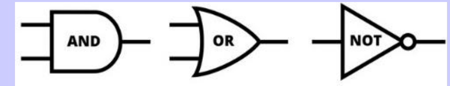
# J277 - 2.4 Boolean logic




## 2.4 – Boolean logic

| Sub topic | Guidance |
|---|---|
| **2.4.1 Boolean logic** | |

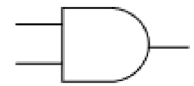
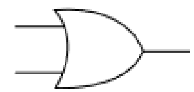
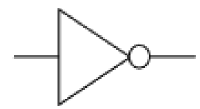
**Sub topic**

- ☐ Simple logic diagrams using the operators AND, OR and NOT
- ☐ Truth tables
- ☐ Combining Boolean operators using AND, OR and NOT
- ☐ Applying logical operators in truth tables to solve problems

**Guidance**

**Required**

- ✓ Knowledge of the truth tables for each logic gate
- ✓ Recognition of each gate symbol
- ✓ Understanding of how to create, complete or edit logic diagrams and truth tables for given scenarios
- ✓ Ability to work with more than one gate in a logic diagram

| Boolean Operators | Logic Gate Symbol |
|---|---|
| AND *(Conjunction)* |  |
| OR *(Disjunction)* |  |
| NOT *(Negation)* |  |

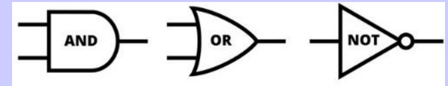
**Truth Tables**

| AND | | | OR | | | NOT | |
|---|---|---|---|---|---|---|---|
| A | B | A AND B | A | B | A OR B | A | NOT A |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | | |
| 1 | 1 | 1 | 1 | 1 | 1 | | |

**Alternatives**

- Use of other valid notation will be accepted within the examination, e.g. Using T/F for 1/0, or V for OR, etc.

# J277 - 2.4 Boolean logic

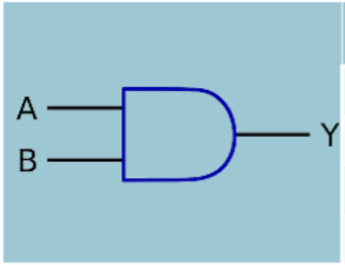



## Simple logic diagrams using the operations AND, OR and NOT

Used to change bits and perform calculations within a computer. They are created by using transistors.
There are 3 basic logic gates:

### The AND Gate

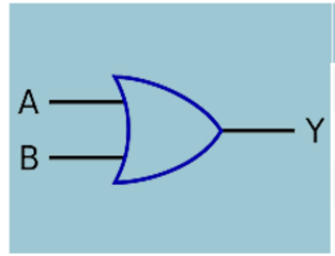
- Only has an output of 1 if both A AND B are 1
- We write this as: **A ∧ B**




| A | B | A ∧ B or Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

### The OR Gate

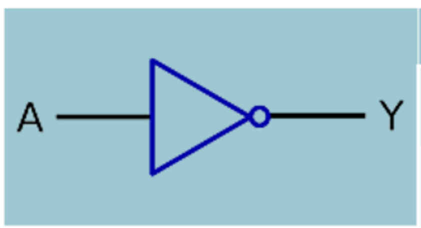
- Has an output of 1 if either of A OR B are 1
- We write this as: **A ∨ B**




| A | B | A ∨ B or Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

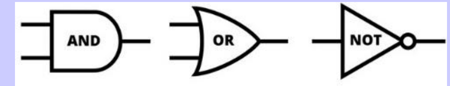
### The NOT Gate

- Has an 1 input only
- It **INVERTS** or swaps the input
- We write this as: ¬ **A**




| A | ¬ A or Y |
|---|---|
| 0 | 1 |
| 1 | 0 |

# J277 - 2.4 Boolean logic




## 2.4 – Boolean logic

| Sub topic | Guidance |
|---|---|
| **2.4.1 Boolean logic** | |

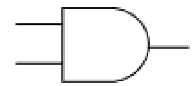
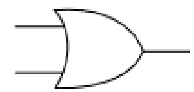
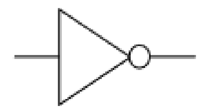
**Sub topic**

- ☐ Simple logic diagrams using the operators AND, OR and NOT
- ☐ Truth tables
- ☐ Combining Boolean operators using AND, OR and NOT
- ☐ Applying logical operators in truth tables to solve problems

**Guidance**

**Required**

- ✓ Knowledge of the truth tables for each logic gate
- ✓ Recognition of each gate symbol
- ✓ Understanding of how to create, complete or edit logic diagrams and truth tables for given scenarios
- ✓ Ability to work with more than one gate in a logic diagram

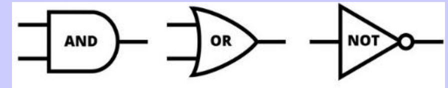
| Boolean Operators | Logic Gate Symbol |
|---|---|
| AND<br>*(Conjunction)* |  |
| OR<br>*(Disjunction)* |  |
| NOT<br>*(Negation)* |  |

**Truth Tables**

| AND | | | OR | | | NOT | |
|---|---|---|---|---|---|---|---|
| A | B | A AND B | A | B | A OR B | A | NOT A |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | | |
| 1 | 1 | 1 | 1 | 1 | 1 | | |

**Alternatives**

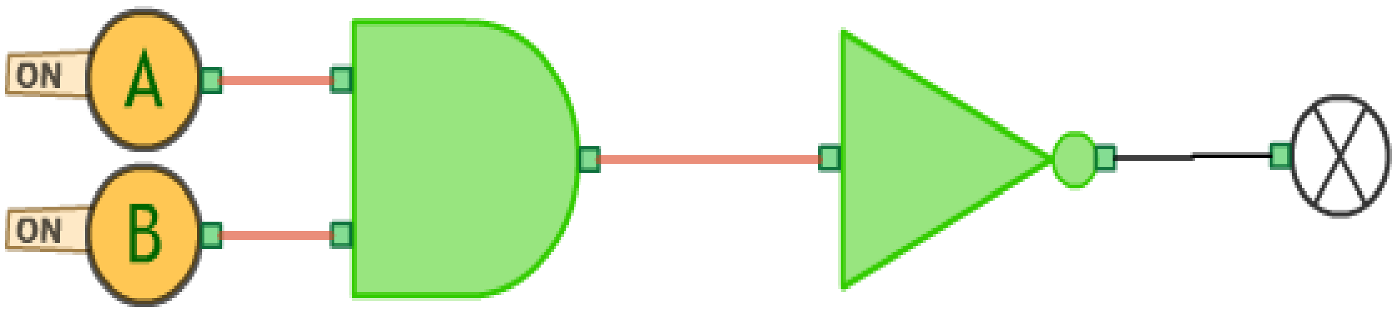
- Use of other valid notation will be accepted within the examination, e.g. Using T/F for 1/0, or V for OR, etc.

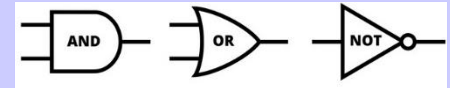



## Combining Boolean operators using AND, OR and NOT to two levels

Logic diagram for the expression: NOT (A AND B)

Alternative notation : $\neg$ **( A $\wedge$ B )**

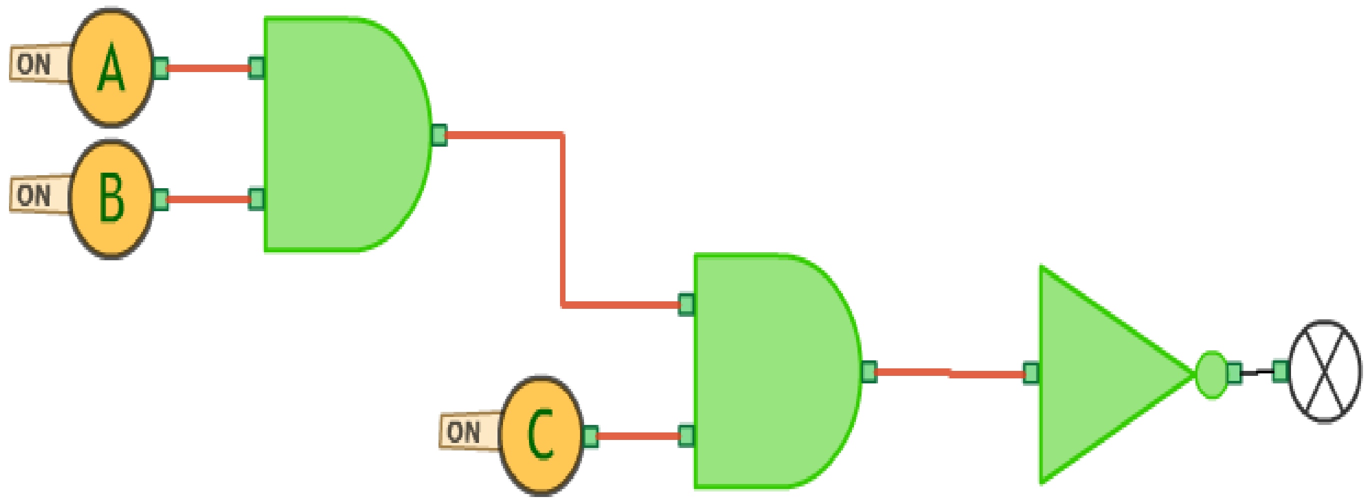




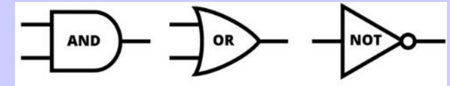

## Combining Boolean operators using AND, OR and NOT to two levels

Logic diagram for the expression: NOT(A AND B) AND C)

Alternative notation : $\neg \mathbf{((A \wedge B) \wedge C)}$



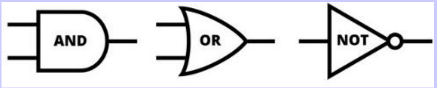
# J277 - 2.4 Boolean logic




## Applying logical operators in truth tables to solve problems

Draw the truth table for the following logic statement: **¬ ( A ⋀ B )**

| A | B | A ⋀ B | ¬ ( A ⋀ B ) |
|---|---|---|---|
| **0** | 0 | 0 | 1 |
| **0** | 1 | 0 | 1 |
| **1** | 0 | 0 | 1 |
| **1** | 1 | 1 | 0 |

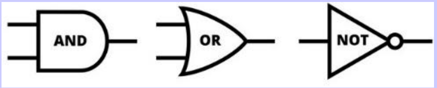



## Applying logical operators in truth tables to solve problems

Draw the truth table for the following logic statement: **¬ (( A ⋀ B ) ⋀ C)**

| A | B | C | A ⋀ B | ( A ⋀ B ) ⋀ C) | ¬ (( A ⋀ B ) ⋀ C) |
|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 1 |
| **0** | 0 | 1 | 0 | 0 | 1 |
| **0** | 1 | 0 | 0 | 0 | 1 |
| **0** | 1 | 1 | 0 | 0 | 1 |
| **1** | 0 | 0 | 0 | 0 | 1 |
| **1** | 0 | 1 | 0 | 0 | 1 |
| **1** | 1 | 0 | 1 | 0 | 1 |
| **1** | 1 | 1 | 1 | 1 | 0 |

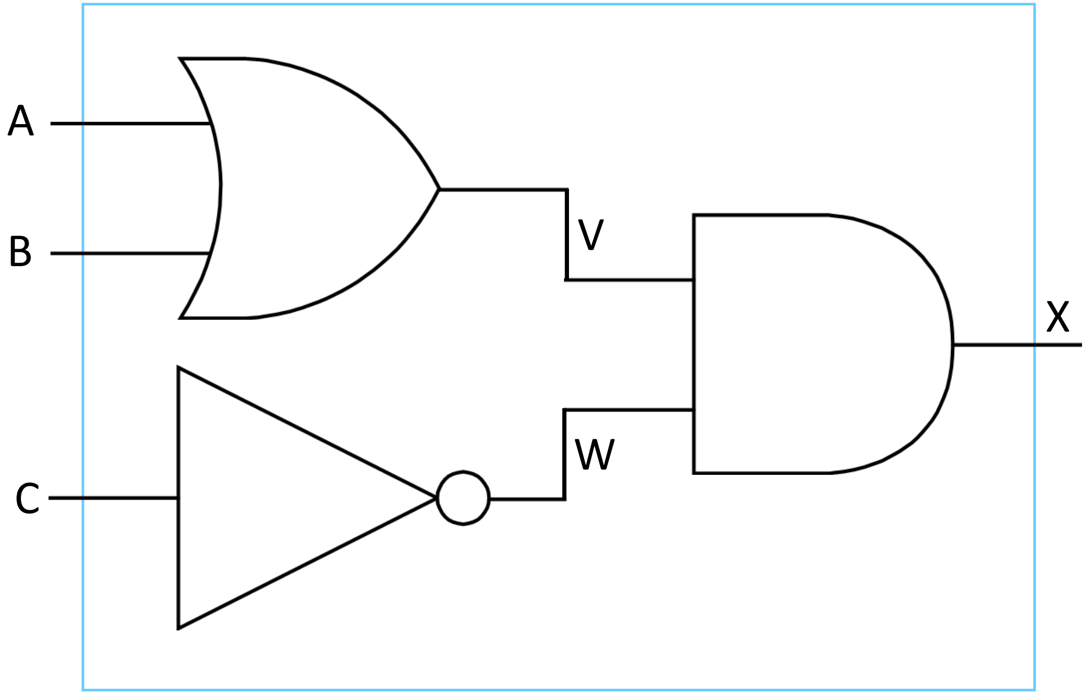
# J277 - 2.4 Boolean logic




## Create, complete or edit logic diagrams and truth tables for given scenarios

Logic diagram and truth table for the following scenario:

“A factory has an automated manufacturing system which operates in an “ON” state (OUTPUT X) if either it is manually switched on by an operator (INPUT A), OR a computer system triggers a scheduled production run (INPUT B). The system also has an emergency override (INPUT C) which in its normal operating state is feeding no signal to the computer system, when it is pressed however it triggers a positive “TRUE” state (1) which should result in the system shutting down.”




| A | B | C | V = A OR B | W = NOT C | X = V AND W |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 |