ADDIS ABABA
SCIENCE AND
TECHNOLOGY
UNIVERSITY
UNIVERSITY FOR INDUSTRY

COLLEGE OF ELECTRICAL AND MECHANICAL ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Course Title: - Digital Logic Design

Presentation Pointer

Group members

1. Netsanet Tesfaye	1300/14
2. Rakeb Yared	1336/14
3. Rediet Bekele	1346/14
4. Rediet Sileshi	1344/14
5. Ruth Shimels	1385/14
6. Ruth Akalu	1377/14

ABSTRACT

This project is about designing and simulating a presentation pointer using Multisim. The pointer will have a timer, slide navigation, and laser pointer activation. We will create a digital circuit using basic logic gates. With Multisim's simulation tools, we have designed, tested, and improved the circuit to meet all needed functions.

We start by creating the core circuit diagrams in detail, then move on to connecting and testing them in Multisim. We will integrate features like timing and wireless control using simple logic gates like (AND, OR, XOR, NOT gates) to more complex combinational logics (comparator, d latch, jk flip flop) to show how digital logic design works in practice.

Throughout the project, we will keep detailed records of each step, from the first design to final testing. We plan on recording documentation which will serve as a guide and a record of the design process.

The goal is to create a working virtual model of a presentation pointer that can later be made into a physical device. This device will make presentations more efficient and effective by offering easy-to-use slide navigation and timing features. Along with a built in timer that sets a buzzer at a given set time to avoid going over our give time.

WORKING MECHANISM

We divided our circuit into 5 core parts when integrated together will achieve the desired pointer. We will go through each one and show the derivation of their logic;

1. BASIC CIRCUIT

Composed of 3 inputs that are connected together with logic gates such as AND, OR, NOT Gates: used for processing these input signals from buttons (next slide, previous slide, LED on/off). Truth table used:

Timer	bt1	bt2	bt3	next	pre	flash
0	0	0	0	0	0	0
0	0	0	1	1	0	0
0	0	1	0	0	0	1
0	1	0	0	0	1	0
1	bt1	bt2	bt3	0	0	0

Logic: Next: b1’.b2’.b3 Previous: b1.b2’.b3’ Flash: b1’.b2.b3’

Notice here that if our Timer is on the buttons won't be able to function as per stated. This is because the Timer on the buttons has another designated function which we will see in the next section.

2, TIMER CIRCUIT

The timer circuit help us set the desired time for the buzzer to go off. In order to do that the Timer button has to be on. Now the three buttons which we used previously as next, pre and flash will consequently act as 10, 15 and 20 min respectively. We can used each buttons individually or together to set a time of the sum of the activated buttons. To achieve such a function we used the following button.

when button1 and button2 are activated, the set time = 10 min + 15 min = 25 min total

converting this into a binary number we get: 11001. In the same manner the every possible combination of buttons are decoded like this:

Bt1 (10 min)	Bt2 (15 min)	Bt3 (20 min)	y0	y1	y2	y3	y4	y5
0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	1	0
0	1	0	1	1	1	1	0	0
0	1	1	1	1	0	0	0	1

1	0	0	0	1	0	1	0	0
1	0	1	0	1	1	1	1	0
1	1	0	1	0	0	1	1	0
1	1	1	1	0	1	1	0	1

Using Kmap we have evaluated the logic of each output to be

y0 = B	y2 = B'C + AC + A'BC'	y4 = B'C + ABC'
y1 = AB' + A'B = A XOR B	y3 = A + BC'	y5 = BC

You'll see in the file named 'Timer' that these 6 outputs will be input to a d latch to store their data in order to use the buttons like we normally did.

3, D LATCH / SR LATCHES

Is a type of temporary storage device that has two stable states. Latches are similar to flip-flops because they are Bistable devices that can reside in either of two states using a feedback arrangement. Although latches are useful for storing binary information and for the design of asynchronous sequential circuits, they are not practical for use as storage elements in synchronous sequential circuits. The basic difference between latches and flip-flops is the way in which they are changed from one state to the other. However, Latches are the building blocks of flip-flops.

Six d latches are used to store the 6 bits resting our time. The reset input is activated with the Timer button so unless the Timer is on no data will be saved into our latches.

4. 6 BIT JK FLIP FLOP

6-bit JK flip-flop counters can be used to generate timing signals, control sequences, and count events The 6-bit size provides a counter range of 0 to 63, which is useful for many practical applications.

However this flip flop has a twist. The set and reset sides of the ff are also connected. The set has a constant value of 0 which will set the value of all our flip flops to zero if the reset is given the value of 1. In this project we use this connection to set the value of our counter to zero once the desired min has been reached and causes the buzzer to go off at the same time.

5, COMPARATOR CIRCUIT

In this project we used a 74F85N comparator to compare 6 bits of our time set and the counter. Once the we gate an equal value the buzzer is activated along with the reset input of the flip flop to reset the values of all the flip flop to zero, as stated above.

We made a slight adjustment to the buzzer circuit to avoid the buzzer going off when the time set is zero and the timer hasn't started yet. In this case the buzzer continuously goes off causing the buzzer to not work perfectly. To avoid this complication we used this logic and came up with the right circuit:

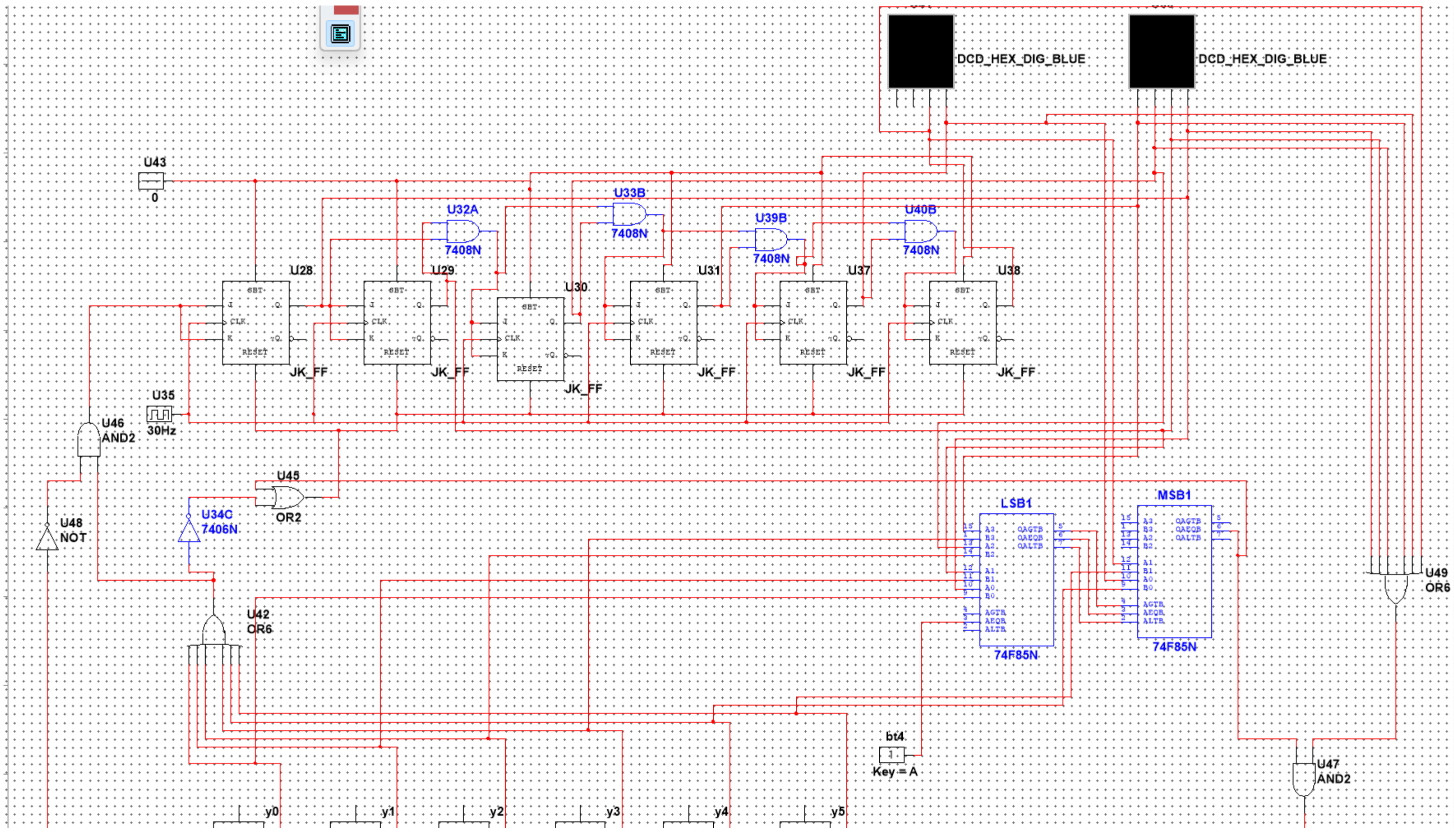
is equal	all zero	buzzer
1	0	0
1	1	1
0	0	0
0	1	0

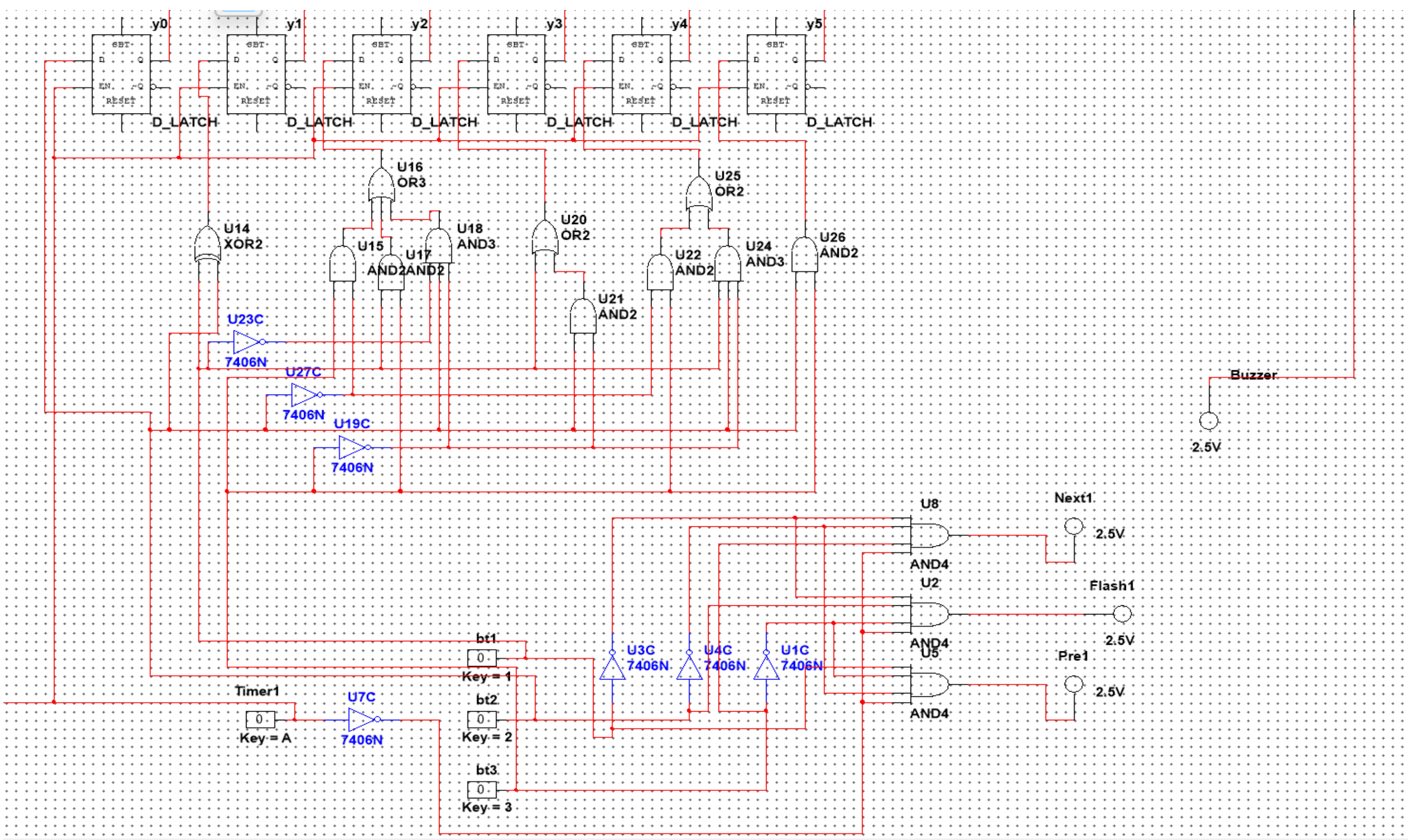
Is equal : the comparator is high because the time set matches the counter
all zero : an OR gate connecting the all the bits of the counter
Logic : AB

Circuit connecting the Timer button with the and the flip flop
In order to set and save the value of our time the Timer button has to be on since it controls the reset input of the latches. However the counter won't start until the Timer is back off, this will give us the chance to change our time. To achieve this we used the following logic and came up with the right circuit:

Timer	Value	Flipflop
0	0	0
0	1	1
1	0	0
1	1	0

Value: an OR gate connecting all the bits of the timer, to see if any time has been set.
Logic : A’B


DCD_HEX_DIG_BLUE
DCD_HEX_DIG_BLUE
U43
0
U32A
7408N
U33B
7408N
U39B
7408N
U40B
7408N
U28
U29
U30
U31
U37
U38
JK_FF
JK_FF
JK_FF
JK_FF
JK_FF
JK_FF
U35
30Hz
U46
AND2
U45
OR2
U34C
7406N
U48
NOT
U42
OR6
LSB1
74F85N
MSB1
74F85N
U49
OR6
bt4
Key = A
U47
AND2
y0
y1
y2
y3
y4
y5


y0
y1
y2
y3
y4
y5
D_LATCH
D_LATCH
D_LATCH
D_LATCH
D_LATCH
D_LATCH
U16
OR3
U25
OR2
U14
XOR2
U15
U17
AND2AND2
U18
AND3
U20
OR2
U22
AND2
U24
AND3
U26
AND2
U21
AND2
U23C
7406N
U27C
7406N
U19C
7406N
Buzzer
2.5V
U8
AND4
Next1
2.5V
U2
AND4
Flash1
2.5V
U5
AND4
Pre1
2.5V
U3C
7406N
U4C
7406N
U1C
7406N
bt1
Key = 1
bt2
Key = 2
bt3
Key = 3
Timer1
Key = A
U7C
7406N