Draw a single-cycle CPU that can do the instruction
\[
\text{MULT}_4 \ Rd, \ Rn = [Rd] = [Rn] \times 4.
\]
Take the single cycle processor developed in class, and add the following instruction. Show the controls for the new design.

**MULIF Rd, Rn, Rm:**

Multiply Rn with Rm and keep the result in Rd only if multiplying them is going to give valid/good data (result doesn’t overflow meaning result can be expressed using not more than 64 bits) otherwise skip to next instruction while writing all zeros to Rd.
Midterm Review Question

a) Compute the CPI for the single-cycle processor developed in class

b) Compute the CPI of machine 1

<table>
<thead>
<tr>
<th>Type</th>
<th>Cycles</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>load</td>
<td>2</td>
<td>20%</td>
</tr>
<tr>
<td>store</td>
<td>1</td>
<td>70%</td>
</tr>
<tr>
<td>branch</td>
<td>3</td>
<td>10%</td>
</tr>
</tbody>
</table>

c) Compute the CPI of machine 2

<table>
<thead>
<tr>
<th>Type</th>
<th>Cycles</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>load</td>
<td>2</td>
<td>10%</td>
</tr>
<tr>
<td>store</td>
<td>2</td>
<td>80%</td>
</tr>
<tr>
<td>branch</td>
<td>2</td>
<td>10%</td>
</tr>
</tbody>
</table>

d) If machine 1 & 2 process the same amount of instructions, what does the clock rate for machine 2 need to be in order to be faster than machine 1? Given: machine 1 clock rate = 1GHz
Imagine that we modify our processor from class.
Instead of five stages, let's use three.
Let's unify the Reg/Dec and Exec stages.
Also, put together the Mem and WB stages.

What kinds of structural hazards?
Do we still need delay slots to handle branch calculations or load structural hazards?
What kind of forwarding logic is necessary? Why?
, if any
The following code is being executed on the pipelined processor:

```
ADDI   X2,    X31,    #6
ADD    X2,    X2,    X2
LDUR   X2,    [X2,    #4 ]
ADD    X2,    X2,    X2
ADDI   X2,    X2,    #2
```

(1) During which cycle(s), the forwarding unit has overridden the output of the register file?

(2) During which cycle(s), the register file has forwarded its input to its output?

(3) Assuming Mem[16] = 20, for each cycle, write down the value of Dw (aka. WriteData) of the register file. If it cannot be determined, write down “?”.

<table>
<thead>
<tr>
<th>Cycle #</th>
<th>Dw</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>
Explain what is happening in each stage of the pipelined processor during cycle 4 and 5 in the code below. If there is forwarding happening in the code, explain it.

LDUR X2, [X1, #3]
ADD X1, X2, #5
ADD X4, X2, X1
STUR X1, [X1, #3]
CBNZ X1, END
Sequence of instructions:
LDUR   x2,  [x1,  #0]
LDUR   x1,  [x6,  #40]
SUB    x6,  x1,  x2
ADD    x6,  x2,  x2
OR     x3,  x6,  x31
STUR   x6,  [x1,  #50]

1. List the Read-After-Write data dependencies.
2. Assume the 5-stage pipeline with no forwarding, and each stage takes 1 cycle, you let the processor stall on hazards. How many times does the processor stall? How long is each stall (in cycles)? What is the execution time (in cycles) for the whole program?
Consider a pipelined CPU running the code below:

LDUR X1, [X7, #0]
ADD X9, X8, X1
STUR X9, [X6, #0]
LDUR X5, [X10, #0]
ADD X10, X5, X1

With the forwarding unit from class implemented to our pipelined CPU, what changes does the code need to run correctly?
Write the machine code for a program that swaps the values at Mem[32] and Mem[40]
Test question

Why does a single cycle cpu not need a Forwarding Unit?
Convert the following Java code into assembly language

```java
public int returnMax (int[] array) {
    int max = 0;
    for (int i = 0; i < array.length; i++) {
        if (max < array[i]) {
            max = array[i];
        }
    }
    return max;
}
```