EasyChair Preprint

№ 5317

A Logic Circuit Simulation on Marriage Problem Predicate with Timing Diagrams

Frank Appiah

June 7, 2021

A LOGIC CIRCUIT SIMULATION ON MARRIAGE PROBLEM PREDICATE WITH TIMING DIAGRAMS.

Frank Appiah

Abstract—This is about logic circuits on the digital logic studies on Marriage Problem Predicate with its defined digital set over [0,1] values. The timing diagrams of the designed circuits are shown. Again is a look at different input simulations and their circuit simulations.

Index Terms—digital circuit, logic circuit, design, simulation, truth table, timing diagrams.

• *Frank Appiah is with King's College London, King's Engineer Group, UK. E-mail: appiahnsiahfrank@gmail.com*

1 Introduction

In answering queries whether a particular predicate is a Digital-1 or not. The next predicate is to determine if a sentence is a question or not. There is only one question in all the five sentences. It is represented as mpsentenceask predicate sentence. This category predicate is important in this work.
This will take on two passing values of sentence number and an indicator of a question or not. 1 indicates a pass value whiles 0 does not. The following question stances are:

1. mpsentenceask(1, no).
2. mpsentenceask(2, yes).
3. mpsentenceask(3, no).
4. mpsentenceask(4, no).
5. mpsentenceask(5, no).

General Predicate : mpsentenceask (sentence _no, response).

In generating a set for mpsentenceask(named as MbA) , It will give:

MbA={0, 1, 0, 0, 0}.

MbA is a binary set. The number of words of a sentence is now represented with mpwordsize predicate
sentences. . The following details are as follows :
1. mpwordsize(1, 2).
2. mpwordsize(2, 2).
3. mpwordsize(3, 6).
4. mpwordsize(4, 5).
5. mpwordsize(5, 5).

This category predicate is important in this work. The set theoretic form is represented as :

MbWs={1, 1, 1, 1, 1}.

This predicate took its arguments to be the sentence number and the number of words.
General predicate is represented as:
General Predicate : mpwordsize (sentence_no, word_number).
Further details on negation sentences are looked at. This will have the predicate sentence,
mpnegation. This is explicitly sentences with a not word.
The problem solution are as follows :
1. mpnegation(1, no).
2. mpnegation(2, no).
3. mpnegation(3,no).
4. mpnegation(4, yes).
5. mpnegation(5, yes).

General Predicate : mpnegation (sentence _no, response).

The set representation of Mpnegation is

MbNg={0, 0, 0, 1, 1}.

MP example has only two negation statements in total. Statements like "damn it" creates a feeling of regret or disappointment. What's wrong did create sudden worry but does not bring the negation that is not interesting. The

predicate sentence is represented as mpregret.
These are as follows :

1. mpregret(1, yes).
2. mpregret(2, no).
3. mpregret(3, no).
4. mpregret(4, no).
5. mpregret(5, no).

General Predicate : mpregret (sentence _no, response).

The set theoretical form is given by:

MbR={1, 0, 0, 0, 0}

mpworry is the predicate sentence for sudden worry. These includes the following :

- mpworry(1, no).
- mpworry(2, yes).
- mpworry(3, no).
- mpworry(4, no).
- mpworry(5, no).

General Predicate : mpworry (sentence _no, response).

The set theoretical form is given by:

MbW={0, 1, 0, 0, 0}.

The problem solver took on statement 3 to bring out an approach. The predicate for this will be mpsolver. The knowledge needed to be programmed are as follows:

1. mpsolver(1, no).
2. mpsolver(2, no).
3. mpsolver(3, yes).
4. mpsolver(4, no).
5. mpsolver(5, no).

General Predicate : mpsolver (sentence _no, response).

The set theoretical form is given by:

MbS={0, 0, 1, 0, 0}.

The third round tried to bring out a solution in the context of problem solving. The 4 and 5 statements are involved with names of female sex. These are Akua and Pokua. The fact base for this representation is captured with predicate sentences, mpnamsex. These will
include the following :

- mpnamsex(1, no).
- mpnamsex(2, no).
- mpnamsex(3, no).
- mpnamsex(4, yes).
- mpnamsex(5, yes).

General Predicate : mpnamsex (sentence _no, response).

The set theoretical form is given by:

MbX={0, 0, 0, 1, 1}.

The following are used in forming binary set on tabular representation :
MbA={0, 1, 0, 0, 0}.
MbX={0, 0, 0, 1, 1}.
MbS={0, 0, 1, 0, 0}.
MbR={1, 0, 0, 0, 0}
MbW={0, 1, 0, 0, 0}.
MbNg={0, 0, 0, 1, 1}.

Tabular Representations On Binary Set
Table 1.

B	MbA	MbNg	MbR	MbW	MbX	MbS
N1	0	0	1	0	0	0
N2	1	0	0	1	0	0
N3	0	0	0	0	0	1
N4	0	1	0	0	1	0
N5	0	1	0	0	1	0

The parameters in the table of binary set with any pair forms a Digital set.

For example Digital set (MbA, MbNg) will show the following :

Digital set (MbA, MbNg)={(0,0), (1,0), (0,0), (0,1), (0,1)}.

Truth Table 2.

The binary logic will now be treated on the above Digital set.

MbA	MbNg	And	Or	Xor
0	0	0	0	0
1	0	0	1	1
0	0	0	0	0
0	1	0	1	1
0	1	0	1	1

The Digital logic((MbA, MbNg)=(and,or,xor)) ={(0,0)=(0,0,0), (1,0)=(0,1,1), (0,1)=(0,1,1)}

For example Digital set (MbA, MbR) will show the following :

set (MbA, MbR)={(0,1), (0,0), (0,0), (0,0), (0,0)}.

The binary logic will now be operator treated on the above Digital set.

Truth Table 3.

MbA	MbR	And	Or	Xor
0	1	0	1	1
1	0	0	1	1
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

The Digital logic((MbA, MbR)=(and,or,xor)) ={(0,1)=(0,1,1), (1,0)=(0,1,1), (0,0)=(0,0,0)}

For example Digital set (MbA, MbW) will show the following :

Digital set (MbA, MbW)={(0,0), (0,1), (0,0), (0,0), (0,0)}.

The binary logic will now be operator treated on the above Digital set.

Truth Table 4.

MbA	MbW	And	Or	Xor
0	0	0	0	0
1	1	1	1	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

The Digital logic((MbA, MbR)=(and,or,xor)) ={(0,0)=(0,0,0), (1,1)=(1,1,0)}

For example Digital set (MbA, MbX) will show the following :

Digital set (MbA, MbX)={(0,0), (0,0), (0,0), (0,1), (0,1)}.

The binary logic will now be operator treated on the above Digital set.

Truth Table 5.

MbA	MbX	And	Or	Xor
0	0	0	0	0
1	0	0	1	1
0	0	0	0	0
0	1	0	1	1
0	1	0	1	1

The Digital logic((MbA, MbR)=(and,or,xor)) ={(0,0)=(0,0,0), (1,0)=(0,1,1),(0,1)=(0,1,1)}

For example Digital set (MbA, MbS) will show the following :

Digital set (MbA, MbS)={(0,0), (0,0), (0,1), (0,0), (0,0)}.

The binary logic will now be operator treated on the above Digital set.

Truth Table 6.

MbA	MbS	And	Or	Xor
0	0	0	0	0
1	0	0	1	1
0	1	0	1	1
0	0	0	0	0
0	0	0	0	0

The Digital logic((MbA, MbS)=(and,or,xor)) ={(0,0)=(0,0,0), (1,0)=(0,1,1),(0,1)=(0,1,1)}

For example Digital set (MbNg, MbR) will show the following :

Digital set (MbNg, MbR)={(0,1), (0,0), (0,0), (1,0), (1,0)

MbNg	MbR	And	Or	Xor
0	1	0	1	1
0	0	0	0	0
0	0	0	0	0
1	0	0	1	1
1	0	0	1	1

Truth Table 7.

Digital logic((MbNg, MbR)=(and,or,xor)={(0,1)=(0,1,1), (1,0)=(0,1,1)}

For example Digital set (MbNg, MbW) will show the following :

Digital set (MbNg, MbW)={(0,0), (0,1), (0,0), (1,0), (1,0)}.

Truth Table 8.

MbNg	MbW	And	Or	Xor
0	0	0	0	0
0	1	0	1	1
0	0	0	0	0
1	0	0	1	1
1	0	0	1	1

Digital logic((MbNg, MbW)=(and,or,xor)={(0,1)=(0,1,1), (1,0)=(0,1,1)}

For example Digital set (MbNg, MbX) will show the following :

Digital set (MbNg, MbW)={(0,0), (0,0), (0,0), (1,1), (1,1)}.

Truth Table 9.

MbNg	MbX	And	Or	Xor
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1	1	1	1	0
1	1	1	1	0

Digital logic((MbNg, MbX)=(and,or,xor)={(1,1)=(1,1,0)}

For example Digital set (MbNg, MbS) will show the following :

Digital set (MbNg, MbS)={(0,0), (0,0), (0,1), (1,0), (1,0)}.

Truth Table 10.

MbNg	MbS	And	Or	Xor
0	0	0	0	0
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	0	0	1	1

Digital logic((MbNg, MbX)=(and,or,xor)={(0,1)=(0,1,1), (1,0)=(0,1,1)}

For example Digital set (MbR, MbW) will show the following :

Digital set (MbR, MbW)={(1,0), (0,1), (0,0), (0,0), (0,0)}

Truth Table 11..

MbR	MbW	And	Or	Xor
1	0	0	1	1
0	1	0	1	1
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Digital logic((MbNg, MbX)=(and,or,xor)={(1,0)=(0,1,1), (0,1)=(0,1,1)}

For example Digital set (MbR, MbS) will show the following :

Digital set (MbR, MbS)={(1,0), (0,0), (0,1), (0,0), (0,0)}.

Truth Table 12.

MbR	MbS	And	Or	Xor
1	0	0	1	1
0	0	0	0	0
0	1	0	1	1
0	0	0	0	0
0	0	0	0	0

Digital Logic ((MbR,MbS)=(and,or,xor)={(1,0)=(0,1,1), (0,1)=(0,1,1)}

This research is organised as follows :

1. Review on the digital logic studies on Marriage Problem Predicate Task
2. Design logic circuits based on the 8 or more digital set.
3. Optimise on design techniques by using less logic elements.
4. Identify similar features of each circuit.
5. Produce timing diagrams on circuit simulation.
6. Stimulate each circuit on different input digital values.
7.Trace each output by line.

2 LOGIC CIRCUITS DESIGN AND TIMING DIAGRAMS

The design of logic circuit was done with Logic Circuit Sim Professional. This is not the full version. It starts by creating a project with the tool. Here, I am actually using a Android mobile version. Two projects are created as shown below:

Figure 1

In the pictorials, you will see the number of logic elements, number of connections and date of modifications. The project names are MbAMbNg and MbAMbW.

I will describe the design of MbAMbW circuit that has same name as the project. The variables are MbA and MbW.

Logic Function :
(1) *MbA.MbW*
(2)*MbA+MbW*

This is deduced from table 4.

Circuit Design of MbAMbW

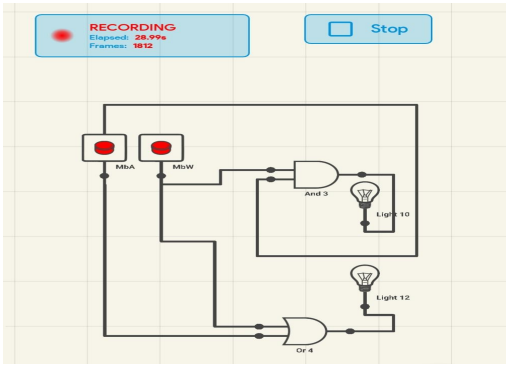


Figure 2

Timing Diagram of MbAMbW

Timing is done by recording the entry connections entered by selecting the points.

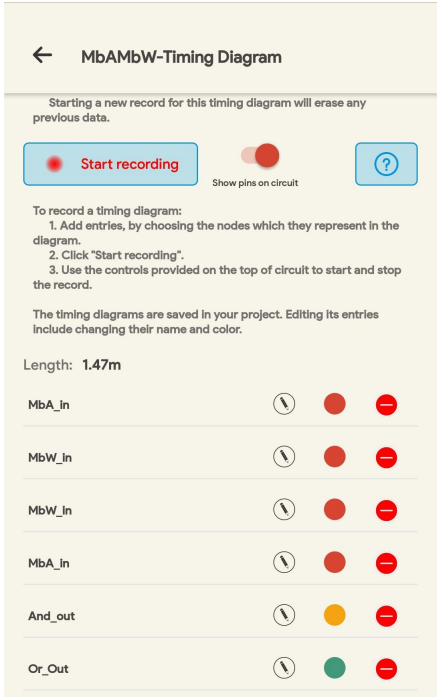


Figure 3

This in particular has 1.47 minutes of recording activity with the off/on Switch.

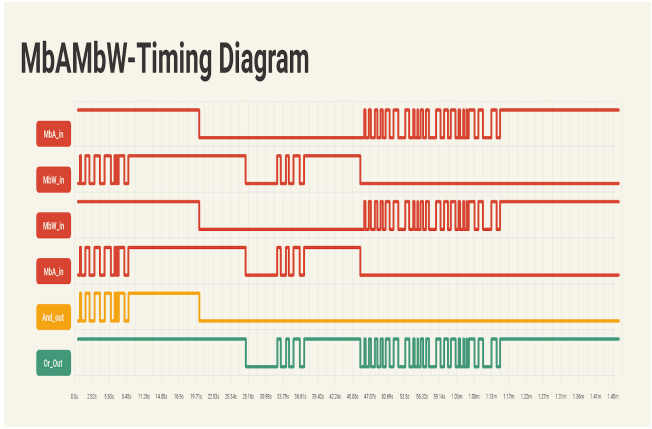


Figure 4

The above diagram is for each entry connection selected. The and function has timing diagram And_out. The or function has timing diagram, or_Out. :

Trace Outputs and Input Simulation

MbA=1, MbW=0. 1 is button pushdown activity whilst 0 is not. The record of states is as shown :

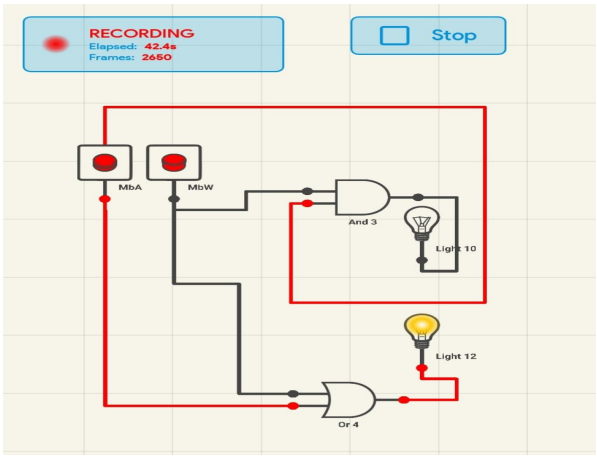


Figure 5

With the button MbA pushdown and MbW button not pressed, the light goes on. The trace of activity are shown with the red lines and no activity with black lines.

MbA=0, MbW=1. 1 is button pushdown activity whilst 0 is not. The record of states is as shown :

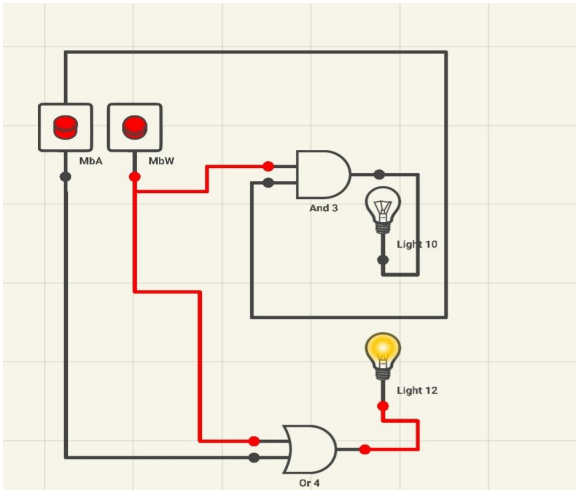


Figure 6

With the button MbW pushdown and MbA button not pressed, the light goes on.

MbA=1, MbW=1. 1 is button pushdown activity. With the button MbW pushdown and MbA button pushdown , the light goes on. The record of states is as shown :

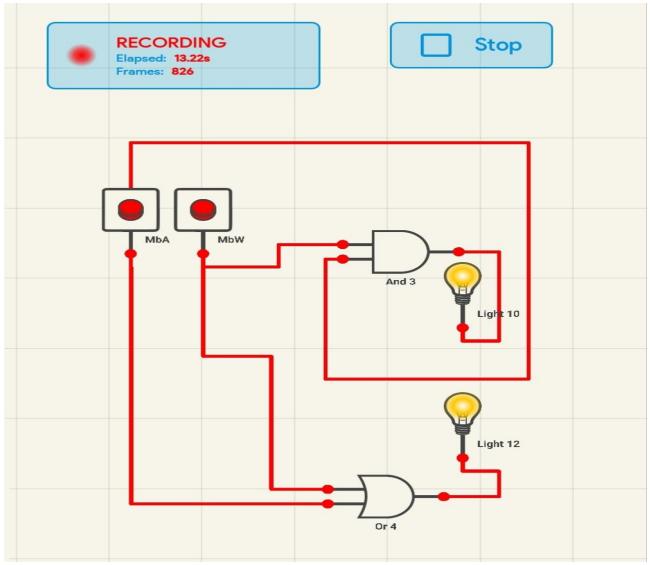


Figure 7

The next is based on table 2.

With variables MbA and MbNg,

The logic functions are:

(1) $\overline{MbA}.\ MbNg + MbA.\overline{MbNg}$

(2) $MbA.\ \overline{MbNg} + \overline{MbA}.MbNg$

CIRCUIT Design of MbAMbNg

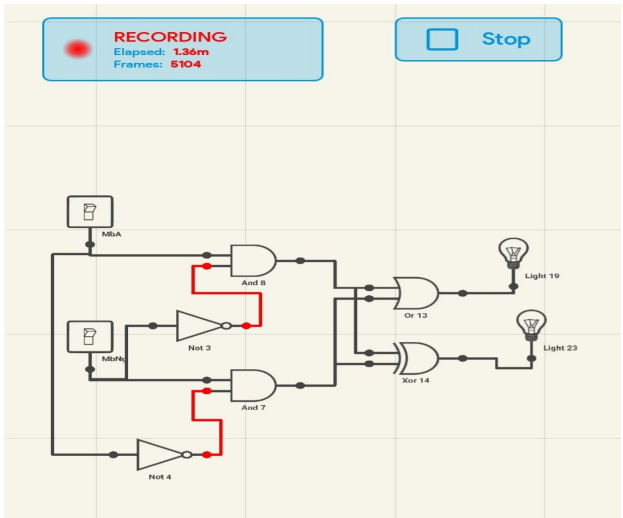


Figure 8

Timing Diagram on MbAMbNg

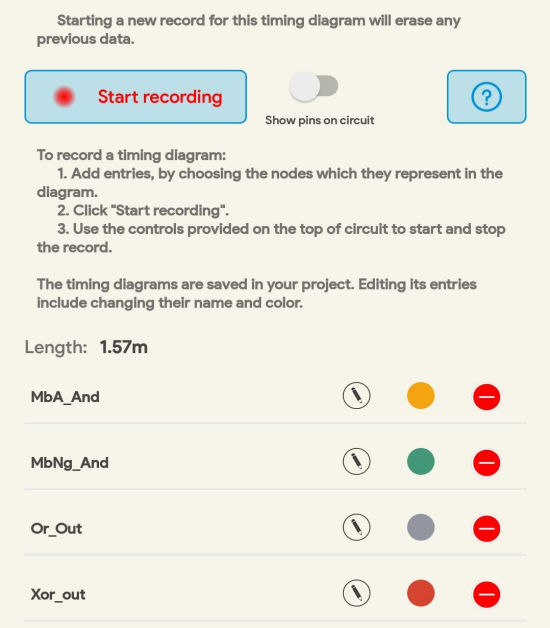
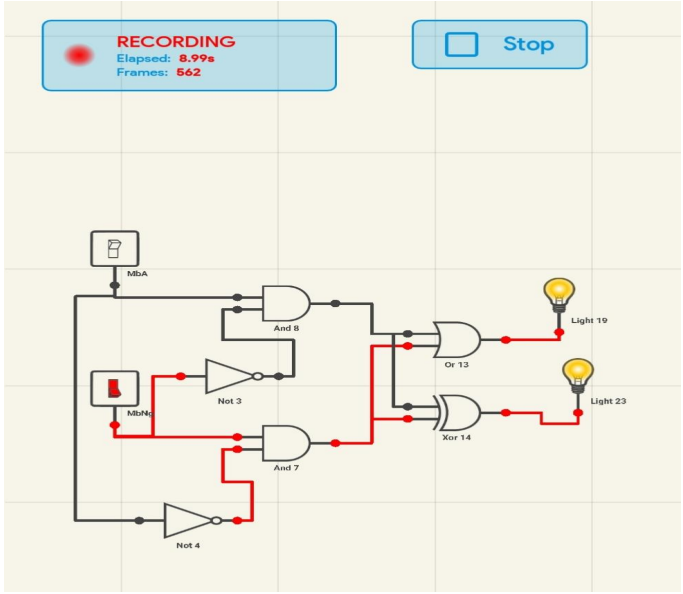


Figure 9.

The connection points are labelled as MbA_And and MbNg_And. Or_Out and Xor_out are both outputs.



Fugure 10.

MbA=1 and MbNg =0. 1 is an on switch whilst 0 is an off switch. With MbA switch on and MbNg switch off, both output lights on.
The switch MbNg causes the lines or wires to show as red to indicate activation.
It is as shown below :

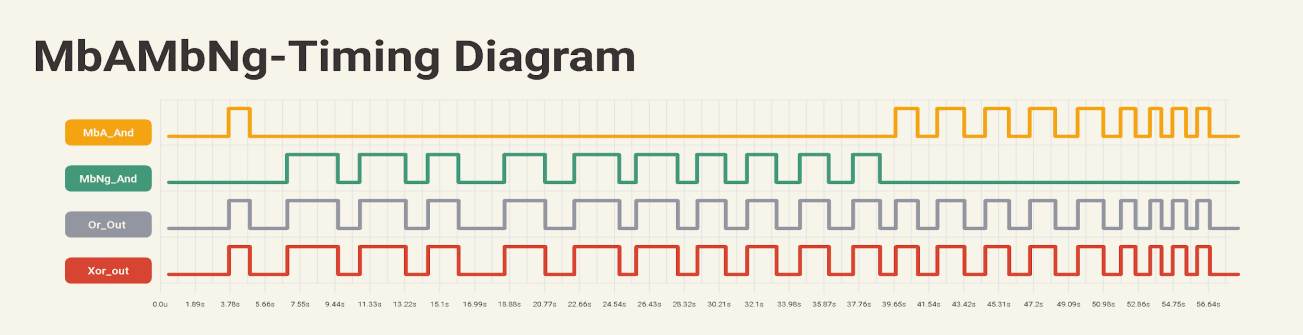


Figure 11.

Trace Output and Input Simulation

MbA=0 and MbNg =1. 1 is an on switch whilst 0 is an off switch. With MbA switch off and MbNg switch on, both output lights on.
The switch MbNg causes the lines or wires to show as red to indicate activation.

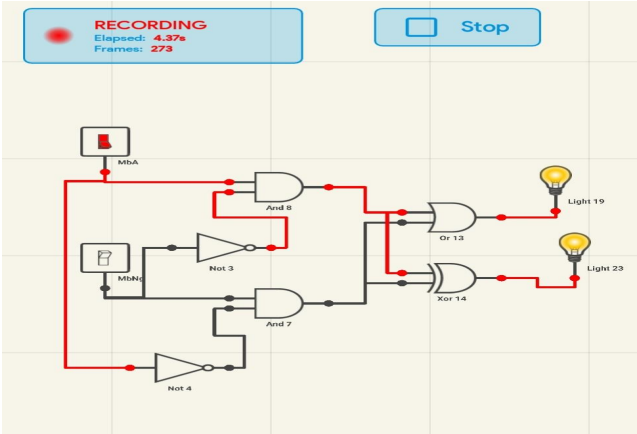


Figure 12

MbA=1 and MbNg =1. 1 is an on switch whilst 0 is an off switch. With MbA switch off and MbNg switch off, both output lights off.
The switch MbNg causes the lines or wires to show as red to indicate activation.

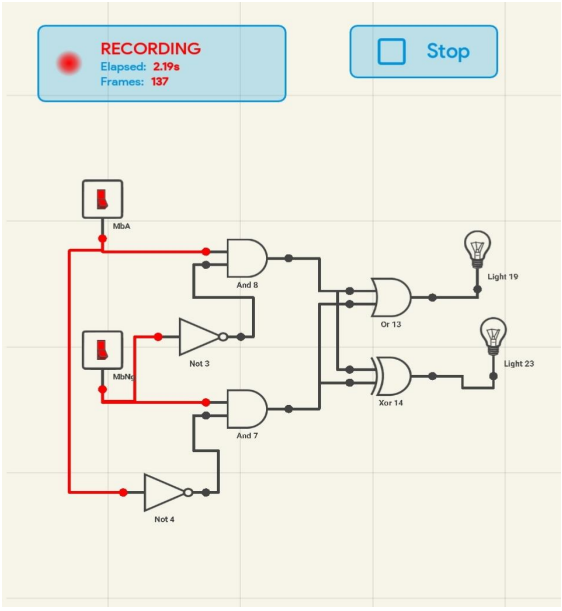


Figure 13.

These tables have similar logic circuit to Table 2 *(MbAMbNg* Circuit Design) namely:

- Table 3
- Table 5
- Table 6
- Table 7
- Table 8
- Table 9
- Table 10
- Table 11
- Table 12.

These tables have no similar logic circuit to Table 4(MbAMbW Circuit Design).

3 Figures

3.1 Appendices

```
Figure 1          Figure 8
Figure 2.         Figure 9
Figure 3.         Figure 10
Figure 4.         Figure 11
Figure 5.         Figure 12
Figure 6.         Figure 13
Figure 7
```

4 Conclusion

This research looks mainly at logic circuit design based on predicate task involving a Marriage Problem [2,3]. The logic circuits are stimulated to make different scenarios of simulation. Finally, a view on timing diagrams.

References

[1] Vingron, S. P. (2012). Logic circuit design: Selected methods. Springer Science & Business Media.
[2] Appiah Frank. Letter Combinatorics : Theory on counting problems. EPSRC UK TuringAI Letter. 2020
[3] Appiah Frank. Letter Combinatorics : Theory on counting problems. Marriage Problem. Mendeley Publication. 2020
[4] Fisler, K. (1999). Timing diagrams: Formalization and algorithmic verification. Journal of Logic, Language and Information, 8(3), 323-361.
[5] Appiah, F. (2021). Digital Logic on Marriage Problem Predicate Task (No. 5317). EasyChair.
[6] Gates, S. C. (2004). Gate Design.

Dr. Frank Appiah. He is a holder of Bsc(Hon) from Kwame Nkrumah University of Science and Technology in 2018, Msc in Advanced Software Engineering from King's college London in 2010 and PhD in computer science and engineering from both KCL (2012/2014) and KNUST (2014) respectively. Frank Appiah has professional certificates in Management and engineering since 2011 from IEEE. He developed StreamEPS - Stream Event Processing System in 2011 which is hosted at Github.