National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

BCD Multiplier

Semester Project: Digital Logic Design

Course: BEE-14D

Teacher: Sir Arshad Nazir

Submitted by:

Irfa Farooq 412564

Sataish Elahi 423621

Dedication

Throughout our end-semester project, both my groupmate and I displayed unwavering dedication, forging a relentless commitment toward our shared goals. We consistently invested substantial time and effort, often going above and beyond to ensure the project's success. Our dedication was evident in our tireless collaborative efforts, spending late nights refining concepts, meticulously analyzing data, and meticulously iterating through numerous prototypes to achieve optimal results. Despite challenges and setbacks, our dedication remained steadfast, propelling us to overcome obstacles with resilience and determination. Our unwavering focus on the project not only showcased our passion but also exemplified our commitment to delivering a project that reflects our utmost dedication and hard work.

Abstract

This project presents a novel approach to Binary Coded Decimal (BCD) multiplication, employing a down counter to regulate the shifting mechanism within a series register and integrating an adder to compute the final product. The design involves utilizing a down counter to manage the shifting cycles within the series register, allowing precise control over the multiplication process. The down counter's terminal count acts as a signal to halt the shifting sequence, effectively storing the resultant product in the series register. Furthermore, an adder unit is integrated to accumulate and compute the overall product by summing up the partial products stored within the register. This approach optimizes the multiplication process, leveraging the controlled series register and adder to generate accurate BCD multiplication results efficiently. The project underscores the synergy between the down counter, series register, and adder, showcasing their integration for enhanced precision and efficiency in BCD multiplication operations.

List of Figures:

Table of Contents

Contents

Chapter 1: Introduction

1.1 Overview of Project

The Sequential BCD Multiplier project embodies an intricate fusion of combinational and sequential logic circuits, aiming to execute efficient Binary Coded Decimal (BCD) multiplication operations while strictly adhering to BCD constraints.

Central to the project's architecture is the robust combinational logic circuit, meticulously engineered to filter and validate input data. This circuit serves as the gatekeeper, meticulously examining incoming numbers to ensure adherence to BCD standards. Only valid BCD numbers within the range of 0 to 9 are permitted for multiplication. Any input exceeding this range, such as the number 1001, triggers the circuit to halt the operation, substituting all zeros to maintain precision in subsequent calculations.

The project's sequential logic circuit consists of three key components: the parallel in parallel out register for the multiplier, the parallel in series out register for the multiplicand and the final register for storing product. The multiplier, representing the first BCD number, is channeled into the parallel in parallel out register, where it is efficiently stored and prepared for manipulation. Simultaneously, the multiplicand, the second BCD number, enters the parallel in series out register, initially storing the value and subsequently engaging a controlled shifting mechanism. When activated, each bit of the multiplicand sequentially shifts rightward, orchestrating the multiplication process. Upon completion, the register resets to an all-zero state, ensuring readiness for subsequent calculations. The final product register contains a combinational logic at the input of each flip flop input. When the circuit is enabled, the register acts as a parallel series out register. As soon as the counter reaches zero, the combinational circuit forces the register to act as a parallel in parallel out register that restores its own value forcing it to stop the shift and store the overall value in the register.

Integral to the project's functionality is the inclusion of a counter designed to truncate the final answer stored within the shift register. After the multiplication process concludes and the resultant product is accumulated within the series register, the counter triggers an operation to truncate or crop the final answer to the desired length, ensuring precision and accuracy in the output.

1.2 Block Diagram of Complete System (without using ICs, just use simple blocks)

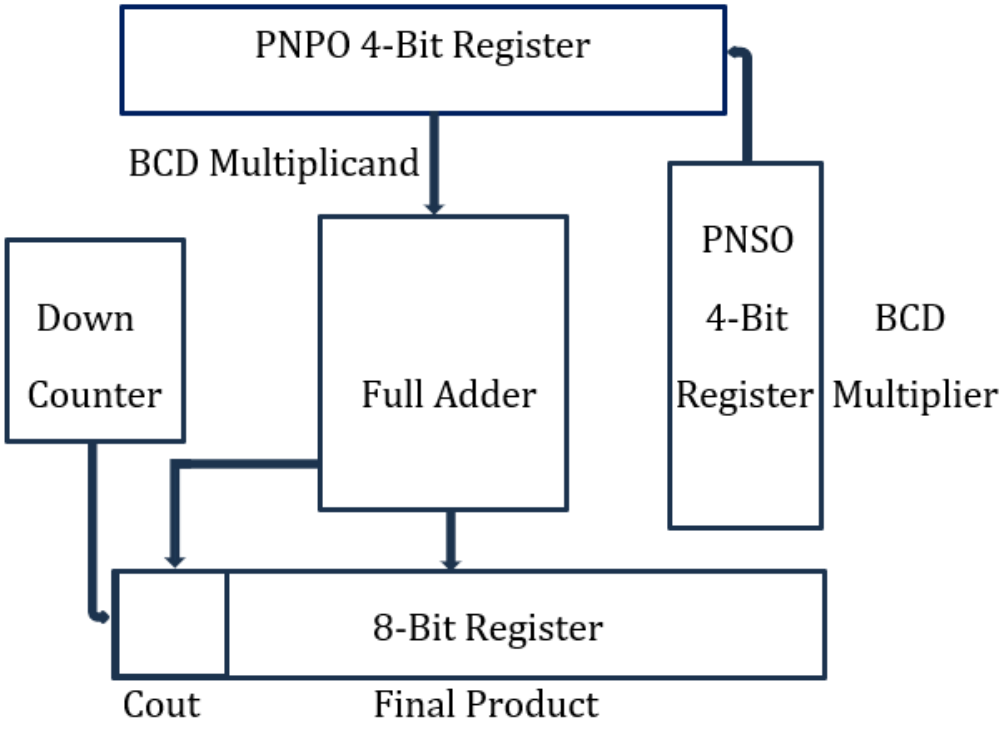


Figure 1: Block Diagram

1. Clear Work Division

Clearly describe the work of each member with the help of a block diagram if possible. Or give a new figure illustrating work division.

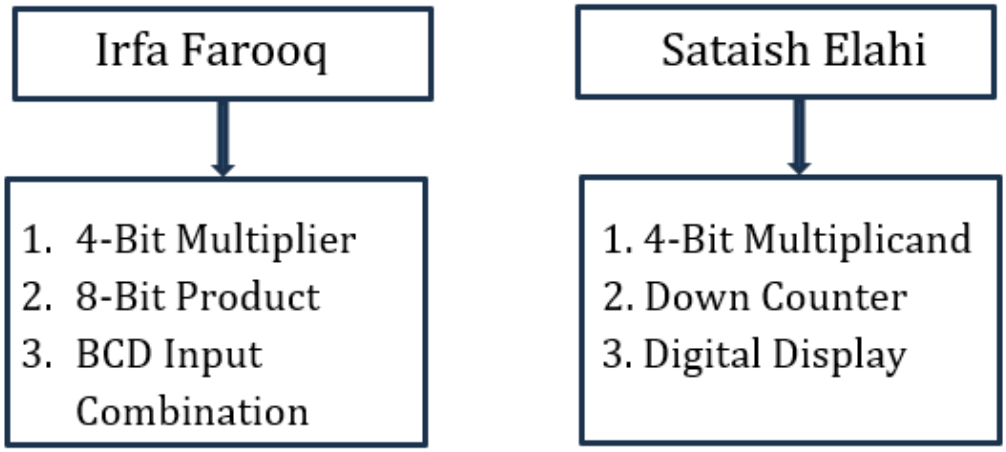


Figure 2: Work Division

Chapter 2: Design

2.1 Problem Statement

Design a 4-bit Binary Coded Decimal (BCD) sequential multiplier circuit capable of displaying the multiplication result on a digital display. The circuit should efficiently process two 4-bit BCD-encoded numbers, utilizing sequential logic components to perform accurate multiplication operations. The multiplier must incorporate a parallel in parallel out register for one BCD input and a parallel in series out register for the other input, facilitating controlled shifting and storage mechanisms. Ensure the output of the multiplication process is displayed on a digital display, adhering strictly to BCD standards. The circuit should validate inputs to accept only valid BCD numbers (0-9) and truncate the final result if it exceeds the 4-bit display capacity. Aim for precision, efficiency, and clear visualization of the multiplication outcome on the digital display.

2.2 Truth Table / State Diagram

Truth Table for accepting only BCD values as inputs in registers.

Input	Binary Code				BCD Input			
	W	X	Y	Z	A3	A2	A1	A0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	0
3	0	0	1	1	0	0	1	1
4	0	1	0	0	0	1	0	0
5	0	1	0	1	0	1	0	1
6	0	1	1	0	0	1	1	0
7	0	1	1	1	0	1	1	1
8	1	0	0	0	1	0	0	0
9	1	0	0	1	1	0	0	1
10	1	0	1	0	0	0	0	0
11	1	0	1	1	0	0	0	0
12	1	1	0	0	0	0	0	0
13	1	1	0	1	0	0	0	0
14	1	1	1	0	0	0	0	0
15	1	1	1	1	0	0	0	0

Figure3: Truth Table for valid BCD Input

State diagram for BCD down counter:

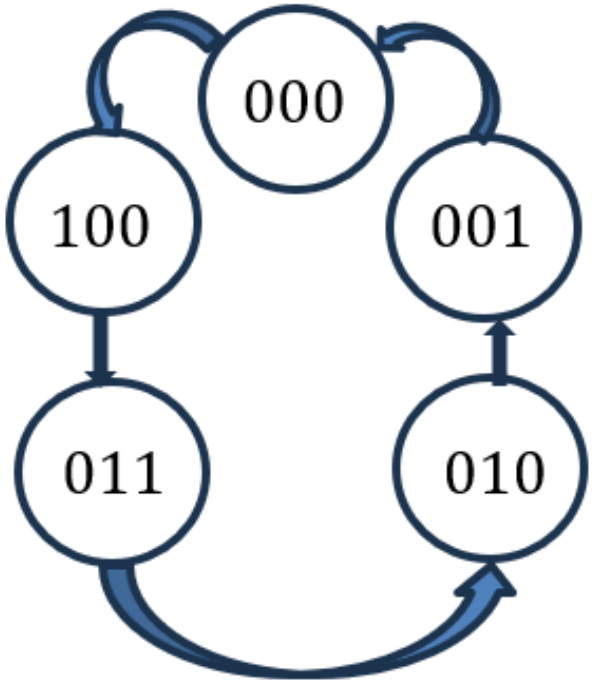


Figure 4: State Diagram for 4-0 BCD Down counter

2.3 State Table If Applicable

State Table for Output 8-Bit Shift Register's one D Flip Flop.

Present State	Inputs		Next State	Output
Cout	PCout	Counter	Cout*	S
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	1
1	0	0	1	1
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

Figure 5: State Table for Output 8-Bit Shift Register

State Table for BCD Down Counter

Minterms	Present State			Inputs			Next State			Output
	C2	C1	C0	T2	T1	T0	C2*	C1*	C0*	
0	0	0	0	1	0	0	1	0	0	0
4	1	0	0	1	1	1	0	1	1	0
3	0	1	1	0	0	1	0	1	0	0
2	0	1	0	0	1	1	0	0	1	0
1	0	0	1	0	0	1	0	0	0	1

Figure 6: State Table for BCD Down Counter

2.4 Simplification of Functions / K-Maps & Equations

1. KMaps for BCD Input Validation:

yz / wx	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	1	1	0	0
A3 = wx'y'				

yz / wx	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	0	0	0	0
A2 = w'x				

yz / wx	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	0	0
A1 = w'y				

yz / wx	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	0	0	0	0
10	0	1	0	0
A0 = w'z + x'y				

2. Product 8-Bit Register D Flip Flop Inputs:

Counter \ PCout.Cout	00	01	11	10
0	0	1	1	0
1	0	0	1	1
Cout* = Counter.PCout + Counter'.Cout				

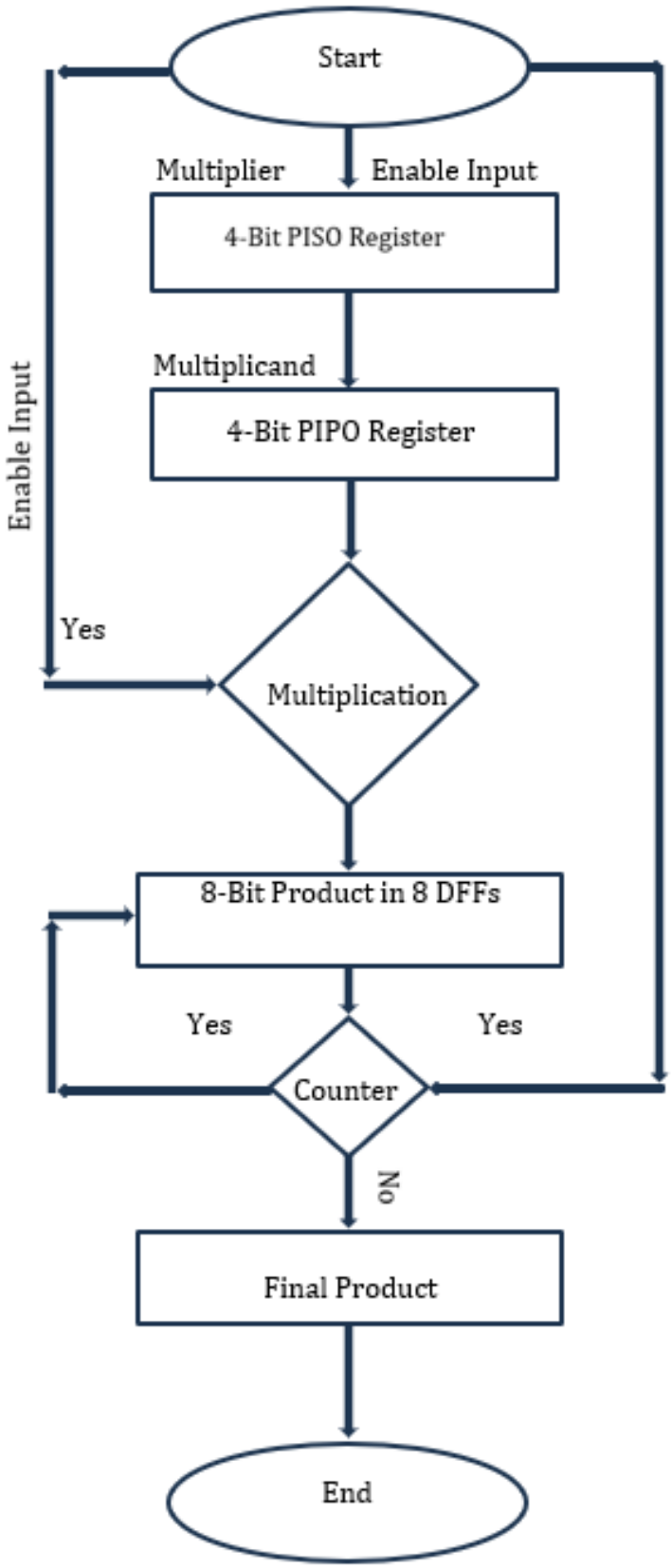
3. BCD Down Counter Inputs:

C2 \ C1C0	00	01	11	10
00	1	0	0	0
01	1	0	0	0
T2 = C1'C0'				

C2 \ C1C0	00	01	11	10
00	0	0	1	0
01	1	0	0	0
T1 = C2'C1C0' + C2C1'C0'				

C2 \ C1C0	00	01	11	10
00	0	1	1	1
01	1	1	1	1
T0 = C2 + C1 + C0				

2.5 Complete Logic Diagram



2.6 Simulations in parts

1. Parallel in parallel out 4-Bit Register for BCD Input and Checker:

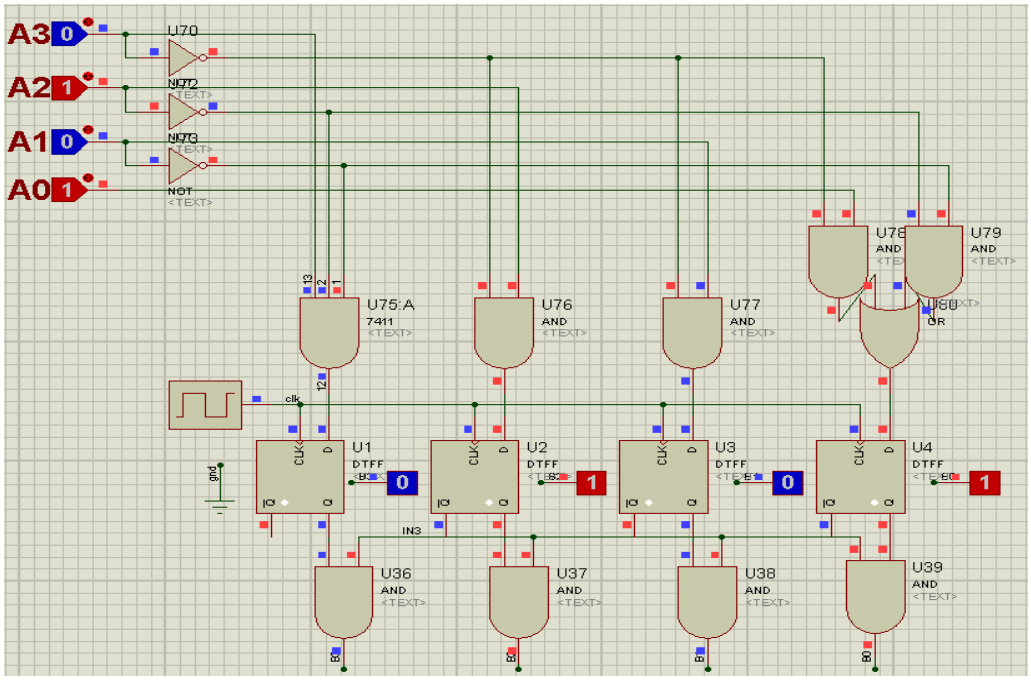


Figure 7: 4-Bit Multiplicand

2. Parallel in serial out 4-Bit Register for BCD Input and Switch enable:

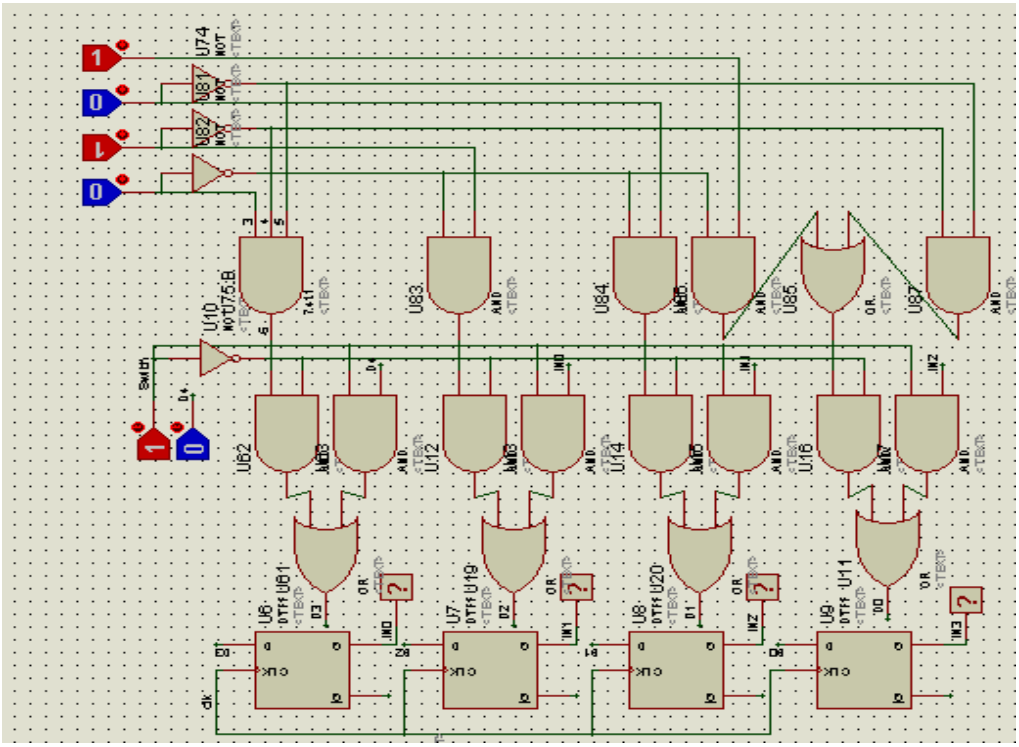


Figure 8: 4-Bit Multiplier

3. Full Adder with enable input and right shift:

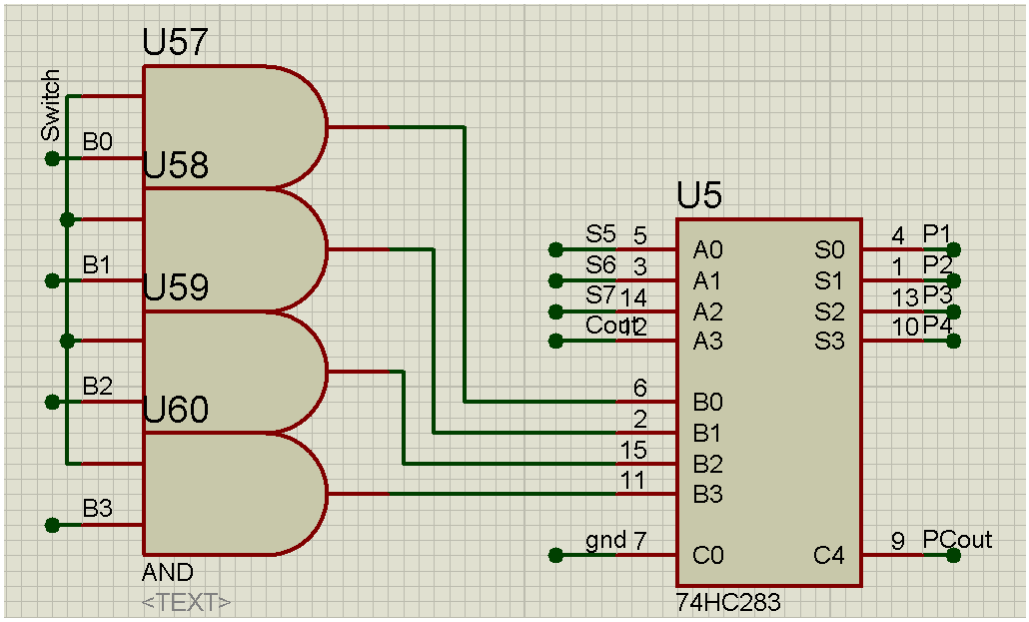


Figure 9: 4-Bit adder with enable input

4. 4 – 0 Down Counter that acts as enable input for 8-Bit Product:

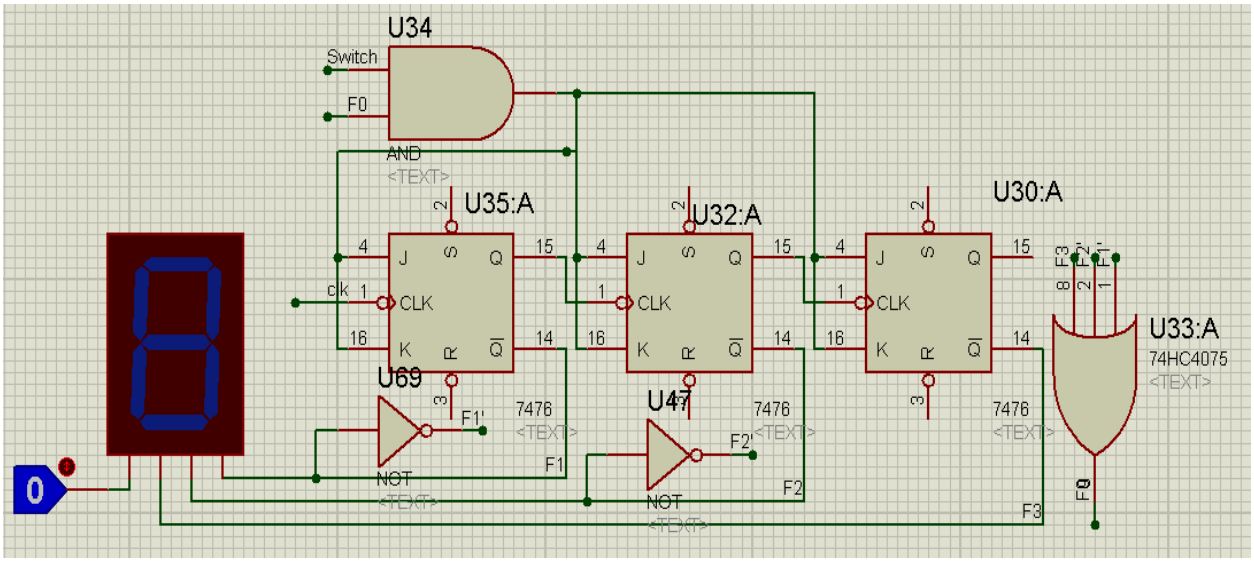


Figure 10: 4-0 down Counter

5. Final Product 8-Bit register using 8 DFFs:

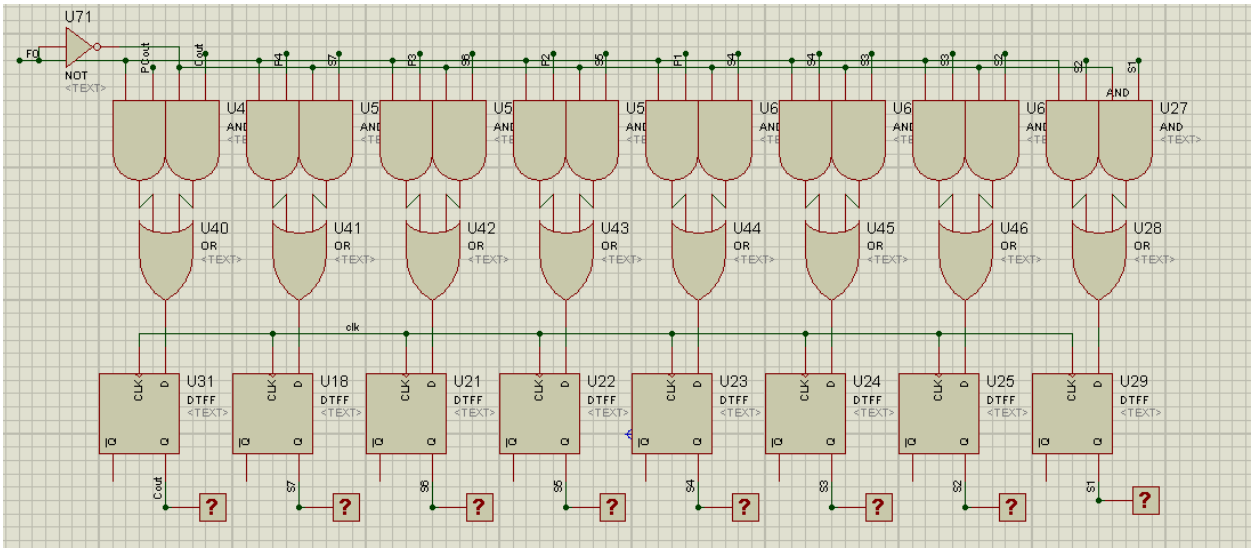


Figure 11: 8-Bit Product Register with count enable

6. 8-Bit to BCD converter and Display

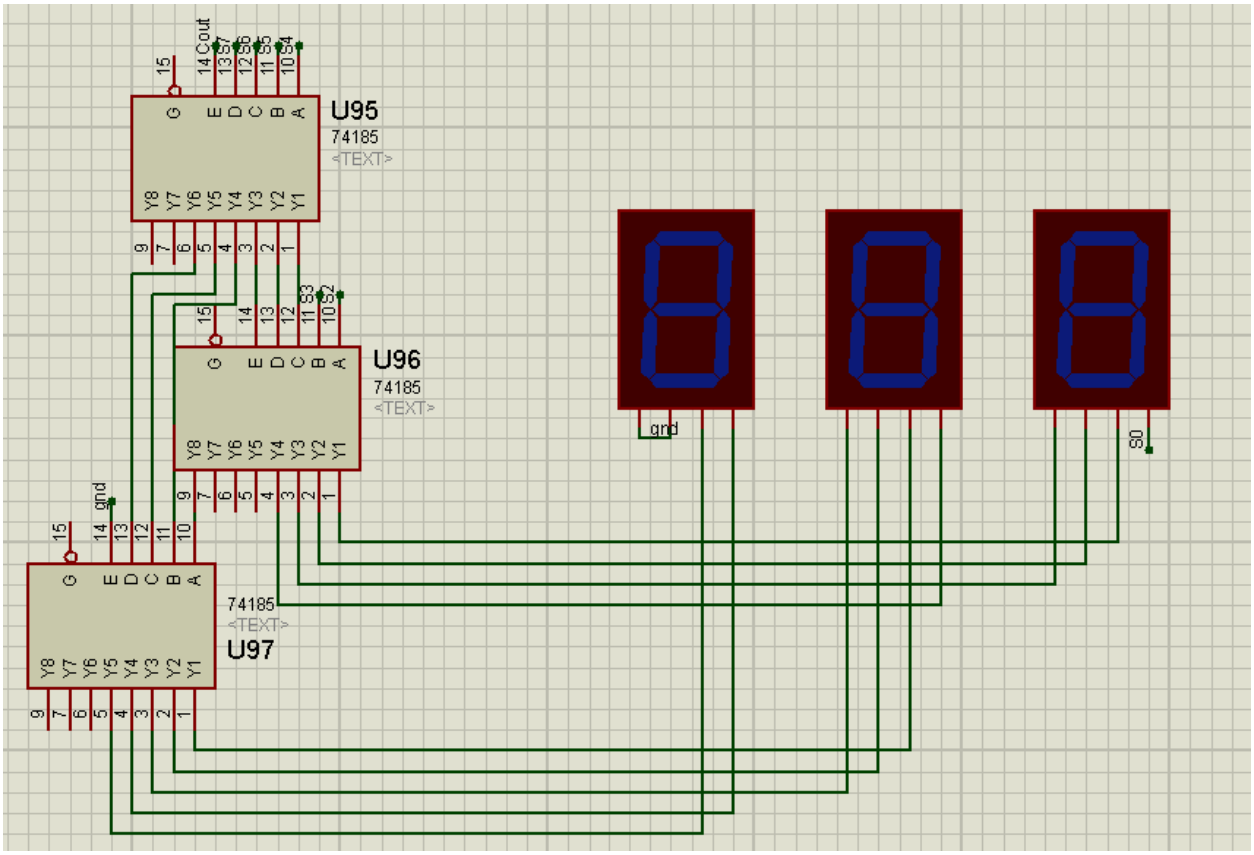


Figure 10: 8-Bit Binary to BCD Converter and Display

Chapter 3: Hardware Implementation

3.1 Detailed Schematic of Design and its Description

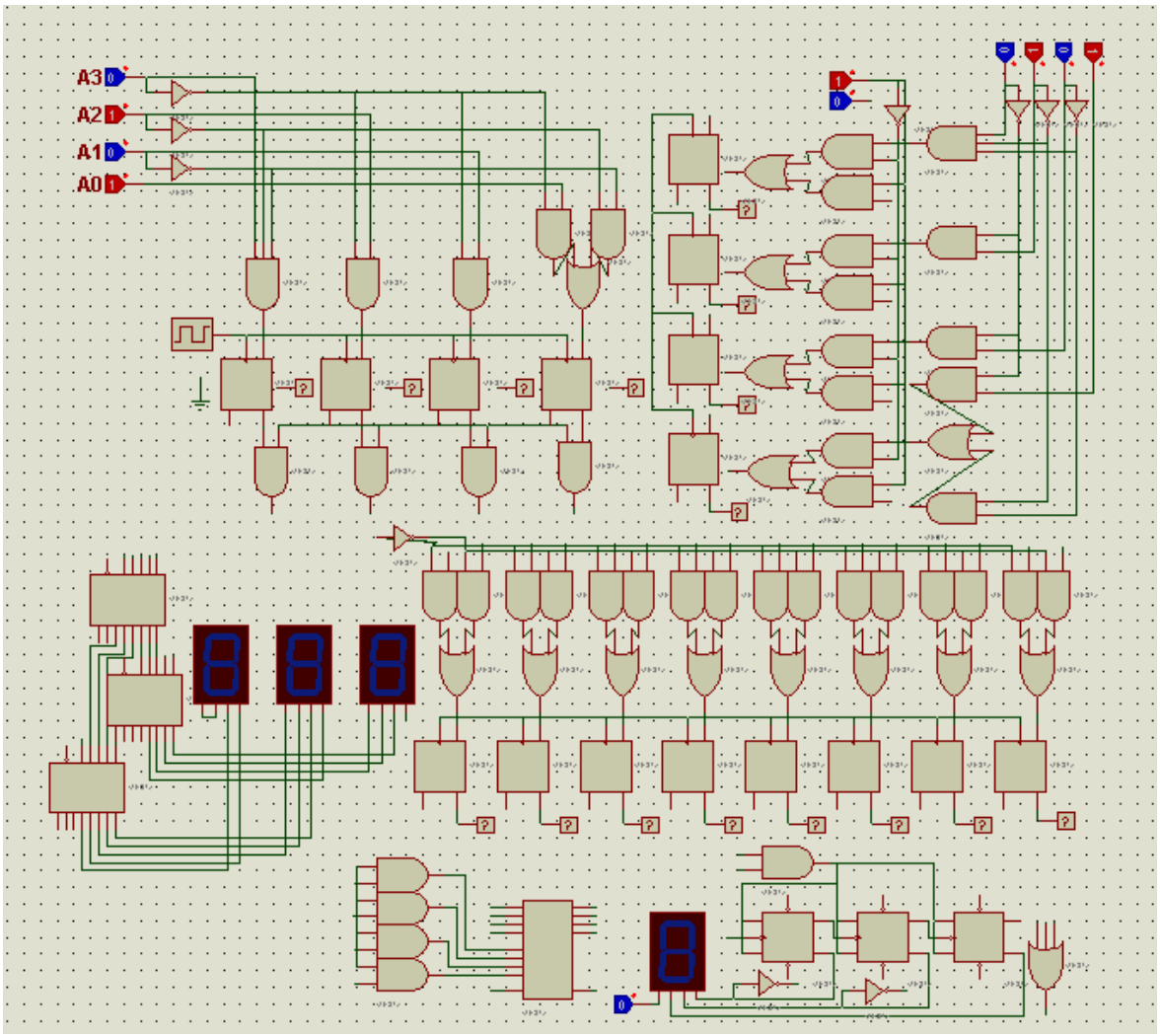


Figure 13: Complete Schematic Diagram

i. **BCD Input in PIPO 4-Bit Register:** Toggle switches are used to display the input from user. The input logic of the combinational circuit allow only 0 to 9 BCD numbers and inputs all zeros for any other input considering other inputs to be invalid. The combinational circuit attached at the output of this register performs multiplication with the LSB of PISO Register.

ii. **BCD input in PISO 4-Bit Register:** Just as the PIPO input Register mentioned above, this register also allows only 0 to 9 valid BCD input numbers and passes all zeros for the rest of inputs. Other than that, this register uses the LSB to control the inputs in PIPO register. Whenever the LSB is 1, the PIPO Register shows the output stored in the multiplicand. On the contrary, whenever the LSB is 0, the PIPO passes a value of all zeros. Hence controlling the overall inputs from the PIPO register.

iii. **Full adder for multiplication:** The combinational circuit connected before the full adder prevents input from PIPO Register whenever the LSB from PISO Register is low and inputs the multiplicand from PIPO Register whenever the LSB from PISO Register is high. The output from full adder IC is then again fed into the IC itself. The LSB from full adder is passed onto the 8-Bit Register and the remaining bits are shifted while **Cout** from the full adder is used as the MSB in the full adder second input. This part of the circuit performs the shifting as well as an additional part of the overall problem circuit.

iv. **4-0 Down Counter:** This counter counts down from 4 to 0 and once the circuit reaches 0, the OR IC indicate a low signal setting the clock to a 0 state and truncates serial shift in output shift register forcing it to hold its value.

v. **8-Bit Register using 8 DFFs and Digital Display:** This register acts as a PISO Register when the counter outputs a non-zero value. On the other hand, when the counter strikes zero, the conversion shifts to a PIPO Register that feeds its value to itself and infinitely repeats itself allowing the output to be stored and seen. The display circuit converts 8-Bit Binary to BCD and used a 7-segment display to show the desired product.

vi. **Circuit Progress:**
The logic used for the multiplication and process is as follows:

Let be,
Multiplicand: B3 B2 B1 B0
Multiplier: A3 A2 A1 A0

			B3	B2	B1	B0
		X	A3	A2	A1	A0
			A0B3	A0B2	A0B1	A0B0
		A1B3	A1B2	A1B1	A1B0	X
	A2B3	A2B2	A2B1	A2B0	X	X
A3B3	A3B2	A3B1	A3B0	X	X	X
S7	S6	S5	S4	S3	S2	S1

Solved Example for B = 1001 and A = 0111 (9 x 7 = 63 = 00111111)

Explanation		Cout	S7	S6	S5	S4	S3	S2	S1
Initially Stored		0	0	0	0	0	0	0	0
1001 Input for A0 = 1	+	1	0	0	1				
Sum 1	=	1	0	0	1	0	0	0	0
Shift 1	->	0	1	0	0	1	0	0	0
1001 Input for A1 = 1	+	1	0	0	1				
Sum 2	=	1	1	0	1	1	0	0	0
Shift 2	->	0	1	1	0	1	1	0	0
1001 Input for A2 = 1	+	1	0	0	1				
Sum 3	=	1	1	1	1	1	1	0	0
Shift 3	->	0	1	1	1	1	1	1	0
0000 Input for A3 = 0	+	0	0	0	0				
Sum 4	=	0	1	1	1	1	1	1	0
Shift 4 /Answer	->	0	0	1	1	1	1	1	1

3.2 Details of ICs used

a. 7400 Series

7408 Quad 2- input AND gates

7432 Quad 2- input OR gates

7404 Six 1- input NOT gates

7476 Dual input JK Flip Flops with Clock and Preset

7474 Dual input D Flip Flops with Clock and Preset

7447 BCD to 7-Segment Decoder

74185 Binary to BCD Decoder

b. Other Components

Jumper Wires

Arduino Nano for 8-Bit Binary to BCD Conversion

Common Cathode for Digital Display

3.3 Hardware Issues

During the implementation phase, several hardware issues emerged that impacted the circuit's functionality. Firstly, sourcing components according to the schematic proved challenging. Substituting some components led to slight discrepancies in size and tolerance, potentially affecting the circuit's performance. Wiring errors were another hurdle. Interpreting the schematic and correctly establishing wiring connections demanded precision. Mistakes in wiring resulted in intermittent shorts and open circuits, causing unexpected behavior in the circuit.

Power supply considerations were crucial. Issues with voltage levels and grounding led to instability in the circuit's operation. Ensuring appropriate power supply ratings and grounding arrangements emerged as critical aspects for reliable circuit functionality.

Furthermore, component placement played a significant role. Incorrectly placed components and orientation errors hindered the signal paths and made troubleshooting more challenging.

Testing and debugging presented considerable challenges. Identifying discrepancies between the schematic and the physical circuit demanded meticulous attention to detail and thorough testing procedures.

3.4 Results

Upon successful patching and implementation of the 4-bit Binary Coded Decimal (BCD) multiplier circuit, several significant observations and results were obtained:

1. **Functionality Validation:** The circuit successfully processed two 4-bit BCD-encoded numbers as intended. The implemented sequential logic components efficiently facilitated the multiplication operation, showcasing controlled shifting mechanisms for accurate processing of BCD numbers.
2. **Validated BCD Constraints**: The combinational logic circuit effectively filtered and validated input data, permitting only valid BCD numbers (0-9) for multiplication. Inputs exceeding this range were properly rejected, maintaining precision in subsequent calculations.
3. **Digital Display Output:** The multiplication results were accurately displayed on the digital display unit, providing a clear visualization of the outcome. The circuit adhered strictly to BCD standards, ensuring precise representation of the multiplication results on the display.

4. **Truncation for Display:** The counter function integrated into the circuit effectively truncated the final answer stored within the shift register to fit the 4-bit digital display capacity. This ensured that the output displayed on the digital screen remained within the designated length, avoiding overflow or distortion.
5. **Operational Stability:** The circuit demonstrated stable and reliable operation throughout the multiplication process. Power supply considerations and grounding arrangements were appropriately addressed, contributing to the circuit's consistent functionality.
6. **Troubleshooting and Refinement:** Several rounds of testing and debugging were conducted to identify and rectify initial hardware issues encountered during implementation. Refinement of wiring connections, component placements, and adherence to schematic specifications were crucial in achieving operational stability.

These results indicate the successful implementation of the 4-bit BCD multiplier circuit, highlighting its efficiency in performing BCD multiplication operations while ensuring adherence to BCD constraints and accurate representation of results on the digital display.

3.5 Observations

Our circuit's shifting part smoothly handled the multiplication of BCD numbers. It managed to shuffle things around just right. The other part, checking the numbers before they jumped into math, did its job perfectly. Our digital display showed the answers clear as day, exactly how we wanted. We did some testing and tweaking to make sure our circuit worked without a hitch. This practical project helped us understand how these circuits really work, making our classroom lessons practical and clear.

This hands-on experience was an eye-opener, showing us how each part of the circuit played its role in getting accurate results. It was like connecting the dots between theory and reality, giving us a solid understanding of how these circuits function in the real world.

Chapter 4: Project Applications

Some Applications of our project are discussed as under:

Real-Life Applications:

- **Digital Calculators**: BCD multipliers are fundamental components in digital calculators, aiding in precise arithmetic calculations. They ensure accurate multiplication of BCD-encoded numbers, vital for everyday mathematical computations.
- **Financial Systems**: In banking and accounting, BCD multipliers play a pivotal role in processing financial data. They assist in handling monetary calculations with high accuracy, ensuring precise computations for transactions, interests, and investments.
- **Embedded Systems**: The circuit finds usage in embedded systems for various applications like digital clocks, where accurate timekeeping necessitates efficient multiplication operations for BCD-encoded time values.
- **Digital Signal Processing**: In telecommunications and signal processing, BCD multipliers contribute to accurate data manipulation. They aid in filtering, encoding, and decoding signals, ensuring reliable communication and data processing.

Theoretical Implications:

- **Boolean Algebra**: Understanding BCD multipliers provides insights into the practical application of Boolean algebra in digital circuits. It demonstrates the implementation of logical operations to perform arithmetic tasks.
- **Sequential and Combinational Logic**: The circuit emphasizes the role of sequential and combinational logic in digital systems. It showcases how these components work together to execute arithmetic operations efficiently.
- **Data Representation**: BCD multiplication illustrates data representation techniques. It showcases how binary-coded decimal numbers are manipulated and processed, offering insights into various data encoding methods.
- **Circuit Design Principles:** Implementing the BCD multiplier reinforces concepts of circuit design, highlighting the importance of precise wiring, component placement, and testing procedures for reliable operation.

The practical applications and theoretical implications of the 4-bit BCD multiplier circuit span across diverse fields, demonstrating its significance in both real-world applications and theoretical understanding of digital circuits and arithmetic operations.

Chapter 5: Future Recommendations

Some recommendations can be made to ensure a smoother circuit. As such:

1. **Arduino Integration:** Incorporating an Arduino microcontroller can enhance the project's functionality by enabling interaction with external devices, data logging, or implementing a user-friendly interface. The Arduino can manage input validation, display output, or even control additional functions, providing a more versatile and interactive experience.
2. **FPGA Implementation**: Utilizing a Field-Programmable Gate Array (FPGA) can significantly enhance the circuit's performance and scalability. FPGA implementation allows for hardware acceleration, increased speed, and parallel processing capabilities, which could optimize the BCD multiplication operations and facilitate more complex arithmetic functions.
3. **Optimized Integrated Circuits**: Exploring specialized integrated circuits or components designed explicitly for BCD arithmetic could lead to more efficient and dedicated hardware solutions, ensuring accuracy and speed in multiplication tasks.
4. **Parallel Processing Architectures**: Implementing parallel processing architectures could enhance the circuit's capabilities. This could involve exploring parallel processing methodologies or utilizing multi-core architectures to improve computational speed and efficiency.

References / Bibliography

Some trustworthy websites that we consulted are mentioned as under:

- Sequential Multiplier - Digital System Design
- Microsoft Word - Lab13_Revd.doc (kfupm.edu.sa)
- (3) Sequential multiplier - Unsigned numbers - YouTube