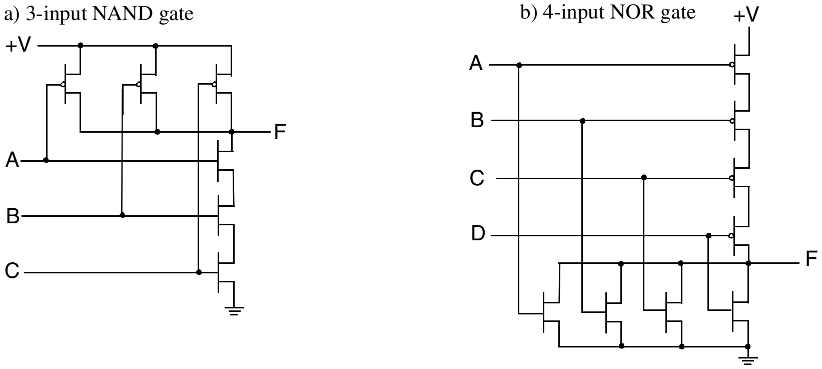
CHAPTER 6

6-1.*

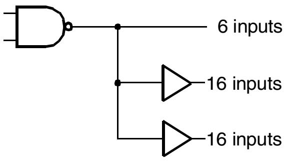
a) $F = (\overline{A} + B)\,C\,D$

b) $G = (A + \overline{B})\,(\overline{C} + D)$

6-2.



6-3.

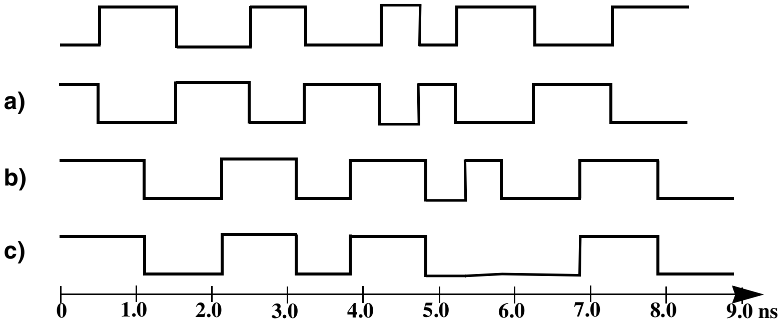


6-4.*

The longest path is from input C or $\overline{D}$.

0.073 ns + 0.073 ns + 0.048 ns + 0.073 ns = 0.267 ns

6-5.



6-6.

a) $t_{PHL\text{-}C,D \text{ to } F} = 2\ t_{PLH} + 2t_{PHL} = 2(0.36) + 2(0.20) = 1.12$ ns

$t_{PLH\text{-}C,D \text{ to } F} = 2t_{PHL} + 2t_{PLH} = 2(0.20) + 2(0.36) = 1.12$ ns

$t_{pd} = 1.12$ ns

$t_{PHL\text{-}\overline{B} \text{ to } F} = 2t_{PHL} + t_{PLH} = 2(0.20) + (0.36) = 0.76$ ns

$t_{PLH\text{-}\overline{B} \text{ to } F} = 2t_{PHL} + t_{PLH} = 2(0.36) + (0.20) = 0.92$ ns

$t_{pd\text{-}\overline{B} \text{ to } F} = 0.76 + 0.92 = 0.84$ ns

$t_{PHL\text{-}A,B,\overline{C} \text{ to } F} = t_{PLH} + t_{PHL} = 0.36 + 0.20 = 0.56$ ns

$t_{PLH\text{ - }A,B,\overline{C} \text{ to } F} = t_{PHL} + t_{PLH} = 0.20 + 0.36 = 0.56$ ns

$t_{pd\text{ - }A,B,\overline{C} \text{ to } F} = 0.56$ ns

b) $t_{pd\text{-}C,D \text{ to } F} = 4\ t_{pd} = 4(0.28) = 1.12$ ns

$t_{pd\text{-}\overline{B} \text{ to } F} = 3\ t_{pd} = 3(0.28) = 0.78$ ns

$t_{pd\text{-}A,B,C}$ to F = 2 $t_{pd} = 2(0.28) = 0.56$ ns

c) For paths through an odd number of inverting gates with unequal gate t_{PHL} and t_{PLH}, path t_{PHL}, t_{PLH}, and t_{pd} are different. For paths through an even number of inverting gates, path t_{PHL}, t_{PLH}, and t_{pd} are equal.

6-7.

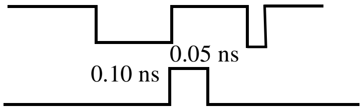
If the rejection time for inertial delays is greater than the propagation delay, then an output change can occur before it can be predicted whether or not it is to occur due to the rejection time.

For example, with a delay of 2 ns and a rejection time of 3 ns, for a 2.5 ns pulse, the initial edge will have already appeared at the output before the 3 ns has elapsed at which whether to reject or not is to be determined.

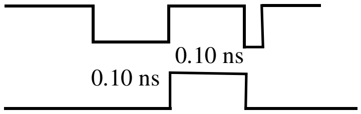
6-8.+

a) The propagation delay is $t_{pd} = \max(t_{PHL} = 0.05, t_{PLH} = 0.10) = 0.10$ ns.

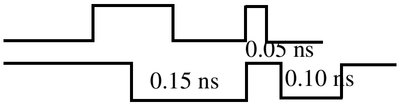
Assuming that the gate is an inverter, for a positive output pulse, the following actually occurs:



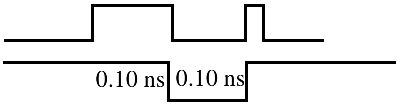
If the input pulse is narrower than 0.05 ns, no output pulse occurs so the rejection time is 0.05 ns. The resulting model predicts the following results, which differ from the actual delay behavior, but models the rejection behavior: :



b) For a negative output pulse, the following actually occurs:



The model predicts the following results, which differs from the actual delay behavior and the actual rejection behavior:



Overall, the model is inaccurate for both cases a and b, and provides a faulty rejection model for case b. Using an average of t_{PHL} and t_{PLH} for t_{pd} would improve the delay accuracy of the model for circuit applications, but the rejection model still fails.

6-9.

a) There is a setup time violation at 28 ns. There is an input combination violation around 24 ns.

b) There is a setup time violation just before 24 ns, There is an input combination violation around 24 ns.

c) There is a setup time violation at 28ns.

d) There is a hold time violation at 16ns and a setup time violation at 24ns.

6-10.*

a) The longest direct path delay is from input X through the two XOR gates to the output Y.

$$t_{delay} = t_{pdXOR} + t_{pdXOR} = 0.20 + 0.20 = 0.40 \text{ ns}$$

b) The longest path from an external input to a positive clock edge is from input X through the XOR gate and the inverter to the B Flip-flop.

$$t_{delay} = t_{pdXOR} + t_{pd\ INV} + t_{sFF} = 0.20 + 0.05 + 0.1 = 0.35 \text{ ns}$$

c) The longest path delay from the positive clock edge is from Flip-flop A through the two XOR gates to the output Y.

$t_{delay} = t_{pdFF} + 2\ t_{pdXOR} = 0.40 + 2(0.20) = 0.80$ ns

d) The longest path delay from positive clock edge to positive clock edge is from clock on Flip-flop A through the XOR gate and inverter to clock on Flip-flop B.

$$t_{delay\text{-}clock\ edge\ to\ clock\ edge} = t_{pdFF} + t_{pdXOR} + t_{pdINV} + t_{sFF} = 0.40 + 0.20 + 0.05 + 0.10 = 0.75 \text{ ns}$$

e) The maximum frequency is $1/t_{delay\text{-}\ clock\ edge\ to\ clock\ edge}$. For this circuit, $t_{delay\text{-}clock\ edge\ to\ clock\ edge}$ is 0.75 ns, so the maximum frequency is 1/0.75 ns = 1.33 GHz.

Comment: The clock frequency may need to be lower due to other delay paths that pass outside of the circuit into its environment. Calculation of this frequency cannot be performed in this case since data for paths through the environment is not provided.

6-11.

a) The longest direct path delay is from input X through the four XOR gates to the output Y.
$t_{delay} = 4\ t_{pdXOR} = 4(0.20) = 0.80$ ns

b) The longest path from an external input to a positive clock edge is from input X through three XOR gates and the inverter to the clock of the second B Flip-flop.

$$t_{delay} = 3\ t_{pdXOR} + t_{pd\ INV} + t_{sFF} = 3(0.20) + 0.5 + 0.1 = 0.75 \text{ ns}$$

c) The longest path delay from the positive clock edge is from the first Flip-flop A through the four XOR gates to the output Y.

$$t_{delay} = t_{pdFF} + 4\ t_{pdXORR} = 0.40 + 4(0.20) = 1.2 \text{ ns}$$

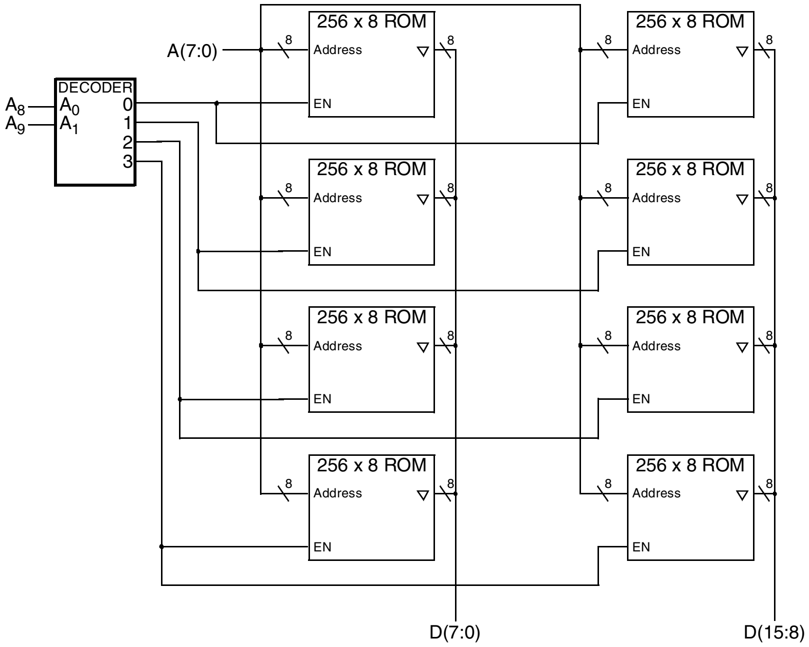
d) The longest path delay from positive clock edge to positive clock edge is from the first Flip-flop A through three XOR gates and one inverter to the clock of the second Flip-flop B.

$$t_{delay\text{-}clock\ edge\ to\ clock\ edge} = t_{pdFF} + 3\ t_{pdXOR} + t_{pdINV} + t_{sFF} = 0.40+ 3(0.20) + 0.5 + 0.1 = 1.15 \text{ ns}$$

e) The maximum frequency is $1/t_{delay\text{-}clock\ edge\ to\ clock\ edge}$. For this circuit, the delay is 1.15 ns so the maximum frequency is 1/1.15 ns = 870 MHz.

Comment: The clock frequency may need to be lower due to other delay paths that pass outside of the circuit into its environment. Calculation of this frequency cannot be performed in this case since data for paths through the environment is not provided.

6-12.



6-13.* (Errata: Change "32 X 8" to "64 X 8" ROM)

IN	OUT	IN	OUT	IN	OUT	IN	OUT
000000	0000 0000	010000	0001 0110	100000	0011 0010	110000	0100 1000
000001	0000 0001	010001	0001 0111	100001	0011 0011	110001	0100 1001
000010	0000 0010	010010	0001 1000	100010	0011 0100	110010	0101 0000
000011	0000 0011	010011	0001 1001	100011	0011 0101	110011	0101 0001
000100	0000 0100	010100	0010 0000	100100	0011 0110	110100	0101 0010
000101	0000 0101	010101	0010 0001	100101	0011 0111	110101	0101 0011
000110	0000 0110	010110	0010 0010	100110	0011 1000	110110	0101 0100
000111	0000 0111	010111	0010 0011	100111	0011 1001	110111	0101 0101
001000	0000 1000	011000	0010 0100	101000	0100 0000	111000	0101 0110
001001	0000 1001	011001	0010 0101	101001	0100 0001	111001	0101 0111
001010	0001 0000	011010	0010 0110	101010	0100 0010	111010	0101 1000
001011	0001 0001	011011	0010 0111	101011	0100 0011	111011	0101 1001
001100	0001 0010	011100	0010 1000	101100	0100 0100	111100	0110 0000
001101	0001 0011	011101	0010 1001	101101	0100 0101	111101	0110 0001
001110	0001 0100	011110	0011 0000	101110	0100 0110	111110	0110 0010
001111	0001 0101	011111	0011 0001	101111	0100 0111	111111	0110 0011

6-14.

a) 16 + 16 + 1 = 33 address bits and 16 + 1 = 17 output bits, 8G × 17

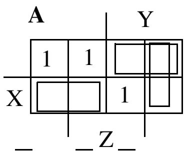
b) 8 + 8 + 1 + 1 = 18 address bits and 8 + 1 = 9 output bits

c) 4 × 4 = 16 address bits and 14 output bits are needed, 64K × 14

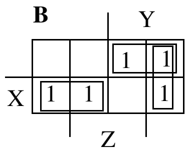
6-15.

Input			Output			
X	Y	Z	A	B	C	D
0	0	0	1	0	0	0
0	0	1	1	0	0	1
0	1	0	0	1	1	1
0	1	1	0	1	0	1
1	0	0	0	1	0	0
1	0	1	0	1	0	1
1	1	0	0	1	1	1
1	1	1	1	0	0	1

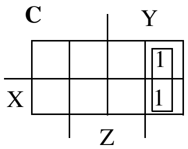
6-16.



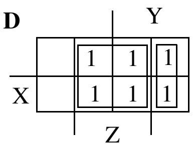
$\overline{A} = X\overline{Y} + \overline{X}Y + Y\overline{Z}$



$B = X\overline{Y} + \overline{X}Y + Y\overline{Z}$



$C = Y\overline{Z}$



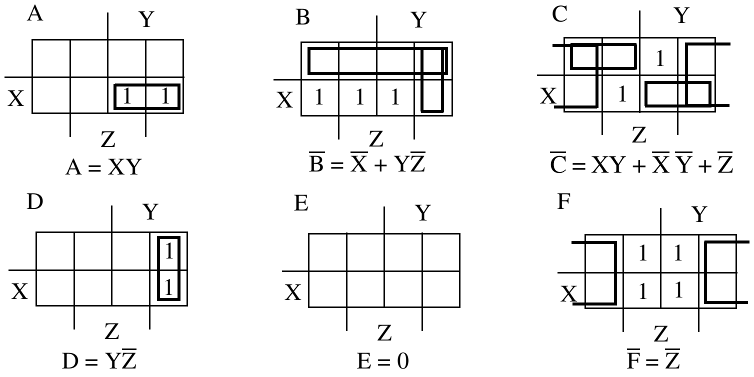
$C = Y\overline{Z} + Z$

By using $\overline{A}$ instead of A and $Y\overline{Z}$ instead of Y in D, $Y\overline{Z}$ can be shared by all four functions. Further, since A is the complement of B, terms $X\overline{Y}$ and $\overline{X}Y$ can be shared between $\overline{A}$ and B. Thus, only four product terms $Y\overline{Z}$, $X\overline{Y}$, $\overline{X}Y$, and Z are required.An inversion must be programmed for A.

6-17.

Find the truth table and K-maps:

X	Y	Z	A	B	C	D	E	F
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1
0	1	0	0	0	0	1	0	0
0	1	1	0	0	1	0	0	1
1	0	0	0	1	0	0	0	0
1	0	1	0	1	1	0	0	1
1	1	0	1	0	0	1	0	0
1	1	1	1	1	0	0	0	1



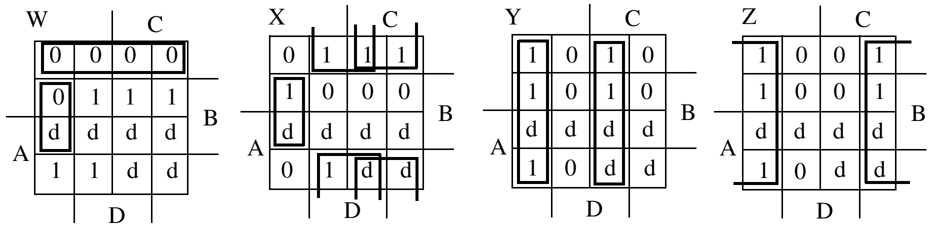
$A = XY$ $\quad \overline{B} = \overline{X} + Y\overline{Z}$ $\quad \overline{C} = XY + \overline{X}\,\overline{Y} + \overline{Z}$

$D = Y\overline{Z}$ $\quad E = 0$ $\quad \overline{F} = \overline{Z}$

Implementation of A, D, and E requires only two terms, XY and $Y\overline{Z}$. Straightforward implementation of B, C, and F requires four terms, XY, XYZ, XYZ, and Z. By implementing $\overline{B}$, $\overline{C}$, and $\overline{F}$, only three additional terms $\overline{X}$, $\overline{X}\,\overline{Y}$, and $\overline{Z}$ are required. So we form the solution using five product terms: XY, $Y\overline{Z}$, $\overline{X}\,\overline{Y}$, $\overline{X}$, and $\overline{Z}$. The solution is described by the equations given with the six K-maps.

6-18.

The values given in the four K-maps come from Table 3-1 on page 99.

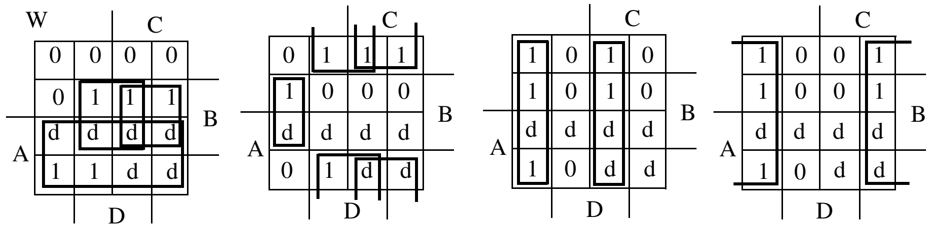


$\overline{W} = \overline{A}\,\overline{B} + B\overline{C}\,\overline{D}$ $\quad X = B\overline{C}\,\overline{D} + \overline{B}C + \overline{B}D$ $\quad Y = CD + \overline{C}\,\overline{D}$ $\quad Z = \overline{D}$

In this case, shared terms are limited. One such term $B\,\overline{C}\,\overline{D}$ is generated in $\overline{W}$.

6-19.*

Assume 3-input OR gates.

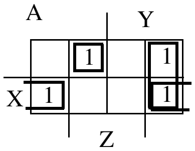


$W = A + BC + BD$ $\quad X = B\overline{C}\,\overline{D} + \overline{B}C + \overline{B}D$ $\quad Y = CD + \overline{C}\,\overline{D}$ $\quad Z = \overline{D}$

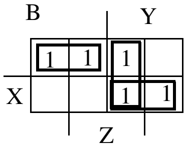
Each of the equations above is implemented using one 3-input OR gate. Four gates are used.

6-20.

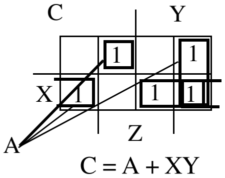
Figure 6-23 uses 3-input OR gates.



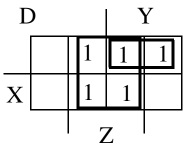
$A = X\overline{Z} + Y\overline{Z} + \overline{X}\,\overline{Y}Z$



$B = XY + YZ + \overline{X}\,\overline{Y}$



$C = A + XY$

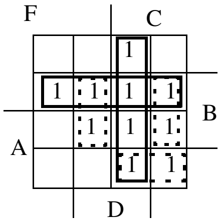


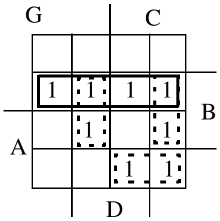
$D = \overline{X}Y + Z$

A, B, and D each require three or fewer product terms so can be implemented with 3-input OR gates.
C requires four terms so cannot be implemented with a 3-input OR gate. But because the first PAL device output can used as an input to implement other functions it can be assigned to A and A can then be used to implement C using just two inputs of a 3-input OR gate.

6-21.

Figure 6-23 uses 3-input OR gates.





Straightforward implementation of F requires five prime implicants and of G requires four prime implicants, but only 3 inputs are available on the PAL OR gates. So sum-of-products that can be factored from F and G or both and implemented by the other PAL cells are needed. A single sum of products that will work is $H = A\overline{B}C + B\overline{C}D + BC\overline{D}$. The terms of H are shown with dotted lines on the K-maps. Using H:
$F = H + CD + \overline{A}B$
$G = H + \overline{A}B$
There are other possible functions for H and corresponding results for F and H.