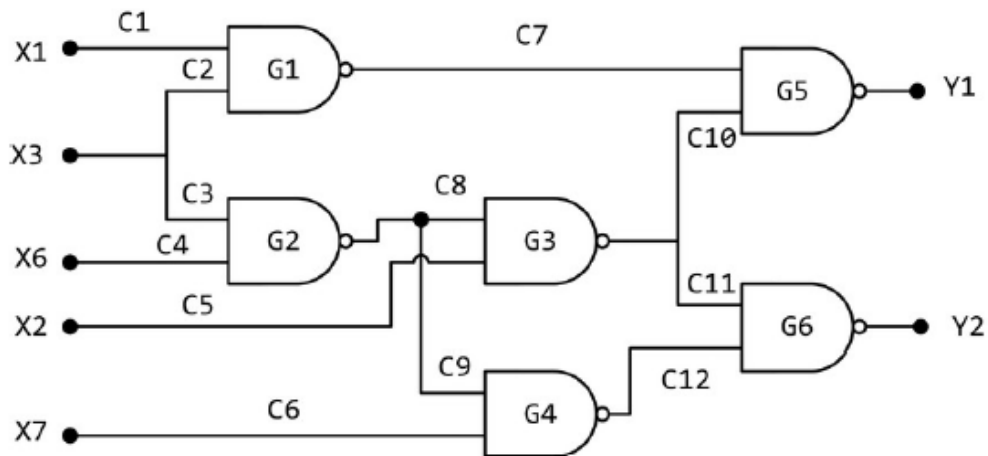


Final project:

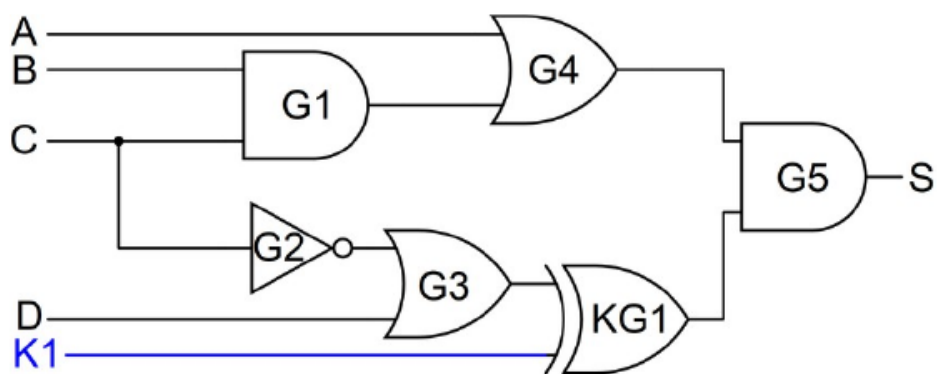
Giving circuits as following Fig 1. (a) and (b) show, please answer the following questions:

(1) Please find the test pattern for these circuits

(2) Please implement a STUMPS scheme to test the circuits as Fig. 1 (a) and (b) show.



(a)



(b)

1. (a)

根據題目提供的圖 (a) 的邏輯，電路的邏輯關係和操作如下：

- 理解圖 (a) 的邏輯結構

輸入變數：X1, X3, X6, X2, X7。

中間變數 (C1 到 C12) 和邏輯閘操作：

$$C1 = X1$$

$$C2 = X3$$

$$C3 = X3$$

$$C4 = X6$$

$$C5 = X2$$

$$C6 = X7$$

邏輯閘 (NAND 閘) 計算如下：

$$G1: C1 \text{ NAND } C2 \rightarrow C7$$

$$G2: C3 \text{ NAND } C4 \rightarrow C8$$

$$G2: C3 \text{ NAND } C4 \rightarrow C9$$

$$G3: C8 \text{ NAND } C5 \rightarrow C10$$

$$G3: C8 \text{ NAND } C5 \rightarrow C11$$

$$G4: C9 \text{ NAND } C6 \rightarrow C12$$

$$G5: C7 \text{ NAND } C10 \rightarrow Y1$$

$$G6: C11 \text{ NAND } C12 \rightarrow Y2$$

- 測試模式的設計

測試模式是對電路進行檢驗的一組輸入數據，我們需要針對圖 (a) 中所有可能的輸入組合來計算輸出。這樣做是為了確認電路能夠對每個輸入組合做出正確的反應。

輸入組合：圖 (a) 的電路有 5 個輸入變數 (X1, X3, X6, X2, X7)，每個輸入變數可以是 0 或 1。因此，總共有 $2^5 = 32$ 種不同的輸入組合。

輸出結果：對每組輸入，我們利用 python 程式將計算出對應的輸出 Y1 和 Y2，相結果輸出成 csv 檔。

| index | X1 | X3 | X6 | X2 | X7 | Y1 | Y2 |
|-------|----|----|----|----|----|----|----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 2 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 3 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 6 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 7 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 8 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 10 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 11 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 12 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 13 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 14 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 15 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 16 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 17 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 18 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 19 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 20 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 21 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 22 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 23 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 24 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 25 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 26 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 27 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 28 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 29 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 30 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 31 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

● 解析數據

數據列出了 32 個輸入組合和相對應的輸出 Y1 和 Y2。這些數據可以幫助我們理解每個輸入組合下的邏輯運行結果。這是從圖（a）所計算出來的測試模式。

最小測試向量的選擇

我們根據測試模式的輸入組合進行選擇。以下是根據圖（a）中邏輯運行，

推導出的最小測試向量（每一組測試向量後面是其對應的輸出）：

| X1 | X3 | X6 | X2 | X7 | Y1 | Y2 |
|----|----|----|----|----|----|----|
| 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 |

1. (b)

為此電路中的所有可能故障（如某節點卡 0 或卡 1）生成測試向量，確保可以檢測每個故障。

故障模型：單一故障模型（SAF）

考慮單一故障模型，即某節點的信號卡死為 0 或 1（SA0 或 SA1）。逐一分析每個節點的故障對應的測試模式。

節點清單

1. 輸入節點：A, B, C, D, K1A
2. 內部節點：
 - G1 的輸出
 - G2 的輸出
 - G3 的輸出
 - G4 的輸出
 - KG1 的輸出
 - G5 的輸出（S）

逐步分析與測試向量生成

步驟 1：測試 G1 的輸出故障

- 目標：檢測 G1 的輸出卡 0 或卡 1
- $G1=B \cdot C$
 - 若 G1 的輸出卡 0，需施加測試向量讓 $G1=1$ ：
 - 測試向量： $B=1, C=1$ ，此時 $G1=1$ 。
 - 若 G1 的輸出卡 1，需施加測試向量讓 $G1=0$ ：
 - 測試向量： $B=0$ 或 $C=0$ ，此時 $G1=0$ 。

步驟 2：測試 G2 的輸出故障

- 目標：檢測 G2 的輸出卡 0 或卡 1
- $G2=\neg C$
 - 若 G2 的輸出卡 0，需施加測試向量讓 $G2=1$ ：
 - 測試向量： $C=0$ 此時 $G2=1$ 。
 - 若 G2 的輸出卡 1，需施加測試向量讓 $G2=0$ ：
 - 測試向量： $C=1$ ，此時 $G2=0$ 。

步驟 3：測試 G3 的輸出故障

- 目標：檢測 G3 的輸出卡 0 或卡 1
- $G3=G2+D$ (OR 閘)
 - 若 G3 的輸出卡 0，需施加測試向量讓 $G3=1$ ：
 - 測試向量： $D=1$ 或 $G2=1$ (例如 $C=0$)。
 - 若 G3 的輸出卡 1，需施加測試向量讓 $G3=0$ ：
 - 測試向量： $D=0, G2 = 0$ (例如 $C=1$)。

步驟 4：測試 G4 的輸出故障

- 目標：檢測 G4 的輸出卡 0 或卡 1
- $G4=A+G1$ (OR 閘)

- 若 G4 的輸出卡 0，需施加測試向量讓 $G4=1$ ：
 - 測試向量： $A=1$ 或 $G1=1$ （例如 $B=1, C=1$ ）。
- 若 G4 的輸出卡 1，需施加測試向量讓 $G4=0$ ：
 - 測試向量： $A=0, G1=0$ （例如 $B=0$ 或 $C=0$ ）。

步驟 5：測試 KG1 的輸出故障

- 目標：檢測 KG1 的輸出卡 0 或卡 1
- $KG1 = G3 \oplus K1$ （XOR 閘）
 - 若 KG1 的輸出卡 0，需施加測試向量讓 $KG1=1$ ：
 - 測試向量： $G3=1, K1=0$ 或 $G3=0, K1=1$ 。
 - 若 KG1 的輸出卡 1，需施加測試向量讓 $KG1=0$ ：
 - 測試向量： $G3=0, K1=0$ 或 $G3=1, K1=1$ 。

步驟 6：測試 G5 的輸出故障

- 目標：檢測 G5 的輸出卡 0 或卡 1
- $G5 = G4 \cdot KG1$ （AND 閘）
 - 若 G5 的輸出卡 0，需施加測試向量讓 $G5=1$ ：
 - 測試向量： $G4=1, KG1=1$ 。
 - 例如： $A=1, K1=0, D=1, C=0$ 。
 - 若 G5 的輸出卡 1，需施加測試向量讓 $G5=0$ ：
 - 測試向量： $G4=0$ 或 $KG1=0$ 。
 - 例如： $A=0, K1=0, D=0 \Rightarrow A=0, K1=0, D=0$ 。

總結測試向量

綜合上述分析，以下是一些重要的測試向量：

1. $A=1, B=1, C=0, D=1, K1=0$ (檢測 $G5=1$)
 2. $A=0, B=0, C=1, D=0, K1=0$ (檢測 $G5=0$)
 3. $A=1, B=1, C=1, D=0, K1=1$ (檢測 $KG1$ 故障)
 4. $A=0, B=1, C=0, D=1, K1=0$ (檢測 $G4$ 和 $G3$)
-

2. (a)

完整步驟：設計與模擬 STUMPS 架構

1. 設計 LFSR (測試向量生成器)

- 使用 7 個 D 型觸發器，依次級聯，構成移位寄存器。
- 根據多項式 $P(x)=x^7+x^3+1$ $P(x) = x^7 + x^3 + 1$ 設計反饋：
 - 第 7 位和第 3 位的輸出 (Q 腳) 經 XOR 閘反饋至第 1 位的 D 腳。
- 確保所有觸發器的時鐘 (Clock 腳) 連接到同一個時鐘源。
- 初始化：模擬開始時，使用 Reset 信號設置觸發器的初始狀態為非零。

作用：LFSR 生成偽隨機測試向量，並連接到 CUT 的輸入 ($X1 \sim X7$)。

2. 設計 CUT (被測電路)

- 按提供的邏輯電路圖，使用 AND、OR、NAND 等邏輯閘實現。
- 確保 CUT 的輸入來自 LFSR 的輸出：
 - $X1$ 、 $X2$ 、 $X3$ 、 $X6$ 、 $X7$ 分別連接到對應的 LFSR Q 腳。
- 確保 CUT 的輸出 ($Y1$ 和 $Y2$) 正確連接到下一模組 (MISR)。

作用：CUT 接收偽隨機測試向量，生成響應信號 $Y1$ 和 $Y2$ 。

3. 設計 MISR (響應壓縮器)

- 使用 3 個 D 型觸發器，構成 3 位元的移位寄存器。

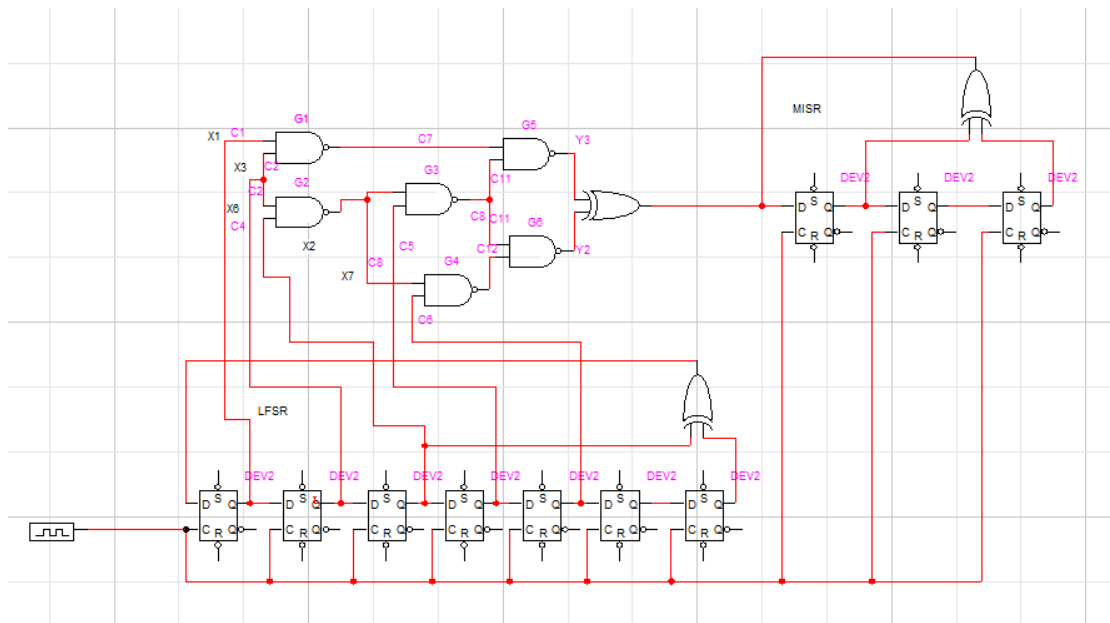
- 添加 XOR 閘：
 - 將 Y1 和 Y2 的輸出經 XOR 閘壓縮，連接到第一個 D 型觸發器的 D 腳。
- 根據多項式 $Q(x)=x^3+x+1$ 設計反饋：
 - 第 3 位和第 1 位的 Q 腳經 XOR 閘反饋至第一個觸發器的 D 腳。
- 確保 MISR 的時鐘信號與整體電路同步。

作用：MISR 壓縮 CUT 的輸出，生成最終的簽名值。

4. 模擬與驗證

1. 初始化：
 - 使用 Reset 信號初始化 LFSR 和 MISR 的狀態。
 - 確保觸發器的初始狀態非零。
2. 啟動模擬：
 - 啟動時鐘脈衝，讓 LFSR 生成偽隨機測試向量，傳遞給 CUT。
 - 同時，觀察 CUT 的輸出 (Y1 和 Y2)，以及 MISR 的簽名值變化。
3. 觀察與比對：
 - 使用探針 (Probe) 檢查：
 - LFSR 是否生成正確的偽隨機測試向量。
 - CUT 是否根據測試向量生成合理的響應。
 - MISR 的輸出簽名值是否與預期值一致。
 - 如果簽名值不匹配，CUT 可能存在故障。

5. 完成架講圖



2. (B)

完整步驟

1. 設計 LFSR (測試向量生成器)

1. 選擇位數：

- LFSR 需要 5 個 D 型觸發器，對應電路的輸入 A, B, C, D, K1

2. 連接：

- 將觸發器依次級聯：每個觸發器的 Q 腳連接到下一個觸發器的 D 腳。
- 根據多項式 $P(x)=x^5+x^3+1$ 將第 5 位 (Q5) 和第 3 位 (Q3) 的輸出通過 XOR 閘反饋到第 1 個觸發器的 D 腳。

3. 初始化：

- 使用 Reset 信號將 LFSR 初始化為非零狀態 (如 00001)。

4. 輸出連接：

- LFSR 的 5 個輸出 (Q1 ~ Q5) 分別連接到主電路 (CUT) 的輸入 A, B, C, D, K1。

2. 設計 CUT（被測電路）

1. 輸入：

- CUT 的輸入端 A, B, C, D, K1 接收來自 LFSR 的測試向量。

2. 內部邏輯：

- 根據電路圖設計邏輯閘組合（如 AND、OR、NOT、XOR 等）。
- CUT 的輸出為 S。

3. 輸出連接：

- 將 CUT 的輸出 S 連接到響應壓縮器（MISR）的輸入端。

3. 設計 MISR（響應壓縮器）

1. 選擇位數：

- 使用 3 個 D 型觸發器構成響應壓縮器。

2. 連接：

- 將 CUT 的輸出 S 與反饋信號通過 XOR 閘壓縮，連接到第 1 個觸發器的 D 腳。
- 根據多項式 $Q(x)=x^3+x+1$ ，將第 3 位（Q3）和第 1 位（Q1）的輸出通過 XOR 閘反饋到第 1 個觸發器的 D 腳。

3. 時鐘信號：

- LFSR 和 MISR 的所有觸發器共享同一時鐘源，保證同步運行。

4. 輸出：

- MISR 的 3 個觸發器的 Q1, Q2, Q3 為壓縮後的簽名值。

4. 模擬與驗證

1. 初始化：

- 使用 Reset 信號初始化 LFSR 和 MISR 的初始狀態，確保 LFSR 為非零狀態。

2. 啟動模擬：

- 啟動時鐘信號，讓 LFSR 生成偽隨機測試向量，驅動 CUT 生成輸出 S，MISR 壓縮響應。

3. 觀察輸出：

- 使用探針（Probe）觀察：
 - LFSR 的輸出是否生成偽隨機序列。
 - CUT 的輸出 S 是否正確。
 - MISR 的簽名值 Q1, Q2, Q3 是否更新。

4. 記錄最終簽名：

- 在模擬完成後，記錄 MISR 的最終狀態，與預期簽名值比對，確認 CUT 是否正常。

