Chapter 4

Sequential Logic Circuits

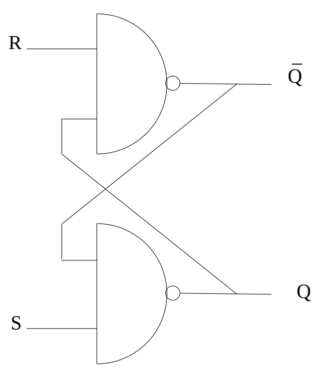
4.1 Problems

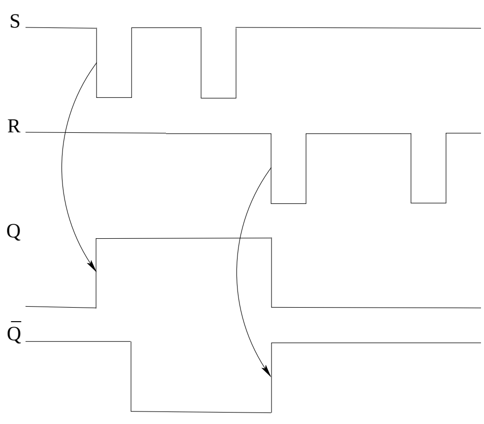
Problem 4.1 (SR-FF using two NOR gates) *It is not allowed to have $S = 1$ and $R = 1$ at the same time for a SR-FF that is implemented with NOR gates as shown in Figure **??**. What are the values of Q and $\overline{Q}$ when $S = 1$ and $R = 1$ if by mistake or during power-up?*

(ans:
$S = 1$ causes S-OR $X = 1$, making $\overline{Q} = 0$. $R = 1$ causes R-OR $Y = 1$, making $Q = 0$.
Hence $S = 1$ and $R = 1$ produce $\overline{Q} = 0$ and $Q = 0$. This violates the rule that $\overline{Q}$ is the complement of Q.
)

Problem 4.2 (Implementing an SR-FF using two NAND gates) *Implement an SR-FF using two NAND gates and describe its behavior using a timing diagram and state transition table.*

(ans:
When NAND gates are used, both S and R are normally $= 1$, which is the stable condition, and it is not allowed for both S and R to be 1 at the same time. The figure below shows the case with $Q = 0$.





When S makes a $1 \rightarrow 0$ transition either $Q \rightarrow 1$, making $\overline{Q} \rightarrow 0$, or Q stays at 1.

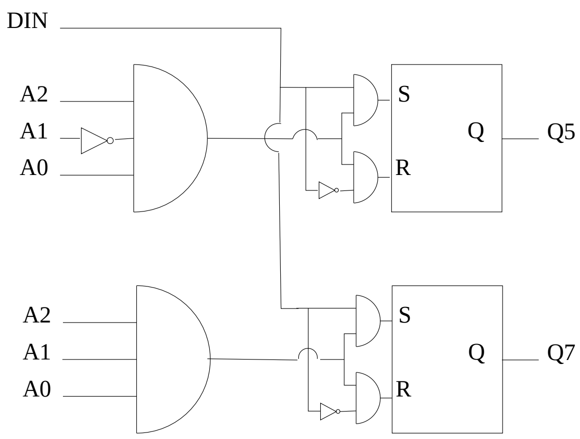
When R makes a $1 \rightarrow 0$ transition either $\overline{Q} \rightarrow 1$, making $Q \rightarrow 0$, or $\overline{Q}$ stays at 1.

S	R	Q(cur)		Q(next)
1	1	0		0
1	1	1		1
0	1	0		1
0	1	1		1
1	0	0		0
1	0	1		0

)

Problem 4.3 (Random-access digital memory input) *Design a random-access digital memory that* stores *single bits in two addressable SR-FFs having 3-bit addresses 5 and 7. The memory has the address input lines $A2, A1, A0$ and data input line DIN and the two SR-FF outputs are Q5 and Q7.*

(ans:

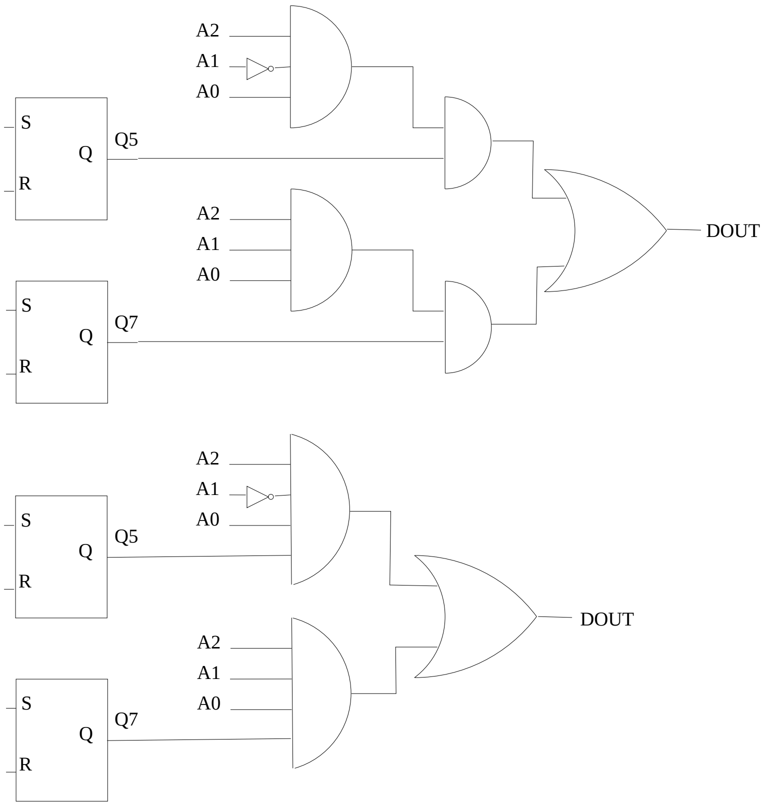


)

Problem 4.4 (Random-access digital memory output) *Design a random-access digital memory that* reads *single bits stored in two addressable SR-FFs having 3-bit addresses 5 and 7. The memory has address input lines $A2, A1, A0$, and data output line DOUT.*

(ans:

Two implementations are shown. The first is a direct extension of that described in the book. The second is a more efficient use of AND gates.

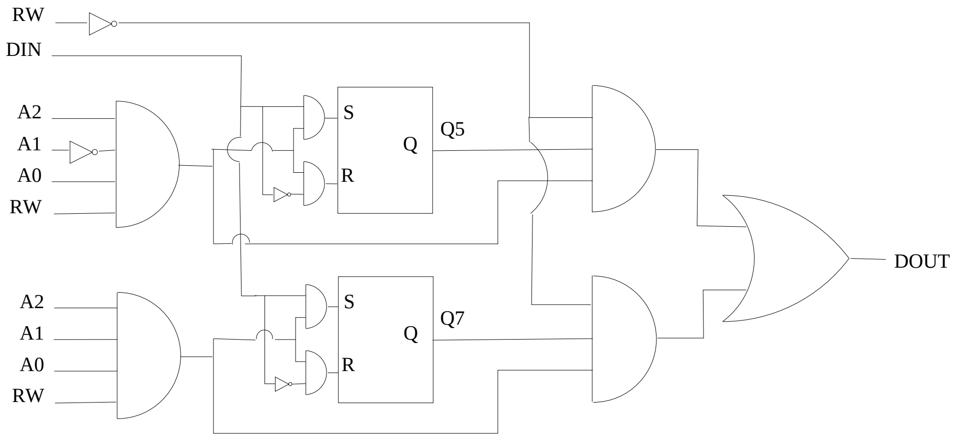


)

Problem 4.5 (Random-access digital memory input/output) *Design a random-access digital memory that can write or read single bits stored in two addressable SR-FFs having 3-bit addresses 5 and 7. The memory has the following inputs and outputs*

- *Address input lines* $A2, A1, A0$
- *Data input line* DIN
- *Data output line* $DOUT$
- *Read/Write input line* RW *(* $RW = 0$ *stores* DIN *into addressed memory.* $RW = 1$ *puts addressed memory data on* $DOUT$*).*

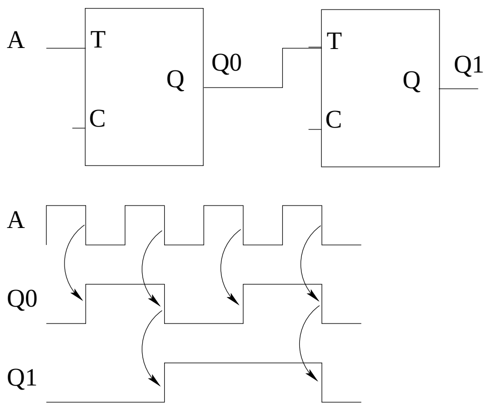
(ans:



)

Problem 4.6 (Counting chain containing two T-FFs) *Sketch the counting chain containing two T-FFs and its timing diagram, starting with $Q_1Q_0 = 00$. Show the transitions and the count values over a complete count cycle.*

(ans:



)

Problem 4.7 (Modulo-365 counter) *Implement an efficient modulo-365 counter using the following steps.*

1. *How many T-FFs are needed?*
2. *What is the binary pattern representing 365?*
3. *Show that only the 1's in the pattern representing 365 need to be inputs to the recognition AND gate by replacing one of the 0s in the pattern with 1s and computing the resulting decimal value.*
4. *Draw the simplified Modulo-365 counter circuit.*

(*ans:*

1.

$$M = ceiling(\log_2 365) = ceiling\left(\frac{\log_{10} 365}{\log_{10} 2}\right) = ceiling(8.5) = 9$$

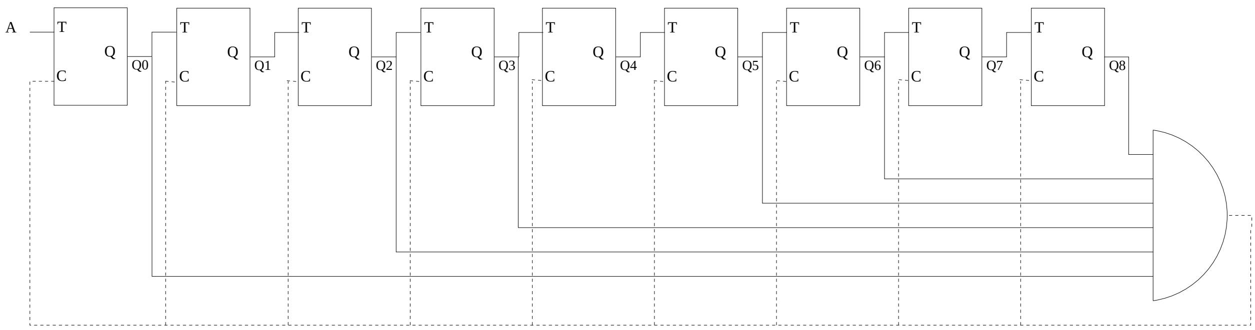
2.

$$365_{10} = 101101101_2 = 256 + 64 + 32 + 8 + 4 + 1$$

3. *With bit pattern denoted as Q_8 (MSB) to Q_0 (LSB),*
   Q_7 $0 \to 1$:
   $$1\ 1\ 1101101_2 = 256 + 128 + 64 + 32 + 8 + 4 + 1 = 493$$
   Q_4 $0 \to 1$:
   $$1011\ 1\ 1101_2 = 256 + 64 + 32 + 16 + 8 + 4 + 1 = 381$$
   Q_1 $0 \to 1$:
   $$1011011\ 1\ 1_2 = 256 + 64 + 32 + 8 + 4 + 2 + 1 = 367$$

4. *Draw the simplified Modulo-365 counter circuit.*

   Inputs from Q_7, Q_4, and Q_1 to the AND are not needed. Technically, since these outputs are already at 0, the inputs to the clear (C) of their T-FFs are not necessary, but they must then the Cs must be connected to ground (not left floating). So, the AND gate output should connect to the C inputs of all T-FFs.



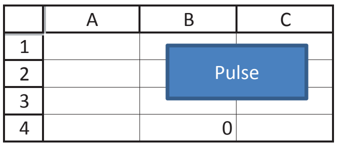
)

4.2 Excel Projects

Project 4.1 (Generate a two-second pulse) *Modify Example 13.19 to generate a two-second pulse in cell B4 each time a shape labeled* pulse *is clicked.*

(*ans: The VBA Macro changes from a "1" to "2" in the* `TimeSerial` *function and "A1" to "B4" the* `Range` *function, as*

```
Sub pulse()
' Generate a 1 second pulse in cell B4
Range("B4").Value = 1           ' set pulse high (=1)
Application.Wait Now + TimeSerial(0, 0, 2) '  2 s delay
Range("B4").Value = 0           ' set pulse low (=0)
End Sub
```



)

Project 4.2 (Set Reset Flip-Flop) *Modify Example 13.20 to generate a two-second pulse in B1 each time "Set" is clicked, and in B2 each time "Reset" is clicked.*

(ans: The VBA Macros change from a "1" to "2" in the `TimeSerial` *functions, as*

```
Sub Set_SR_FF()
' Sets the Set-Reset FF with pulse on Set input
' B1 is set, B2 is reset, E1 is Q, E2 is ~Q
Range("B1").Value = 1 ' Set 0 -> 1
Range("E1").Value = 1 ' Q -> 1
Range("E2").Value = 0 ' ~Q -> 0
Application.Wait Now + TimeSerial(0, 0, 2) ' 2 s delay
Range("B1").Value = 0 ' Set 1 -> 0
End Sub
------------------------------------------------------
Sub Reset_SR_FF()
' Resets the Set-Reset FF with pulse on Reset input
' B1 is set, B2 is reset, E1 is Q, E2 is ~Q
Range("B2").Value = 1   ' Reset 0 -> 1
Range("E1").Value = 0   ' Q -> 0
Range("E2").Value = 1   ' ~Q -> 1
Application.Wait Now + TimeSerial(0, 0, 2) ' 2 s delay
Range("B2").Value = 0   ' Reset 1 -> 0
End Sub
```

)

Project 4.3 (Three Toggle Flip-Flop chain) *Extend Example 13.21 from 2 T-FF to 3 T-FFs. The chain state increments with every click of the "A" input and clears the T-FF chair to zero by clicking on the* Clear *shape. Each click on A maintains the 0 value for one second, followed by a 1 value that lasts for one second, and followed by a downward transition back to 0.*

(ans: The worksheet is modified by including Q2 in Column E. Note the T-FFs change on downward transitions on the T input and on the upward transition on the CLR inputs.

	A	B	C	D	E
1	A		Q0	Q1	Q2
2	0		0	0	0
3					
4	A	Clear		C	
5				0	

The `Clear_3T_FF` *Macro includes a clear in E2, as*

```
Sub Clear_3T_FF()
Range("D5").Value = 1   ’ Clear 0 -> 1
Range("C2").Value = 0   ’ FF0-Q -> 0
Range("D2").Value = 0   ’ FF1-Q -> 0
Range("E2").Value = 0   ’ FF2-Q -> 0
Application.Wait Now + TimeSerial(0, 0, 1) ’ 1 s delay
Range("D5").Value = 0   ’ Clear 1 -> 0
End Sub
```

The `Pulse_3T_FF_Chain` *Macro includes the Boolean value for Q2* `DW2` *and the code delimited by* `’---`.

```
Sub Pulse_3T_FF_Chain()
Dim DW0 As Boolean ’ downward transition on Q0-T
Dim DW1 As Boolean ’ downward transition on Q1-T
Dim DW2 As Boolean ’ downward transition on Q2-T
Application.Wait Now + TimeSerial(0, 0, 1) ’ 1s delay
Range("A2").Value = 1   ’ FF0-T = 1
Application.Wait Now + TimeSerial(0, 0, 1) ’ 1s delay
Range("A2").Value = 0   ’ FF0-T 1 -> 0
DW0 = True              ’ DW transition on FF0-T
If DW0 = True Then  ’ Toggle FF0 (First FF always True but makes code simpler)
    If Range("C2").Value = 0 Then
        Range("C2").Value = 1    ’ 0 -> 1
        DW1 = False              ’ upward transition
    Else
        Range("C2").Value = 0    ’ 1 -> 0 DW transition
        DW1 = True               ’ downward transition on FF1-T
    End If
End If
If DW1 = True Then  ’ Toggle FF1
    If Range("D2").Value = 0 Then
        Range("D2").Value = 1    ’ 0 -> 1
        DW2 = False              ’ upward transition
    Else
        Range("D2").Value = 0    ’ 1 -> 0 DW transition
        DW2 = True               ’ downward transition on FF2-T (if any)
    End If
End If
’---
If DW2 = True Then  ’ Toggle FF2
    If Range("E2").Value = 0 Then
```

```
        Range("E2").Value = 1   ' 0 -> 1
     Else
        Range("E2").Value = 0   ' 1 -> 0 DW transition
     End If
End If
'---
End Sub
```

)

Project 4.4 (Modulo-6 counter) *Modify Example 13.22 to implement an efficient* $modulo - 6$ *counter.*

(ans: The "Demo" button shows a free-running demonstration of the counter. The Macro is terminated by pressing "Esc" twice.

The `Mod6Counter` *Macro includes the code to recognize a 6 delimited by* `'---`*.*

```
Sub Mod6Counter()
Dim Q0 As Boolean
Dim Q1 As Boolean
Dim Q2 As Boolean
Dim C As Boolean
Clear_3TFF
Do While Range("A2").Value < 2  ' infinite loop
    Pulse_3TFF_Chain
    ' check state
    Q0 = True            ' default value
    If Range("C2").Value = 0 Then ' change if needed
     Q0 = False
    End If
    Q1 = True            ' default value
    If Range("D2").Value = 0 Then ' change if needed
     Q1 = False
    End If
    Q2 = True            ' default value
    If Range("E2").Value = 0 Then ' change if needed
     Q2 = False
    End If
    ' compute C
'---
    C = Q1 And Q2      ' logic equation for clearing on 110
'---
    If C = True Then
        Range("D5").Value = 1
    Else
        Range("D5").Value = 0
    End If
    If Range("D5").Value = 1 Then   ' check if Clear = 0
        Clear_3TFF                  ' reset T-FF chain
    End If
Loop
End Sub
```

)

Project 4.5 (Modulo-14 counter) *Modify Example 13.22 to implement a mod-14 counter by extending the T-FF chain to operate with four T-FFs.*

(ans: The "Demo" button shows a free-running demonstration of the counter. The Macro is terminated by pressing "Esc" twice.

	A	B	C	D	E	F
1	A		Q0	Q1	Q2	Q3
2	0		1	0	0	1
3	Demo Mod-14 Counter					
4				C		
5				0		

The `Clear_4TFF` *Macro includes the code to also clear* `F2`*, delimited by* `'---`.

```
Sub Clear_4TFF()
' Clear signal in D5
Range("D5").Value = 1   ' C -> 1
Range("C2").Value = 0   ' Q0 -> 0
Range("D2").Value = 0   ' Q1 -> 0
Range("E2").Value = 0   ' Q2 -> 0
'---
Range("F2").Value = 0   ' Q3 -> 0
'---
Application.Wait Now + TimeSerial(0, 0, 1) ' 1 s delay
Range("D5").Value = 0   ' C -> 0
End Sub
```

The `Pulse_4TFF_Chain` *Macro includes the code to form a fourth T-FF delimited by* `'---` *below.*

```
Sub Pulse_4TFF_Chain()
Dim DW0 As Boolean ' downward transition on Q0-T
Dim DW1 As Boolean ' downward transition on Q1-T
Dim DW2 As Boolean ' downward transition on Q2-T
Dim DW3 As Boolean ' downward transition on Q3-T
'---
Dim DW4 As Boolean ' downward transition on Q4-T
'---
Application.Wait Now + TimeSerial(0, 0, 1) ' 1 s delay
Range("A2").Value = 1    ' FF0-T = 1
Application.Wait Now + TimeSerial(0, 0, 1) ' 1 s delay
Range("A2").Value = 0    ' FF0-T 1 -> 0
DW0 = True               ' DW transition on FF0-T
If DW0 = True Then  ' Toggle FF0 (First FF always True but makes code simpler)
    If Range("C2").Value = 0 Then
        Range("C2").Value = 1    ' 0 -> 1
        DW1 = False              ' upward transition
    Else
        Range("C2").Value = 0    ' 1 -> 0 DW transition
        DW1 = True               ' downward transition on FF1-T
    End If
End If
If DW1 = True Then  ' Toggle FF1
    If Range("D2").Value = 0 Then
        Range("D2").Value = 1    ' 0 -> 1
        DW2 = False              ' upward transition
    Else
        Range("D2").Value = 0    ' 1 -> 0 DW transition
```

```
        DW2 = True              ' downward transition on FF2-T (if any)
    End If
End If
If DW2 = True Then  ' Toggle FF2
    If Range("E2").Value = 0 Then
        Range("E2").Value = 1   ' 0 -> 1
        DW3 = False             ' upward transition
    Else
        Range("E2").Value = 0   ' 1 -> 0 DW transition
        DW3 = True              ' downward transition on FF3-T (if any)
    End If
End If
'-----
If DW3 = True Then  ' Toggle FF3
    If Range("F2").Value = 0 Then
        Range("F2").Value = 1   ' 0 -> 1
        DW4 = False             ' upward transition
    Else
        Range("F2").Value = 0   ' 1 -> 0 DW transition
        DW4 = True              ' downward transition on FF4-T (if any)
    End If
End If
'----
End Sub
```

The `Mod14Counter` *Macro demonstrates the counter. To form a fourth T-FF the code delimited by* `'---` *was included. The test condition for clearing is delimited by* `'***`*.*

```
Sub Mod14Counter()
Dim Q0 As Boolean
Dim Q1 As Boolean
Dim Q2 As Boolean
'---
Dim Q3 As Boolean
'---
Dim C As Boolean
Clear_4TFF
Do While Range("A2").Value < 2  ' infinite loop
    Pulse_4TFF_Chain
    ' check state
    Q0 = True           ' default value
    If Range("C2").Value = 0 Then ' change if needed
     Q0 = False
    End If
    Q1 = True           ' default value
    If Range("D2").Value = 0 Then ' change if needed
     Q1 = False
    End If
    Q2 = True           ' default value
    If Range("E2").Value = 0 Then ' change if needed
     Q2 = False
    End If
'---
    Q3 = True           ' default value
    If Range("F2").Value = 0 Then ' change if needed
     Q3 = False
    End If
'---
    ' compute C

'***
    C = Q3 And Q2 And Q1     ' logic equation for clearing on 1110
'***
    If C = True Then
        Range("D5").Value = 1
```

```
    Else
        Range("D5").Value = 0
    End If
    If Range("D5").Value = 1 Then   ' check if Clear = 0
'---
    Clear_4TFF                  ' reset T-FF chain
'---
    End If
Loop
End Sub
```

)