1.

<table>
<thead>
<tr>
<th>Year</th>
<th>Chip</th>
<th>Transistor Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>2022</td>
<td>M1 Ultra</td>
<td>114,000,000,000</td>
</tr>
<tr>
<td>2021</td>
<td>M1 Max</td>
<td>57,000,000,000</td>
</tr>
<tr>
<td>2020</td>
<td>M1</td>
<td>16,000,000,000</td>
</tr>
</tbody>
</table>

The Apple M1 Ultra, M1 Max, and M1 appear to fit the curve from Lecture 1 slide 19 according to Moore's Law.

2.
3. Prove using Boolean Algebra
   a. \(a + 1 = 1\)
      \[
      \begin{align*}
      a + 1 &= (a + 1)(1) \quad \text{Identity Law} \\
      &= (a + 1)(a + a') \quad \text{Complement Law} \\
      &= a + (1*a') \quad \text{Distributive Law} \\
      &= a + a' \quad \text{Identity Law} \\
      &= 1 \quad \text{Complement Law}
      \end{align*}
      \]
   b. \(a0 = 0\)
      \[
      \begin{align*}
      a0 &= a0 + 0 \quad \text{Identity Law} \\
      &= a0 + aa' \quad \text{Complement Law} \\
      &= a(0 + a') \quad \text{Distributive Law} \\
      &= aa' \quad \text{Identity Law} \\
      &= 0 \quad \text{Complement Law}
      \end{align*}
      \]
   c. \((be + d)(a + c + d')(a + be + c) = (be + d)(a + c + d')\)
      \[
      \begin{align*}
      (be + d)(a + c + d')(a + be + c) &= (be + d)(a + c + d')(a + be + c) \\
      &= (be + d)(a + c + d')(a + be + c + 0) \quad \text{Identity Law} \\
      &= (be + d)(a + c + d')(a + be + c + dd') \quad \text{Complement Law} \\
      &= (be + d)(a + c + d')(a + be + c + d)(a + be + c + d') \quad \text{Distributive Law} \\
      &= (be + d)(a + c + d')(a + be + c + d) \quad \text{Idempotent Law} \\
      &= (be + d)(a + c + d') \quad \text{Idempotent Law}
      \end{align*}
      \]
Idempotent law:
\[ a + a = a \]
\[ a + a = (a + a)(1) \]  
Identity Law
\[ = (a + a)(a + a') \]  
Complement Law
\[ = a + (aa') \]  
Distributive Law
\[ = a + 0 \]  
Complement Law
\[ = a \]  
Identity Law

2. Prove using Shannon’s expansion

a. \( f(a) = a + 1 \)
   \[ = a * f(1) + a' * f(0) \]
   \[ = a(1 + 1) + a'(0 + 1) \]
   \[ = a(1) + a'(1) \]
   \[ = a + a' \]
   \[ = 1 \]

b. \( f(a) = a0 \)
   \[ = (a + f(1))(a' + f(0)) \]
   \[ = (a + (1*0))(a' + (0*0)) \]
   \[ = (a + 0)(a' + 0) \]
   \[ = aa' \]
   \[ = 0 \]

c. Let \( x = be \)
   Let \( y = a + c \)
   \[ f(d, x, y) = (x + d)(y + d')(x + y) \]
   \[ = (d + f(0, x, y))(d' + f(1, x, y)) \]
   \[ = (d + (x + 0)(y + 1)(x + y))(d' + (x + 1)(y + 0)(x + y)) \]
   \[ = (d + (x)(1)(x + y))(d' + (1)(y)(x + y)) \]
   \[ = (d + x(x + y))(d' + y(x + y)) \]
   \[ = (d + xx + xy)(d' + yx + yy) \]
   \[ = (d + x + xy)(d' + yx + y) \]
   \[ = (d + x)(d' + y) \]
   \[ = (d + be)(d' + a + c) \]
   \[ = (be + d)(a + c + d') \]

4. Prove or Disprove Statements

a. Given \( a = bc \) and \( a = b + c \):

   \[ bc = b + c \]  
   Substitution for \( a \)
   \[ b'+bc=b'+b+c \rightarrow b'+c=1 \]  
   OR both sides with \( b' \)
   \[ b'bc=b'(b+c) \rightarrow b'c=0 \]  
   AND both sides with \( b' \)
   \[ c \text{ is } b'”, \text{ thus } c=b \]
   Therefore the statement is true

b. We can prove by counterexample that the statement is false:
   Let \( b = 1 \) and \( c = 0 \).
   \[ a = 1*0 = 0 \]
a' = 0 + 1 = 1 → a = 0
However, b is not equal to c. Therefore a=bc and a'=b'+c' does not imply b=c and the statement is false.
c. (a + bc + d)(b' + c' + ef + g)(ef + g)
= (a + bc + d)(ef + g)

We can then prove by counterexample that the statement is false:
Let a = 0, b = 1, c = 1, d = 0, e = 1, f = 1, g = 1
LHS: (a + bc + d)(b' + c' + ef + g)(ef + g) = (a + bc + d)(ef + g) = (0 + 1 + 0)(1 + 1) = 1
RHS: (a + d)(ef + g) = (0 + 0)(1 + 1) = 0
In this case, the left hand side evaluates to 0 while the right hand side evaluates to 1. Therefore the two expressions are not equivalent and the statement is false.

5. Translate from digital problem to boolean expressions
a. Two pipes are open and two pipes are closed at any given time
OR together all cases with 2 open and 2 closed
\[ a_0 \overline{a}_1 a_2 a_3 + a_0 a_1 a_2 a'_3 + a_0 a_1 a'_2 a'_3 + a_0 a_1 a'_2 a_3 + a_0 a'_1 a'_2 a_3 + a_0 a'_1 a_2 a_3 \]

<table>
<thead>
<tr>
<th>( a_3 )</th>
<th>( a_2 )</th>
<th>( a_1 )</th>
<th>( a_0 )</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
b. Number of pipes open cannot be an odd number
Similar to part a, but include cases where number of pipes open is 0 or 4.
\[
a_0a_1a_2a_3 + a_0'a_1a_2a_3 + a_0'a_1a_2a_3 + a_0'a_1a_2a_3 + a_0'a_1a_2a_3 + a_0'a_1a_2a_3 + a_0'a_1a_2a_3 + a_0'a_1a_2a_3
\]

<table>
<thead>
<tr>
<th>$a_3$</th>
<th>$a_2$</th>
<th>$a_1$</th>
<th>$a_0$</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

6. Boolean algebra and implementation
   i. \( y(a, b, c, d, e) = a(a' + b)(a' + b' + c)(a' + b' + c' + d)(a' + b' + c' + d' + e) \)
      = \( a(a' + b)(a' + b' + c)(a' + b' + c' + d) \)
      = \( a(a' + b)(a' + b' + c) \)
      = \( a(a' + b)(a' + b' + cde) \)
\[
\begin{align*}
  &= a(a' + bcde) \\
  &= abcde
\end{align*}
\]

b. 

\[A\] \[B\] \[C\] \[D\] \[E\]

- 5 literals, 1 operation, 1 gate, 6 nets, 6 pins

ii. \(y(a, b, c, d, e, f, g, h) = a'b + ab'(c'd + cd' (ef' + e'(gh' + g'h)))\)

  a. Cannot be simplified further
  
  b. 

\[F\] \[E'\] \[F'\] \[G\] \[G'\] \[H\] \[H'\] 

- 16 literals, 15 operators, 15 gates, 23 nets, 45 pins