Software Design Document

Base Conversion Program (MIPS)

Page 1|5 Programming Fundamentals 3

Arnold Archaga

1. Introduction

1.1. Goals and Requirements

* Convert a user‑entered integer from base **b** (2–10) to its decimal equivalent (base 10).
* Prompt the user for the number of digits **n**.
* Prompt the user for the base **b** (must be between 2 and 10).
* Prompt the user **n** times for each digit **Xᵢ** (from least significant, i = 0, to most significant, i = n−1).
* Validate every digit: if **Xᵢ ≥ b**, display an error and reprompt for that same digit.
* Compute the decimal result using: Num₁₀ = Σᵢ₌₀ⁿ⁻¹ (Xᵢ × bⁱ)
* Display “Decimal result = ” upon successful conversion.
* Implementation in MIPS Assembly; must assemble and run without errors in QtSpim or MARS.

1.2. Product Scope This console application converts positional numbers in bases 2 through 10 into decimal. It demonstrates MIPS proficiency with loops, branching, arithmetic operations, and basic error handling. Users interact via simple prompts in the simulator’s console.

P a g e 2 | 5 Programming Fundamentals 3. Arnold Archaga

1. Design Overview

2.1. Pseudo Code

BEGIN program

PRINT "Enter number of digits (n):"

READ n

PRINT "Enter base (2..10):"

READ b

decimalResult ← 0

i ← 0

WHILE i < n DO

PRINT "Enter digit X" + i + ":"

READ Xi

IF Xi ≥ b THEN

PRINT "ERROR: Digit not valid in this base."

CONTINUE # repeat same i

ENDIF

power ← 1

counter ← 0

WHILE counter < i DO

power ← power × b

counter ← counter + 1

ENDWHILE

decimalResult ← decimalResult + (Xi × power)

i ← i + 1

ENDWHILE

PRINT "Decimal result = " + decimalResult

END program

2.2. Flow Charts

* **Step 1:** Read the number of digits (n) and base (b).
* **Step 2:** Initialize decimalResult = 0 and i = 0.
* **Step 3:** While i < n:
  + Prompt “Enter digit Xi:” and read Xi.
  + If Xi ≥ b:
    - Print error message and repeat the same i.
    - Continue to next iteration of the loop.
  + Else:
    - Compute power = b^i by:
      * power = 1; counter = 0;
      * While counter < i: power \*= b; counter++.
    - Add Xi \* power to decimalResult.
    - Increment i.
* **Step 4:** After loop ends, print “Decimal result = ”.

P a g e 3 | 5 programming Fundamentals 3. Arnold Archaga

1. Class Diagram

***MAIN PROGRAM Module***  
• Reads inputs: n (number of digits), b (base), and each digit Xi.  
• Validates each digit (< b).  
• Accumulates decimalResult using power\_function.

***power\_function (inline)***  
• Computes b^i by repeated multiplication (loop).

P a g e 4 | 5 Programming Fundamentals 3 Arnold Archaga

1. User Interface

* **Prompts**:
  1. “Enter number of digits (n):”
  2. “Enter base (2..10):”
  3. Repeatedly: “Enter digit Xi:”
* **Validation**: If digit ≥ base, prints “ERROR: Digit not valid in this base.” and reprompts the same digit index.
* **Output**: After valid sequence, prints “Decimal result = ”.
* **Execution**: Run in QtSpim or MARS; user interacts via the simulator’s console.

P a g e 5 | 5 Programming Fundamentals 3 Arnold Archaga

5. Testing

**Test Case 1**

* **Input:** n = 4, b = 2, digits = 1, 0, 1, 0
* **Expected:** Decimal result = 10
* **Actual:** Decimal result = 10
* **Status:** Pass

**Test Case 2**

* **Input:** n = 3, b = 5, digits = 4, 3, 2
* **Expected:** Decimal result = 69
* **Actual:** Decimal result = 69
* **Status:** Pass

**Test Case 3**

* **Input:** n = 1, b = 10, digits = 7
* **Expected:** Decimal result = 7
* **Actual:** Decimal result = 7
* **Status:** Pass

**Test Case 4 (Invalid Digit)**

* **Input:** n = 3, b = 2, digits = 1, 2, 0
* **Expected:** Error printed: “ERROR: Digit not valid in this base.”, then Decimal result = 1
* **Actual:** Error printed once; Decimal result = 1
* **Status:** Pass

**Discussion:** These scenarios cover normal conversions (binary, base‑5, single digit) and error handling (invalid digit). All actual outputs matched expectations, confirming the program meets the requirements from Section 1.

Below is the Base Conversion Skeleton and Final

https://github.com/arnoldarchaga/MIPSBaseConversion-Final