

**Problem 1:**

**Computer Science or Information Technology**

Instructor: Dr. G.E. Antoniou

Day, Month, Year

Day

CSIT 502

Department of CSIT

Assessment

Module-3

Hidalgo, Rafael

A heart rate monitor measures an individual’s heart rate and blood pressure. Both sensors output zero (0) if they are within safety range. An alarm will sound if either sensor indicates an unsafe condition is present. Set–up the appropriate truth table, simplify (minimally) using K-maps. Implement, using LogiSim, the simplified logic circuit with optimal number of logic gates.

**Solution**

Set up the truth table for the problem:

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **X** |
| **0** | **0** | **0** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **1** |

For the output the corresponding logic expression is written as follows:

**X = A’B + A’B + AB**

Now we must simplify X using K-Map. Set up the 2-variable K-map table as follows:

|  |  |  |
| --- | --- | --- |
| **A\B** | **0** | **1** |
| **0** | **0** | **1** |
| **1** | **1** | **1** |

K-map looping:

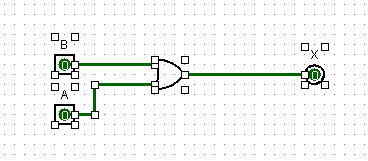
|  |  |  |
| --- | --- | --- |
| **A\B** | **0** | **1** |
| **0** | **0** | **1** |
| **1** | **1** | **1** |

Therefore the simplified expression for **X** is: **A + B**

Therefore, we have the following logic equation to implement with basic logic gates:

**X = A+B**

**Logic circuit (LogiSim)**



The above logisim circuit is running according to the problem specifications.

**Problem 2:**

Three light-emitting diodes (LEDs) [one Red, one Green, one Blue] turn on when a number 0–7 is passed through. Red turns on with even numbers, green turns on with odd numbers, blue turns on with multiples of 3. Zero means they are all off, seven means they are all on. Set–up the appropriate truth table, simplify (minimally) using K-maps. Implement, using LogiSim, the simplified logic circuit with optimal number of logic gates.

**Solution**

Set up the truth table for the problem:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **R** | **G** | **B** |
| **0** | **0** | **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **0** | **1** | **0** |
| **0** | **1** | **0** | **1** | **0** | **0** |
| **0** | **1** | **1** | **0** | **1** | **1** |
| **1** | **0** | **0** | **1** | **0** | **0** |
| **1** | **0** | **1** | **0** | **1** | **0** |
| **1** | **1** | **0** | **1** | **0** | **1** |
| **1** | **1** | **1** | **1** | **1** | **1** |

For the output the corresponding logic expression is written as follows:

**R = A’BC’ + AB’C’ + ABC‘ + ABC**

**G= A’B’C + A’BC + AB’C + ABC**

**B= ABC + ABC + ABC + ABC**

We must simplify, using K-Map, the output R, G, and B. Set up the 3-variable K-map table as follows:

K-Map for R

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A\B,C** | **00** | **01** | **11** | **10** |
| **0** | **0** | **0** | **0** | **1** |
| **1** | **1** | **0** | **1** | **1** |

K-map looping for R:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A\B,C** | **00** | **01** | **11** | **10** |
| **0** | **0** | **0** | **0** | **1** |
| **1** | **1** | **0** | **1** | **1** |

Simplified expression for R: AB + AC’ + BC’

K-Map for G:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A\B,C** | **00** | **01** | **11** | **10** |
| **0** | **0** | **1** | **1** | **0** |
| **1** | **0** | **1** | **1** | **0** |

K-Map looping for G:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A\B,C** | **00** | **01** | **11** | **10** |
| **0** | **0** | **1** | **1** | **0** |
| **1** | **0** | **1** | **1** | **0** |

Simplified expression for G: C

K-Map for B:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A\B,C** | **00** | **01** | **11** | **10** |
| **0** | **0** | **0** | **1** | **0** |
| **1** | **0** | **0** | **1** | **1** |

K-Map looping for B

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A\B,C** | **00** | **01** | **11** | **10** |
| **0** | **0** | **0** | **1** | **0** |
| **1** | **0** | **0** | **1** | **1** |

Simplified Expression for B: BC + AB

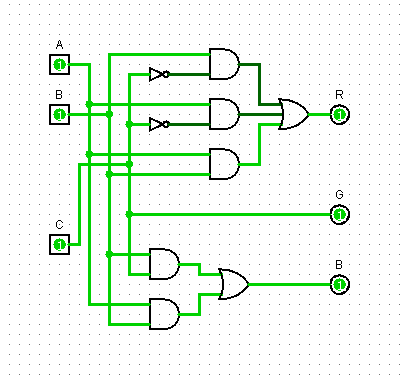
Therefore, we have the following logic equations to implement with basic logic gates:

**R =** AB + AC’ + BC’

**G = C**

**B =** BC + AB

**Logic circuit (LogiSim)**



The above logisim circuit is running according to the problem specifications.

**Problem 3:**

A teacher is grading the students in 4 subjects (Math, Spelling, English, and History) to see whether or not they will graduate. If a student passes Math and Spelling, they will graduate. If a student passes either English or History, they will graduate. All other students will not graduate. Set–up the appropriate truth table, simplify (minimally) using K-maps. Implement, using LogiSim, the simplified logic circuit with optimal number of logic gates.

**Solution**

Set up the truth table for the problem:

A= Math (1 = Pass, 0 = not Pass)

B= Spelling (1 = Pass, 0 = not Pass)

C= English (1 = Pass, 0 = not Pass)

D = History (1 = Pass, 0 = not Pass)

G = Graduate (1 = graduate, 0 = not graduate)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **G** |
| **0** | **0** | **0** | **0** | **0** |
| **0** | **0** | **0** | **1** | **1** |
| **0** | **0** | **1** | **0** | **1** |
| **0** | **0** | **1** | **1** | **1** |
| **0** | **1** | **0** | **0** | **0** |
| **0** | **1** | **0** | **1** | **1** |
| **0** | **1** | **1** | **0** | **1** |
| **0** | **1** | **1** | **1** | **1** |
| **1** | **0** | **0** | **0** | **0** |
| **1** | **0** | **0** | **1** | **1** |
| **1** | **0** | **1** | **0** | **1** |
| **1** | **0** | **1** | **1** | **1** |
| **1** | **1** | **0** | **0** | **1** |
| **1** | **1** | **0** | **1** | **1** |
| **1** | **1** | **1** | **0** | **1** |
| **1** | **1** | **1** | **1** | **1** |

For the output the corresponding logic expression is written as follows

**G= A’B’C’D + A’B’CD’+ A’B’CD+ A’BC’D+ A’BCD’+ A’BCD+ AB’C’D+ AB’CD’+ AB’CD+ ABC’D’+ ABC’D+ ABCD’+ ABCD**

We must simplify, using K-Map, the output G. Set up the 4-variable K-map table as follows:

K-Map for R

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A,B\C,D** | **00** | **01** | **11** | **10** |
| **00** | **0** | **1** | **1** | **1** |
| **01** | **0** | **1** | **1** | **1** |
| **11** | **1** | **1** | **1** | **1** |
| **10** | **0** | **1** | **1** | **1** |

K-map looping for G:

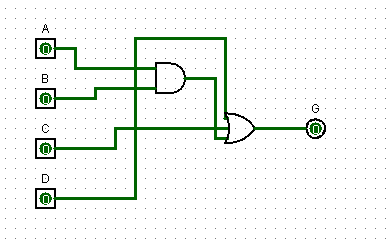
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A,B\C,D** | **00** | **01** | **11** | **10** |
| **00** | **0** | **1** | **1** | **1** |
| **01** | **0** | **1** | **1** | **1** |
| **11** | **1** | **1** | **1** | **1** |
| **10** | **0** | **1** | **1** | **1** |

Simplified expression for G: AB + C + D

Therefore, we have the following logic equations to implement with basic logic gates:

**G = AB + C + D**

**Logic circuit (LogiSim)**



The above logisim circuit is running according to the problem specifications.

**Problem 4:**

Design a digital logic circuit (LookUp Table or LUT) with three inputs, x, y and z, and the three outputs, A, B, and C. When the binary input is 0, 1, 2, or 3, the binary output is 2 greater than the input. When the binary input is 4, 5, 6, or 7, the binary output is 2 less than the input. Set–up the appropriate truth table, simplify (minimally) using K-maps. Implement, using LogiSim, the simplified logic circuit with optimal number of logic gates.

**Solution**

Set up the truth table for the problem:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **x** | **y** | **z** | **A** | **B** | **C** |
| **0** | **0** | **0** | **0** | **1** | **0** |
| **0** | **0** | **1** | **0** | **1** | **1** |
| **0** | **1** | **0** | **1** | **0** | **0** |
| **0** | **1** | **1** | **1** | **0** | **1** |
| **1** | **0** | **0** | **0** | **1** | **0** |
| **1** | **0** | **1** | **0** | **1** | **1** |
| **1** | **1** | **0** | **1** | **0** | **0** |
| **1** | **1** | **1** | **1** | **0** | **1** |

For the output the corresponding logic expression is written as follows

**A= x’yz’ + x’yz + xyz’ + xyz**

**B= x’y’z’ + x’y’z + xy’z’ + xy’z**

**C= x’y’z + x’yz + xy’z + xyz +**

We must simplify, using K-Map, the outputs A,B,C. Set up the 3-variable K-map table as follows:

K-Map for A

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **x\y,z** | **00** | **01** | **11** | **10** |
| **0** | **0** | **0** | **1** | **1** |
| **1** | **0** | **0** | **1** | **1** |

K-map looping for A :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **x\y,z** | **00** | **01** | **11** | **10** |
| **0** | **0** | **0** | **1** | **1** |
| **1** | **0** | **0** | **1** | **1** |

Simplified expression for A: y

K-Map for B

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **x\y,z** | **00** | **01** | **11** | **10** |
| **0** | **1** | **1** | **0** | **0** |
| **1** | **1** | **1** | **0** | **0** |

K-map looping for B :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **x\y,z** | **00** | **01** | **11** | **10** |
| **0** | **1** | **1** | **0** | **0** |
| **1** | **1** | **1** | **0** | **0** |

Simplified expression for B: y’

K-Map for C

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **x\y,z** | **00** | **01** | **11** | **10** |
| **0** | **0.** | **1** | **1** | **0** |
| **1** | **0** | **1** | **1** | **0** |

K-map looping for C :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **x\y,z** | **00** | **01** | **11** | **10** |
| **0** | **0.** | **1** | **1** | **0** |
| **1** | **0** | **1** | **1** | **0** |

Simplified expression for C: z

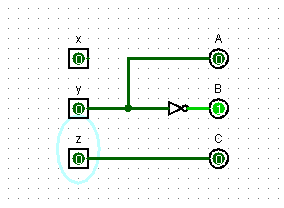
Therefore, we have the following logic equations to implement with basic logic gates:

**A = y**

**B = y’**

**C = z**

**Logic circuit (LogiSim)**



The above logisim circuit is running according to the problem specifications.