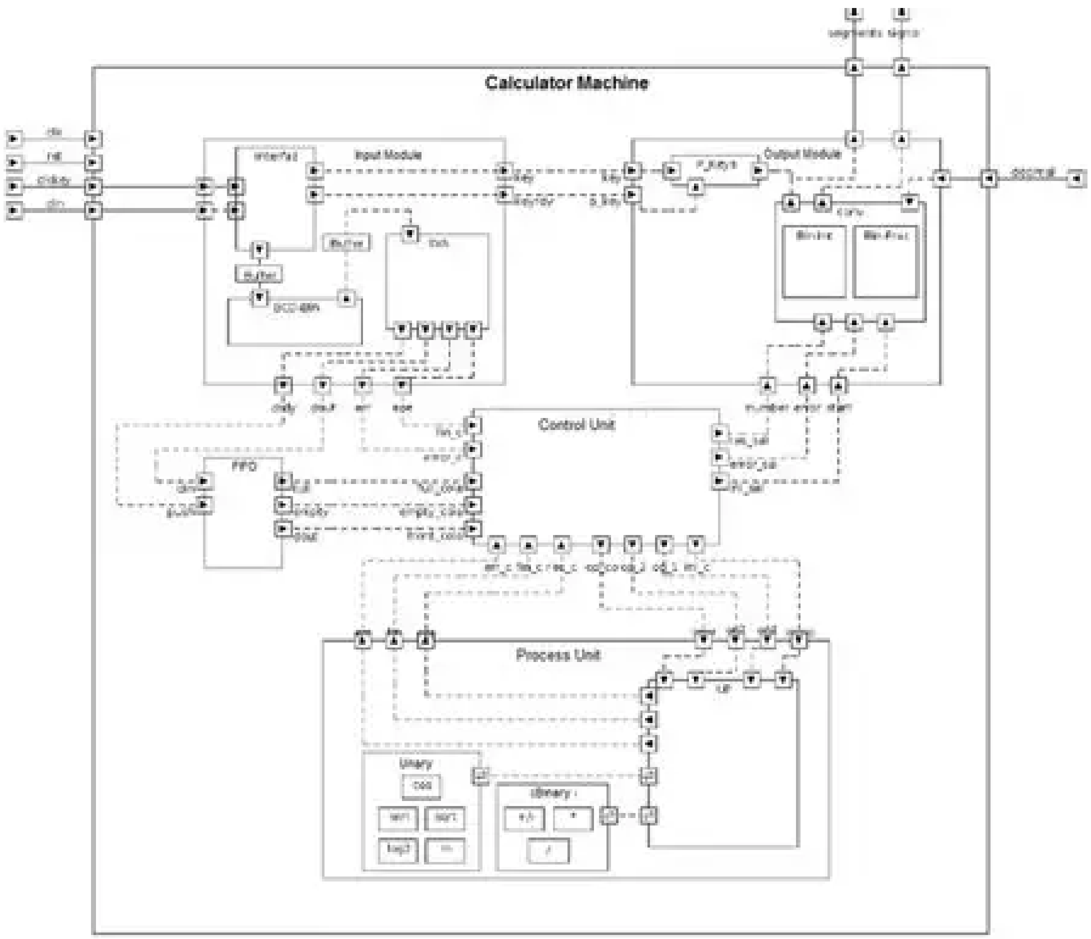
INTERNAL COMPONENTS AND WORKING FLOW

INTRODUCTION:

A **calculator** is typically a portable electronic device used to perform calculations, ranging from basic arithmetic to complex mathematics.

The first solid-state electronic calculator was created in the early 1960s. Pocket-sized devices became available in the 1970s, especially after the Intel 4004, the first microprocessor, was developed by Intel for the Japanese calculator company Busicom. Modern electronic calculators vary from cheap, give-away, credit-card-sized models to sturdy desktop models with built-in printers. They became popular in the mid-1970s as the incorporation of integrated circuits reduced their size and cost. By the enddecade, prices had dropped to the point where a basic calculator was affordable to most acceptable in school.

INTERNAL BLOCK DIAGRAM(FUNCTIONAL VIEW)


Calculator Machine
Input Module
Output Module
Control Unit
FIFO
Process Unit
Unary
Binary
segments
clk
rst
Buffer
number
error
start
empty
dout
full

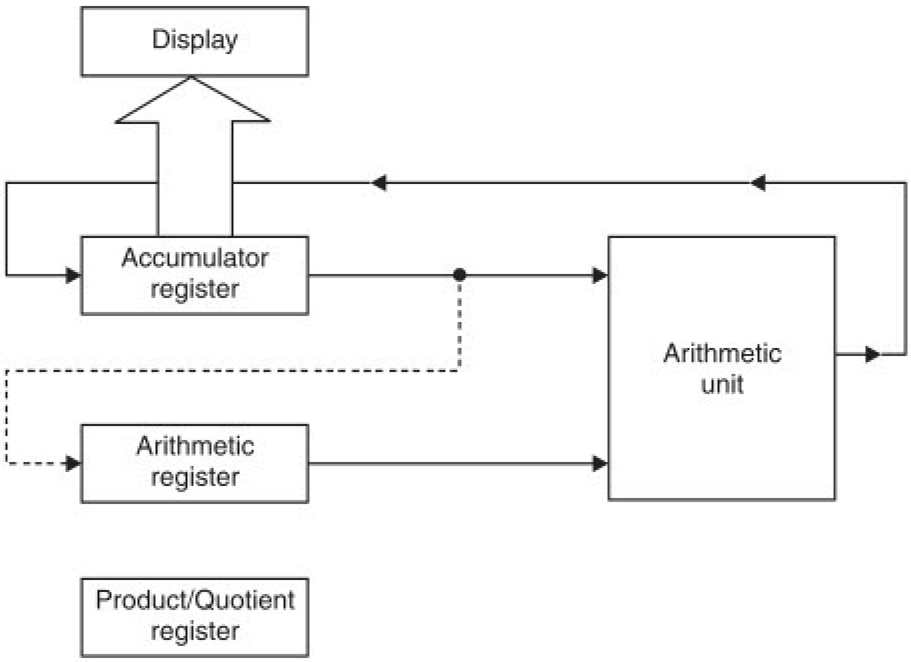
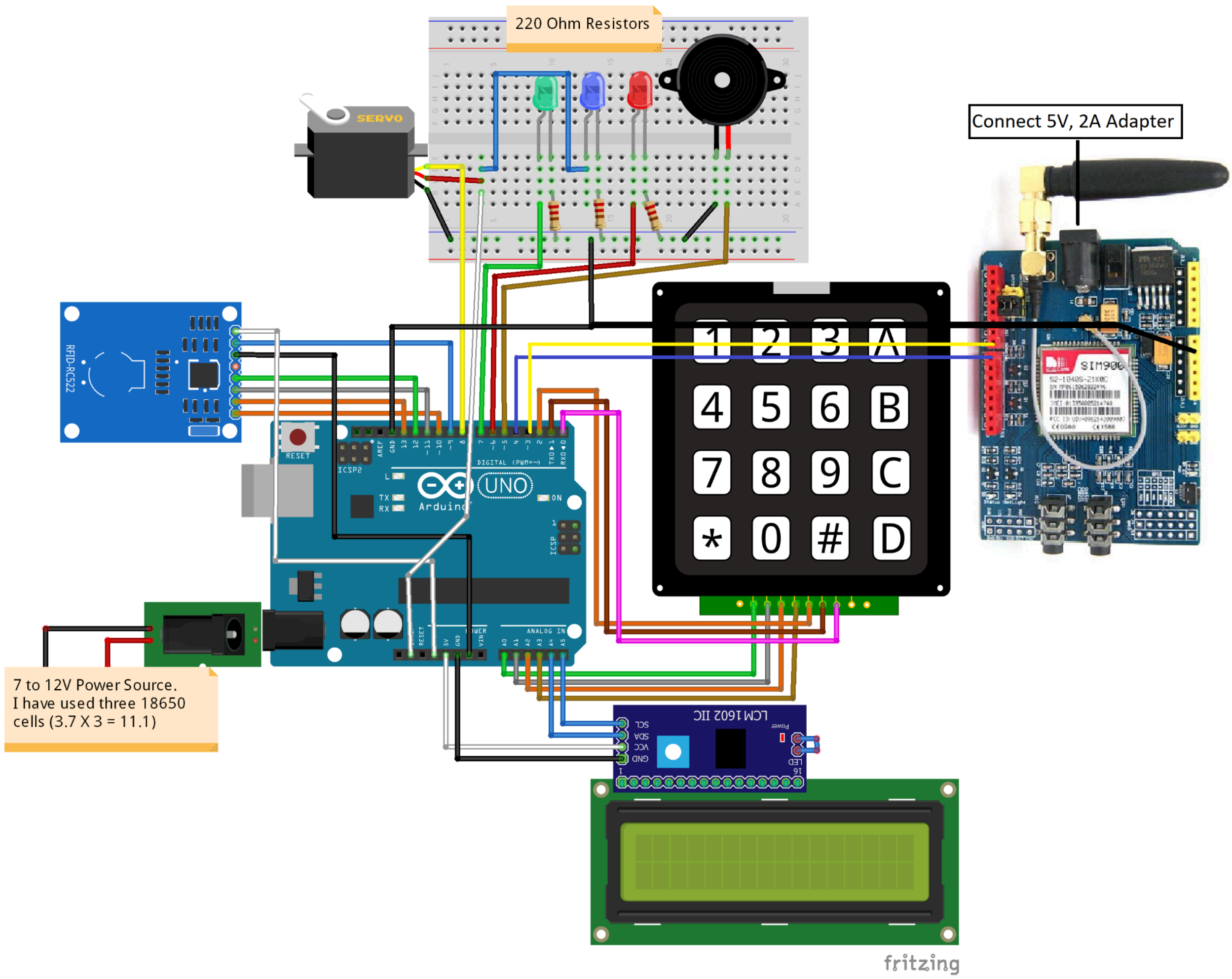


Fig. 46.2 *Block diagram showing register structure of a calculator*

Internal Components of a Calculator (Simple)

1. **Keypad**
   - Used to press numbers and symbols.
2. **Processor (CPU)**
   - Thinks and solves the sums.
3. **ALU**
   - Does the math like +, −, ×, ÷.
4. **Memory**
   - Remembers numbers while calculating.
5. **Display Screen**
   - Shows the numbers and answers.

6. **Battery / Solar Cell**
   - Gives power to the calculator.



INTERNAL COMPONENTS &DETAILED EXPLANATIONS

Below is a **clear breakdown of the internal components of a calculator** with a **detailed explanation of how each part works together**. This applies mainly to **basic electronic calculators** (the kind used in schools), but I'll also briefly note differences for scientific calculators.

---

1. Power Supply

Purpose: Provides electrical energy to run the calculator.

Components

- **Battery (button cell / AAA)** or **solar cell**
- **Voltage regulator**

How it works

- The battery or solar panel supplies DC power.
- The voltage regulator ensures a stable voltage for sensitive electronic components.
- In solar calculators, excess light charges the battery or directly powers the circuit.

---

2. Keyboard (Input Unit)

Purpose: Allows the user to enter numbers and operations.

Components

- **Keypad matrix**
- **Conductive rubber contacts**
- **Printed circuit traces**

How it works

- Keys are arranged in rows and columns.
- When a key is pressed, it connects a specific row and column.
- The processor scans the matrix to identify which key was pressed (e.g., 5, +, =).

---

3. Processor / Integrated Circuit (IC)

Purpose: The “brain” of the calculator.

Components

- **Microcontroller or custom calculator IC**
- **Control logic**

- **Timing circuit (clock)**

How it works

- Receives key inputs from the keyboard.
- Interprets the operation (addition, subtraction, etc.).
- Executes arithmetic logic instructions.
- Sends results to the display.

👉 In **scientific calculators**, the IC also handles:

- Trigonometric functions
- Logarithms
- Memory operations
- Floating-point arithmetic

---

4. Arithmetic Logic Unit (ALU)

Purpose: Performs mathematical calculations.

Operations performed

- Addition
- Subtraction
- Multiplication
- Division
- Logical operations (comparisons, shifts)

How it works

- Numbers are converted into binary.
- The ALU processes them using logic gates.
- Results are passed back to the processor for display or further operations.

---

5. Memory Unit

Purpose: Stores data temporarily or permanently.

Types of memory

- **RAM:** Stores current calculations and user inputs.
- **ROM:** Stores the calculator’s firmware (built-in instructions).
- **Registers:** Store intermediate results.

How it works

- ROM contains the program that tells the calculator how to function.
- RAM holds numbers during calculations.
- Memory buttons (M+, M-, MR, MC) interact with memory registers.

---

6. Display Unit (Output Unit)

Purpose: Shows numbers and symbols to the user.

Types of displays

- **LCD (Liquid Crystal Display)** – most common
- **LED display** – older calculators

Components

- **Segmented digits (7-segment or 14-segment)**
- **Driver circuits**

How it works

- The processor sends signals to specific segments.
- LCD crystals align to block or allow light.
- Numbers, decimal points, and symbols appear.

---

7. Clock / Timing Circuit

Purpose: Synchronizes operations.

How it works

- Generates clock pulses.
- Ensures instructions are executed in the correct order.
- Controls display refresh and keypad scanning.

---

8. Printed Circuit Board (PCB)

Purpose: Physically connects all components.

How it works

- Copper traces act as electrical paths.
- Components are soldered or mounted.
- Ensures compact and reliable operation.

---

9. Casing and Mechanical Parts

Purpose: Protection and usability.

Components

- Plastic casing
- Keycaps
- Display window

How it works

- Protects internal electronics.
- Provides ergonomic key layout.
- Shields components from dust and damage.

Working Principle (Step-by-Step Example)

Input: `8 + 5 =`

1. Keyboard sends key signals to the processor.

2. Processor stores 8 and 5 in memory.
3. ALU performs addition (`8 + 5 = 13`).
4. Result is stored briefly in registers.
5. Display driver shows 13 on the LCD.

Difference Between Basic and Scientific Calculators

Feature	Basic Calculator	Scientific Calculator
Processor	Simple IC	Advanced microcontroller
Functions	+, −, ×, ÷	Trig, log, exp, stats
Memory	Small	Larger RAM/ROM
Display	7-segment	Multi-line dot-matrix

Absolutely! Let's go **deep dive, step by step**, into the **complete working flow of a calculator**, from turning it on to getting the result, including the internal mechanisms. I'll break it into **fine-grained steps**.

COMPLETE WORKING FLOW(STEP-BY-STEP)

Step 1: Power On / Initialization

1. User presses the **ON** button.
2. The calculator's **power circuit** supplies voltage to:
   - Microcontroller / CPU
   - Display (LCD or LED)
   - Keypad interface
3. **Internal memory/registers** are initialized to 0:
   - `Operand1 = 0`
   - `Operand2 = 0`
   - `Operator = None`
4. The display shows `0` or blank, indicating readiness for input.

Step 2: Input First Number

1. User presses numeric keys (`0–9`).
2. Each key generates a **digital signal** via the **keypad matrix**.
3. The signal is interpreted by the **controller** and stored sequentially in a buffer:
    - If user presses `1` then `2` → buffer = `12`
4. **Display updates** in real time to show the number being typed.
5. Optional features at this stage:
    - Decimal point (`.`) → allows floating-point numbers
    - Negation (`+/-`) → allows negative numbers

Step 3: Input Operator

1. User presses an operator key (`+`, `−`, `×`, `÷`).
2. Controller stores:
    - `Operand1` = buffer value (first number entered)
    - `Operator` = the operator key pressed
3. Buffer for **second number** is cleared to prepare for the next input.
4. Display may optionally show the operator to indicate the ongoing operation.

Step 4: Input Second Number

1. User enters the second operand via numeric keys.
2. The second number is stored in a **new buffer** (Operand2).
3. Display shows the second number in real time.
4. Optional features:
    - Decimal input
    - Sign change
    - Backspace for corrections

Step 5: Calculation Execution (Equals `=` Pressed)

1. User presses the `=` key.
2. Controller sends data to the **Arithmetic Logic Unit (ALU)**:
    - `Operand1`, `Operand2`, `Operator`

3. ALU performs the calculation based on the operator:
    - `+` → Addition
    - `-` → Subtraction
    - `×` → Multiplication
    - `÷` → Division
4. Special cases:
    - Division by zero → triggers **error handling**
    - Overflow / underflow → may trigger `Error` or `Overflow`

Step 6: Result Handling

1. ALU returns the result to the controller.
2. Controller updates **internal registers**:
    - `Operand1 = Result` (useful for chain calculations)
    - `Operand2 = 0`
    - `Operator = None`
3. Result is displayed on the screen.
4. User can now:
    - Start a **new calculation**
    - Use the result as `Operand1` for further calculations
    - Clear the calculator (`C` / `AC`)

Step 7: Continuous / Chained Operations

1. After a calculation, user can press another operator directly:
    - Calculator takes the **previous result as Operand1** automatically
2. Repeat steps 4–6 for new calculation.
3. Display shows updated results in real time.

Step 8: Error Handling

1. During any operation, the controller checks for:
    - Division by zero → displays `Error`
    - Invalid input sequences (e.g., `++`, `÷=`) → ignore or beep
    - Number overflow → displays `Overflow` or `Error`

2. If an error occurs, user may press `C` to reset.

Step 9: Power Off

1. User presses `OFF` button.
2. Controller:
   - Clears all buffers
   - Powers down ALU
   - Turns off display

Step 10: Optional Advanced Features (Scientific Calculator)

- Trigonometric functions (`sin`, `cos`, `tan`)
- Logarithmic / exponential calculations
- Memory functions (`M+`, `MR`, `MC`)
- Parentheses for complex expressions
- These add additional steps:
  - Parse function/operator
  - Compute using internal formulas
  - Update memory and display

Step-by-Step Flow Diagram (Text)

Power On

↓

Enter First Number

↓

Press Operator

↓

Enter Second Number

↓

Press "="

↓

ALU Computes Result

↓

Display Result

↓

Continue / Chain Calculation OR Clear

↓

Error Handling (if needed)

↓

Power Off

Component	Description	Example / Notes
Input (x)	Values chosen to plug into the function (independent variable).	x = 0, 1, 2, 3…
Output (f(x) or y)	Values calculated from the function for each input.	If $f(x) = x^2$, then $f(2)=4$
Function Rule	The formula or equation that defines the function.	$f(x) = 2x + 3$

Step / Increment (Δx)	The amount by which x increases for each row in the table.	1, 0.5, etc.
Table Range (Start / End)	The starting and ending x-values displayed in the table.	Start = 0, End = 5
Additional Functions	Ability to display multiple functions simultaneously in the table.	f(x) and g(x)
Auto-calculated Column	Some calculators show extra calculations like $f(x)^2$, $\sqrt{f(x)}$, or f(x)+g(x).	Optional column
Row Number / Index	Some tables number each row for reference.	Row 1, Row 2, …
Formatting Options	Options for displaying decimal places, fractions, or scientific notation.	3.14, 22/7, 1e3
Error Indicators	Shows if a value is undefined (like division by 0) or exceeds calculator limits.	"ERR" or blank
Graph Link / Shortcut	Many graphing calculators allow jumping from the table to the corresponding graph point.	Trace feature
Memory / Storage	Some calculators allow saving the table for later review or export.	Recall table