1. Recall that the sum-of-products (SOP) Boolean formula for the carry-out \( c_{i+1} \) of a 1-bit adder was: 
\[ c_{i+1} = x_i y_i + x_i c_i + y_i c_i. \]

a) Complete the following table showing the gate delays for different types of adders assuming a 9-input limit into any gate.

<table>
<thead>
<tr>
<th>Type of Adder</th>
<th># of product terms in SOP expression to be OR'ed</th>
<th>Most # of inputs in any of the product terms</th>
<th># gate delays due to product/AND terms</th>
<th># gate delays due to sum/OR</th>
<th>Gate Delay per Adder</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-bit</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2-bit</td>
<td>7</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3-bit</td>
<td>15</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-bit</td>
<td>31</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-bit</td>
<td>63</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-bit</td>
<td>127</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-bit</td>
<td>255</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-bit</td>
<td>511</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9-bit</td>
<td>1,023</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

b) Consider the SOP Boolean formula for the carry-out \( c_{i+9} \) of a 9-bit adder:

Give an example of each of the following:

i) a product term with only two terms

ii) a product term with 10 terms
2) Complete the below diagram of a 4-bit register so that it is able to perform the following operations:
   - parallel read/output of all bits (just look at the Q values)
   - Control the MUXs using the following codes:
     00₂ - parallel write/input of all bits
     01₂ - circular shift left one bit position (value shifted out of most-significant bit is shifted into the least-significant bit)
     10₂ - arithmetic shift right (sign-extend the most-significant bit)
     11₂ - logical shift right one bit position (value shifted out of least-significant bit is lost and a “0” is shifted into the most-significant bit)

Note: For each D-flip flop, the output of a MUX is used as the D-input.

---

```
D   D   D   D
Q   Q   Q   Q
```

---

```
0  1  2  3
MUX
0  1  2  3
MUX
0  1  2  3
MUX
0  1  2  3
MUX
```

---

Data to Write in Parallel

Data to Read in Parallel