Develop a single-cycle CPU that can do B and CBZ (only).
Make it as simple as possible.

(OR)
You can use any standard logic units.
Datapath, page 2: show the control

Write the Truth Table for SUB Function with inputs: A, B & Cin for a full adder.
Use Assembly language to write a program that could recognize if the value of X1 is the multiple of 3; if so, store it in certain register. You can use other registers as temporary storage.

\[
XO \equiv \left(\frac{\text{val}}{3}\right) \times 3 = \text{val1}
\]

```
addi x0, x31, #3  // set X0 to 0
addi x2, x31, #3  // X2 = 0
sdw x3, x1, x2   // divide
mul x3, x3, x2  // mult
cmp x3, x1  // check equal
beq x3, x1, DONE

addi x0, x0, #1  // set to 1
```

DONE:
Turn the following machine code to assembly language and fill in the control setting for the instruction.

Machine code:

```
31  1 1 1 1 1 0 0 0 0 1 0 0 0 0 0 0 1 1 1 1 0 0 0 1 0 1 1 1 0 1 0 1 1 1
```

Assembly language:

```
LDR R X3, [X15, #15]
```

Control setting:

```
<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reg2Loc</td>
<td>X</td>
</tr>
<tr>
<td>ALUSrc</td>
<td>1</td>
</tr>
<tr>
<td>MemToReg</td>
<td>YES 1</td>
</tr>
<tr>
<td>RegWrite</td>
<td>1</td>
</tr>
<tr>
<td>MemWrite</td>
<td>0</td>
</tr>
<tr>
<td>BrTaken</td>
<td>0</td>
</tr>
<tr>
<td>UncondBr</td>
<td>X</td>
</tr>
<tr>
<td>ALUOp</td>
<td>Add</td>
</tr>
</tbody>
</table>
```
Given the following signed 5-bit values (2's comp), compute the result.
Your answer should be a 9-bit signed value.

A: 01011
B: 11100

\[
\begin{array}{c}
\text{A} \times \text{B} \\
\hline
11100 \\
\times 01011 \\
\hline
1111100 \\
+ 1111000 \\
\hline
100110100
\end{array}
\]