Bazla Bilquees

U3312671

Introduction To Information Technology

**ASSIGNMENT # 02 – PYTHON FUNDAMENTAL**

**Case Study 3 – A Boolean Circuit Equivalence**

## Step 1 — Understand the Problem:

We receive Boolean circuits and have to model their logic as Boolean expressions, which must be performed in Python. Lastly, we check that the Python code is acting properly under all input combinations.

## Step 2 — Inputs & Outputs

**Circuit a)**

**Inputs:** A, B, C (True/False or 0/1)

**Output:** Y (Boolean: True/FALSE or 0/1)

**Circuit b)**

**Inputs:** P, Q, R (Boolean: True/False or 0/1)

**Output:** Z (Boolean: True/False or 0/1).

*Units:* All are Boolean values; Python will use True/False.

## Step 3 — Algorithm

**Steps for Circuit a:**

* Read inputs A, B, C.
* Use logic gates in a series (e.g. AND, OR, NOT) to obtain an intermediate result.
* Use the output of the intermediate results based on the Boolean expression to generate the output Y.
* Return/print Y.

**Steps for Circuit b:**

* Read inputs P, Q, R.
* Use logic gates to find intermediate results.
* Add the results to obtain output Z.
* Return/print Z.

Reference I/O: The input variables of each step generate intermediate Boolean values to the final output.

## Step 4 — Flow Chart



## Step 4 – PSEUDOCODE

**Circuit a:**

START

INPUT A, B, C

X1 = NOT A

X2 = B AND C

Y = X1 OR X2

OUTPUT Y

END

**Circuit b:**

START

INPUT P, Q, R

X1 = P OR NOT Q

X2 = Q AND R

Z = X1 AND X2

OUTPUT Z

END

## Step 5 Python Code

**For Circuit a:**

def circuit\_a(A, B, C):

X1 = not A

X2 = B and C

Y = X1 or X2

return Y

# Test all combinations

for A in [True, False]:

for B in [True, False]:

for C in [True, False]:

print(f"A={A}, B={B}, C={C} => Y={circuit\_a(A,B,C)}")

**For Circuit b:**

def circuit\_b(P, Q, R):

X1 = P or not Q

X2 = Q and R

Z = X1 and X2

return Z

# Test all combinations

for P in [True, False]:

for Q in [True, False]:

for R in [True, False]:

print(f"P={P}, Q={Q}, R={R} => Z={circuit\_b(P,Q,R)}")

## Step 6 -Testing: handwritten expected results + test runs & notes

**For Circuit A:**

A screenshot of a computer

AI-generated content may be incorrect.

**For Circuit B:**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

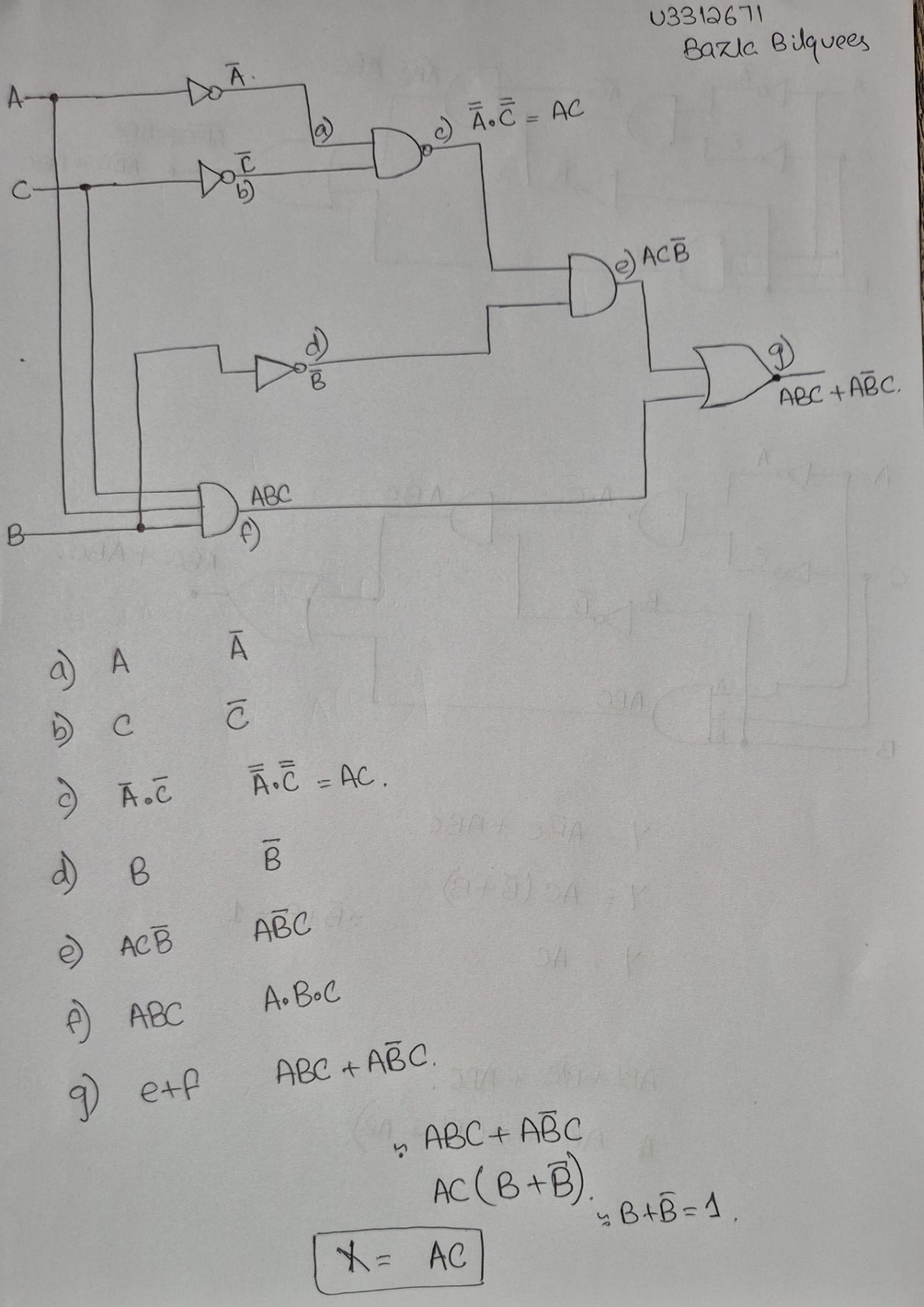
AI-generated content may be incorrect.

## Step 7- Boolean expression:

**For Circuit A:**

A diagram of a circuit

AI-generated content may be incorrect.

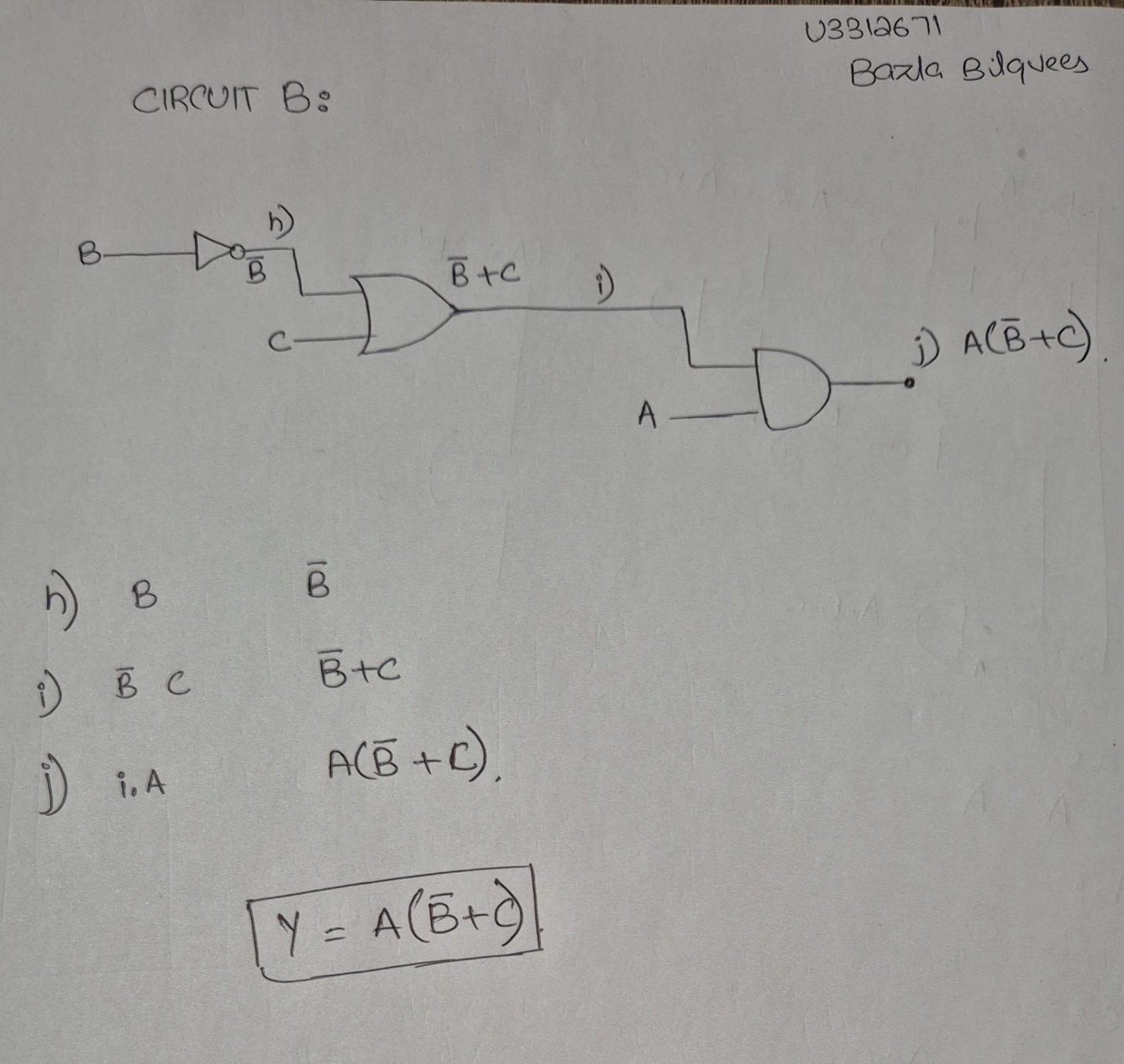


So, the Boolean expression is **X = AC.**

**For Circuit B:**

A diagram of a circuit

AI-generated content may be incorrect.



## Step 8 – Truth Table of Both Circuits:

Circuit A:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **a= NOT A** | **b= NOT C** | **c = NOT A AND NOT C** | **d = NOT B** | **e = c AND d** | **f = AND ABC** | **g = e OR f** |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

A screenshot of a computer

AI-generated content may be incorrect.

Circuit B:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **NOT B (h)** | **B OR C (i)** | **OUTPUT Y = h AND i** |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 |

A screenshot of a computer

AI-generated content may be incorrect.

## Step 9 - Check that both circuits are equivalent.

A screenshot of a computer

AI-generated content may be incorrect.

**Both Circuit are not equivalent to each other**

# Each element is a tuple (X, Y)

truth\_a = [

(0,0), # A=0 B=0 C=0

(1,0), # A=0 B=0 C=1

(0,1),

(1,1),

(0,0),

(1,0),

(0,1), # example row 110

(1,1),

]

truth\_b = [

(0,0),

(1,0),

(0,1),

(1,1),

(0,0),

(1,0),

(1,1), # differs from truth\_a at row index 6

(1,1),

]

assert len(truth\_a) == len(truth\_b), "Both truth tables must have same length"

diffs = []

for i, (out\_a, out\_b) in enumerate(zip(truth\_a, truth\_b)):

A = (i >> 2) & 1

B = (i >> 1) & 1

C = i & 1

if out\_a != out\_b:

diffs.append((i, A, B, C, out\_a, out\_b))

if not diffs:

print("Equivalent: all outputs match for every input.")

else:

print("NOT equivalent. Differences found:")

for idx, A, B, C, out\_a, out\_b in diffs:

print(f" Row {idx}: A={A} B={B} C={C} CircuitA={out\_a} CircuitB={out\_b}")

A screenshot of a computer

AI-generated content may be incorrect.