2026 Digital IC Design

Homework 2: Modular Multiplier

1. Introduction:

This homework aims to design a resource-efficient modular multiplier for two 23-bit inputs under a fixed modulus Q=8380417. The circuit avoids costly division by using constant multipliers and bitwise operations. You will implement a combinational module **Mul_Mod** that outputs **Z=(A×B) mod Q**, suitable for cryptographic applications like Kyber or Dilithium.

1.1 23-bit Dual-Input Mul_Mod

The logic diagram of the Mul_Mod circuit is shown in Fig. 1, and its I/O specification is listed in Table I. The module performs modular multiplication between two 23-bit unsigned inputs A and B, producing a 24-bit result Z = (A × B) mod 8380417. The design uses partial multipliers, shifts, and constant multiplications to efficiently implement the operation without full-width multiplication or division. **In this homework, you must design the adder module in Mul_Mod using Ripple Carry Adder (RCA) in homework 1.**

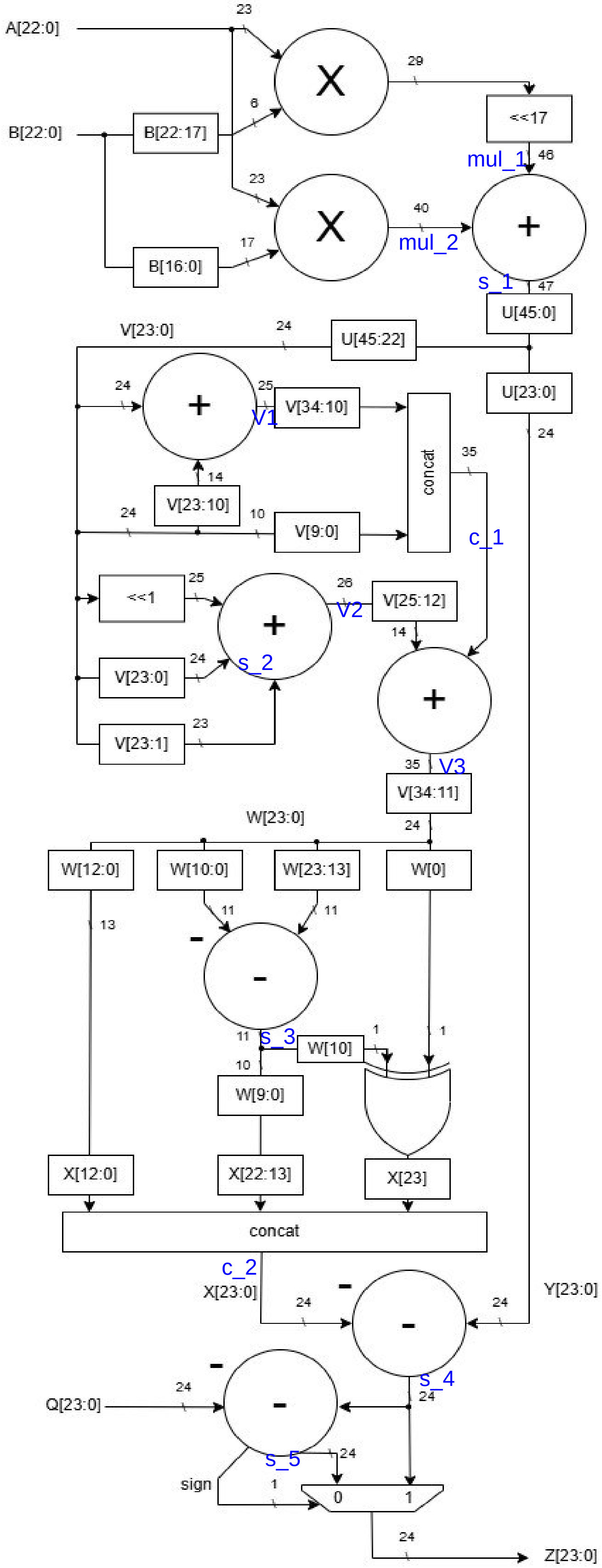


Fig. 1 Logic diagram of the Mul_Mod.

Table I. Specification and I/O interface of Mul_Mod

Signal Name	I/O	Width	Description
A	I	23	First 24-bit input operand
B	I	23	Second 24-bit input operand
Z	O	24	The result of modular multiplier

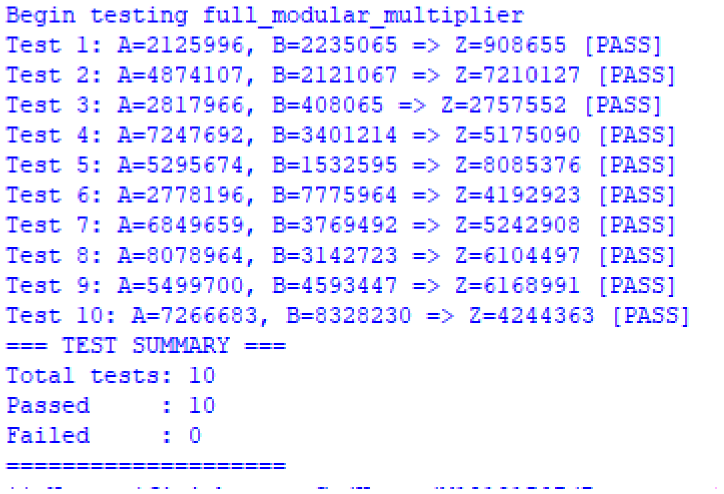
1.2 File Description

File Name	Description
Mul_Mod.v	The module of 23-bit Modular Multiplier.
testfixture.sv	Testbench file. This file **is not allowed** to be modified.
golden.dat	Test and golden data file for Mul_Mod.

2. Scoring:

2.1 23-bit Dual-Input Mul_Mod [100%]

The result should be generated correctly, and you will get the following message in ModelSim simulation.



```
Begin testing full_modular_multiplier
Test 1: A=2125996, B=2235065 => Z=908655 [PASS]
Test 2: A=4874107, B=2121067 => Z=7210127 [PASS]
Test 3: A=2817966, B=408065 => Z=2757552 [PASS]
Test 4: A=7247692, B=3401214 => Z=5175090 [PASS]
Test 5: A=5295674, B=1532595 => Z=8085376 [PASS]
Test 6: A=2778196, B=7775964 => Z=4192923 [PASS]
Test 7: A=6849659, B=3769492 => Z=5242908 [PASS]
Test 8: A=8078964, B=3142723 => Z=6104497 [PASS]
Test 9: A=5499700, B=4593447 => Z=6168991 [PASS]
Test 10: A=7266683, B=8328230 => Z=4244363 [PASS]
=== TEST SUMMARY ===
Total tests: 10
Passed     : 10
Failed     : 0
====================
```

Fig. 2 Simulation result for Mul_Mod

3. Submission

3.1. File Submission

You should classify your files into two directories and compress them to .zip format. The naming rule is StudentID.zip. If your file is not named according to the naming rule, you will lose five points.

	RTL
*.v	All of your Verilog RTL code
	Documentary
Report.pdf	The report file of your design (in pdf).

Fig. 3 File hierarchy

- 📎 StudentID.zip
  - 📁 StudentID
    - 📁 RTL
      - 🗎 *.v (All of your Verilog RTL code)
    - 📁 Documentary
      - 🗎 Report.pdf

3.2. Report File

Please follow the spec of report. You are asked to describe how the circuit is designed as detailed as possible.

3.3. Notes

a. Please submit your .zip file to folder HW2 in moodle.
   Deadline: 2025/10/12 23:55

b. Late submission will only be accepted within a week and will result in a penalty of 5 points per day.

c. TA email: p76134799@gs.ncku.edu.tw