1. A four-processor SMP system uses a shared bus to access main memory. The bus is 128 bits wide and operates at a clock rate of 100 MHz. Each processor has a private write-back cache. The internal clock rate of each processor is 200 MHz and cache access is 0.25 instruction fetches and 0.25 data fetches per instruction. This external cache uses 4 words per line and has an access time of 1 clock (10 ns). The main memory uses 100 MHz SDRAM with a RAS access time of 5 clocks. Assume that the processing rate is limited by the data access rate (that is, 100% utilization of the processor-to-cache bus), and that the processor can proceed immediately after the requested word is delivered from main memory after a miss.

What is the effective MIPS rate of each processor running as a uniprocessor with a cache hit rate of 100%?

- $0.25 + 0.25 = 0.5$ accesses per instruction. $10 \text{ ns/access} \times 0.5 \text{ accesses per instruction} = 5 \text{ ns per instruction} \rightarrow 200 \text{ MIPS}$.

2. For the SMP system in question 1, what is the effective MIPS rate of each processor running as a uniprocessor with a cache hit rate of 97%?

- $t_{\text{eff}} = 10 + (1 - 0.97)(50) = 11.5 \text{ ns}$. $11.5 \text{ ns per access} \times 0.5 \text{ accesses per instruction} = 5.75 \text{ ns per instruction} \rightarrow 174 \text{ MIPS}$.

3. Without considering 2nd order effects (that is, don’t consider lengthening the memory access time because of contention), what is the average memory bus utilization with the four processors running in the multiprocessor configuration with a hit rate of 97%?

- Each processor uses the bus for 50 ns access plus 30 ns cache fill time three times out of every 100 accesses (0.03 miss rate). Out of the 100 accesses, 97 are at the cache hit time of 10 ns and three are at the miss time of 10 + 50 ns main memory access time. Therefore, each processor is using the bus $(3)(50 + 30)/((100)(10) + (3)(50)) = 21\%$. With four processors, the bus utilization is $(4)(21\%) = 83\%$. 
4. We split the shared memory into eight banks and connect the processors to the memories with a non-blocking crossbar switch rather than a shared bus. What is the average utilization of each of the memories?

- Each processor still uses memory 21% of the time, but they are now spread out over 8 independent modules. So we have an average of \( \frac{4 \times 21\%}{8} = 10.5\% \) utilization for each module.

5. Bonus 10%! What would be the optimum way to set up the addressing to these independent memory modules? That is, which bits are used to address a byte within a word, which are used to address the memory module, etc.?

- The interesting thing here is that we want to block interleave the memory modules, not word interleave. We have a bus width of 128 bits = 16 bytes, so the low-order 4 bits are used to address the byte within a word. We want to read out 4 sequential words from the synchronous DRAM chips in a memory module in order to load a cache line in the minimum time of 5 + 3 clocks, so the next two bits are used to address those four sequential words within a single memory module. We want to interleave the memory modules on sequential block addresses to distribute the accesses evenly over the eight modules, so the next three bits are used for the module address. The two sequential word-address bits are concatenated to the remainder of the address bits to form the complete address within a module.
Assuming a 32-bit address, following is a diagram that shows this ordering in two different ways:

**Main Memory Address**

- 23 bits
- 3 bits
- 2 bits
- 4 bits

32 bit address

- 16 Bytes within a word
- 4 sequential words within a module
- One of 8 modules
- 8,388,608 blocks of 4 words within a module

23 bits

- 2 bits
- 3 bits

word address within module

32 bit address

- 2 bits
- 4 bits

module address

- 3 bits

byte address within word
This diagram shows how the bytes are actually numbered within the memory modules when we have 16 bytes per word, 4 words per block, and we block interleave the 8 modules:

<table>
<thead>
<tr>
<th>Word address within the module</th>
<th>Module 0</th>
<th>Module 1</th>
<th>Module 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000→</td>
<td>0 1 2 ... 15</td>
<td>64 65 66 ... 79</td>
<td>448 449 450 ... 463</td>
</tr>
<tr>
<td>0001→</td>
<td>16 17 18 ... 31</td>
<td>80 81 82 ... 95</td>
<td>464 465 466 ... 479</td>
</tr>
<tr>
<td>0002→</td>
<td>32 33 34 ... 47</td>
<td>96 97 98 ... 111</td>
<td>480 481 482 ... 495</td>
</tr>
<tr>
<td>0003→</td>
<td>48 49 50 ... 63</td>
<td>112 113 114 ... 127</td>
<td>496 497 498 ... 511</td>
</tr>
<tr>
<td>0004→</td>
<td>512 513 514 ... 527</td>
<td>576 577 578 ... 591</td>
<td>960 961 962 ... 975</td>
</tr>
<tr>
<td>0005→</td>
<td>528 529 530 ... 543</td>
<td></td>
<td>976 etc.</td>
</tr>
<tr>
<td>0006→</td>
<td>544 545 546 ... 559</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0007→</td>
<td>560 561 562 ... 575</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0008→</td>
<td>1024 etc.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>