Boolean Algebra Lesson Plan

Wednesday 1st February - Y12 & 13

2 Hours including Campus Tour

The aim of this lesson is to introduce Year 12 to boolean algebra simplification and to help year 13 improve their ability to solve these problems.

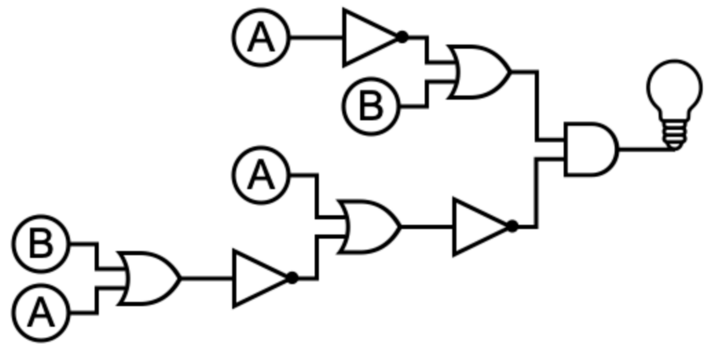
After this lesson students should be able to:
- Understand the reasoning behind the common boolean simplification rules.
- Apply these rules to make expressions simpler.
- Implement these in electronics to understand the reasoning behind boolean algebra and why we use simplification.

2:00	Student's arrive at Imperial We take a short 20-30min tour. Aiming to be back in Huxley225 by 2:30.

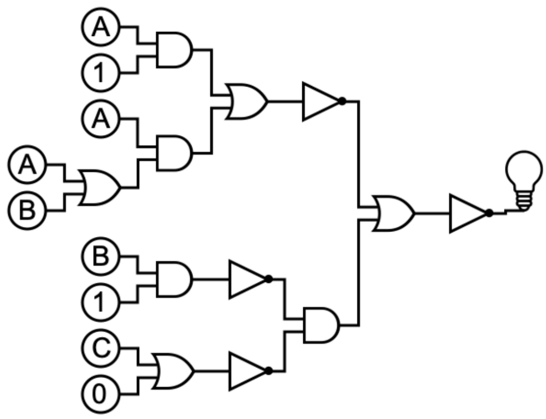
2:30	**Boolean Electronics Introduction** Split the group into pairs. For this activity Year 13's should be with other Year 13's. Year 12's should be paired with another Year 12. Each group will be given one of the following Boolean algebra expressions. They will then need to build the circuit for this logic and fill out a truth table using their circuit:

Year 12 - $(\overline{A}+B).\overline{(A+\overline{B}+A)}$

A	B	Output
0	0	F
0	1	T
1	0	F
1	1	F



Year 13 - $\overline{\overline{((A.1)+(A.(A+B)))}+\overline{(B.1)}.\overline{(C+0)}}$



A	B	C	Output
0	0	0	F
0	0	1	F
0	1	0	F
0	1	1	F
1	0	0	F
1	0	1	T
1	1	0	T
1	1	1	T

2:45	**Introduction to Boolean algebra** A brief presentation going through all the rules of Boolean Algebra. Slides attached

3:00	**Boolean Algebra Game** A simplification game. Splitting the students up into cross year pairs i.e. a year 12 pupil paired with a year 13 pupil. The year 12 is the “writer” and the year 13 is the “runner”. Writer -> Must solve the Boolean expression given by the runner. Runner -> Must deliver the Boolean expression to the teacher for checking and get given the next one. The runners cannot talk or write. They can only communicate with the writer by holding up signs with the boolean rules on them. The team that solves the most in the time available will win a small Imperial prize. This can be adjusted based on time but can have the following ready: 1. $(A\,.\,B) + (A\,.\,\overline{B}) = A$ 2. $\overline{\overline{A}\,.\,\overline{B}} = A + B$ 3. $\overline{A\,.\,B} + A = 1$ 4. $(A\,.\,B\,.\,\overline{C}) + (A\,.\,\overline{C}) = A\,.\,\overline{C}$ 5. $(\overline{A\,.\,B}) + \overline{A\,.\,\overline{B}} = 1$ 6. $\overline{\overline{A} + \overline{B\,.\,A}} = A\,.\,B$ 7. $A\,.\,B\,.\,(A + B) = A\,.\,B$ 8. $\overline{\overline{A} + \overline{B}} + B\,.\,\overline{A} = B$
3:15	**Boolean Programming** Students will split back up into the groups they had for the Boolean Electronics exercise. And log into the computers and do the Collab notebook given. This will help them visualise and try out different Boolean Algebra rules. 1. valid_expression -> Check that a string input only contains letters A, B, C, ., +, ! 2. evaluate -> Given an expression and a value for A, B, C return the boolean result. 3. equivalent -> Check if two boolean expressions are equivalent. They’re equivalent if for each combination of A, B, C then they evaluate to the same. 4. generate_truth_table -> Generate a truth table for the expression given. 5. swap_algorithm -> Allow a user to input an identity. Check that its a correct identity and if so then swap all occurrences of the left with the right of the expression. Then loop and ask for more expressions until the user enters done. Extensions: Identity simplifications are almost always good. Can you make your program always perform these identity improvements whenever it can?
3:40	**Boolean Hardware** Can you use your program to simplify the boolean algebra rule you were given at the start of the lesson. And then build that solution in hardware. Write out the truth table in hardware to verify that it’s the same as the one before.
3:55	**Survey** Give the students a short survey to help with evaluation. https://forms.gle/hgeBo3AfD1KMacqb8