Convert the Verilog code into circuit using basic gates and muxes

```
logic [1:0] a;
logic [1:0] b, c, d;

always_comb begin
    case (a)
        2'b00:  b = c;
        2'b01:  b = d;
        2'b10:  b = 2'b11;
        2'b11:  b = 2'b01;
    endcase
end
```
Give answer to following Two’s Complement numbers:

\[ 0100 + 0001 = \overbrace{0101}^\text{overflow} \]

\[ 0111 + 1010 = \overbrace{0100} \]

\[ 1011 + 1001 = \overbrace{0100} \]

\[ 1111 - 0110 = 1111 + (-0110) = \overbrace{1001} \]

\[ 0001 - 1111 = \]

\[ 1110 - 1010 = \]

\[ 0100 - 0010 = \]
Build a full adder using only two 4:1 muxes and as few inverters as possible.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Count</th>
<th>Sum</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>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
a) Produce the minimum Sum of Products of the following K-map

\[ A \overline{D} + CD + \overline{A \overline{B}} \]

A

\[ \begin{array}{cccc}
& 1 & 0 & 1 & X \\
C & \cdot & 1 & 0 & 0 \\
D & 1 & \cdot & X & X \\
0 & 1 & 1 & 1 & 1
\end{array} \]

b) Draw the Circuit Diagram for the minimum Sum of Product equation above using only NANDs, NORs and Inverters.

[Diagram of the circuit with NAND and NOR gates]
Design a mediocre 4-bit calculator using circuit and block diagrams. It has 3 buttons: "+1", "x2", and "clear". Whenever "+1" is pressed, the display (output) should go up by 1. Whenever "x2" is pressed, the output should double. If "clear" is pressed, the output goes to "00000".

Ignore overflow.

If multiple pressed, do nothing.