



The University of Azad Jammu and Kashmir, Muzaffarabad

Name**Kamal Ali Akmal**

Course Name	Computer Architecture and Logic Design
Submitted to	Engr. Sidra Rafique
Semester	2nd
Session	2024-2028
Roll No	2024-SE-38
Lab	Open Ended Lab
Submission date	05 September 2025

Task 1: Parity Generator and Checker

1. 4-bit Parity Generator Circuit

- Inputs: A, B, C, D (4-bit data word).
- Output: Parity Bit (P).
- **Even parity rule:** The parity bit is chosen such that the total number of 1's (data bits + parity) is even.

Boolean Equation:

$$P = A \oplus B \oplus C \oplus D$$

2. Implementation

- Use XOR gates in PLA or a digital logic simulator like Electronic Work Bench..
- Connect inputs A, B, C, D → XOR chain → Output P.

3. Testing the Generator

- Try all 16 combinations of A, B, C, D.
- Verify that $(A \oplus B \oplus C \oplus D) + P$ always gives **even number of 1s**.

4. Parity Checker Circuit

- Inputs: A, B, C, D, and received Parity Bit (P).
- Output: Error signal (E).

Boolean Equation:

$$E = A \oplus B \oplus C \oplus D \oplus P$$

- If E = 0 → No error.
- If E = 1 → Error detected.

5. Implementation and Testing

- Connect inputs to XOR chain.
- Apply different data with parity bit.
- Check if circuit correctly detects error when a bit flips.

4-bit Parity Generator Circuit

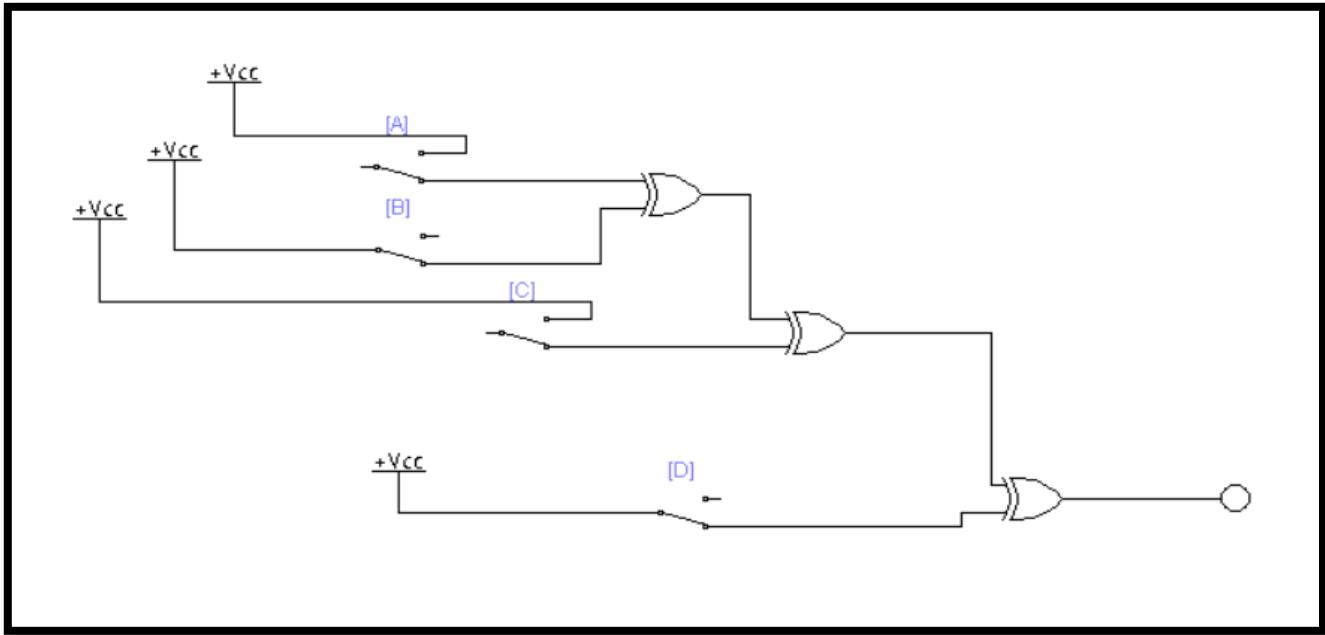


Figure 1: When two Switch is OFF and two ON means inputs (A, B, C, D) = (0 ,1, 0,1) then output is 0.

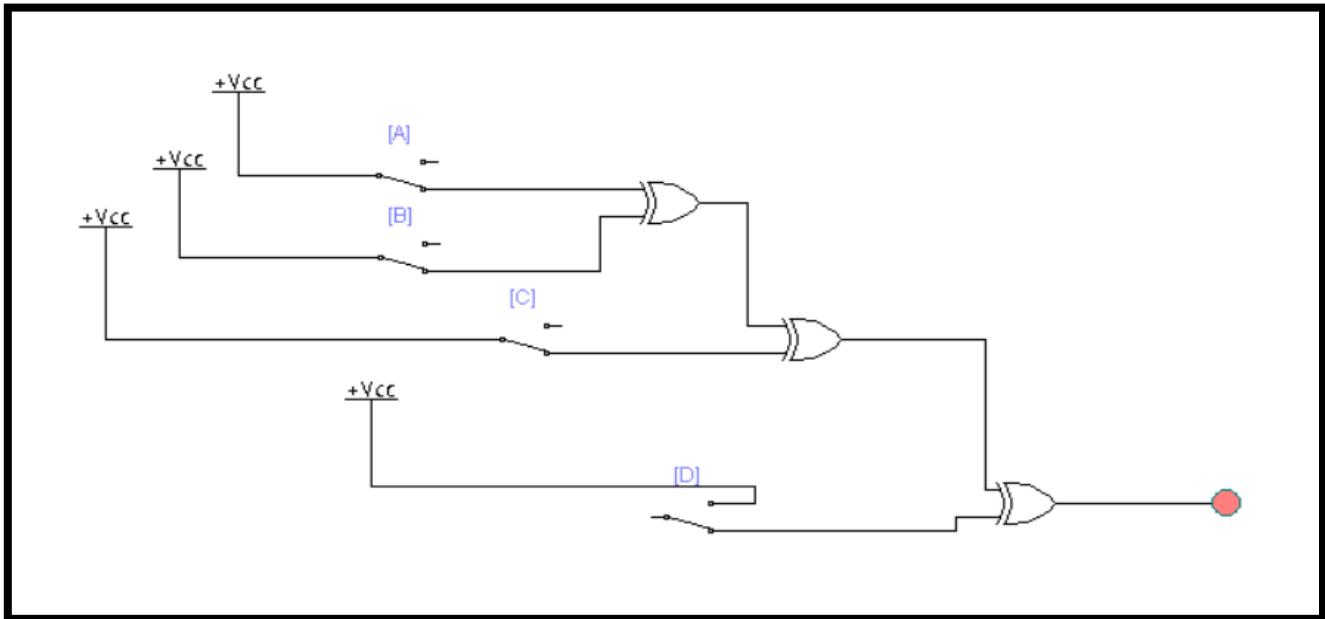
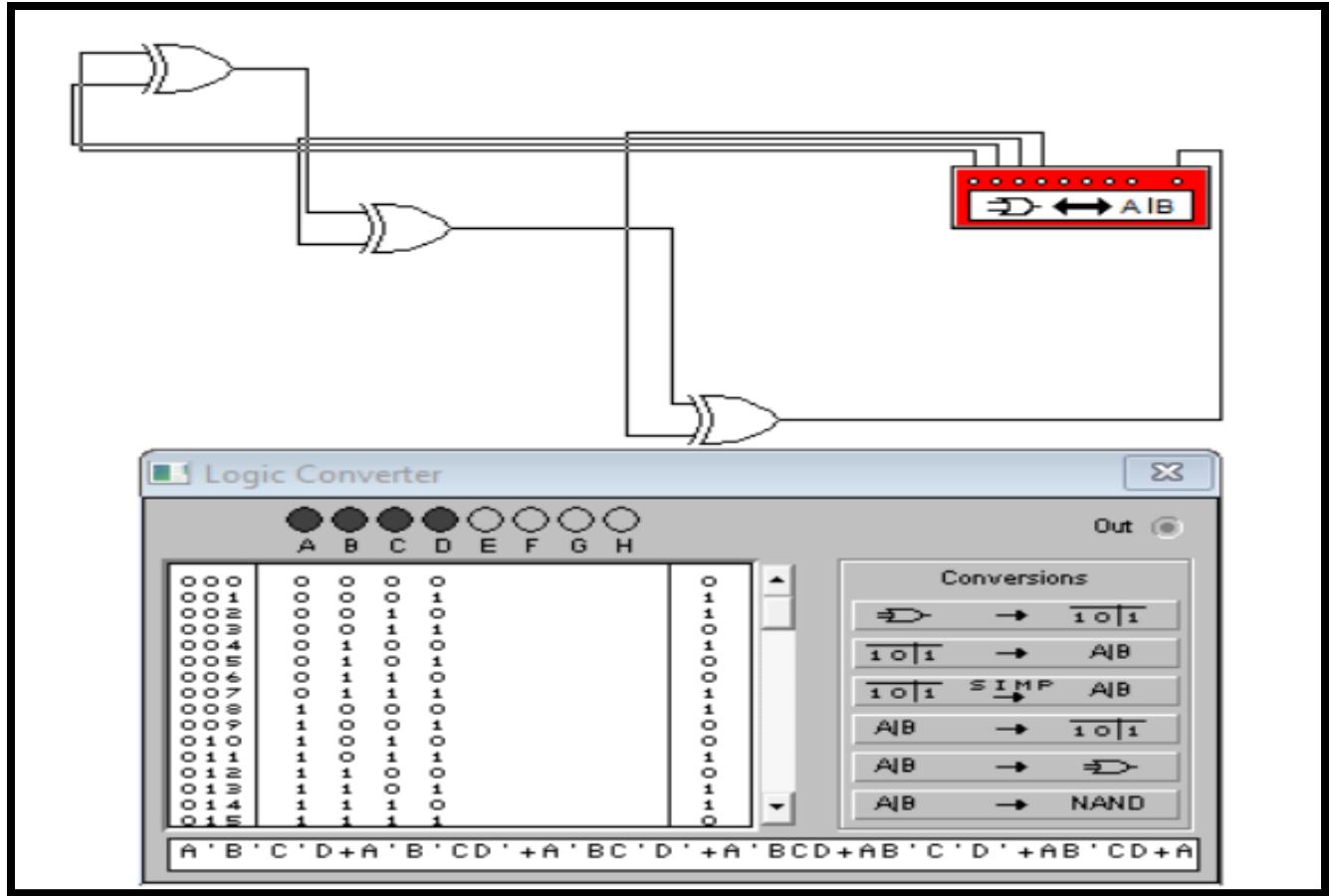


Figure 2: When three Switches ON and one OFF means all inputs (A, B, C, D) = (1,1,1,0), then output is 1.

4-bit Parity Generator (Even Parity): -Formula: $P = A \oplus B \oplus C \oplus D$

A	B	C	D	P (Parity)
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0



4-bit Parity Checker (Error Detection)

- Inputs: A, B, C, D, and received Parity Bit (P).
- Output: Error signal (E).

Boolean Equation:

$$E = A \oplus B \oplus C \oplus D \oplus P$$

- If $E = 0 \rightarrow$ No error.
- If $E = 1 \rightarrow$ Error detected.

Implementation and Testing

- Connect inputs to XOR chain.
- Apply different data with parity bit.
- Check if circuit correctly detects error when a bit flips.

4-bit Parity Checker (Error Detection)

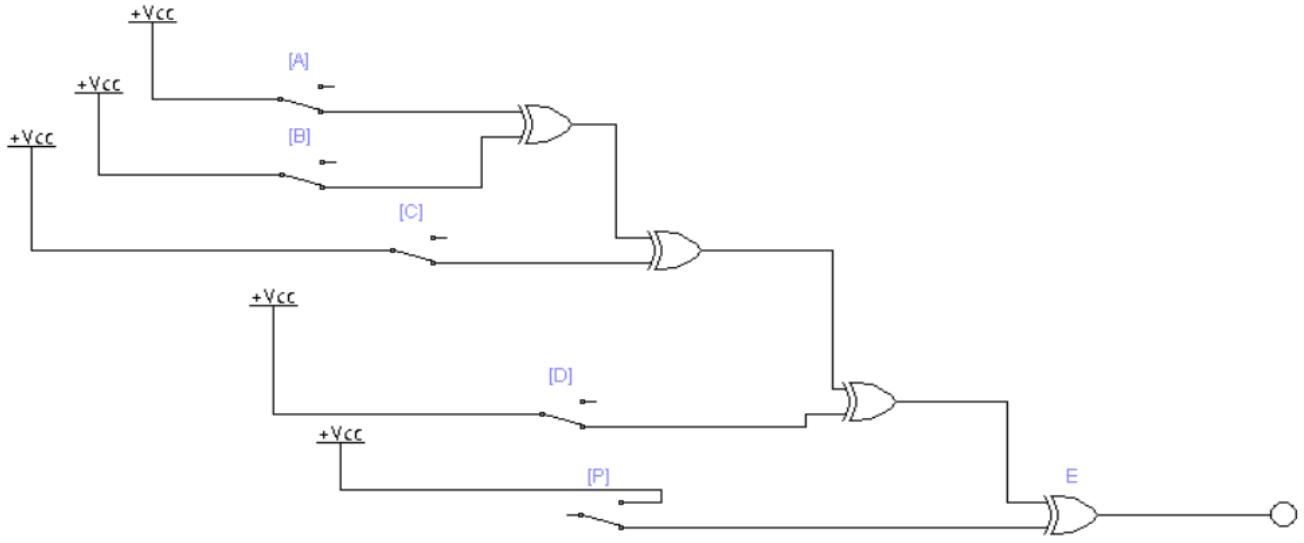


Figure 1 : When four Switches ON and one OFF means all inputs (A, B, C, D, P) = (1,1,1,1,0), then output is 0.

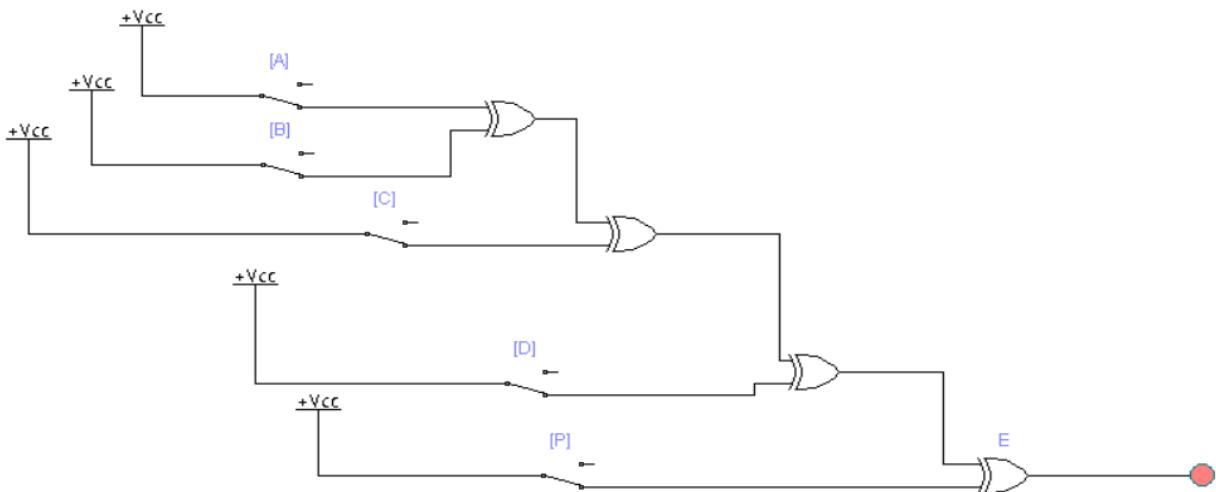


Figure 2: When all Switches ON means all inputs (A, B, C, D, P) = (1,1,1,1,1), then output is 1

Parity Checker Truth Table

Boolean Equation:

$$E = A \oplus B \oplus C \oplus D \oplus P$$

Inputs: **A, B, C, D, P** → Output: **E (Error)**

A	B	C	D	P (Expected)	E (Output)	Meaning
0	0	0	0	0	0	<input checked="" type="checkbox"/> No error
0	0	0	0	1	1	<input checked="" type="checkbox"/> Error
1	0	1	0	0	0	<input checked="" type="checkbox"/> No error
1	0	1	0	1	1	<input checked="" type="checkbox"/> Error
1	1	1	1	0	0	<input checked="" type="checkbox"/> No error
1	1	1	1	1	1	<input checked="" type="checkbox"/> Error

Task 2: Low-Level Program for Shift Instructions

1. **SHL (Shift Left Logical)** – Multiplies number by 2.
2. **SHR (Shift Right Logical)** – Divides unsigned number by 2.
3. **SAR (Shift Arithmetic Right)** – Divides signed number by 2, keeps sign.
4. **SAL (Shift Arithmetic Left)** – Similar to SHL.

MOV AL, 00001101b ; Load data = 13 (binary)

SHL AL, 1 ; Shift Left → result = 26

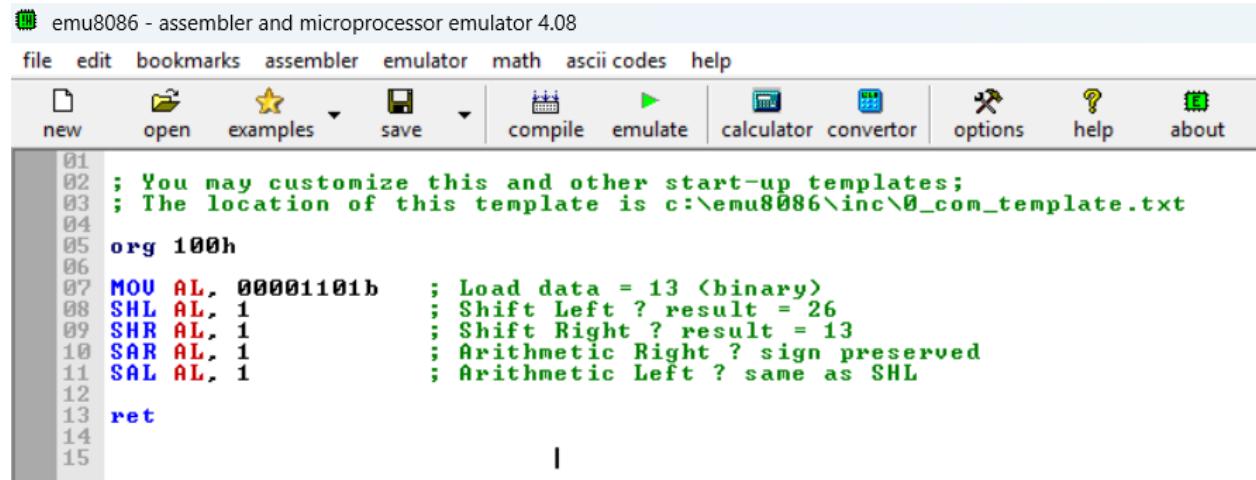
SHR AL, 1 ; Shift Right → result = 13

SAR AL, 1 ; Arithmetic Right → sign preserved

SAL AL, 1 ; Arithmetic Left → same as SHL

By using emulator,

Step #01:

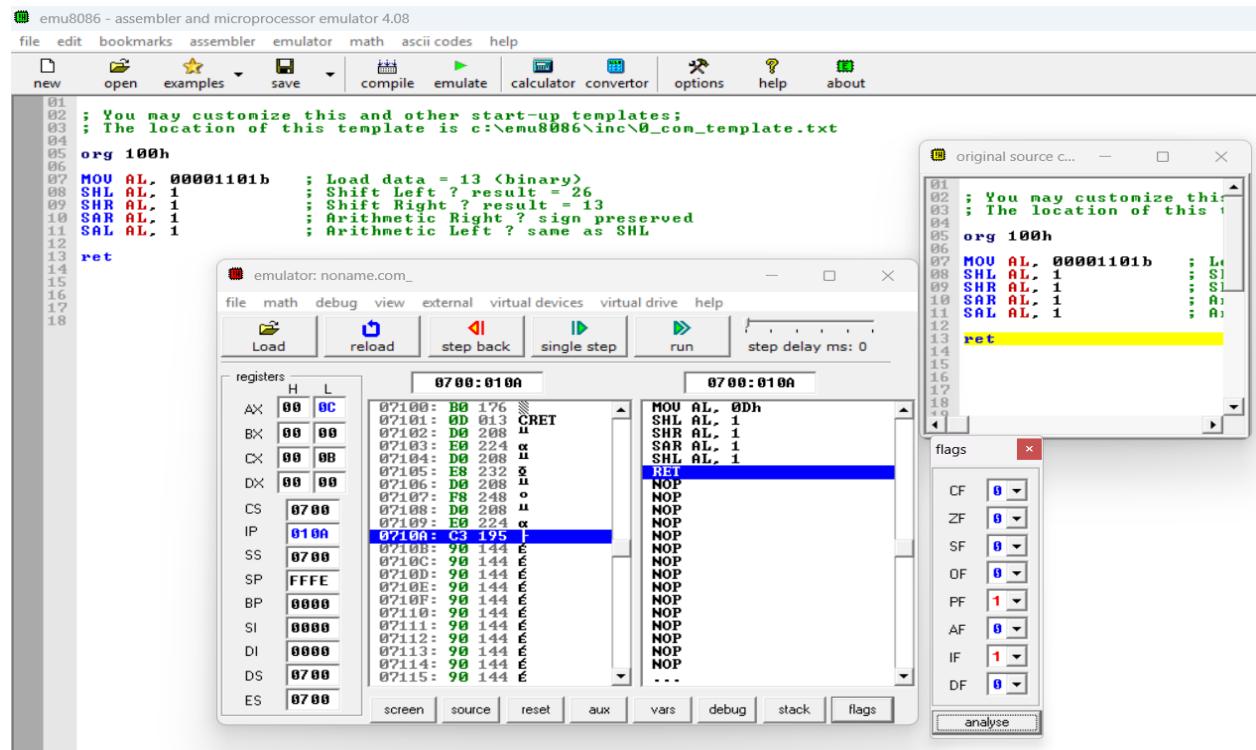


```

01 ; You may customize this and other start-up templates;
02 ; The location of this template is c:\emu8086\inc\0_com_template.txt
03
04
05 org 100h
06
07 MOU AL, 00001101b ; Load data = 13 (binary)
08 SHL AL, 1 ; Shift Left ? result = 26
09 SHR AL, 1 ; Shift Right ? result = 13
10 SAR AL, 1 ; Arithmetic Right ? sign preserved
11 SAL AL, 1 ; Arithmetic Left ? same as SHL
12
13 ret
14
15

```

Result:



The screenshot shows the emu8086 emulator interface with the assembly code loaded. The CPU registers and flags are displayed at the bottom. The registers show the following values:

Register	H	L
AX	00	0C
BX	00	00
CX	00	0B
DX	00	00
CS	0700	
IP	0100	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

The flags register shows the following values:

Flag	Value
CF	0
ZF	0
SF	0
OF	0
PF	1
AF	0
IF	1
DF	0

Deliverables

1. Working Circuits

- 4-bit parity generator and checker implemented in simulator.

2. Truth Table & Boolean Equations

A	B	C	D	Parity P = A⊕B⊕C⊕D
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
...

Boolean Equation:

$$E = A \oplus B \oplus C \oplus D \oplus P$$

3. Summary

- Designed a 4-bit parity generator using XOR gates.
- Verified functionality with truth table and simulation.
- Designed a parity checker for error detection.
- Implemented and tested both circuits in simulator.
- Wrote Assembly program for shift operations SHL, SHR, SAR, SAL.