter Q output of the 2nd combination 2 and 6 of the unit (D) part of the leap years is shown in Table 12.

Table 12: Karnaugh map of the 4-bit of the 2nd combination 2 and 6

$\frac{DCBA}{0010}$
$\frac{0110}{\bar{D}B\bar{A}}$

a)

DC / BA	00	01	11	10
00				
01				
11				
10	1	1		

b)

The resulting minimum sum of product expression for both the two combinations is given as;

$$SUM = \bar{D}\bar{B}\bar{A} + C\bar{B}\bar{A} + \bar{D}B\bar{A}$$

The expression is implemented in Figure 23 however additional gates are added to incorporate the other units.

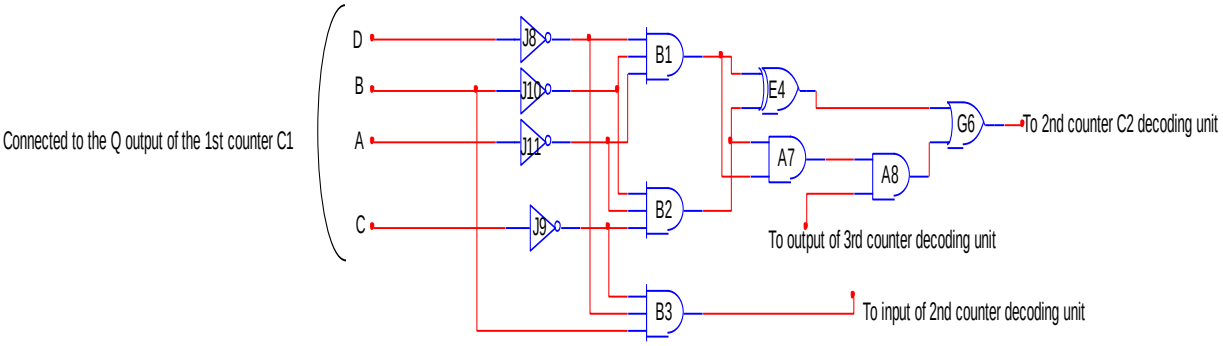

Figure 23: Implemented expression for decoding 00, 04 and 08.

Similarly, Table 13 gives the possible values in decimal of the tenth (C) part from the counter C2 separately from the corresponding outputs of the counter in their equivalent 4-bit binary coded decimal (BCD) group.

Table 13: Possible decimal values of the tenth (C) part from the counter C2 in their equivalent 4-bit binary coded decimal (BCD) group

Decimal Combination	4-Bit Binary Coded Decimal (BCD) Group form Q output of 1st counter C1.			
Tenth PART (C) of 1ST COMBINATION	**D**	**C**	**B**	**A**
1	0	0	0	1
3	0	0	1	1

5	0	1	0	1
7	0	1	1	1
9	1	0	0	1

The karnaugh map of the 4-bit group taking from the 2nd counter Q output of the 1st combination 1, 3 5, 7 and 9 of the tenth (C) part of the leap years is shown in Table 14.

Table 14: karnaugh map of the 4-bit group 1st combination 1, 3 5, 7 and 9

$\overline{DCBA}\ \overline{DCBA}$

$\overline{0001}\ \overline{0001}$

$\overline{0011}\ \overline{1001}$

$\overline{0101}\ \overline{\bar{C}\bar{B}A}$

$\overline{0111}$

$\overline{\bar{D}A}$

a)

DC / BA	00	01	11	10
00				
01	1	1		1
11	1	1		
10				

b)

The resulting minimum sum of product expression the combination is given as;

$$SUM = \bar{D}A + \bar{C}\bar{B}A$$

Since the combination is consistent with the base 10 (decimal) system inverting the resulting sum will give us the 2nd combination. Thus the expression is implemented in Figure 24 however additional gates are added to incorporate the other units.

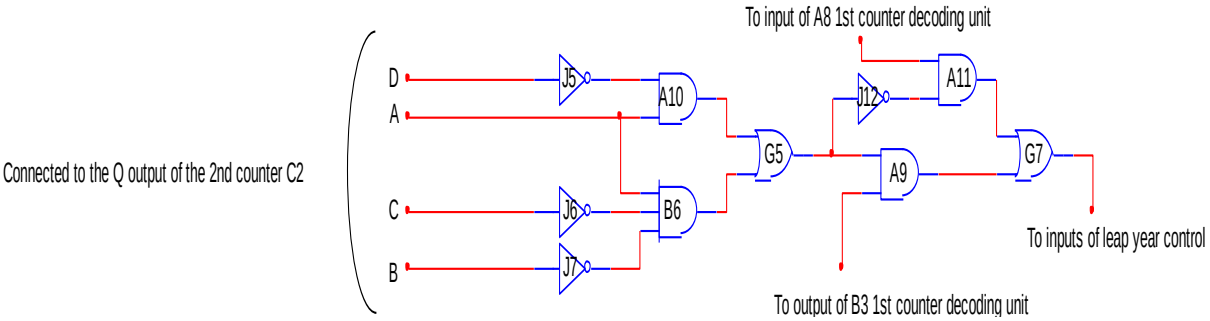

Figure 24: Implemented expression for decoding 1,3,5,7 and 9.

For the hundred (B) part, that is the 3rd counter all combination is similar to the 1st counter combination but only changes every 100 years. Since leap year occurs only after 4 years not 100 years and then occurs 400 years the following gates are implemented as in Figure 25 to detect that change using the same procedure as similar to the unit (A) part from the 1st counter .

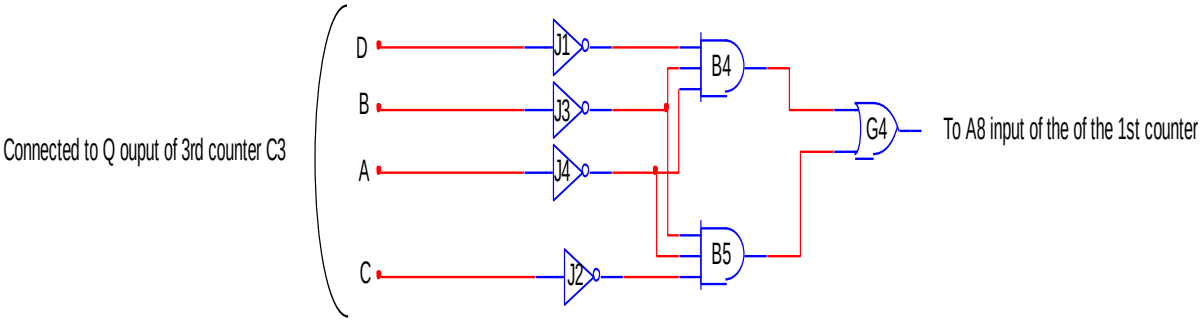

Figure 25: 400 years decoding gate

2.7.2 MAIN CIRCUIT SECTION

As mention earlier the architectural design of the main circuit is carried out and produce by cascaded arrangements of the four 4-bit synchronous decade counters with the leap year detection unit and some addition gates to accommodate other units. Figure 26 shows the complete year's circuit section.

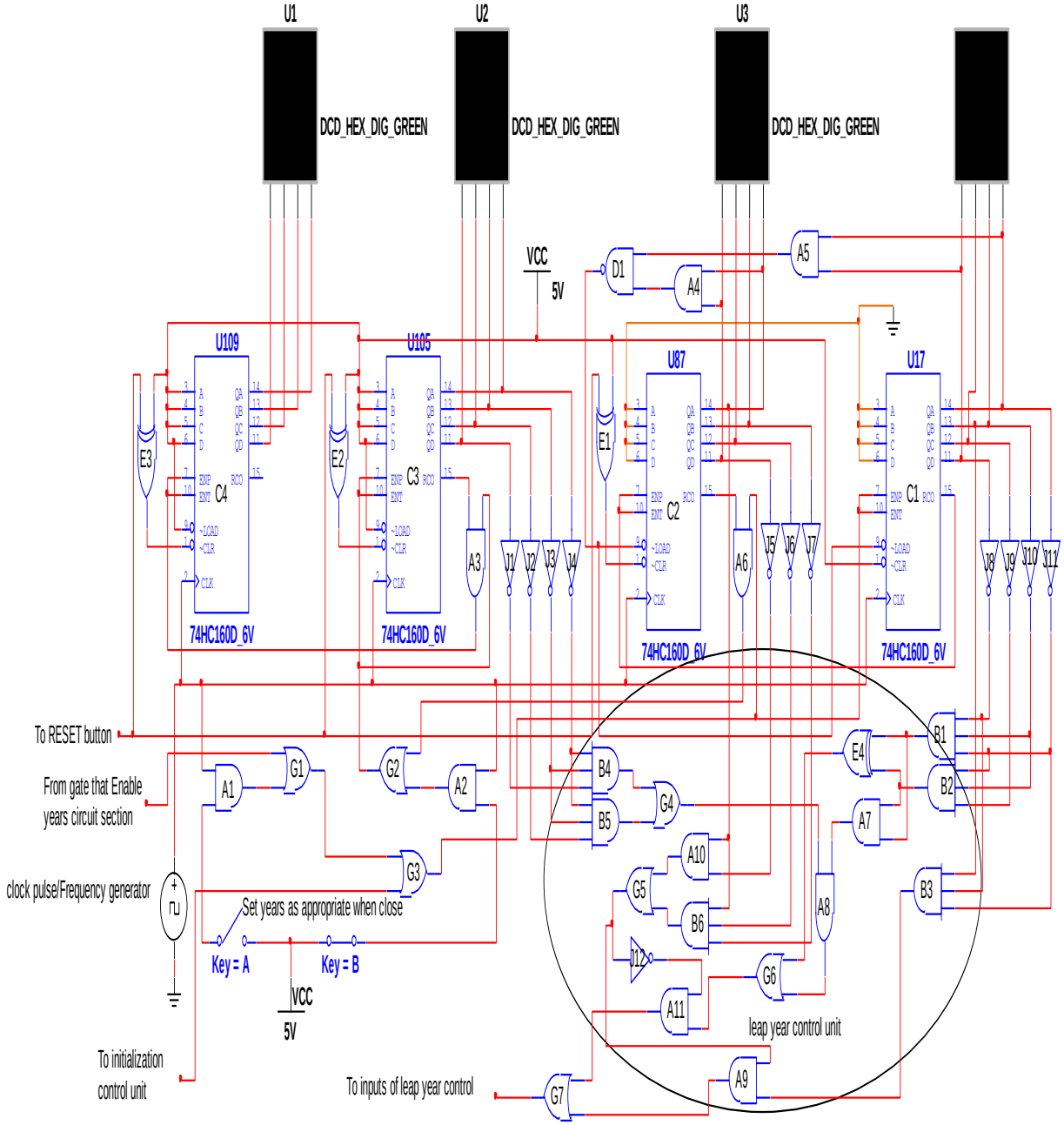

Figure 26: Years Circuit Section

2.8 INITIALIZATION CONTROL UNIT

To have appropriate counts in all the previously discussed section a control unit is design, this control unit consist of two 4-bit synchronous counters C1 and C2 connected as a divide by-3 through a NAND gate say D1 and D2 for both counters respectively. In other to fully explain the initialization process, we assume switch A to be the enable input of the 1st counter C1 in the second circuit section. When HIGH logic level is produce from the enable input of second circuit (by closing switch A) in other to sequence through its state, counter C1 advances through its state by the HIGH level provided at its enable input (ENP and ENT) through the output of the gate G1 whose input is control by the logic level produce by the gate E1 and A1. When the switch A is close the enable input (ENP and ENT) of C1 is control by the logic level provided at output of A1 and the counter is

cleared through F1 as appropriate. The gates J2, A3, A4, E2, G2 and H2 controls the enable input of the month circuit section, while A2 only controls the hour's circuit. Generally this control section allows the digital clock to display a 1 in the hours, days, month and years circuit block only once unless the clock is switch OFF and then back ON. This allows all the sections to advances through their state and RESET the clock as desired. Figure 27 show the complete stop watch control section [1, 2, 3, 12].

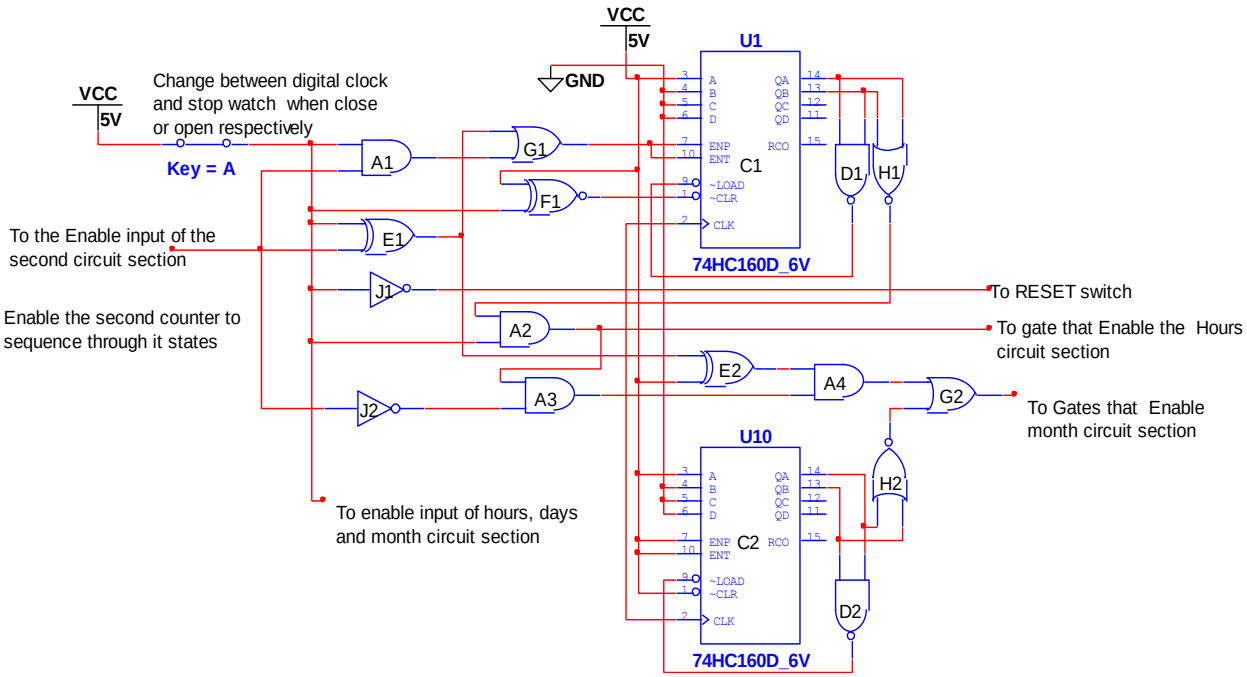

Figure 27: initialization control unit.

2.9 STOP WATCH SECTION

The design of the stop watch section comprises of 5 4-bit synchronous decade counter. The operation and architectural design of the stop watch is similar to that of the second and minute circuit section of the digital clock as shown from the given flow chart in Figure 28.

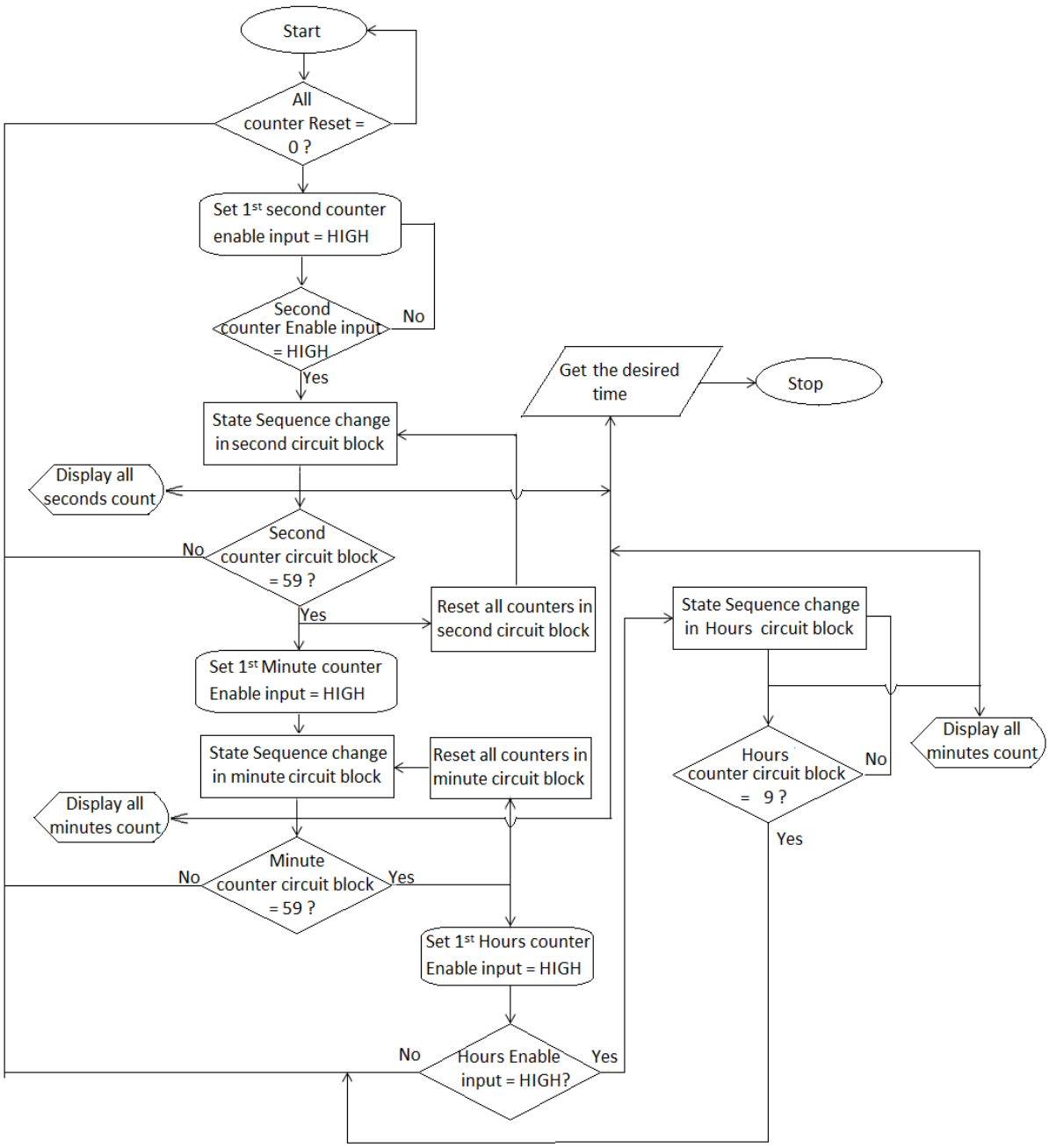

Figure 28: Flow chart of stop watch circuit section

A counter is added such that the Ripple clocking output of the minutes counter can enable it to advance through its state. This counter gives up to 9 hours count and then recycles back to 0. The switch A can be used to activate/start or disable/stop the stop watch as desired while the push button switch can RESET all the counters for a new count. It should be noted that all clock pulse calibration for approximate timing precision is similar to that of the digital clock. When the desired time required is achieved the switch A is open and the displayed time is still retained in the seven segment display due to the clock pulse that is still passing through the counter until the RESET button is depressed [1,2,3]. Figure 29 gives the complete section of the stop watch.

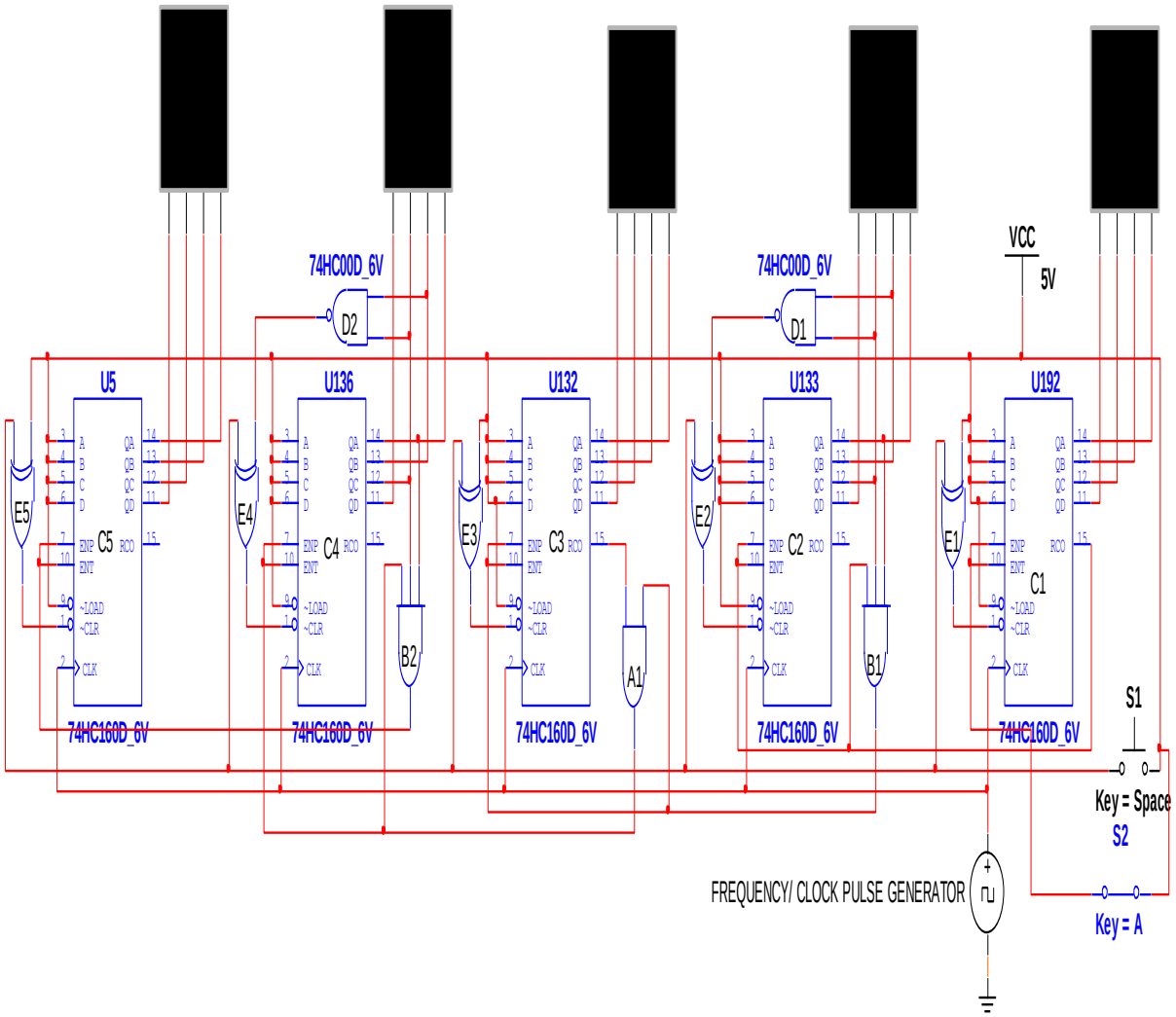

Figure 29: Stop watch circuit section.

a. RESULTS AND DISCUSSIONS

To find the exact or approximate frequency required for the digital clock to have an accurate timing precision with every other clock the Direct Comparison Method is used. Different stop watches are used to compare the timing precision of the design clock in sixty seconds (60s) for different frequencies [14]. The main aim of doing this analysis is to know the exact frequency that will enable the designed digital clock to have an approximate or close timing precision with any clock in question. Table 15 gives the frequency and the time obtained for some clocks.

Table 15: Frequency and the time obtained for different timers.

Frequency Used in hertz (Hz).	Time precision of five different digital stop watches in seconds(s).				
	T_1	T_2	T_3	T_4	T_5
20	67.72	67.79	67.89	67.79	67.74
22	63.77	63.95	63.86	63.77	63.77
23	61.08	61.09	61.06	61.10	61.05
23.3	60.01	60.00	60.00	59.99	60.01
23.5	60.27	60.21	60.27	60.24	60.30
24	59.66	59.69	59.67	59.72	59.69

Furthermore, average timing precision between the five different stop watches used in the experiment can also be used in analyzing the range of frequencies/frequency that are/is close or approximate to the required

frequency for accurate timing sequence of the digital clock/ stop watch. The difference between the Average time (T_A) and the Time precision (T_D) of the design digital clock $T_A - T_D$ is what helps in giving an insight of the frequency range that can be more suitable.

Table 16: Average time (T_A) and Time precision (T_D) analysis.

Frequency Used in hertz (Hz).	Time precision of the design digital clock (T_D) in seconds(s)	Average time (T_A) of the experimental stop watches In seconds(s). $(T_1 + T_2 + T_3 + T_4 + T_5)/5$	Difference between average time (T_A) and time precision of the design digital clock (T_D). $T_A - T_D$
20	60.00	67.786	7.786
22	60.00	63.824	3.824
23	60.00	61.076	1.076
23.3	60.00	60.002	0.002
23.5	60.00	60.258	0.258
24	60.00	59.682	-0.318

Table 16 gives the Average time (T_A) of the stop watches used and the difference between Average time (T_A) and Time precision (T_D) of the design digital clock ($T_A - T_D$). Note the positive sign on differences ($T_A - T_D$) indicate that the design clock is moving at faster rate than the experimental rate to which it can be calibrated while the negative sign indicate that design clock is moving at slower rate than the experimental rate to which it can be calibrated [7,13, 14]. From the table it can be seen that the best choice of frequency to which it can be calibrated is around 23.3 Hertz (Hz) with timing of 60.002s. The percentage error and percentage accuracy can be calculated using the approximate measured value to which the digital clock can be calibrated.

Absolute error $= measured\ value - true\ value$ (1)

$$\text{Absolute error} = 60.002 - 60.000 = 0.002$$

$$\text{Percentage error} = \frac{measured\ \ value - true\ \ value}{measured\ \ value} \times 100 \quad (2)$$

$$\text{Percentage error} = \frac{60.002 - 60.000}{60.002} \times 100 = 0.0033\ \%$$

$$\text{Relative accuracy} = 1 - \left|\frac{measured\ \ value - true\ \ value}{measured\ \ value}\right| \quad (3)$$

$$\text{Relative accuracy} = 1 - \left|3.3 \times 10^{-05}\right| = 0.99997 \approx 0.99$$

$$\text{Percentage accuracy} = 0.99 \times 100\ \% \quad (4)$$

$$\text{Percentage accuracy} = 9.9\ \%$$

From the above calculated values and from Table 16 it can be seen that the error is about 0.002sec in 1 Mins, 0.12 in 1 hour and 2.88sec in 1 day.

III. CONCLUSION

In conclusion the Design of the digital clock/stopwatch was successfully carried out using synchronous counter and basic logic gates. The designed system was implemented and simulated using nation instrument multisim version 13.0. The Simulation result shows that the system functions as desired, where for every 60 second there is one minute and for every 60 minutes there is 1 hour, until the clock reach 12 hours or 24 before it cycle back to 1 hour. Also every 24 hours or at 12 pm we have a day changing successively which is determining by the month and available days in that specific month. Every 31st day of December when the clock is recycling back to 1st day of January we have a year, and every 4 years and every 400 years we have a leap and hence the available days for February becomes 29. Analysis shows that the design digital clock and stop watch have a percentage error of *0.0033%*.

REFERENCES

[1] Thomas L. Floyd, *Digital Fundamentals 9th Edition*, *Pearson Prentice Hall*, Pearson Education international New Jersey, 2007.
[2] M. Morris Mano, *Digital Design*, 3rd Edition, Prentice Hall, Upper Saddle River, New Jersey, 2001
[3] Paul Scherz, *Practical Electronics for Inventors*, McGraw-Hill Companies, New York, 2000.
[4] William Kleitz, *Digital Electronics with VHDL*, Quartus II version, Prentice Hall, Upper Saddle River, New Jersey, 2006
[5] Md. Rabiul Islam and Md. Jobayer Hossain, *Gate Level Design of a Digital Clock with Asynchronous-Synchronous Logic*, Global

Journal of researches in engineering Electrical and electronics engineering, USA, March 2012

[6] I. Ochala, O. Y. Momoh, and F. Gbaorun, *Pelagia Research Library Advances in Applied Science Research*, USA2011

[7] MykePredko, *Digital Electronics Demystified*, McGraw-Hill Companies, Inc., New York, 2005.

[8] Mark Balch, *Digital Design - A Comprehensive Guide to Digital Electronics and Computer System Architecture*, McGraw-Hill Companies, Inc, New York ,2003.

[9] Barry Paton, *Fundamentals of Digital Electronics*, National Instruments Corporation, Texas, 1998.

[10] Anil K. Maini, *Digital Electronics: Principles*, Devices and Applications, John Wiley & Sons, Ltd., 2007.

[11] Ronald J. Tocci *Digital Systems Principles and Applications 8th Edition*, Prentice Hall, Upper Saddle River, New Jersey, 2001

[12] John F. Wakerly, *Digital Design Principles & Practices 3rd Edition*, Prentice Hall, Upper Saddle River, New Jersey, 1999.

[13] G.S.M. Galadanci, S. M. Gana and M. Abdullahi, *Design and Implementation of Digital Clock using Frequency Dividers and 555 timer*, Journal of Nigerian Association of Mathematical Physics, Nigeria, Volume 23, march 2013.

[14] Jeff C. Gust, *Stop Watch And Timers Calibration*, National Institute of Standards and Technology, Washington, U. S. A, May 2004.

[15] Phillip a. Laplante and Seppo j. Ovaska, *Real-Time Systems Design And Analysis 4th Edition*, John Wiley & Sons, Ltd. Hoboken, New jersey, 2012.

[16] Unknown, leap year, 2014. Retrieved on 20/02/14, http://kalender-365.de/leap-years.php

[17] Unknown, Timer, 2014. Retrieved on 29/05/14 from http://en.wikipedia.org/wiki/Timer

Aknowledgement
The authors wish to aknowledge the contribution of Bayero University, Kano toward the success of this research through the award of research grant to one of them.