Problem 1

```verilog
code
rule stage1;
    fifo.enq(f0(inQ.first));
inQ.deq;
endrule
rule stage2;
    outQ.enq(f1(fifo.first));
fifo.deq;
endrule
```

a) Fill in the conflict matrix for this design. You may assume that the fifo we’ve provided allows for concurrent calls to enq and deq.

<table>
<thead>
<tr>
<th></th>
<th>r1</th>
<th>r2</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>C</td>
<td>CF</td>
</tr>
<tr>
<td>r2</td>
<td>CF</td>
<td>C</td>
</tr>
</tbody>
</table>
b) Draw a hardware circuit for this design, ignoring the internals of the fifo design. Include the internals of the bluespec scheduler generated for this design.

![Diagram](image1)

(c) Repeat the previous two parts, but this time you may assume that the fifo we’ve provided doesn’t allow for concurrent calls to enq and deq.

<table>
<thead>
<tr>
<th></th>
<th>r1</th>
<th>r2</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>r2</td>
<td>C</td>
<td>C</td>
</tr>
</tbody>
</table>

![Diagram](image2)
Problem 2

```
rule ra(p1);
  x <= y + 1;
endrule
rule rb(p2);
  y <= z + 2;
endrule
rule rc(p3);
  z <= x + 2;
endrule
```

a) Fill in the Conflict Matrix for this design.

<table>
<thead>
<tr>
<th></th>
<th>ra</th>
<th>rb</th>
<th>rc</th>
</tr>
</thead>
<tbody>
<tr>
<td>ra</td>
<td>C</td>
<td>&lt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>rb</td>
<td>&gt;</td>
<td>C</td>
<td>&lt;</td>
</tr>
<tr>
<td>rc</td>
<td>&lt;</td>
<td>&gt;</td>
<td>C</td>
</tr>
</tbody>
</table>

b) Can all three rules execute concurrently?
   No, ra < rb, rb < rc but rc < ra

c) Can any two rules execute concurrently?
   Yes