CLOCK PROJECT

1. Overview

As mentioned in the Ppm Project handout, the ppm project has three different versions. Students will develop the machine vs. human : **ppmvsh** version of the Ppm. This handout describes the clock project. The clock project is intuitive and is similar to clocks that we use daily. The clock project has two modes : A digital clock and a timer. One can switch between the two modes. It is described in this Clock Project handout. Both projects are also described in lab presentations.

When the Clock project is downloaded or reset, it enters the reset state and waits there until the user sets the seconds+minutes+hours or starts the timer. Then, the clock can be **reset** anytime where all eight displays show 0s. The clock can time while the digital clock is keeping track of the time. However, if the mode is switched from the timer mode to the clock mode, the timer pauses. This is a feature that can be enhanced by the students. One can pause the timer and resume it as many times as needed. One can set the clock as many times as needed. Note that centiseconds are not set ! In summary, the clock allows a timer and a digital clock to work in parallel, with clock set options. There is no alarm feature for the clock which can be an enhancement option for students.

The clock circuit is a **digital system**. A digital system consists of digital circuits. Today's digital systems are numerous. Examples of digital systems are microprocessors, computers, DVD players and iPhones. They are also complex. Consequently, special emphasis is given to the coverage of digital systems in computer science and computer engineering curricula. For example, digital systems is a major topic of **CS 2214**, as well as **upper level ECE courses**.

The clock digital system consists of four blocks on four schematic sheets. All blocks are **core** blocks : They are already designed and provided with on the four schematics. Schematic 2 has a black box or **macro**. Students cannot see its schematic implementation, but only its VHDL code. Students can use the clock project and combine with the machine vs human or ppmmvsh project.

The experiments during the semester are such that students work on three phases of the term project, or three experiments : Experiment 1 through 3. Experiments 1 and 2 are for learning the hardware and software fundamentals of the lab as well understanding the ppm project. Experiment 3 completes the term project. All projects affect the term grade. In Experiment 3, teams finalize how they enhance ppmmvsh project. Experiment 3 is collected in Lab 14 of each lab section.

The experiments follow the class coverage where initial circuits are combinational circuits. Then, blocks with sequential circuits are designed. Below, we first describe the black box view of Clock and then the individual blocks. The descriptions are in the context of digital systems. The description is also based on the Digilent NEXYS-4 DDR FPGA board.

2. Black-Box View and the Input/Output Relationship (Operation) of the Clock

The clock project is a digital system. The black-box view of the clock is shown in Figure 1. It has 17 inputs and 21 outputs. One of the inputs is a clock signal generated on the board. All the remaining signals are connected to I/O devices such as **switches**, **push buttons**, **7-Segment displays**, and **LED lights** which are used to set the time, to switch the mode or to start and pause the times.

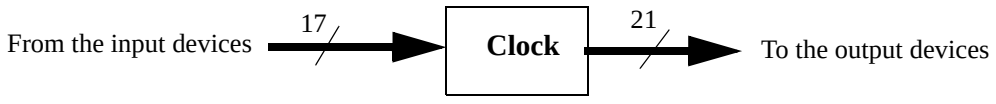

Figure 1. The clock black box view.

2.1. The Clock Input/Output Devices

This handout describes the use of the eight 7-segment displays, six of the 16 LED lights and the rightmost three switches of the board. The other input/output devices can be used by the students to enhance the clock project and also to do hardware debugging. The clock I/O devices used, other than the clock are shown in Figure 2. The detailed view of the clock inputs and outputs is shown in Figure 3. The description of the inputs and outputs of the clock are shown on Table 1 and Table 2, respectively. 11 inputs are from the switches and five inputs are from push buttons. 15 outputs are for the 7-segment displays and six are for the LED lights. One signal that is not connected to an I/O device is the **Clock** signal. It is at 100MHz and used for timing in the digital system.

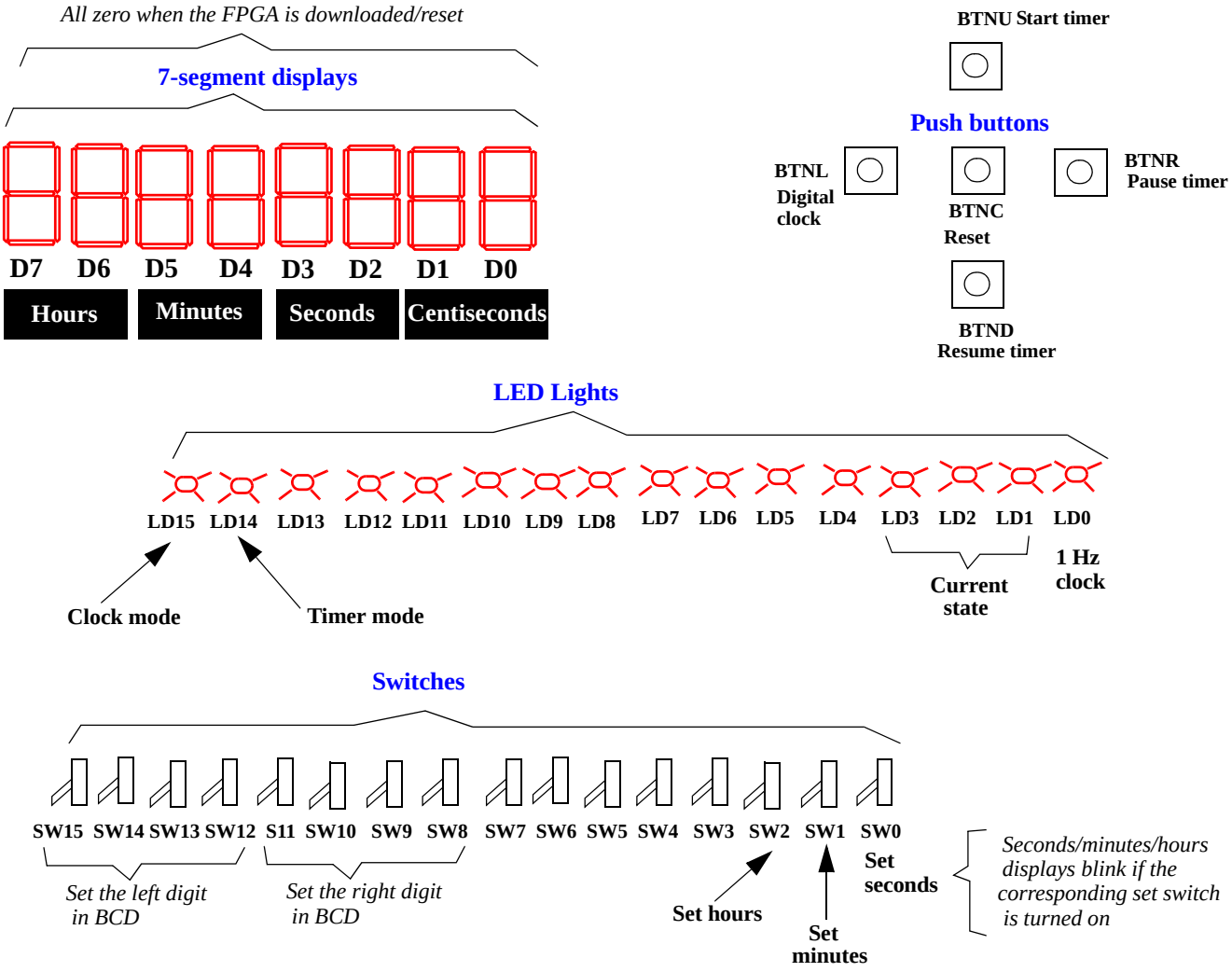

Figure 2. FPGA Board Input/Output device utilization of the clock.

The five push buttons on the right, BTNR, BTNL, BTNU, BTND and BTNDC, are used to switch between the digital clock and timer modes and to start and pause the timer. Normally, when these push buttons are not pressed, they generate logic 0. As long as they are pressed, they generate logic 1. For the Clock project, the default starting mode is the Digital Clock mode after download/reset. All 7-segment displays show 0s. One presses push button left, BTNL, to start digital clock counting. That means the digital clock starts at time 0:00 exactly. Otherwise, one goes through the Digital Clock set operations with the switches described above to set the Digital Clock. As we observe the current time, the Clock is in the Clock mode and the leftmost LED light is turned on.

In order to use the timer, one presses BTNU. The timer starts at time 00:00:00:00. As long as the Pause push button, BTNR, is not pressed, we see the timer output counting. When BTNR is pressed, the timer stops counting and stays as it is until the Resume Timer push button, BTND, is pressed. The timer resumes its counting from the count it was

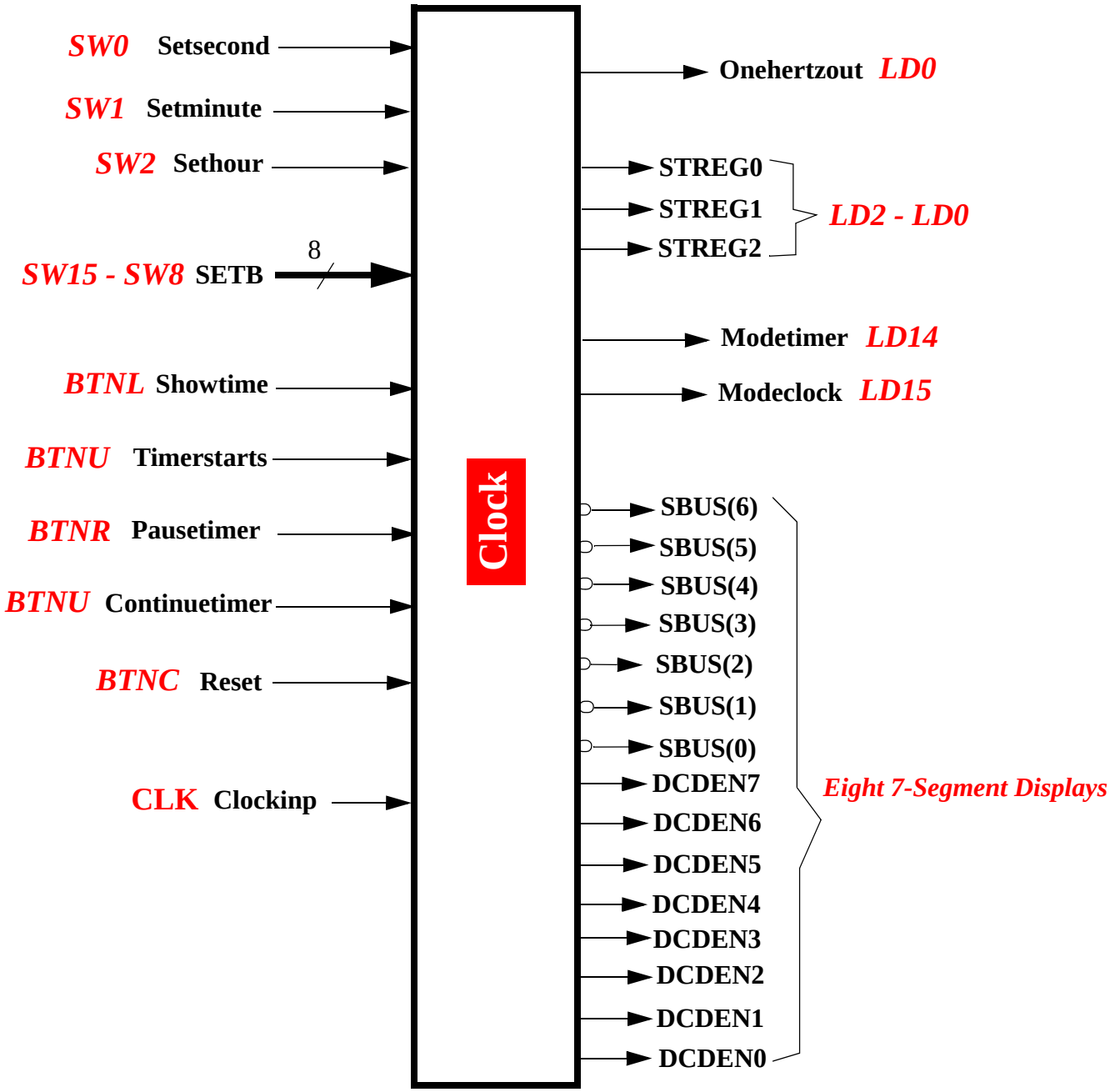

Figure 3. Inputs and outputs of the clock.

paused on. Overall, when it is the Timer mode, LED light 14, next to the leftmost light, is on. While the timer counting is observed, the Digital Clock is working in parallel. In order to see the current time, one has to change the mode to the Clock mode, by pressing push button left, BTNL. However, when we switch to the Clock mode, the timer is paused ! This is a feature students can enhance for their Experiment 4.

The 11 switches are used to set the current time. If the switch handle is closer to the human, it generates logic 0, otherwise it generates logic 1. These 11 switches are leftmost eight switches, SW15-SW8 and rightmost three switches, SW2-SW0. The rightmost switch, SW0, sets seconds on displays, **D3 and D2** ; SW1 sets minutes on displays, **D4 and D5**, and SW2 sets hours on displays **D6 and D7**. If one of these switches is turned on, the corresponding displays blink, indicating that they are being set. Finally, push button center, **BTNC**, resets the system, clearing all displays, the current time and timer go back to 00:00:00:00. One can press BTNC anytime to reset the system.

The leftmost four switches, SW15-SW12, set the left display in BCD and the four switches to the right, SW11-SW8, set the right display in BCD. As an example, to set seconds, one turns on SW0 and then adjusts switches SW15 to SW8 to set the seconds in BCD and then turns off SW0. Adjusting SW15 to SW8 in BCD means that if they are adjusted to decimal digits. For example if they are (0101 0110), it means, the seconds are set to $(56)_{10}$. Then, one turns on SW1 and adjusts SW15 to SW8 to set the minutes in BCD and turns off SW1, and so on.

Input device	Description
SW15 - SW8	Set the two digits of hours/minutes/seconds. Switches SW15-SW12 determine the value of the left digit in BCD and switches SW11-SW8 determine the value of the right digit in BCD.
SW2	Set hours displays. Turn it on then adjust SW15-SW8 then turn it off
SW1	Set minutes displays. Turn it on then adjust SW15-SW8 then turn it off
SW0	Set seconds displays. Turn it on then adjust SW15-SW8 then turn it off
BTNL	Show time. When it is pressed the current time is shown. It can be pressed after downloaded/reset or after setting displays or in the Timer mode
BTNU	Timer starts. By pressing the push button the timer starts counting at 00:00:00:00
BTNR	Pause timer. After the push button is pressed, the timer pauses. All display stay unchanged until BTND is pressed
BTND	Continue timer. Once pressed, the timer resumes its count from the count left off
BTNC	Reset. When pressed, the Clock project returns to its reset state where all displays are 0, the current time is 00:00:00:00 and the timer is also at 0.
Clockinp	The clock signal. It is a 100 MHz clock signal generated by an FPGA board circuit

Table 1: Inputs of the Clock project.

Output device	Description
LD15	Mode clock. When it is bright (on), it indicates that the Clock project is in the digital clock mode, showing the current time
LD14	Mode timer. When it is bright (on), it indicates that the Clock project is in the timer mode, showing the current timer value
LD2 - LD0	The **state** of the Clock project. The value shown is between 0 and 7. State 0 is the reset state. States 4 indicates the current time is shown, advancing every 100th of a second. States 7 indicates the current timer value is shown, advancing every 100th of a second.
LD0	One Hertz out. The LED light blinks once every second
Eight 7-Segment displays	By default, the displays show the time and LED15 indicates it is the clock mode. In the timer mode, they show the timer output. The centiseconds displays are PD1 and PD0. The seconds displays are PD3 and PD2. The minutes displays are PD5 and PD4. The hours displays are PD7 and PD6.

Table 2: Outputs of the Clock project.

Note that, one can just set the hours or minutes seconds after the Digital Clock is set. For example, in order to change from the summer time to the winter time, one needs to change the hours, by decreasing the hour value by one. It is done by turning on SW2, then adjusting SW15-SW8 followed by pressing BTNL to observe the current time. However, as the hours value is set, the Digital Clock stops, meaning the centiseconds, seconds and minutes pause which is not correct. This is another feature students can enhance about for the Clock project.

There are 15 outputs that are used to show digits on the eight 7-segment displays. These displays PD7, PD6, PD5, PD4, PD3, PD2, PD1 and PD0 show 0s when the Clock project is downloaded/reset by default and the mode is the

Clock mode. They show the timer output when BTNU or BTND is pressed and the mode is switched to Timer mode. If BTNL is pressed they show the current time and the mode is the Clock mode again. PD7 and PD6 show the hours, PD5 and PD4 show the minutes, PD3 and PD2 show the seconds and PD1 and PD0 show centiseconds.

Eight of the 15 outputs are to turn on and off the eight 7-segment displays in a round robin way. These eight outputs are named DCDEN7 to DCDEN0, one for each display. These outputs are **active-low**. The remaining seven outputs are the seven segment outputs shared by all eight displays. The seven outputs names SBUS(7) to SBUS(0) are used by the displays turned on at the moment. For example, if DCDEN2 is 1 at the moment, the seven SBUS outputs are used to show a digit on display. The eight displays are turned on and off at a high speed to show eight digits such that the human eye cannot realize that at any time only one display is on. The reason why all eight displays are not on simultaneously is that we save output signals. That is, if all are on, we have to output seven times eight 56 signals, not 15 that we have now.

Parallel to the 16 switches are 16 LED lights, LD0 through LD15. Only six LED are used by the Clock project. LED0 blinks at 1 Hz to signal the seconds advancing with each blink. That is, the LED light blinks once a second. The three lights to the left, LED1, LED2 and LED3, show the current state of Clock. That is, they show the **state** the Clock system is in. The values are between 0 through 7. The state concept is explained in detail below.

2.2. The Clock Input/Output Description

The input/output relationship is the operation description of the Clock project, including how to set the clock and how to use it as a timer. A good understanding of the input/output relationship is needed to design the Clock digital system correctly. Formally, the input/output relationship should be given by an **operation diagram** since the Clock has sequential circuits. However, in order to simplify the discussion of the Clock system, we will first describe the way to use it in detail and then give the operation diagram.

2.2.1. The Clock Usage Description

The Clock is reset when it is downloaded or reset. It is also in the clock mode. In this mode, the clock hours, minutes, seconds are set and the clock is started. If the timer is started, the displays show the timer counter outputs and the mode is the timer mode. While the timer is counting, the clock is also advancing, meaning they work in parallel. Switching to the clock mode however, pauses the timer though. Overall, the clock works as follows after a reset/ download :

- The user has two options in the beginning
    - Set the seconds, minutes and hours and look at the current time
- Start the timer and look at the times counting
    - While the timer is counting, one can switch to the clock mode and see the current time
        - This pauses the timer
- The user can set the seconds or minutes or hours independently any time

2.2.1.1. An overview

After the FPGA chip is downloaded, all eight 7-segment displays show zero, since by default, the clock enters the reset state. The clock is at 00:00:00:00 and the timer is at 00:00:00:00. All LED lights are blank except the leftmost and rightmost. The leftmost LED is on, indicating Clock is in the Digital clock mode. The rightmost LED light blinks at 1 Hz.

While the displays show 00:00:00:00, if BTNU is pressed the timer starts counting. If BTNR is pressed the timer pauses until BTND is pressed after which it resumes counting. One can press BTNL to see the current time. If we turn on SW0 in the reset state, we can set the seconds then we turn the switch off. Then, we can set minutes and hours by turning and off SW1 and SW2 successively. When we set a particular group of digits, those digits blink until their switch is turned off. After reset, setting the time and pressing BTNL will make sure we see the right current time. Note that, if we are in the clock mode, the times is paused if it has been started.

2.2.1.2. The Clock Operation steps

This section shows the operation steps. Although, the critical point is how we use the eight displays, below we list all the steps in the order starting from **reset** :

1) When the FPGA chip is downloaded, all eight displays are zero, the clock timers and the timer counters stay at 0. All 16 LED lights are off, except the leftmost and rightmost. The leftmost LED light is on indicating it is the clock

mode. The rightmost LED light blinks at 1Hz, i.e. blinks once a second. The three LED lights next to it, LED1, LED2 and LED3, are off, indicating it is the reset state at the moment. Note that the FPGA chip can be downloaded and reset anytime, for example when we look at the current time.

2) The user has two options then,

a) Press BTNU to start the timer at 00:00:00:00. We observe the displays show the timer counter outputs. The system enters timer mode. The LED light next to the leftmost LED light is turned on indicating the new mode. LED lights LED1, LED2 and LED 3 indicate it is state 5, the timer has been started. LED0 blinks at 1 Hz. One can pause the timer, by pressing BTNR. The state moves to state 6. Pressing BTND resumes the timer operation. The state moves state 7. Note that, once the timer is started and the state 4, the digital clock is started at time 00:00:00:00. Therefore, one can press BTNL to see the current time in states 5, 6 and 7. But, the time is not right since it has not been set.

b) Turn on SW0 to set the seconds. The state lights indicates it is state 1 in which seconds are set. LED15 is on, indicating it is the clock mode. Use SW8-SW15 to choose values for the two displays of seconds. Then, turn it off. Repeat these for SW1 to set the minutes where the state is 2 and then for SW2 to set the hours where the state is 3. Then press BTNL to see the clock working and showing the current time. The state shown is 4.

3) If option (a) is chosen, i.e., the time mode is chosen in the beginning, the user observes the timer counter values while in the background the clock is also working, but with an incorrect current time value. The state is 5. Then, the user has the following steps to perform :

- Press BTNR to pause the timer. The state is 6. The eight displays show the stopped timer value. The clock counters are working in the background with an incorrect value.
  - Press BTND to resume counting. The state is 7. The eight displays show the timer counter output values changing. The clock counters are working in the background with an incorrect value.
    - One can press BTNR and BTND to pause and resume the timer any time. During that time, the clock counters are working in the background with an incorrect value.
- In state 5, 6 and 7, the user can press BTNL to see the current which is incorrect. The state is 4. But, by using switches SW0, SW1 and SW2, the current time can be set. Also, the user can just set the seconds or minutes or hours. The three do not have to be set wants to be set in succession.

4) If option (b) is chosen, i.e., the clock mode is chosen in the beginning, the user has to set the seconds, minutes and hours **in succession** to be able to see the clock counters counting eventually, i.e. the clock moves from state 0 to 1, then to 2 and then 3, finally reaching state 4 to see the current time. The user cannot just set the seconds and press BTNL to see the current time. The system does not respond to that. That is, the clock does not move from state 0 to 1 and then to 4. While the user sets the time, the timer is at 00:00:00:00. Then, the user has the following steps to perform :

a) Turn on SW0 to start setting the seconds : Use SW8-SW15 to enter the seconds. The state is 1. The setting is in BCD where each group of four switches determine one digit of seconds. For example, if they are 0001 0110, it means the seconds setting is 16 in decimal. After adjusting SW8-SW15, turn of SW0.

b) Turn on SW1 to start setting the minutes : Use SW8-SW15 to enter the minutes. The state is 2. The setting is in BCD where each group of four switches determine one digit of minutes. For example, if they are 0001 0110, it means the minutes setting is 16 in decimal. After adjusting SW8-SW15, turn of SW1.

c) Turn on SW2 to start setting the hours : Use SW8-SW15 to enter the hours. The state is 3. The setting is in BCD where each group of four switches determine one digit of hours. For example, if they are 0001 0110, it means the hours setting is 16 in decimal. After adjusting SW8-SW15, turn of SW2.

d) Press BTNL to start the clock ! The state is 4. Once we observe the current time, we can do the following in **any order** we want :
  - Press BTNU to start the timer. The state is 5. The eight displays show the timer counter values changing starting at 00:00:00:00. The clock counters are working in the background with a correct value.
  - Press BTNR to pause the timer. The state is 6. The eight displays show the stopped timer value. The clock counters are working in the background with a correct value.
  - Press BTND to resume counting. The state is 7. The eight displays show the timer counter output values changing. The clock counters are working in the background with a correct value.
    - One can press BTNR and BTND to pause and resume the timer any time. During that time, the clock counters are working in the background with an incorrect value.
  - Press BTNL any time to switch to the clock mode in states 5, 6 and 6. The eight displays show the current time.

5) The user can press BTNC to reset the clock anytime. Then, one starts with step 1 above.

2.2.2. Clock Modes and Its Operation Diagram

The operation diagram of the clock is shown in Figure 5. The operation diagram shows the operations graphically with respect to time. In order to clearly describe the operations, i.e. what the user does and when, the operation diagram is given in terms of **modes** and **states**. The clock system is always in a specific mode at any time. There are three modes : the **Reset** mode, the **clock** mode and the **timer** mode.

A mode consists of submodes (steps or **states**). A circle is a state in the operation diagram. In each state, the system performs specific operations. Each state takes at least one clock period. To refer to the states easily, numbers are assigned to the states. The assignment is from top to bottom, though, in real life, this is done in a way to reduce the hardware. Each line connects one state to another and is taken if there is a label next to the line and the label is true. For example, if a state has a circled line directed at itself and another line with a label directed at another state, then the system stays in the state until the condition in the label is satisfied. That is, it waits for certain input/event (a push button press/the switch is turned on, etc.) indicated by the label to occur to move to the next state. Also, certain state transitions change the mode to another mode. In the next section, the operation diagram is used to obtain the major operations and blocks of Clock. The concept of state is discussed in detail when sequential circuits are covered in class.

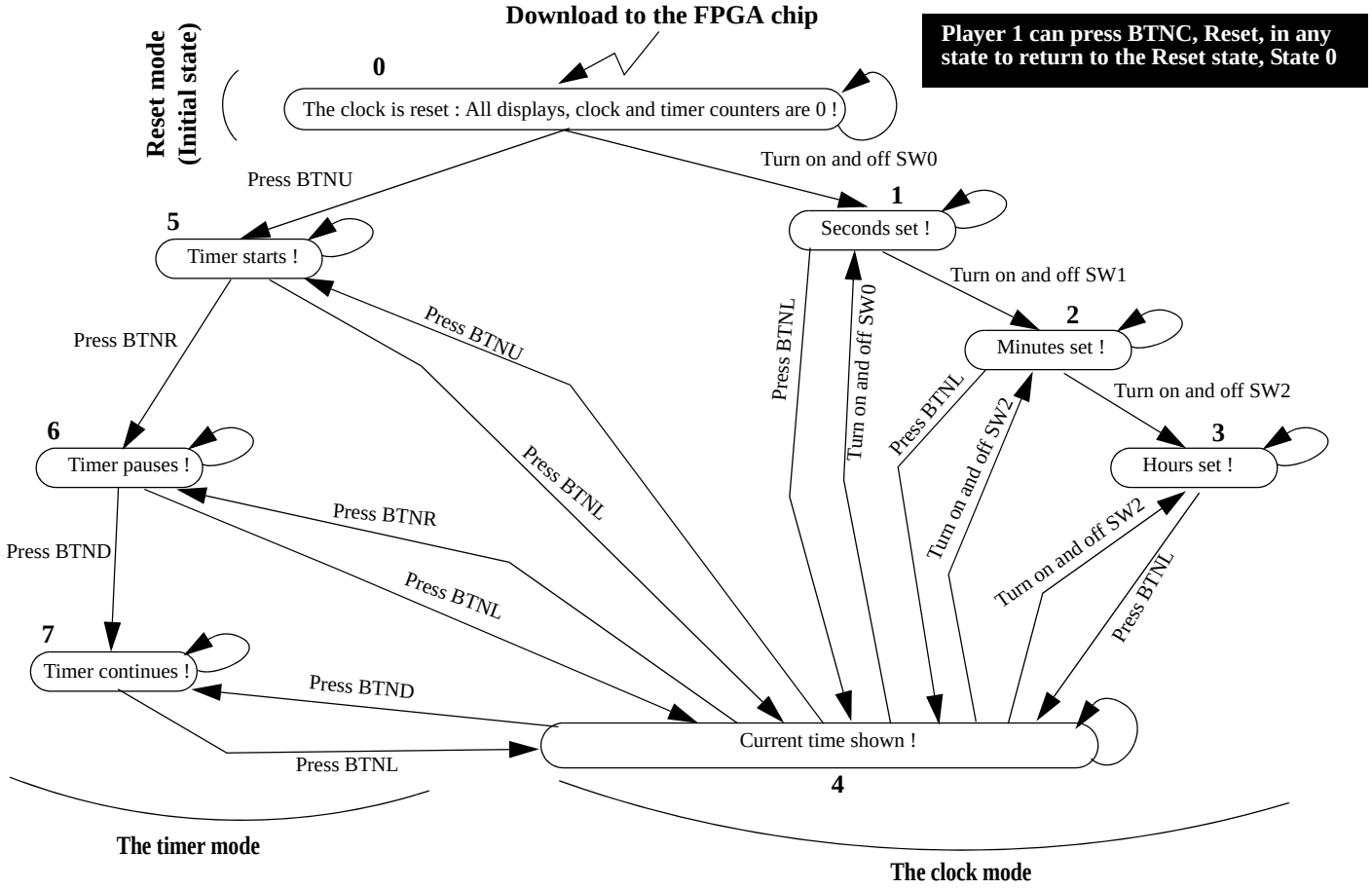

Figure 5. The operation diagram of clock.

At any moment, the current state number is shown on the rightmost three LED lights : LD3 - LD1 (see Figure 3). Thus, if it is the Reset state, the lights show 0. If it is the clock mode, the state is one of 1, 2, 3 or 4. LED15 is on indicating the clock mode. If it is the timer mode, the state is 5 or 5 or 7. LED14 is on, indicating the timer mode. LED 0 blinks at 1 Hz. That is, it blinks once a second.

2.3. Clock Black box Partitioning

We know how to operate the clock and the black-box view. The clock has 17 inputs, 21 outputs and is **sequential**. Because of a large number of inputs, it has to be partitioned into blocks. How can we go about partitioning it ? We have experimented with partitioning complex sequential circuits into blocks : We obtain a list of major operations it performs by studying the operation diagram and also consider the design goals and technology available. Then, we create a block for each major operation. However, thinking of these major operations is not simple. It requires considerable logic design experience. Also, the list of major operations is not fixed, two people can come up with two different lists and therefore two different block partitionings.

By studying the clock operation diagram in Figure 5, we determine the major operations. For example, states 1, 2, 3 and 4, the current time is worked on. Then, we must have a block for keeping track of the time and setting the time. This is the clock major operation. In states 5, 6 and 7 we use the system as a timer, pausing it and resuming it. Then, we need another block as the timer block. This is the timer major operation. A major operation not explicit in the operation diagram, but needed is interfacing the clock to the I/O devices on the FPGA board (switches, displays, etc.). So, we need a major operation and so a block for that : The Input/Output block.

Will we have just three blocks ? No ! We need a way to make sure the clock proceeds as state 0, state 1, state 2, etc. so that the right operation happens at the right time : A **controller** is needed to keep track of the states and indicate what to do when. We will then have one more major operation, the controlling operation and a block, the Control Unit Block. Then, overall, we have four major operations and so four blocks in our clock digital system (Figure 6).

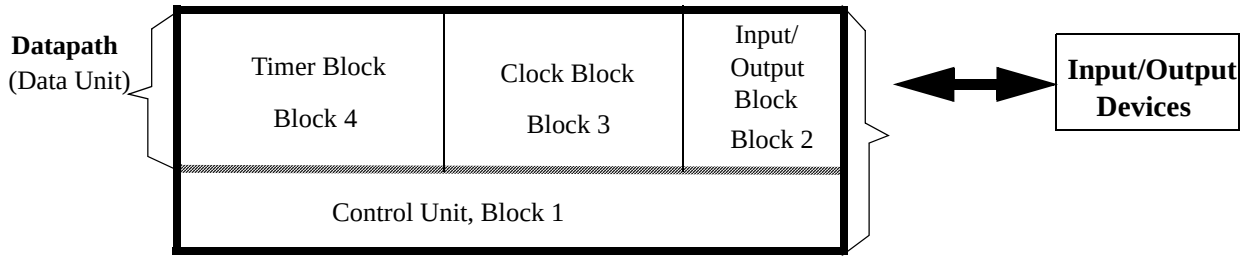

Figure 6. Block partitioning of the clock.

In general, if a circuit has a Control Unit, then it is a **digital system**, since a digital system performs a series of operations and needs to have a circuit to keep track of operations and determine what is to be done next, based on inputs and past operations. This circuit is the Control Unit. So, **the clock is a digital system** that keeps track of time according to the user push button presses and switch operations.

In digital system terminology, all the blocks other than the Control Unit are placed in a super block called **datapath** (Data Unit). Then, the first partitioning of **any** digital system always results in a Control Unit and a Data Unit. The partitioning of the Data Unit follows it and is based on major operations it has to perform. These major operations are the ones we specify above : Interacting with input/output devices on the FPGA board, setting and keeping track of the current time, starting, pausing and resuming a timer.