BENCH Format Manual

Introduction:

BENCH is a text file format used for representing digital circuit netlists in a readable and simple manner. It is commonly used in academic and research contexts for logic synthesis, fault simulation, and testing digital circuits.

File Extension:

`.bench`

Syntax:

1. **Comments:**
   - Begin with a # symbol.
   - Example: `# This is a comment`
2. **Inputs:**
   - Declared with the keyword `INPUT`.
   - No parentheses are used.
   - Example: `INPUT(G1)`
3. **Outputs:**
   - Declared with the keyword `OUTPUT`.
   - No parentheses are used.
   - Example: `OUTPUT(H1)`
4. **Gates:**
   - Gate declarations start with the gate type followed by the output and input(s).
   - Supported gates include `AND`, `OR`, `NAND`, `NOR`, `XOR`, `XNOR`, `NOT`, etc.
   - Example: `AND(A1, G1, G2)`
5. **Wire/Node Names:**
   - Names should be unique for each input, output, and internal node.
   - Example: `NOT(B1, G1)`

Do's and Don'ts:

- **Do:**
  - Start each declaration (input, output, gate) on a new line.
  - Use clear and distinct names for inputs, outputs, and internal nodes.
  - Include comments for clarity and documentation purposes.
  - Follow the syntax strictly for compatibility with tools that parse BENCH files.
- **Don't:**
  - Don't use parentheses for input and output declarations.
  - Avoid using ambiguous or repetitive names for different elements.
  - Don't mix gate declarations with input/output declarations on the same line.
  - Avoid creating circular dependencies in your netlist.

Example:

```
# Full-Adder in BENCH Format

INPUT(A)
INPUT(B)
INPUT(Cin)

OUTPUT(Sum)
OUTPUT(Cout)

XOR(X1, A, B)
AND(A1, A, B)
XOR(Sum, X1, Cin)
AND(A2, X1, Cin)
OR(Cout, A1, A2)
```

```
# c17 in BENCH Format

# 5 inputs
# 2 outputs
# 0 inverter
# 6 gates ( 6 NANDs )

INPUT(1)
INPUT(2)
INPUT(3)
INPUT(6)
INPUT(7)

OUTPUT(22)
OUTPUT(23)

10 = NAND(1, 3)
11 = NAND(3, 6)
16 = NAND(2, 11)
19 = NAND(11, 7)
22 = NAND(10, 16)
23 = NAND(16, 19)
```