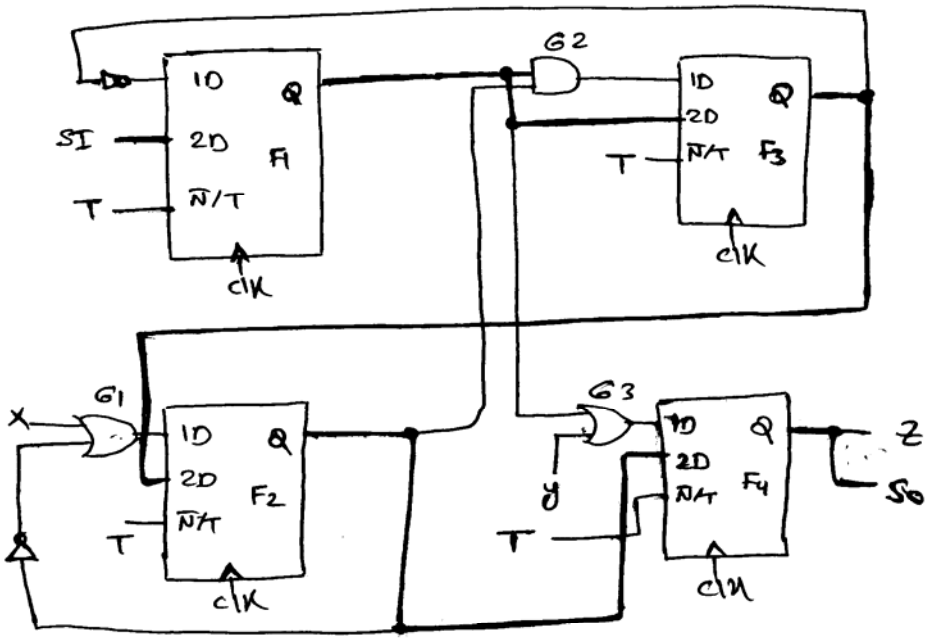
HW#5 Solution

Q1.

(1) Full-Scanned Circuit:

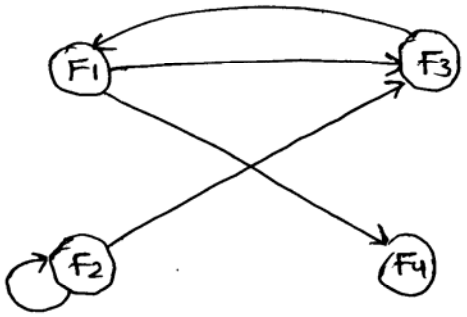


To test the fault G2 s-a-0, we need to set G2=1 which requires F1=1 and F2=1.

Thus, we set T=1 and SI={1,1} and apply 2 clk cycles. This, will shift the values to F1 and F2. Then, we set T=0 and pulse one clk. This will capture the fault effect in F3. Then, we set T=1 and apply one clk to shift the value to F4 and we observe the fault effect at So.

(ii) Partial Scan design

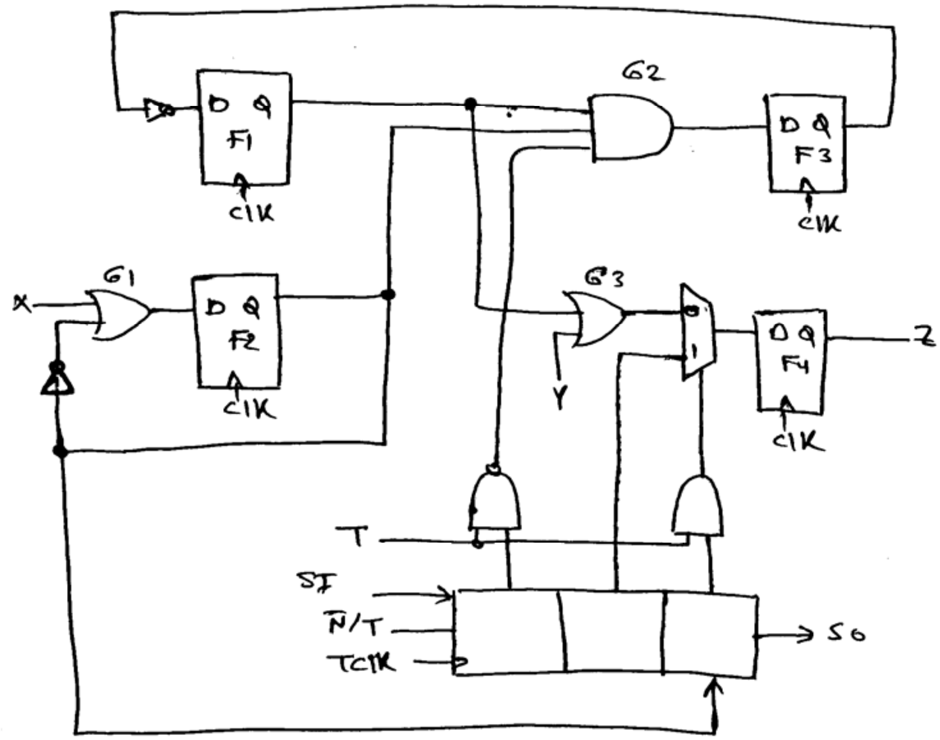
we will select the flip-flop based on breaking the largest number of loops and reducing the sequential depth. To do that, we need to construct the S-graph.



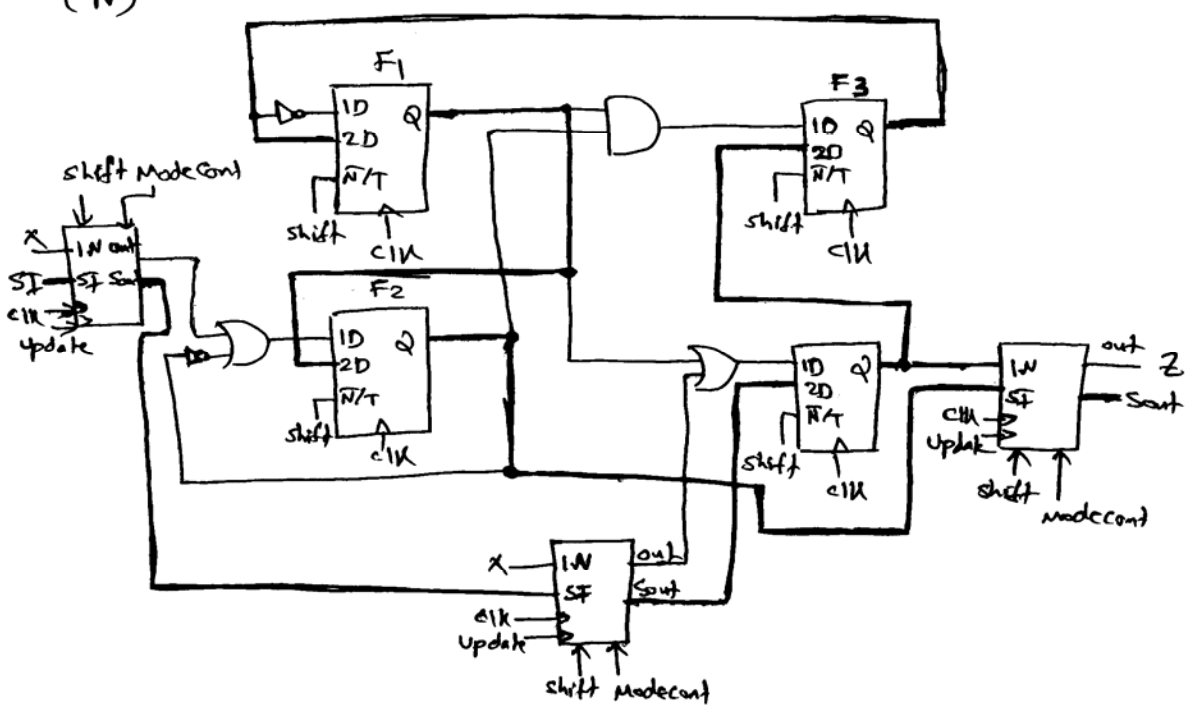
To break the loops, we can select either F1 or F3. Selecting either F1 or F3, the maximum sequential depth is 2. Thus, any one can be selected.

For partial scan design, we need to use a scan flip-flop that uses a separate test clock from the system clock so that the content of sequential elements is not disturbed during scan. An example of this is the two-port dual-clock flop-flop (2p-FF).

(iii)


G2
D Q
F1
clk
D Q
F3
clk
G1
x
D Q
F2
clk
G3
0
1
Y
D Q
F4
clk
Z
T
SI
N̄/T
TClk
SO

(iv)


F1
1D
2D
N̄/T
Q
shift
clk
F3
1D
2D
N̄/T
Q
shift
clk
shift Modecont
x
IN
out
SI
SI
Sout
clk
update
F2
1D
2D
N̄/T
Q
shift
clk
1D
2D
N̄/T
Q
shift
clk
IN
out
Z
SI
Sout
clk
update
shift
modecont
x
IN
out
SI
Sout
clk
update
shift
Modecont

Q2. $p(x) = 1 + x^2 + x^3$

(1) In order to verify or show that $p(x)$ is primitive polynomial, we need to show that it is a maximal length with period = 7.
This, we need to show that the smallest integer K such that $1 - x^K$ divides $p(x)$ evenly is 7.

$$
\begin{array}{r}
x+1 \\
1+x^2+x^3 \;\overline{)\; 1-x^4} \\
x+x^3+x^4 \\
\hline
1+x+x^3 \\
1+x^2+x^3 \\
\hline
x+x^2
\end{array}
$$

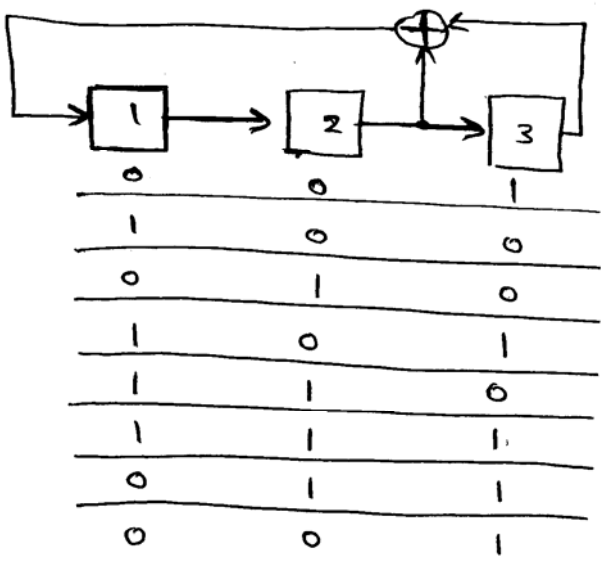
$$
\begin{array}{r}
x^2+x+1 \\
1+x^2+x^3 \;\overline{)\; 1-x^5} \\
x^2+x^4+x^5 \\
\hline
1+x^2+x^4 \\
x+x^3+x^4 \\
\hline
1+x+x^2+x^3 \\
1+x^2+x^3 \\
\hline
x
\end{array}
$$

$$
\begin{array}{r}
x^3+x^2+x \\
1+x^2+x^3 \;\overline{)\; 1-x^6} \\
x^3+x^5+x^6 \\
\hline
1+x^3+x^5 \\
x^2+x^4+x^5 \\
\hline
1+x^2+x^3+x^4 \\
x+x^3+x^4 \\
\hline
1+x+x^2
\end{array}
$$

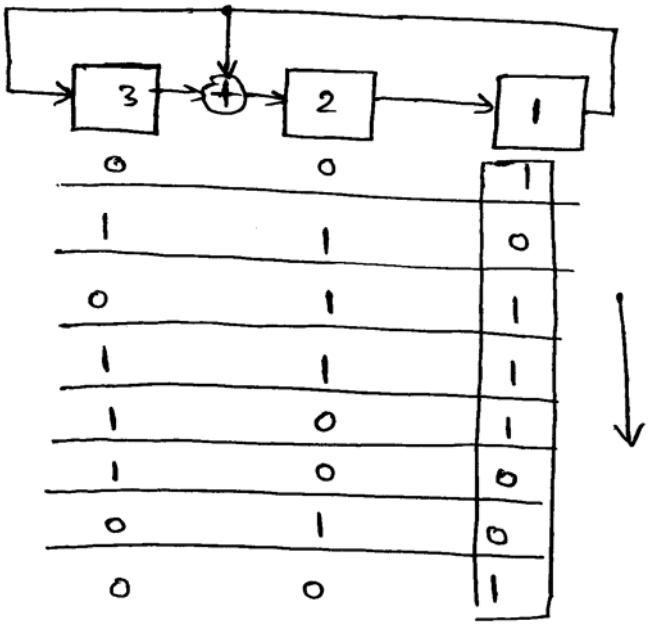
$$
\begin{array}{r}
x^4+x^3+x^2+1 \\
1+x^2+x^3 \;\overline{)\; 1-x^7} \\
x^4+x^6+x^7 \\
\hline
1+x^4+x^6 \\
x^3+x^5+x^6 \\
\hline
1+x^3+x^4+x^5 \\
x^2+x^4+x^5 \\
\hline
1+x^2+x^3 \\
1+x^2+x^3 \\
\hline
0
\end{array}
$$

This, since $K = 7$ is the smallest integer such that $1 - x^K$ divides $p(x)$ evenly, then the period is 7 and $p(x)$ is primitive.

(ii) Type 1 LFSR



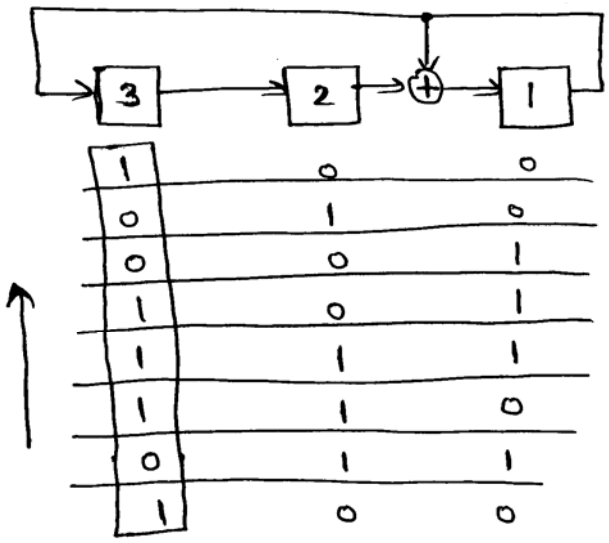
(iii) Type 2 LFSR



(iv) $p^*(x) = x^3 [1 + x^{-2} + x^{-3}]$

$= x^3 + x + 1$

(v) Type 2 LFSR of $p^*(x)$



Thus, we can see the most significant bit in this LFSR produces the reverse squence of the least significant bit of the LFSR of $p(x)$.

Q3. $p(x) = 1 + x + x^2$ $\quad p^*(x) = x^2 + x + 1$

$G(x) = x^7 + x^5 + x^4 + x^2 + x + 1$

(i) 10110000

$e(x) = x^7 + x^5 + x^4$

$e(x) = x^2 + x + 1$

since $e(x)$ is a multiple of $p^*(x)$, the erroneous output will not be detected and the same remainder will be produced.

(ii) 10111010

$G'(x) = x^7 + x^5 + x^4 + x^3 + x$

$e(x) = x^3 + x^2 + 1$

Since $e(x)$ is not a multiple of $p^*(x)$, the error will be detected.

(iii) 01110111

$G'(x) = x^6 + x^5 + x^4 + x^2 + x + 1$

$e(x) = x^7 + x^6$

Since $e(x)$ is not a multiple of $p^*(x)$, the error will be detected

Q4. Assuming that the initial content of the LFSR is a b c, then the content of the scan cham will be as follows:

8	7	6	5	4	3	2	1
c	a ⊕ c	a ⊕ b ⊕ c	a ⊕ b	b ⊕ c	a	b	c

To detect the fault, it is required to have scan cell 2 = 0, scan cell 5 = 1 and scan cell 7 = 1. This implies that $b = 0$, $a \oplus b = 1 \Rightarrow a = 1$ and $a \oplus c = 1 \Rightarrow c = 0$. Thus, by seeding the LFSR with the seed {100} the fault will be detected.

Q5.

(i) **Full scanned circuit (as PI & PO):**

The fault coverage and CPU time after running for one iteration:
Fault coverage = 96.67 %
CPU time = 69s

(ii) **Full scanned circuit (add MUX):**

The fault coverage and CPU time after running for one iteration:
Fault coverage = 86.23%
CPU time = 3933s

As can be seen, although the circuit has full scan, the sequential ATPG spent much more time than running a combinational ATPG and then translating the generated vectors into scan vectors. In addition, the fault coverage achieved is much less. This shows the limitation of sequential ATPGs.

(iii) **Partial Scan:**

Partial scan results are generated after running ATPG for one iteration as follows:

Partial Scan Option	20%=107	30%=160	50%=267
Opus –n -i	Fault Coverage=85.73 CPU Time=2582.13	Fault Coverage=94.46 CPU Time=260.22	Fault Coverage=94.68 CPU Time=175
Opus –t -i	Fault Coverage=91.90 CPU Time=1158.02	Fault Coverage=94.07 CPU Time=402.52	Fault Coverage=94.51 CPU Time=158.53
Opus -t	Fault Coverage=79.75 CPU Time=4071.18	Fault Coverage=85.62 CPU Time=2554.33	Fault Coverage=93.06 CPU Time=578.68

As can be seen from the results, partial scan selection based on iterative testability measures based selection achieves better results in terms of fault coverage and CPU time when the percentage of selected scan flip-flops is small (20% in this case). However, when the percentage of scanned flip-flops is 30% and more, partial scan selection based on cutting loops followed by iterative testability measure based selection achieves the best results in terms of both fault coverage and CPU time. It should be noted these observations are for this circuit and may not apply for other circuits.