1) Consider the following C code:

```c
for ( i = 0; i < MAX; i++ )
{
    a[ i ] = a[ i ] + b[ i ];
}
```

That C code is translated into the following x86-like assembly language: (note: the ++ indicates the autoincrement addressing mode. See page 98 if needed.)

```
mov r1, addr( a )  -- address of a[ 0 ] into r1
mov r2, addr( b )  -- address of b[ 0 ] into r2
mov rx, MAX        -- Number of iterations into rx
l1:  ld r3, (r1)     -- load indirect into r3 through r1
     ld r4, (r2)++  -- what r2 points to loaded in r4
     fadd r5, r3, r4  -- r5 holds sum of two elements
     st r5, (r1)++  -- store result and post-increment
     loop l1        -- does an autodecrement (by 1) of rx
                  -- if rx isnt zero branches to l1
```

And then that assembly code is software pipelined.

```
mov r0, addr( a )  -- Initialization:
mov r1, r0         -- r0 is pointer to a[0]
mov r2, addr( b )  -- r2 is pointer to b[0]
     blank A       
     blank B       
     blank C       
     fadd r5, r3, r4
     ld r3, (r1)++
     ld r4, (r2)++
12:  st r5, (r0)++
     fadd r5, r3, r4
     ld r3, (r1)++
     ld r4, (r2)++
     loop 12       -- decrement rx, if != 0 jump to 12
     blank D       
     fadd r5, r3, r4
     st r5, (r0)
```

a) Supply the missing code for each blank [2 points each]

b) If, in the original C code, MAX is less than ________ the software-pipelined loop will behave incorrectly. [3]
2) Say you are the head architect at "Computers R Us" and your main product is a processor that executes 2000MIPS and runs at 80 Watts. You have some teams of interns that have proposed some improvements to the architecture of this processor in order to save power. Which of these might be worth considering and why? [3 points each]

a) A fairly trivial change to the cache system that drops performance to 1900 MIPS while it drops power to 70 Watts.

b) A fairly complex change to the out-of-order core that drops performance to 1800 MIPS while it drops power to 54 Watts?

c) A very trivial change to the predictor (making it smaller) and the L2 cache (reducing leakage current of some transistors) that drops performance to 1750 MIPS while it drops power to 48 Watts?

3) Consider a design that is clocked at 100MHz. If the clock period is reduced to 50MHz would you expect the performance to decrease by more or less than 50%? Why? (Think about the processor you are designing and what would happen if you dropped the clock speed by a factor of two.) [4 points]

3) Draw the state transition diagram for a three-state MSI (Modified-Shared-Invalid) coherence protocol for a snoopy bus-based symmetric multiprocessor. Draw only the stable states (i.e., your diagram should have three states, “Modified”, “Shared”, and “Invalid”). Include transition arrows for all events that can occur in a state. For each transition, label it with “Event => Reaction”. [6 points]

<table>
<thead>
<tr>
<th>Events</th>
<th>Possible Reactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Read</td>
<td>Invalidate line</td>
</tr>
<tr>
<td>CPU Write</td>
<td>Write Back line</td>
</tr>
<tr>
<td>Bus Read Request (read by another CPU)</td>
<td></td>
</tr>
<tr>
<td>Bus Write Request (write by another CPU)</td>
<td></td>
</tr>
</tbody>
</table>