2024-SE-38

The University of Azad Jammu and Kashmir, Muzaffarabad

Name	**Tayyab Hanif**
Course Name	**Computer Architecture and Logic Design**
Submitted to	**Engr. Sidra Rafique**
Semester	**2nd**
Session	**2024-2028**
Roll No	**2024-SE-11**
Lab	**Open Ended Lab**
Submission date	**05 September 2025**

Department of Software Engineering

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:

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

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

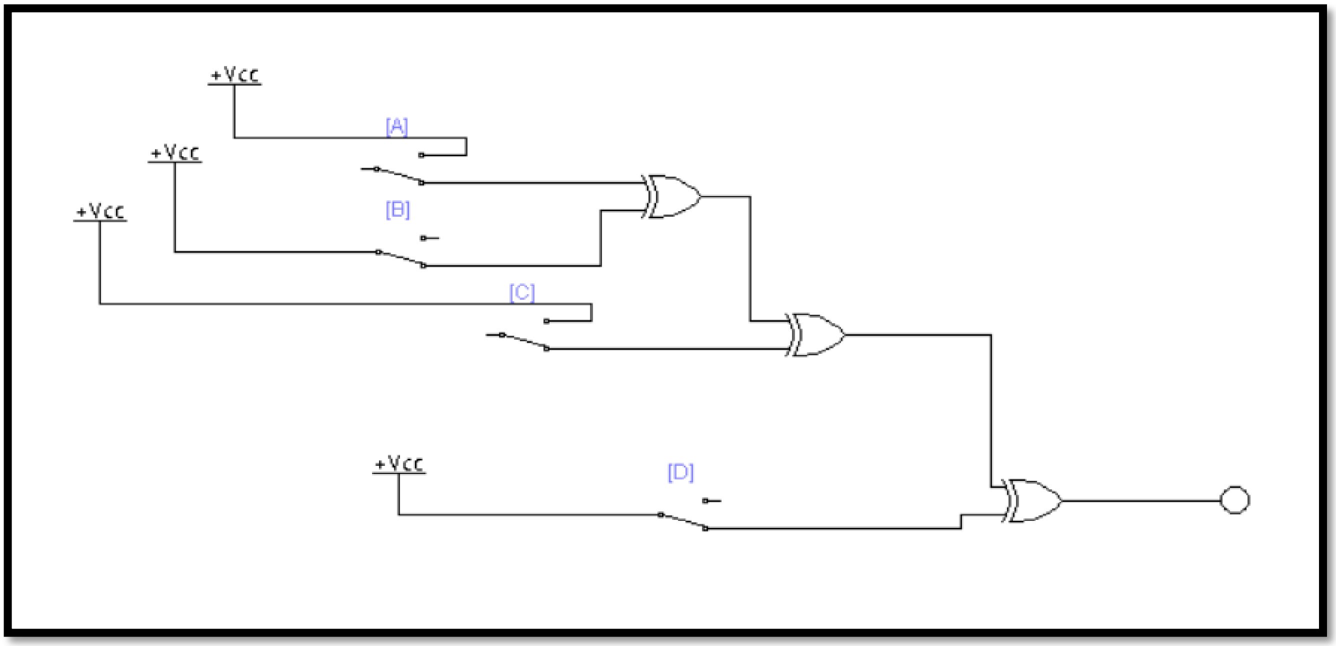


Figure 1: When two Switch is OFF and two ON means inputs (A, B, C, D) = (0 ,1, 0,1) then output is 0.

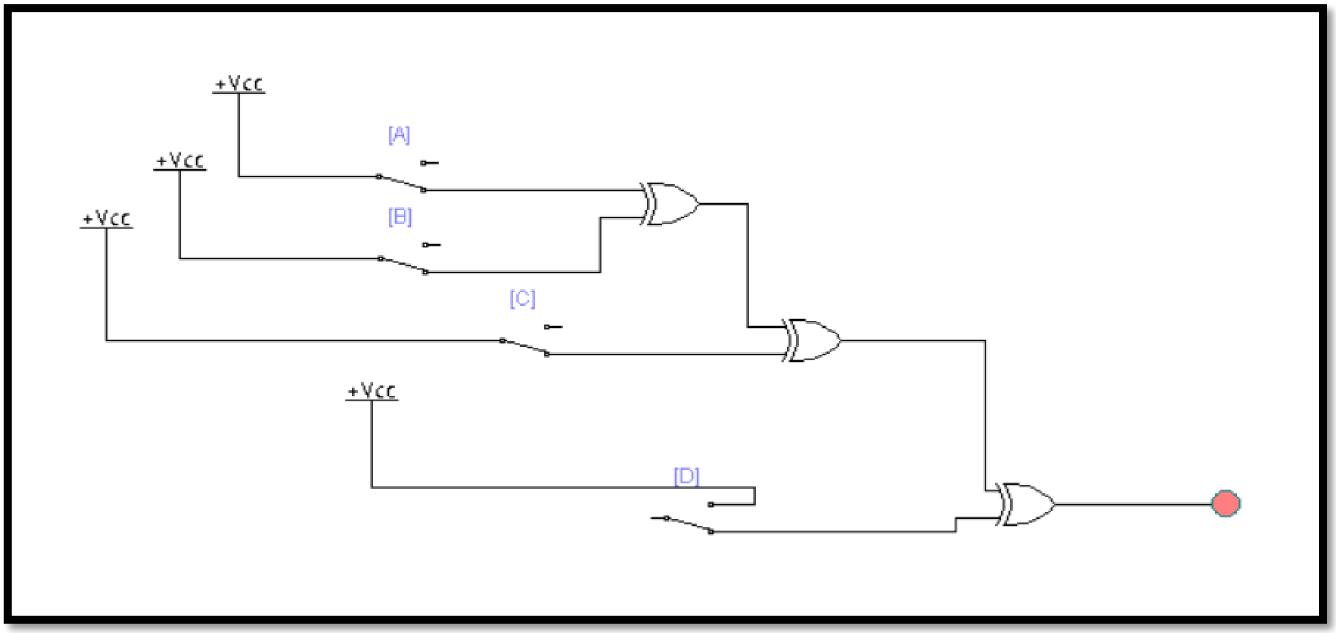
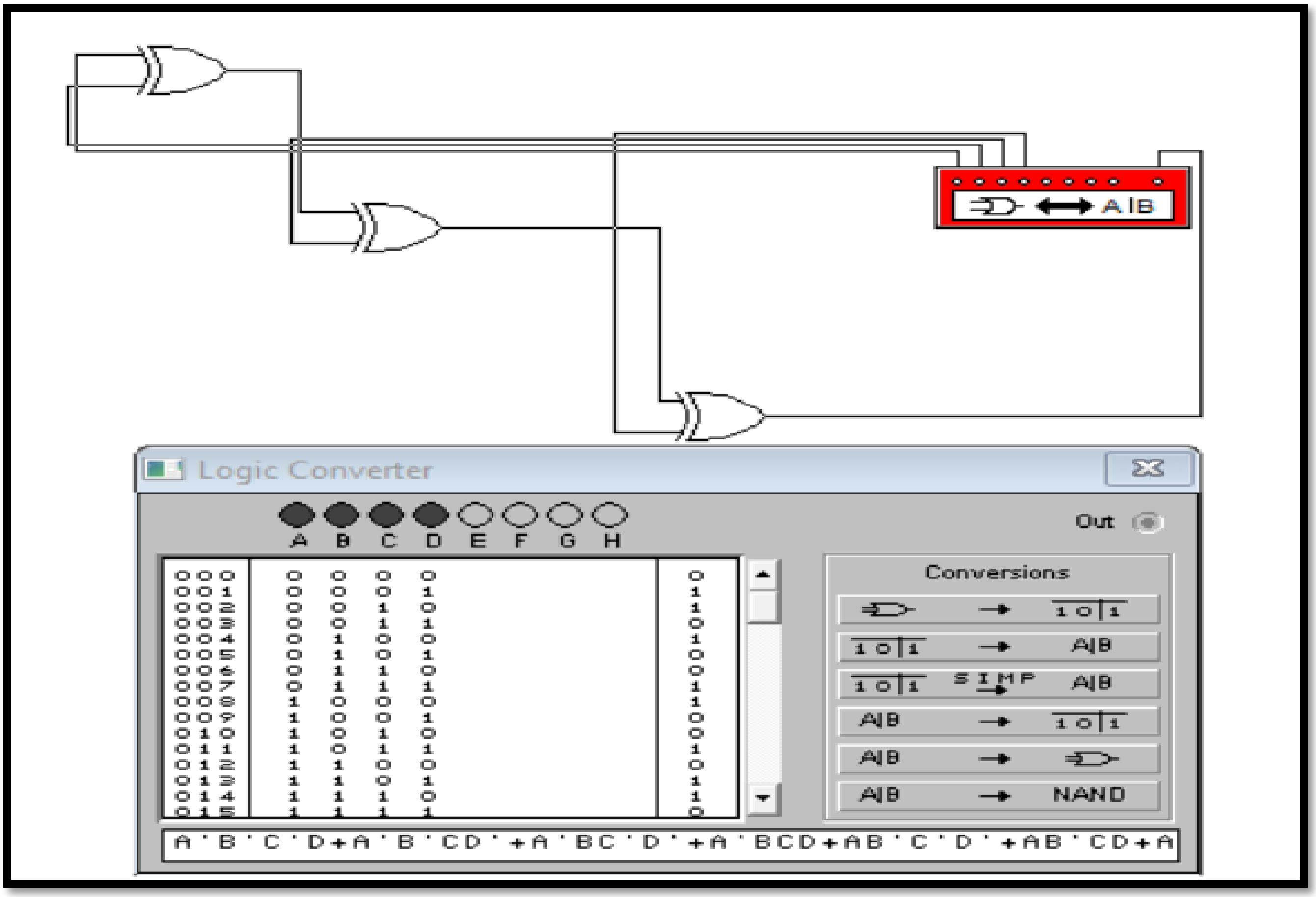


Figure 2: When three Switches ON and one OFF means all inputs (A, B, C, D) = (1,1,1,0), then output is 1.

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



	A	B	C	D	
000	0	0	0	0	0
001	0	0	0	1	1
002	0	0	1	0	1
003	0	0	1	1	0
004	0	1	0	0	1
005	0	1	0	1	0
006	0	1	1	0	0
007	0	1	1	1	1
008	1	0	0	0	1
009	1	0	0	1	0
010	1	0	1	0	0
011	1	0	1	1	1
012	1	1	0	0	0
013	1	1	0	1	1
014	1	1	1	0	1
015	1	1	1	1	0

A'B'C'D+A'B'CD'+A'BC'D'+A'BCD+AB'C'D'+AB'CD+A

4-bit Parity Checker (Error Detection)

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

Boolean Equation:

E=A⊕B⊕C⊕D⊕P

- If **E = 0** → No error.
- If **E = 1** → 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)

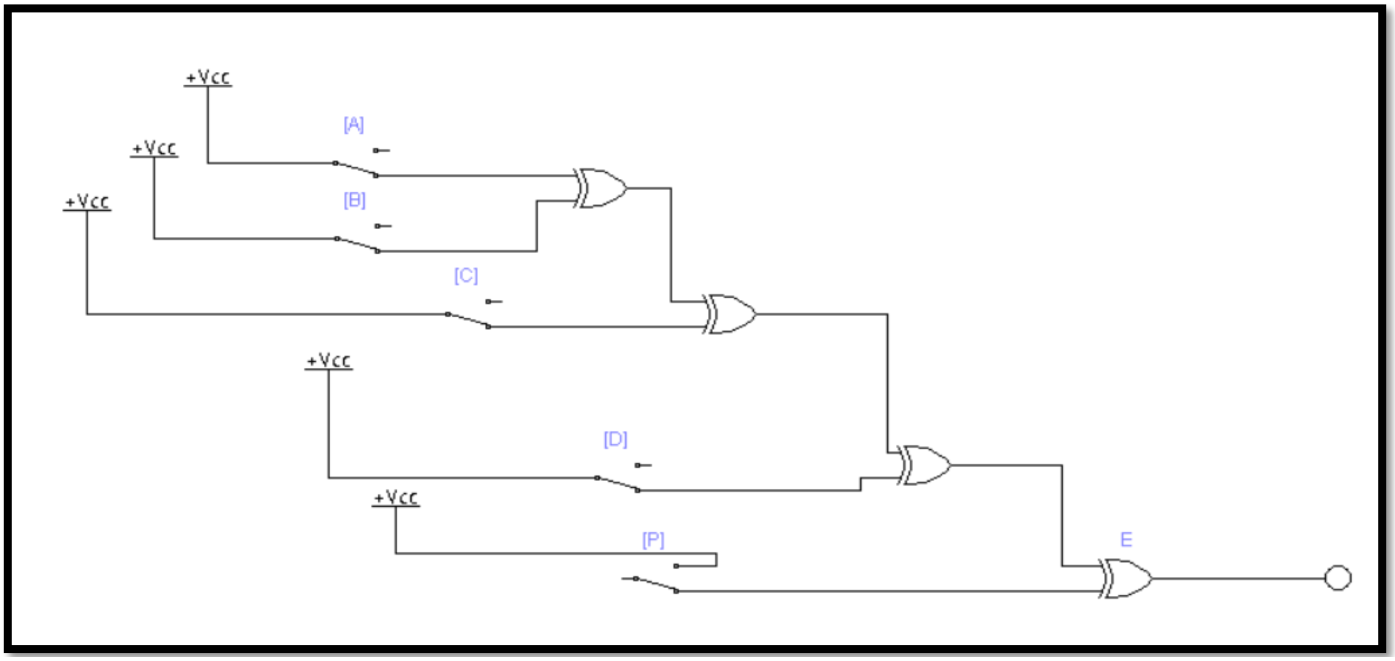


Figure 1 : When four Switches ON and one OFF means all inputs (A, B, C, D, P) = (1,1,1,1,0), then output is 0.

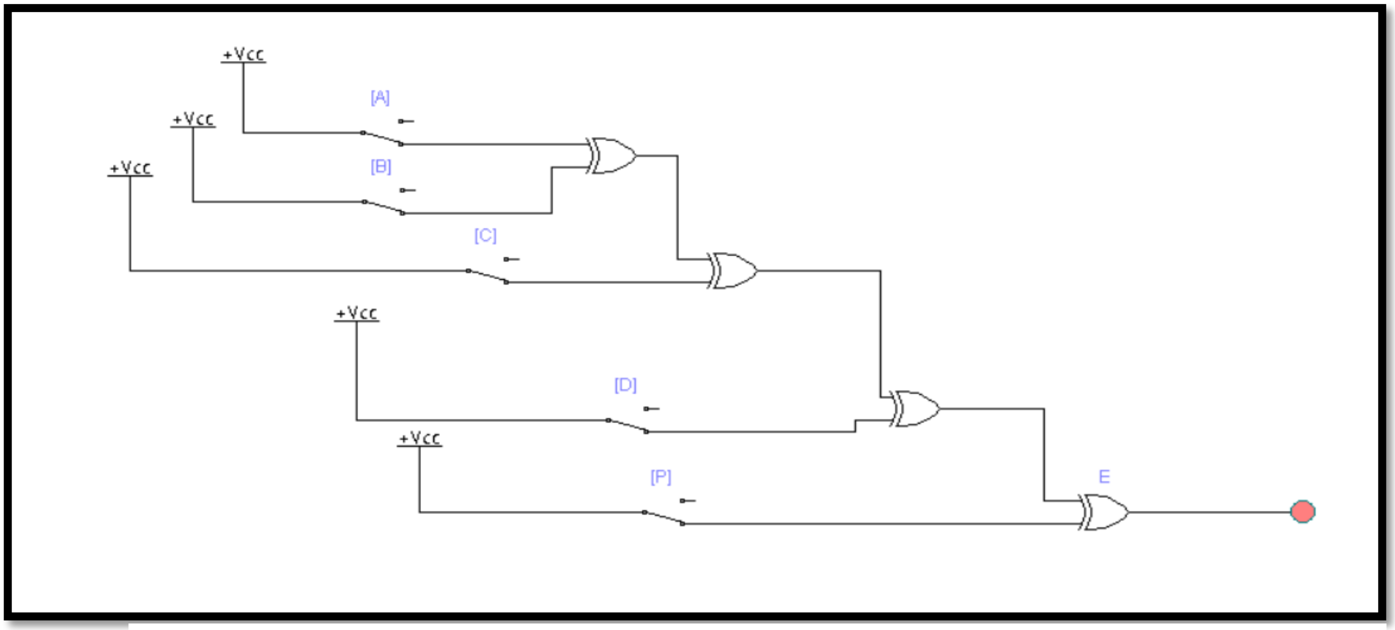


Figure 2: When all Switches ON means all inputs (A, B, C, D, P) = (1,1,1,1,1), then output is 1

Parity Checker Truth Table

Boolean Equation:

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

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

A	B	C	D	P (Expected)	E (Output)	Meaning
0	0	0	0	0	0	☑ No error
0	0	0	0	1	1	✗ Error
1	0	1	0	0	0	☑ No error
1	0	1	0	1	1	✗ Error
1	1	1	1	0	0	☑ No error
1	1	1	1	1	1	✗ Error

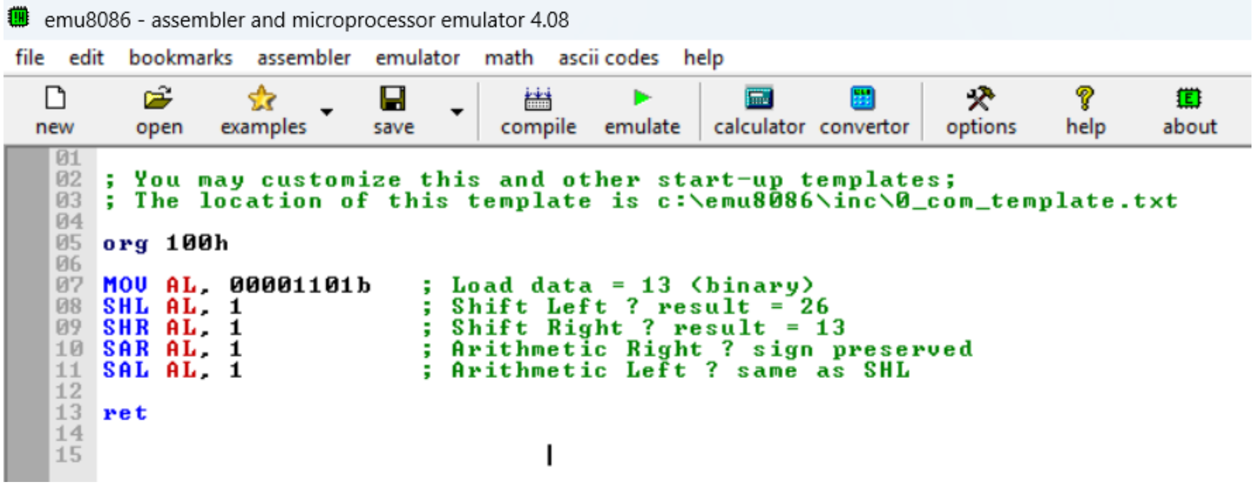
Task 2: Low-Level Program for Shift Instructions

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:


emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about

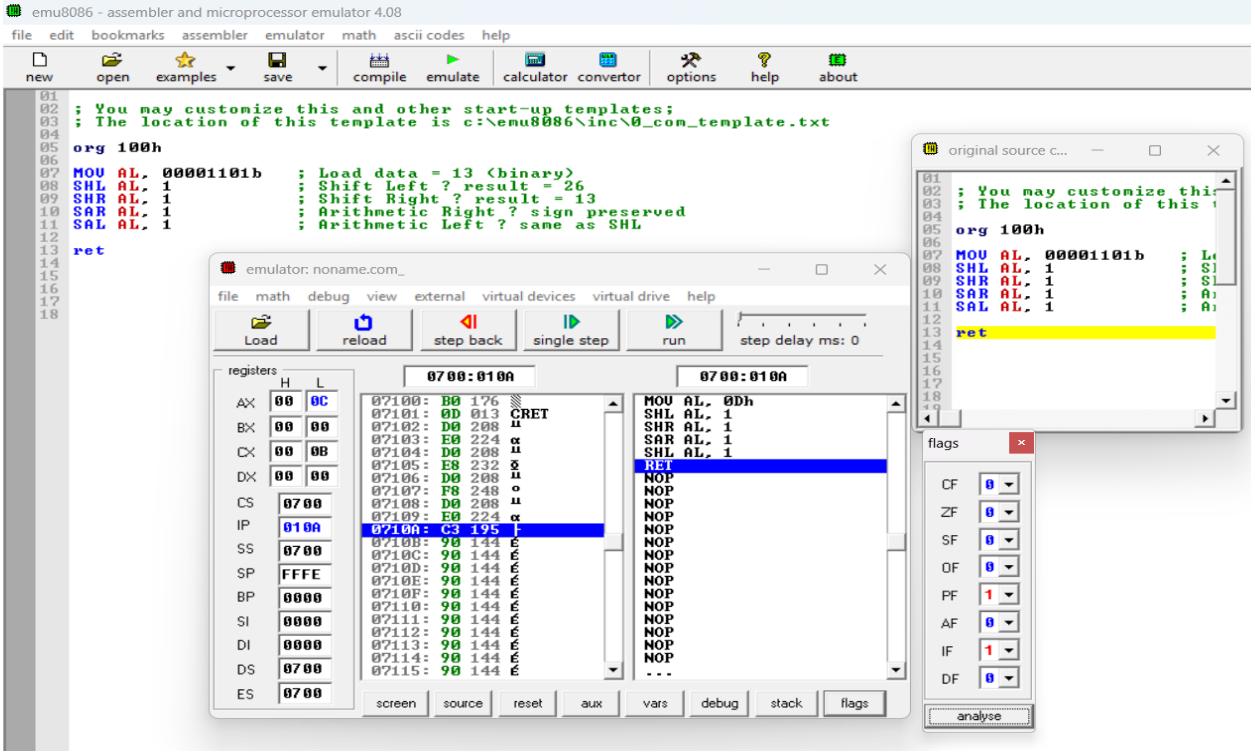
```
; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com_template.txt

org 100h

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

ret
```

Result:


emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
emulator: noname.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
screen source reset aux vars debug stack flags
original source c...
analyse

```
; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com_template.txt

org 100h

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

ret
```

registers

	H	L
AX	00	0C
BX	00	00
CX	00	0B
DX	00	00
CS	0700	
IP	010A	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

```
0700:010A

07100: B0 176 ░
07101: 0D 013 CRET
07102: D0 208 ╨
07103: E0 224 α
07104: D0 208 ╨
07105: E8 232 Φ
07106: D0 208 ╨
07107: F8 248 °
07108: D0 208 ╨
07109: E0 224 α
0710A: C3 195 ├
0710B: 90 144 É
0710C: 90 144 É
0710D: 90 144 É
0710E: 90 144 É
0710F: 90 144 É
07110: 90 144 É
07111: 90 144 É
07112: 90 144 É
07113: 90 144 É
07114: 90 144 É
07115: 90 144 É
```

```
0700:010A

MOV AL, 0Dh
SHL AL, 1
SHR AL, 1
SAR AL, 1
SHL AL, 1
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

flags

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