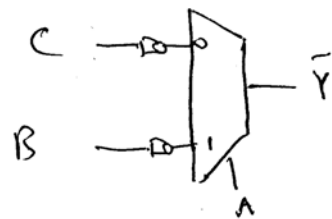
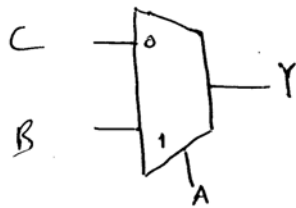


- To implement a function in CMOS, we need to implement the ON-set in pmos and the Off-set in nmos.
In other words, we need to implement the function in pmos and its complement in nmos.
- To find the complement of a function, we can use the Karnaugh-map to guarantee that we have the simplest implementation.
- Useful information:

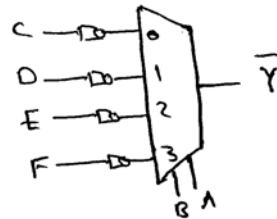
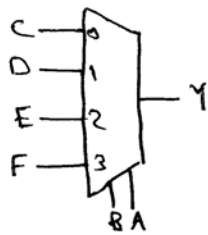
We can quickly find the complement of multiplexor structures as follows.

$$Y = A B + \bar{A} C \quad \bar{Y} = A \bar{B} + \bar{A} \bar{C}$$



$$Y = \bar{A}\bar{B}C + \bar{A}B\bar{D} + A\bar{B}E + ABF$$

$$\bar{Y} = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{D} + A\bar{B}\bar{E} + AB\bar{F}$$



Example

Implement the following function in CMOS

$$Y = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D} \\ + \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} \\ + AB\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}\bar{D}$$

Solution

We first use the Karnaugh map to make sure that we have the simplest implementation for the function.

		CD			
		00	01	11	10
AB	00	1	1	0	1*
	01	1	1	0	1*
	11	0	1	1*	0
	10	1*	1	1*	0

There are two possible implementations that are minimal namely

$$Y = \bar{B}\bar{C} + AD + \bar{A}\bar{D} + \bar{A}\bar{C}$$

$$\text{or } Y = \bar{B}\bar{C} + AD + \bar{A}\bar{D} + \bar{C}D$$

For the complement of the function \bar{Y}

		CD			
		00	01	11	10
AB	00	1	1	0	1
	01	1	1	0	1
	11	0	1	1	0
	10	1	1	1	0

$$\bar{Y} = \bar{A}CD + AB\bar{D} + A\bar{C}\bar{D}$$

Note that we could have found the complement \bar{Y} symbolically as follows.

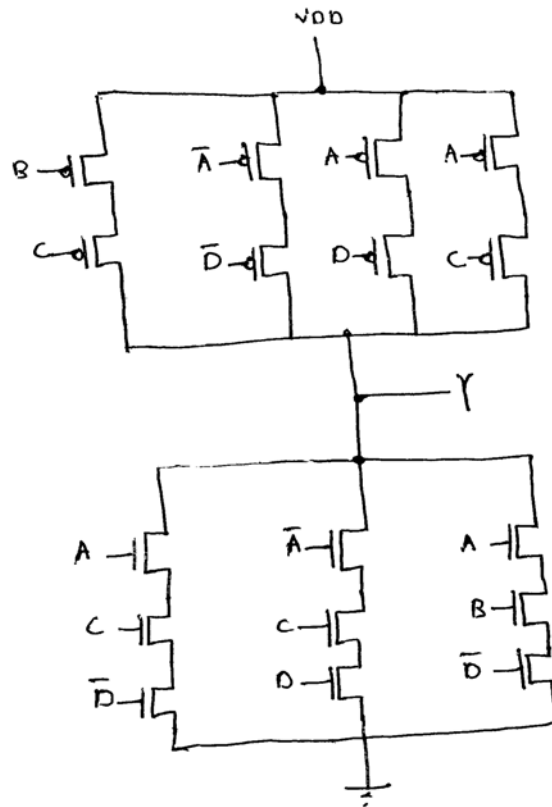
Let us choose $Y = \bar{B}\bar{C} + AD + \bar{A}\bar{D} + \bar{A}\bar{C}$

This can be represented as follows

$$Y = \bar{C}(\bar{B} + \bar{A}) + AD + \bar{A}\bar{D}$$

$$\begin{aligned} \bar{Y} &= [\bar{C}(\bar{B} + \bar{A}) + AD + \bar{A}\bar{D}]' \\ &= [\bar{C}(\bar{B} + \bar{A})]' \cdot [AD + \bar{A}\bar{D}]' \\ &= [C + BA] \cdot [A\bar{D} + \bar{A}D] \\ &= AC\bar{D} + \bar{A}CD + AB\bar{D} \end{aligned}$$

Thus, the CMOS implementation of the function can be as follows:

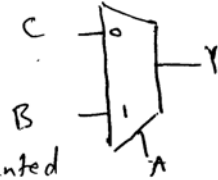


- Multiplexers

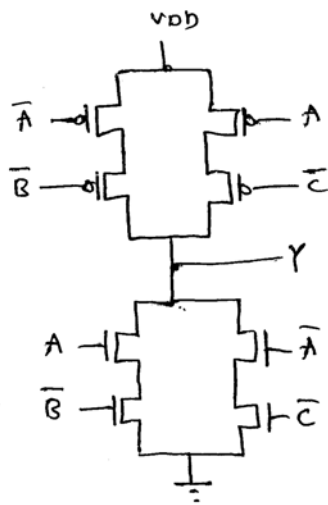
A multiplexer is represented as follows

$$Y = A B + \bar{A} C$$

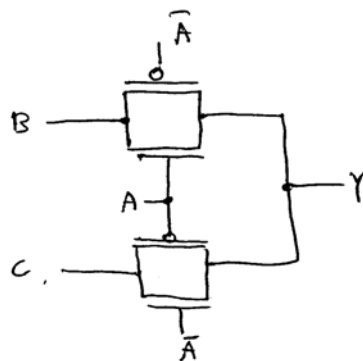
$$\bar{Y} = A \bar{B} + \bar{A} \bar{C}$$



Thus, this can be implemented as follows:



However, a more efficient implementation can be done using transmission gates as follows:



Analyzing this implementation, we can see that when $A=1$, the upper transmission gate will conduct while the lower transmission gate will be off. Hence, $Y=B$. Also, when $A=0$, the lower gate will conduct while the higher gate will be off. So, $Y=C$. Thus, the conditions under which $Y=1$ are AB or $\bar{A}C$. Thus, $Y = AB + \bar{A}C$.

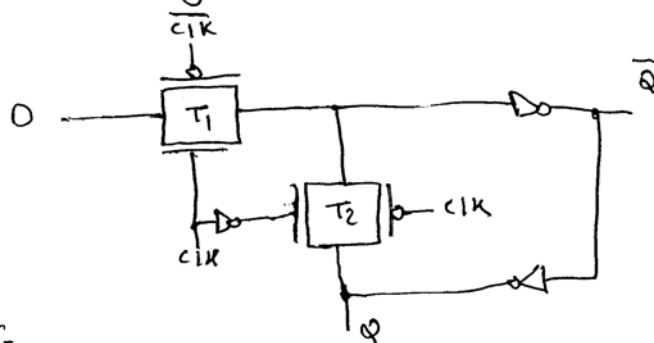
- D-Latch

The characteristic of the d-latch is that when the $clk = 1$, then $Q = D$.



However, when $clk = 0$, then the d-latch keeps or holds the previous value i.e., $Q^t = Q^{t-1}$.

The d-latch can be implemented in CMOS using transmission gates as follows:

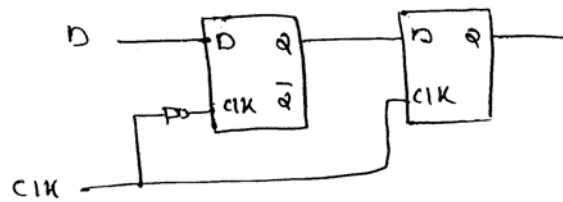


Analyzing this implementation, we can see that when $CLK=1$, the gate T_1 will be on while T_2 will be off, and $Q=D$. When $CLK=0$, then T_1 is off while T_2 is on and Q will hold its value.

- D-flip-flop:

A d-ff can be implemented using two d-latches based on the master-slave structure. The characteristic of a rising-edge d-ff is that $Q=D$ when the CLK changes from 0 to 1, i.e. after the rising edge of the CLK .

rising-edge master-slave d-ff:



The rising-edge d-flip can be implemented in CMOS as follows:

