"Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them."

Co-funded by the European Union

Application of Computers

Formulas and Basic Functions in Excel

PhD Bojan Prlinčević

AASKM

Introduction to Formulas and Functions

What are Formulas?

Formulas are expressions that perform calculations on values in your worksheet. They begin with an equals sign (=) and can include numbers, cell references, operators, and functions.

What are Functions?

Functions are predefined formulas that perform specific calculations using values (arguments) in a particular order. Excel provides hundreds of built-in functions for mathematical, statistical, logical, text, and date operations.

Why Formulas and Functions are Essential:

- **Automation:** Perform calculations automatically without manual computation
- **Accuracy:** Eliminate human calculation errors
- **Efficiency:** Update results instantly when source data changes
- **Scalability:** Apply same calculation to large datasets
- **Complexity:** Handle sophisticated engineering calculations
- **Dynamic:** Results update automatically when inputs change

Introduction to Formulas and Functions

Engineering Applications:

- **Circuit Analysis:** Ohm's Law (V=IR), power calculations (P=VI)
- **Signal Processing:** Frequency calculations, amplitude analysis
- **Statistics:** Mean, standard deviation, variance of measurements
- **Data Analysis:** Trend analysis, correlation, regression
- **Project Management:** Cost calculations, time tracking
- **Quality Control:** Tolerance checking, specification compliance

Formula vs. Function:

- **Formula:** =A1+B1+C1 (manual expression)
- **Function:** =SUM(A1:C1) (predefined operation)
- Functions are specialized formulas with built-in logic

Course Objectives:

Master fundamental Excel formulas and functions to perform engineering calculations, analyze experimental data, and automate repetitive computational tasks.

Formula Basics and Syntax

Co-funded by the European Union

Formula Structure:

All formulas must begin with equals sign (=)

Basic Formula Components:

- **Equals Sign (=):** Indicates formula start
- **Operands:** Values, cell references, or ranges used in calculation
- **Operators:** Symbols indicating operation type (+, -, *, /)
- **Functions:** Predefined calculations (SUM, AVERAGE, etc.)
- **Parentheses:** Control order of operations

Creating a Simple Formula:

1. Click cell where result should appear
2. Type equals sign (=)
3. Enter formula expression
4. Press Enter to execute
5. Result appears in cell, formula visible in Formula Bar

Formula Basics and Syntax

Examples:

Direct Values:

- =5+3 → Result: 8
- =10*2 → Result: 20
- =100/4 → Result: 25

Cell References:

- =A1+B1 → Adds values in cells A1 and B1
- =C2*D2 → Multiplies values in cells C2 and D2
- =E5-F5 → Subtracts F5 from E5

Combined:

- =A1+10 → Adds 10 to value in A1
- =(B2+C2)/2 → Average of B2 and C2
- =A1*B1+C1 → Multiply A1 and B1, then add C1

Formula Basics and Syntax

Editing Formulas:

- **Double-click cell:** Edit directly in cell
- **Click cell, then Formula Bar:** Edit in Formula Bar
- **Press F2:** Enter edit mode
- **Color-coded references:** Excel highlights referenced cells with matching colors

Formula Display:

- Cell shows calculated result
- Formula Bar shows actual formula
- To see all formulas: Ctrl + ` (grave accent) or Formulas tab → Show Formulas

Arithmetic Operators and Order of Operations

Arithmetic Operators:

Addition (+):

- =A1+B1 → Sum of A1 and B1
- =5+3+2 → Result: 10

Subtraction (-):

- =A1-B1 → Difference between A1 and B1
- =100-25 → Result: 75

Multiplication (*):

- =A1*B1* → *Product of A1 and B1*
- *=54* → Result: 20

Division (/):

- =A1/B1 → A1 divided by B1
- =100/4 → Result: 25

Co-funded by the European Union

Arithmetic Operators and Order of Operations

Exponentiation (^):

- =A1^2 → A1 squared
- =2^3 → Result: 8 (2^3)
- =A1^0.5 → Square root of A1

Percentage (%):

- =A1*10% → 10% of A1
- =50% → Result: 0.5

Order of Operations (PEMDAS):

Excel follows standard mathematical order:

1. **Parentheses:** ( )
2. **Exponents:** ^
3. **Multiplication and Division:** *, / (left to right)
4. **Addition and Subtraction:** +, - (left to right)

Arithmetic Operators and Order of Operations

Examples:

Without Parentheses:

- =5+3*2* → *Result: 11 (multiplication first: 3*2=6, then 5+6=11)
- =10-4/2 → Result: 8 (division first: 4/2=2, then 10-2=8)

With Parentheses:

- =(5+3)*2* → *Result: 16 (parentheses first: 5+3=8, then 8*2=16)
- =(10-4)/2 → Result: 3 (parentheses first: 10-4=6, then 6/2=3)

Complex Example:

- =((A1+B1)*C1)^2/D1

Engineering Application - Ohm's Law:

- Voltage: =Current*Resistance* → *=A2*B2
- Current: =Voltage/Resistance → =A2/B2
- Power: =Voltage*Current* → *=A2*B2
- Power: =Voltage^2/Resistance → =A2^2/B2

Cell References - Relative, Absolute, and Mixed

Relative References (Default):

Cell references that change when formula is copied to another location.

Example:

- Formula in C1: =A1+B1
- Copy to C2: Formula becomes =A2+B2 (references adjust)
- Copy to D1: Formula becomes =B1+C1 (references shift right)

Use Case: When you want formula to adjust for each row/column

Absolute References:

Cell references that remain fixed when formula is copied, indicated by dollar signs ($).

Syntax:

- A\1 → Both column and row fixed
- Formula in C1: =A\1+B1
- Copy to C2: Formula remains =A\1+B2 (A1 stays fixed)
- Copy to D1: Formula remains =A\1+C1 (A1 stays fixed)

Cell References - Relative, Absolute, and Mixed

Creating Absolute References:

- Type dollar signs manually: =A\1
- Or press F4 key after typing cell reference (cycles through reference types)

Use Case: Referencing constant values (tax rate, conversion factor, standard value)

Mixed References:

Fix either column or row, but not both.

Column Fixed, Row Relative:

- $A1 \rightarrow ColumnAfixed, rowadjusts \bullet Copydown$:A1, $A2$,A3 (column stays A)
- Copy right: $A1$,B1, $C1 (column stays A)

Row Fixed, Column Relative:

- A$1 → Row 1 fixed, column adjusts
- Copy down: A$1, A$1, A$1 (row stays 1)
- Copy right: A$1, B$1, C$1 (row stays 1)

Use Case: Multiplication tables, lookup tables, matrix calculations

Cell References - Relative, Absolute, and Mixed

F4 Key Cycling:

Press F4 repeatedly to cycle through reference types:

A1 (relative)

A\1 (absolute)

A$1 (mixed - row absolute)

A\1 (mixed - column absolute)

Back to A1

Engineering Example - Voltage Divider:

- Reference voltage in A\1: 5V (constant)
- R1 values in column B
- R2 values in column C
- Vout formula: =A\1*C2/(B2+C2)
- Copy formula down: A\1 stays fixed, B and C adjust

Co-funded by the European Union

SUM Function

What is SUM?

Adds all numbers in a range of cells. Most commonly used Excel function.

Syntax:

=SUM(number1, [number2], ...)

Arguments:

- number1: Required - first number, cell reference, or range
- number2, ...: Optional - additional numbers or ranges (up to 255)

Basic Examples:

Range:

- =SUM(A1:A10) → Adds all values from A1 to A10
- =SUM(B2:B20) → Adds all values from B2 to B20

Multiple Ranges:

- =SUM(A1:A5, C1:C5) → Adds values from both ranges
- =SUM(A1:A10, B1:B10, C1:C10) → Adds three ranges

Co-funded by the European Union

SUM Function

Individual Cells:

- =SUM(A1, B1, C1) → Adds three specific cells
- =SUM(A1, A3, A5) → Adds non-adjacent cells

Mixed:

- =SUM(A1:A5, 10) → Adds range plus constant value
- =SUM(A1:A5, B10, 20) → Combines range, cell, and constant

Entire Column/Row:

- =SUM(A:A) → Adds all values in column A
- =SUM(1:1) → Adds all values in row 1

Quick SUM - AutoSum:

1. Select cell below column of numbers or right of row
2. Home tab → Editing group → AutoSum (Σ)
3. Or Alt + = (keyboard shortcut)
4. Excel suggests range, press Enter to accept
5. Or modify range and press Enter

SUM Function

Engineering Applications:

Total Resistance (Series Circuit):

- R_total = R1 + R2 + R3 + ... + Rn
- =SUM(B2:B10) where B2:B10 contains resistor values

Total Current (Parallel Branches):

- I_total = I1 + I2 + I3 + ...
- =SUM(C2:C5)

Total Power Consumption:

- P_total = P1 + P2 + P3 + ...
- =SUM(D2:D20)

Cumulative Measurements:

- Total voltage drop across components
- Sum of experimental measurements
- Total project costs

AVERAGE, COUNT, MAX, and MIN Functions

Co-funded by the European Union

AVERAGE Function:

Calculates arithmetic mean of numbers in range.

Syntax: =AVERAGE(number1, [number2], ...)

Examples:

- =AVERAGE(A1:A10) → Mean of values in A1:A10
- =AVERAGE(B2:B20) → Average of B2:B20
- =AVERAGE(A1:A5, C1:C5) → Average of combined ranges

Engineering Use: Average voltage, mean current, typical resistance value

COUNT Function:

Counts cells containing numbers in range.

Syntax: =COUNT(value1, [value2], ...)

Examples:

- =COUNT(A1:A10) → Counts numeric cells in A1:A10
- =COUNT(A:A) → Counts all numeric cells in column A

AVERAGE, COUNT, MAX, and MIN Functions

Related Functions:

- **COUNTA:** Counts non-empty cells (numbers and text)
- **COUNTBLANK:** Counts empty cells

Engineering Use: Number of measurements, data points collected, valid readings

MAX Function:

Returns largest value in range.

Syntax: =MAX(number1, [number2], ...)

Examples:

- =MAX(A1:A10) → Largest value in A1:A10
- =MAX(B2:B20) → Maximum value in B2:B20
- =MAX(A1:A5, C1:C5) → Maximum across multiple ranges

Engineering Use: Peak voltage, maximum current, highest temperature, worst-case value

AVERAGE, COUNT, MAX, and MIN Functions

MIN Function:
Returns smallest value in range.
Syntax: =MIN(number1, [number2], ...)

Examples:
- =MIN(A1:A10) → Smallest value in A1:A10
- =MIN(B2:B20) → Minimum value in B2:B20

Engineering Use: Minimum voltage, lowest current, best-case value, threshold detection

AVERAGE, COUNT, MAX, and MIN Functions

Co-funded by the European Union

Practical Example

Measurement Analysis:

Measurement	Voltage (V)
1	4.95
2	5.02
3	4.98
4	5.01
5	4.97
Average	=AVERAGE(B2:B6) → 4.986V
Count	=COUNT(B2:B6) → 5
Max	=MAX(B2:B6) → 5.02V
Min	=MIN(B2:B6) → 4.95V
Range	=MAX(B2:B6)-MIN(B2:B6) → 0.07V

Co-funded by the European Union

Statistical Functions

MEDIAN Function:

Returns middle value in dataset (50th percentile).

Syntax: =MEDIAN(number1, [number2], ...)

Example:

- =MEDIAN(A1:A10) → Middle value when sorted
- Odd count: Actual middle value
- Even count: Average of two middle values

Use: Less affected by outliers than AVERAGE, better for skewed data

MODE.SNGL Function:

Returns most frequently occurring value in dataset.

Syntax: =MODE.SNGL(number1, [number2], ...)

Example:

- =MODE.SNGL(A1:A20) → Most common value

Use: Identify typical or recurring measurement value

Statistical Functions

STDEV.S Function:

Calculates standard deviation of sample (estimates population standard deviation).

Syntax: =STDEV.S(number1, [number2], ...)

Example:

- =STDEV.S(A1:A10) → Standard deviation of sample

Engineering Use: Measurement uncertainty, precision analysis, quality control

STDEV.P Function:

Calculates standard deviation of entire population.

Syntax: =STDEV.P(number1, [number2], ...)

Use: When data represents complete population, not sample

VAR.S and VAR.P Functions:

Calculate variance (square of standard deviation) for sample or population.

Syntax:

- =VAR.S(number1, [number2], ...) → Sample variance
- =VAR.P(number1, [number2], ...) → Population variance

Co-funded by the European Union

Statistical Functions

QUARTILE.INC Function:

Returns specified quartile of dataset.

Syntax: =QUARTILE.INC(array, quart)

- quart: 0 (minimum), 1 (25th percentile), 2 (median), 3 (75th percentile), 4 (maximum)

Example:

- =QUARTILE.INC(A1:A20, 1) → 25th percentile (Q1)
- =QUARTILE.INC(A1:A20, 3) → 75th percentile (Q3)

Engineering Application - Measurement Quality:

Statistical Functions

Engineering Application - Measurement Quality:

Statistic	Formula	Result
Mean	=AVERAGE(A2:A21)	5.00V
Median	=MEDIAN(A2:A21)	4.99V
Std Dev	=STDEV.S(A2:A21)	0.05V
Variance	=VAR.S(A2:A21)	0.0025

IF Function - Logical Testing

What is IF Function?

Performs logical test and returns one value if TRUE, another if FALSE. Essential for conditional calculations and decision-making.

Syntax:

=IF(logical_test, value_if_true, value_if_false)

Arguments:

- **logical_test:** Condition to evaluate (comparison or logical expression)
- **value_if_true:** Value returned if condition is TRUE
- **value_if_false:** Value returned if condition is FALSE

Comparison Operators:

- = (equal to)
- > (greater than)
- < (less than)
- >= (greater than or equal to)
- <= (less than or equal to)
- <> (not equal to)

Co-funded by the European Union

IF Function - Logical Testing

Basic Examples:

Simple Comparison:

- =IF(A1>10, "High", "Low")
  - If A1 is greater than 10, display "High", otherwise "Low"

Pass/Fail Test:

- =IF(B2>=60, "Pass", "Fail")
  - If score in B2 is 60 or above, "Pass", otherwise "Fail"

Numeric Results:

- =IF(C2>100, C2*0.1, 0)
  - If C2 exceeds 100, calculate 10% of C2, otherwise return 0

Engineering Applications:

Voltage Range Check:

- =IF(A2>5, "Over Voltage", "Normal")
  - Check if voltage exceeds 5V limit

Co-funded by the European Union

IF Function - Logical Testing

Tolerance Verification:

- =IF(ABS(B2-5)<=0.1, "Within Tolerance", "Out of Tolerance")
  - Check if value is within ±0.1 of target (5)

Component Selection:

- =IF(C2<1000, "Use Standard", "Use High Power")
  - Select component type based on power requirement

Status Indicator:

- =IF(D2=0, "OFF", "ON")
  - Display device status based on value

Nested IF:

Multiple conditions tested sequentially.

Syntax:

=IF(test1, value1, IF(test2, value2, IF(test3, value3, default_value)))

Co-funded by the European Union

IF Function - Logical Testing

Example - Grade Classification:

- =IF(A2>=90, "A", IF(A2>=80, "B", IF(A2>=70, "C", IF(A2>=60, "D", "F"))))

Engineering Example - Signal Level:

- =IF(A2>4, "High", IF(A2>2, "Medium", "Low"))
  - Classify signal into three levels

Limitation: Excel allows up to 64 nested IF functions, but complex nesting becomes difficult to read and maintain. Consider using IFS function (Excel 2016+) for multiple conditions.

Co-funded by the European Union

AND, OR, and NOT Functions

Logical Functions:

Combine multiple conditions in logical tests, often used with IF function.

AND Function:

Returns TRUE only if ALL conditions are TRUE.

Syntax: =AND(logical1, [logical2], ...)

Examples:

- =AND(A1>10, B1<20) → TRUE if both conditions met
- =AND(A1>=5, A1<=10) → TRUE if A1 between 5 and 10 (inclusive)

With IF:

- =IF(AND(A2>0, A2<5), "Valid Range", "Out of Range")
  - Check if value is between 0 and 5

Engineering Application - Safety Check:

- =IF(AND(Voltage<=5, Current<=100, Temp<80), "Safe", "Warning")
  - All three conditions must be met for safe operation

AND, OR, and NOT Functions

OR Function:

Returns TRUE if ANY condition is TRUE.

Syntax: =OR(logical1, [logical2], ...)

Examples:

- =OR(A1>100, B1>100) → TRUE if either condition met
- =OR(A1<0, A1>10) → TRUE if A1 outside 0-10 range

With IF:

- =IF(OR(A2<0, A2>5), "Error", "Normal")
  - Flag error if value is negative or exceeds 5

Engineering Application - Fault Detection:

- =IF(OR(Voltage>5.5, Voltage<4.5), "Fault", "OK")
  - Detect voltage outside acceptable range (4.5V - 5.5V)

AND, OR, and NOT Functions

NOT Function:

Reverses logical value (TRUE becomes FALSE, FALSE becomes TRUE).

Syntax: =NOT(logical)

Examples:

- =NOT(A1>10) → TRUE if A1 is NOT greater than 10 (equivalent to A1<=10)
- =NOT(B1="Active") → TRUE if B1 does not contain "Active"

With IF:

- =IF(NOT(A2=0), B2/A2, "Error")
  - Perform division only if A2 is not zero

Combining Logical Functions:

Complex Conditions:

- =IF(AND(A2>0, OR(B2="Type1", B2="Type2")), "Valid", "Invalid")
  - A2 must be positive AND B2 must be either Type1 or Type2

AND, OR, and NOT Functions

Engineering Example - Component Qualification:

- =IF(AND(Resistance>=900, Resistance<=1100, OR(Tolerance=5, Tolerance=1)), "Qualified", "Rejected")
    - Resistance must be 900-1100Ω AND tolerance must be either 5% or 1%

Range Checking:

- =IF(AND(A2>=4.5, A2<=5.5), "In Spec", "Out of Spec")
    - Check if voltage is within specification (4.5V - 5.5V)

Multiple Exclusions:

- =IF(OR(A2<0, A2>100, B2="Error"), "Invalid", "Valid")
    - Flag as invalid if any error condition exists

SUMIF and COUNTIF Functions

SUMIF Function:

Adds cells that meet specific criterion.

Syntax: =SUMIF(range, criteria, [sum_range])

Arguments:

- **range:** Range to evaluate against criteria
- **criteria:** Condition that determines which cells to sum
- **sum_range:** Optional - actual cells to sum (if different from range)

Examples:

Simple Criteria:

- =SUMIF(A1:A10, ">50") → Sum values in A1:A10 that are greater than 50
- =SUMIF(A1:A10, "=100") → Sum values equal to 100
- =SUMIF(B1:B10, "<>0") → Sum non-zero values

Text Criteria:

- =SUMIF(A1:A10, "Resistor") → Sum where cell contains "Resistor"
- =SUMIF(A1:A10, "R*") → Sum where cell starts with "R" (wildcard)

SUMIF and COUNTIF Functions

Cell Reference Criteria:

- =SUMIF(A1:A10, ">"&C1) → Sum values greater than value in C1
- =SUMIF(A1:A10, B1) → Sum values equal to B1

Different Sum Range:

- =SUMIF(A1:A10, "Resistor", B1:B10)
  - Check A1:A10 for "Resistor", sum corresponding values in B1:B10

Engineering Application - Component Cost:

- =SUMIF(B2:B5, "Resistor", C2:C5) → Total cost of resistors: $0.20

Component	Type	Cost
R1	Resistor	0.10
C1	Capacitor	0.25
R2	Resistor	0.10
D1	Diode	0.50

SUMIF and COUNTIF Functions

COUNTIF Function:

Counts cells that meet specific criterion.

Syntax: =COUNTIF(range, criteria)

Arguments:

- **range:** Range to evaluate
- **criteria:** Condition that determines which cells to count

Examples:

Numeric Criteria:

- =COUNTIF(A1:A10, ">50") → Count values greater than 50
- =COUNTIF(A1:A10, ">=100") → Count values 100 or greater
- =COUNTIF(B1:B10, 0) → Count cells containing zero

Text Criteria:

- =COUNTIF(A1:A10, "Pass") → Count cells containing "Pass"
- =COUNTIF(A1:A10, "R*") → Count cells starting with "R"

SUMIF and COUNTIF Functions

Cell Reference:

- =COUNTIF(A1:A10, C1) → Count cells equal to value in C1

Engineering Applications:

Pass/Fail Count:

- =COUNTIF(D2:D20, "Pass") → Number of passed tests
- =COUNTIF(D2:D20, "Fail") → Number of failed tests

Tolerance Check:

- =COUNTIF(A2:A50, ">5.5") → Count measurements exceeding 5.5V
- =COUNTIF(A2:A50, "<4.5") → Count measurements below 4.5V

Component Inventory:

- =COUNTIF(B2:B100, "Resistor") → Number of resistors in inventory

Wildcards:

- * (asterisk): Any number of characters
- ? (question mark): Single character
- Example: =COUNTIF(A1:A10, "R?") → Counts "R1", "R2", etc.

Text Functions

CONCATENATE and CONCAT Functions:

Combine text from multiple cells into one cell.

Syntax:

- =CONCATENATE(text1, [text2], ...) → Legacy function
- =CONCAT(text1, [text2], ...) → Modern version (Excel 2016+)

Examples:

- =CONCATENATE(A1, " ", B1) → Combines A1 and B1 with space
- =CONCAT("Resistor ", A1, " Ohm") → "Resistor 1000 Ohm"

Ampersand Operator (&):

Alternative method for concatenation.

- =A1&" "&B1 → Same as CONCATENATE(A1, " ", B1)
- ="Value: "&A1&" V" → "Value: 5 V"

TEXTJOIN Function (Excel 2016+):

Joins text with delimiter, can ignore empty cells.

Syntax: =TEXTJOIN(delimiter, ignore_empty, text1, [text2], ...)

Co-funded by the European Union

Text Functions

Example:

- =TEXTJOIN(", ", TRUE, A1:A5) → Joins A1:A5 with comma separator

LEFT, RIGHT, and MID Functions:

Extract specific characters from text.

LEFT: =LEFT(text, [num_chars])

- =LEFT(A1, 3) → First 3 characters from A1
- =LEFT("Resistor", 4) → "Resi"

RIGHT: =RIGHT(text, [num_chars])

- =RIGHT(A1, 2) → Last 2 characters from A1
- =RIGHT("R1000", 4) → "1000"

MID: =MID(text, start_num, num_chars)

- =MID(A1, 2, 3) → 3 characters starting from position 2
- =MID("Component123", 10, 3) → "123"

Co-funded by the European Union

Text Functions

LEN Function:

Returns number of characters in text.

Syntax: =LEN(text)

Examples:

- =LEN(A1) → Number of characters in A1
- =LEN("Resistor") → 8

UPPER, LOWER, and PROPER Functions:

Change text case.

Examples:

- =UPPER("resistor") → "RESISTOR"
- =LOWER("CAPACITOR") → "capacitor"
- =PROPER("john smith") → "John Smith"

Co-funded by the European Union

Text Functions

TRIM Function:

Removes extra spaces from text.

Syntax: =TRIM(text)

Example:

- =TRIM(" Extra Spaces ") → "Extra Spaces"

Engineering Application - Component Labels:

Code	Formula	Result
R1K5	=LEFT(A2,1)	"R" (Type)
R1K5	=MID(A2,2,1)	"1" (Value)
R1K5	=RIGHT(A2,2)	"K5" (Multiplier)
R1K5	="Component: "&A2	"Component: R1K5"

Co-funded by the European Union

Date and Time Functions

TODAY and NOW Functions:

Return current date and time.

TODAY: =TODAY()

- Returns current date (no time)
- Updates automatically when workbook opens
- Example result: 11/9/2025

NOW: =NOW()

- Returns current date and time
- Updates automatically
- Example result: 11/9/2025 14:30

DATE Function:

Creates date from year, month, day values.

Syntax: =DATE(year, month, day)

Examples:

- =DATE(2025, 11, 9) → 11/9/2025
- =DATE(A1, B1, C1) → Date from separate cells

Co-funded by the European Union

Date and Time Functions

YEAR, MONTH, and DAY Functions:

Extract components from date.

Examples:

- =YEAR(A1) → Extracts year from date in A1
- =MONTH(A1) → Extracts month (1-12)
- =DAY(A1) → Extracts day (1-31)

WEEKDAY Function:

Returns day of week as number (1=Sunday, 7=Saturday).

Syntax: =WEEKDAY(serial_number, [return_type])

Example:

- =WEEKDAY(A1) → Returns 1-7
- =WEEKDAY(TODAY()) → Current day of week

Co-funded by the European Union

Date and Time Functions

TIME Function:

Creates time from hour, minute, second values.

Syntax: =TIME(hour, minute, second)

Example:

- =TIME(14, 30, 0) → 2:30 PM

HOUR, MINUTE, and SECOND Functions:

Extract time components.

Examples:

- =HOUR(A1) → Extracts hour (0-23)
- =MINUTE(A1) → Extracts minute (0-59)
- =SECOND(A1) → Extracts second (0-59)

Date and Time Functions

Date Arithmetic:

Dates stored as numbers (days since 1/1/1900), enabling calculations.

Examples:

- =TODAY()+7 → Date one week from today
- =B1-A1 → Days between two dates
- =NOW()+1/24 → Time one hour from now (1 day = 24 hours)

DATEDIF Function:

Calculates difference between dates in various units.

Syntax: =DATEDIF(start_date, end_date, unit)

Units:

- "D" - Days
- "M" - Months
- "Y" - Years
- "MD" - Days ignoring months and years
- "YM" - Months ignoring years

Example:

- =DATEDIF(A1, TODAY(), "D") → Days since date in A1

Date and Time Functions

Engineering Applications:

Project Timeline:

- Start Date: 1/1/2025
- End Date: 3/31/2025
- Duration: =B2-A2 → 89 days
- Days Remaining: =B2-TODAY()

Measurement Timestamp:

- Date: =TODAY()
- Time: =NOW()
- Elapsed Time: =NOW()-A2 (in days, multiply by 24 for hours)

Equipment Calibration:

- Last Calibration: 1/1/2025
- Days Since: =TODAY()-A2
- Next Due: =A2+365 (annual calibration)

ROUND, ROUNDUP, and ROUNDDOWN Functions

ROUND Function:

Rounds number to specified number of digits.

Syntax: =ROUND(number, num_digits)

Arguments:

- **number:** Value to round
- **num_digits:** Number of decimal places
  - Positive: Decimal places (2 = two decimals)
  - Zero: Round to nearest integer
  - Negative: Round to left of decimal (tens, hundreds, etc.)

Examples:

- =ROUND(3.14159, 2) → 3.14
- =ROUND(3.14159, 0) → 3
- =ROUND(1234.56, -2) → 1200 (round to nearest hundred)
- =ROUND(A1*B1, 2) → Round calculation result to 2 decimals

ROUND, ROUNDUP, and ROUNDDOWN Functions

ROUNDUP Function:

Always rounds up (away from zero).

Syntax: =ROUNDUP(number, num_digits)

Examples:

- =ROUNDUP(3.14159, 2) → 3.15
- =ROUNDUP(3.1, 0) → 4
- =ROUNDUP(1234, -2) → 1300

Use Case: Conservative estimates, worst-case calculations, component quantities (always round up to whole units)

ROUNDDOWN Function:

Always rounds down (toward zero).

Syntax: =ROUNDDOWN(number, num_digits)

ROUND, ROUNDUP, and ROUNDDOWN Functions

Examples:

- =ROUNDDOWN(3.14159, 2) → 3.14
- =ROUNDDOWN(3.9, 0) → 3
- =ROUNDDOWN(1299, -2) → 1200

Use Case: Best-case calculations, available resources (can't use partial units)

Engineering Applications:

Measurement Precision:

- Raw ADC Reading: 3.14159265 V
- Display Value: =ROUND(A2, 2) → 3.14 V (match instrument precision)

Component Quantity:

- Calculated Need: 15.3 resistors
- Order Quantity: =ROUNDUP(A2, 0) → 16 resistors (can't order partial)

ROUND, ROUNDUP, and ROUNDDOWN Functions

Power Calculation:

- P = V × I = 5.123 × 0.0234 = 0.11987820
- Rounded: =ROUND(A2*B2, 3) → 0.120 W

Tolerance Calculation:

- Nominal: 1000 Ω
- Tolerance: 5%
- Max: =ROUNDUP(A2*(1+B2/100), 0) → 1050 Ω
- Min: =ROUNDDOWN(A2*(1-B2/100), 0) → 950 Ω

Related Functions:

CEILING: Rounds up to nearest multiple

- =CEILING(15.3, 5) → 20 (nearest multiple of 5)

FLOOR: Rounds down to nearest multiple

- =FLOOR(15.3, 5) → 15

ROUND, ROUNDUP, and ROUNDDOWN Functions

MROUND: Rounds to nearest multiple

- =MROUND(15.3, 5) → 15

INT: Rounds down to nearest integer

- =INT(3.9) → 3

TRUNC: Truncates to specified decimals (no rounding)

- =TRUNC(3.14159, 2) → 3.14

Error Handling and Troubleshooting

Common Excel Errors:

#DIV/0! Error:

- **Cause:** Division by zero or empty cell
- **Example:** =A1/B1 where B1 is 0 or empty
- **Solution:** Check denominator, use IF to prevent: =IF(B1=0, "Error", A1/B1)

#VALUE! Error:

- **Cause:** Wrong data type (text in numeric formula)
- **Example:** =A1+B1 where B1 contains text "N/A"
- **Solution:** Verify data types, clean data, use ISNUMBER to check

#REF! Error:

- **Cause:** Invalid cell reference (deleted row/column)
- **Example:** Formula referenced C5, but column C was deleted
- **Solution:** Undo deletion or update formula references

Error Handling and Troubleshooting

#NAME? Error:

- **Cause:** Excel doesn't recognize text in formula (misspelled function, missing quotes)
- **Example:** =SOM(A1:A10) instead of =SUM(A1:A10)
- **Solution:** Check spelling, add quotes around text: ="Hello"

#N/A Error:

- **Cause:** Value not available (common in lookup functions)
- **Example:** VLOOKUP can't find match
- **Solution:** Verify lookup value exists, use IFNA to handle

#NUM! Error:

- **Cause:** Invalid numeric value (negative number in SQRT, iteration limit exceeded)
- **Example:** =SQRT(-1)
- **Solution:** Check input values, ensure valid range

Error Handling and Troubleshooting

#NULL! Error:

- **Cause:** Incorrect range operator (space instead of comma or colon)
- **Example:** =SUM(A1 A10) instead of =SUM(A1:A10)
- **Solution:** Use correct range syntax with colon

Display:

- **Cause:** Column too narrow to display value (not actual error)
- **Solution:** Widen column by double-clicking column border

Error Handling Functions:

IFERROR:

Returns specified value if formula results in error.

Syntax: =IFERROR(value, value_if_error)

Examples:

- =IFERROR(A1/B1, 0) → Returns 0 if division error
- =IFERROR(A1/B1, "Invalid") → Returns "Invalid" if error
- =IFERROR(VLOOKUP(A1, Table, 2, FALSE), "Not Found")

Error Handling and Troubleshooting

IFNA:

Returns specified value only if #N/A error (more specific than IFERROR).

Syntax: =IFNA(value, value_if_na)

Example:

- =IFNA(VLOOKUP(A1, Table, 2, FALSE), "Not Found")

ISERROR:

Returns TRUE if value is any error type.

Syntax: =ISERROR(value)

Example:

- =IF(ISERROR(A1/B1), "Error in calculation", A1/B1)

Formula Auditing Tools:

Trace Precedents:

- Formulas tab → Formula Auditing → Trace Precedents
- Shows arrows pointing to cells referenced by formula

Error Handling and Troubleshooting

Trace Dependents:

- Shows arrows pointing to cells that reference selected cell

Evaluate Formula:

- Step through formula calculation to identify issues
- Formulas tab → Formula Auditing → Evaluate Formula

Show Formulas:

- Display formulas instead of results
- Ctrl + ` or Formulas tab → Show Formulas

Error Checking:

- Formulas tab → Formula Auditing → Error Checking
- Identifies potential errors in worksheet

Best Practices and Engineering Applications

Formula Best Practices:

Clarity and Readability:

- **Use Cell References:** =A1*B1 instead of =5*10 (easier to update)
- **Named Ranges:** Define names for important cells/ranges (Formulas → Define Name)
  - Example: =Voltage*Current instead of =A2*B2
    - **Add Comments:** Right-click cell → Insert Comment to document complex formulas
    - **Break Complex Formulas:** Use helper columns for intermediate calculations

Accuracy:

- **Absolute References:** Use $ for constants (tax rate, conversion factors)
- **Parentheses:** Control order of operations explicitly
- **Round Appropriately:** Match precision to measurement accuracy
- **Validate Results:** Test formulas with known values
- **Error Handling:** Use IFERROR to prevent error propagation

Best Practices and Engineering Applications

Efficiency:

- **Avoid Volatile Functions:** NOW(), TODAY(), RAND() recalculate constantly
- **Copy Formulas:** Use fill handle instead of retyping
- **Use Functions:** SUM(A1:A100) instead of =A1+A2+A3+...
- **Structured References:** Use table column names in formulas

Documentation:

- **Header Rows:** Label all columns clearly with units
- **Separate Input/Output:** Distinguish input cells (data) from calculated cells (formulas)
- **Color Coding:** Use cell shading to identify input vs. calculated cells
- **Formula Sheet:** Create separate worksheet documenting key formulas

Best Practices and Engineering Applications

Engineering Applications Summary:

Circuit Analysis:

- Ohm's Law: =Voltage/Resistance (Current)
- Power: =Voltage*Current or =Voltage^2/Resistance
- Series Resistance: =SUM(R1:R10)
- Parallel Resistance: =1/SUM(1/R1, 1/R2, 1/R3)

Data Analysis:

- Average Measurement: =AVERAGE(Data_Range)
- Standard Deviation: =STDEV.S(Data_Range)
- Range: =MAX(Data_Range)-MIN(Data_Range)
- Outlier Detection: =IF(ABS(Value-AVERAGE(Range))>2*STDEV.S(Range), "Outlier", "Normal")

Co-funded by the European Union

Best Practices and Engineering Applications

Quality Control:

- Tolerance Check: =IF(AND(Value>=Min, Value<=Max), "Pass", "Fail")
- Percent Error: =ABS((Measured-Actual)/Actual)*100
- Within Spec: =IF(ABS(Measured-Target)<=Tolerance, "OK", "Reject")

Project Management:

- Total Cost: =SUMIF(Type_Range, "Component", Cost_Range)
- Days Remaining: =Deadline-TODAY()
- Percent Complete: =Completed_Tasks/Total_Tasks*100

Best Practices and Engineering Applications

Practical Example

Voltage Divider Calculator:

Key Takeaways:

- Formulas automate calculations and eliminate manual errors
- Functions provide powerful built-in capabilities for common operations
- Proper cell referencing (relative/absolute) is crucial for copying formulas
- Logical functions enable conditional calculations and decision-making
- Error handling ensures robust, reliable worksheets
- Engineering applications benefit from systematic formula organization

Parameter	Value	Formula
Vin	12	(input)
R1	10000	(input)
R2	5000	(input)
Vout	4.00	=Vin*R2/(R1+R2)
Current	0.80	=Vin/(R1+R2)
P_R1	6.40	=Current^2*R1
P_R2	3.20	=Current^2*R2

WBnet

Co-funded by the European Union

Questions & Answers

"Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them."

Network of centers for regional short study programs in the countries of the Western Balkans

Call: ERASMUS-EDU-2023-CBHE

Project number: 101128813

ATYCC

UNIVERSITY OF L]UBL]ANA
Faculty of Electrical Engineering

POLITÉCNICA

University of Pristina
Kosovska Mitrovica