Exercise Set 3:
Memory Access Rate, Variable-Size-Packet Segmentation

Assigned: Wed. 5 Mar. 2008 (week 3) -- Due: Fri. 14 Mar. 2008 (week 4)

3.1 Maccesses/s versus Gbits/s: Random Address Peak Rate in SRAM's

By increasing the width of a RAM, we can arbitrarily increase its bandwidth. However, when the RAM is operated as a single memory, in a simple and unconstrained mode, all blocks or chips that comprise this wide memory are accessed using the same address at a time, i.e. all blocks or chips are accessed with reference to the same packet or segment or cell at a time. Systems do exist where this is not so, but in those systems the total memory space appears partitioned into banks, and concurrent packet/segment accesses must be carefully scheduled so as to result in non-conflicting bank accesses; we are not considering such more complex systems in this exercise.

Thus, for simple, unconstrained memory operation, besides total memory throughput, the other interesting performance metric is the peak possible rate of **random** accesses to arbitrary, independent locations (i.e. not necessarily sequential or in the same row). This is, in other words, the peak **address** rate for random, independent accesses.

(a) What is this number, for some of the SRAM technologies seen in class, in millions of accesses per second (Maccesses/s)? Consider the following SRAM technologies. the SRAM blocks used in the two on-chip buffer examples in section 2.2.1: *(i)* in the one-port, 40 B wide buffer example, and *(ii)* in the two-port, 256 B wide example; the off-chip examples seen in section 2.2.2: *(iii)* the QDR SRAM chip example (CY7C1545V18), and *(iv)* the shared-bus SRAM chip example (CY7C1550V18 - this is a burst-of-2 chip, capable of receiving one address per clock cycle).

(b) Consider that we build a 64-Byte (512-bit) wide buffer memory out of each of the above technologies in (a). For technologies that provide only burst accesses, the memory "width" is the total size of the *entire* burst that is accessed at a time. This 64-Byte width is a customary segment size in modern networking equipment, because 64 bytes is a "round" number just above the ATM cell size or the minimum IP packet size. (In the case of 18-bit or 36-bit wide parts, the total memory width will be 64x9 = 576 bits, where the extra 64 bits per segment are usually used for parity or ECC, and/or other off-band overhead information, e.g. end-of-packet and other such mark bits).

How many blocks (on-chip) or parts (off-chip) are needed, in each case of (a), for this 64-Byte wide buffer memory to be made? What is their aggregate peak throughput in Gbits/s? What is their total power consumption at peak rate, and their consumption per Gbps of offered throughput?

3.2 DRAM Access Rate

Calculate the peak address rate for random, independent accesses for the old (2001) dynamic RAM (DDR SDRAM) chip (Micron MT46V2M32) seen in the section 2.2.3 slides, in a manner similar to exercise 3.1(a) above.

(a) Find the peak access rate for trully random accesses, i.e. accesses that may fall in the same bank but in a different row relative to the previous access. *Hint:* this is directly linked to the (same-bank) cycle time.

(b) Assuming the peak access rate in (a), what is the chip's peak data throughput (Gb/s)? Answer separately for all-read's and for all-write's. As in class, assume that you may set the burst size to "full page", i.e. "very long", and that the burst goes on continuously until interrupted by the next ACTIVATE command.

3.3 Variable-Size-Packet Bit Rate for given Segment Access Rate

Consider that the 64-Byte-wide buffer memory of exercise 3.1(b) above is used to store incoming (fixed-size) ATM cells, or (variable-size) IP packets that are being segmented into 64-Byte segments, as well as to later read such cells or segments on their way out. Memory utilization is precisely 50% writes and 50% reads. For the shared-bus SRAM technology, which has a bus turn-around penalty, we perform the optimization of arranging read/write accesses in the following fashion: precisely four (4) segments are written consecutively (at 4 arbitrary addresses), then precisely four (4) segments are read consecutively (from 4 arbitrary addresses), then 4 other segments are written, etc. Assume that the above turn-around penalty is one clock cycle lost on every read-to-write change, but no cycle lost on write-to-read changes.

(a) For each SRAM technology in exercise 3.1(a), what is the peak **incoming segment rate** that can be supported, in Msegments/s? *Hint:* Each incoming segment is written into a "random" memory location (address). Thus, for each incoming segment we need to perform an (independent) write memory access. Hence, the peak incoming segment rate that can be supported is one half (50% writes - 50% reads) of the peak (independent) access rate calculated in exercise 3.1(a), except for the shared-bus SRAM that has a bus turn-around penalty where you need to derate their peak Maccesses/s by the turn-around overhead for our specific 4-write-4-read access pattern.

(b) Assume that the incoming traffic is ATM over SONET. For reasons of simplicity of memory management, each ATM cell is written into a different memory segment --hence, approximately 64-53 = 11 bytes in each segment remain unused (the exact number depends on details such as whether the header CRC is stored or just recomputed on the way out, whether any flow ID is stored together with the cell to assist in VP/VC translation in the outgoing path, etc). Thus, the peak incoming cell rate that can be supported is equal to the peak incoming segment rate that you calculated in question (a).

Translate this cell rate into an equivalent "SONET bit rate", for each SRAM technology considered in (a). Of course, SONET bit rates are strictly quantized, as listed in exercise 2.1, but, for the purposes of this exercise, assume that you can linearly scale the SONET bit rate to any number that is needed to provide the desired ATM cell rate; Assume that the percentage of SONET bit rate that is dedicated to SONET overhead (clock recovery, framing, etc) is as in exercise 2.2, i.e. 3.33 percent (3 bytes of overhead in every 90 SONET bytes). Compare the "SONET bit rate" that you find here to the buffer memory aggregate peak throughput in Gbits/s that you found in exercise 3.1(b), for each same technology. How and why do they differ?

(c) Assume, now, that the incoming traffic consists of 40-Byte (minimum sized) IP packets, which are carried in an "IP-over-SONET" technology (**not** IP-over-ATM-over-SONET). These minimum sized IP packets fit within one buffer memory segment (64 bytes), each. For reasons of simplicity of memory management, again, each such IP packet is written into a different memory segment --hence, approximately 64-40 = 24 bytes in each segment remain unused. Thus, the peak incoming packet rate that can be supported is equal to the peak incoming segment rate of question (a), or to the peak incoming cell rate of question (b).

Translate this packet rate into an equivalent "SONET bit rate", for each SRAM technology considered in (a). Unfortunately, I do not know the exact format of IP-over-SONET, so let us assume, for the purposes of this exercise, that the only SONET overhead, above and beyond the 40 bytes times 8 bits/byte = 320 bits of IP packet payload, is the same as for ATM over SONET, i.e. 3 bytes of overhead for every 87 payload bytes in every 90 SONET bytes (BEWARE: do **not** use this number in any real design of yours, because it is most probably **not** the real number!). Also, assume again, contrary to reality, that SONET bit rates are not quantized, and can scale linearly to provide the desired packet rate. Compare the bit rates that you find here to those of question (b) and to those of exercise 3.1(b), and explain the difference.

(d) Next, assume that the incoming traffic consists of 68-Byte IP packets. This is a "bad" size for our buffer memory, because it is just above our segment size (we assume that IP packet sizes are multiples of 4 bytes, otherwise, 65 bytes would be the worst size in this case). In this case, each IP packet needs two (2) memory segments to be written in. For reasons of simplicity of memory management, again, each such IP packet is written into **two** different memory segments --hence,

approximately 128-68 = 60 bytes remain unused in every other segment (30 bytes per segment average fragmentation overhead). In this case, the peak incoming packet rate that can be supported is **half** of what it was in question (c).

Translate this packet rate into an equivalent "SONET bit rate", for each SRAM technology considered in (a), using the same IP-over-SONET assumptions used in question (c). Compare the bit rates that you find to those found earlier, and explain the difference.

(e) *--Optional Question--*
Assume again, as in question (c), that the incoming traffic consists of 40-Byte (minimum sized) IP packets. This time, however, the traffic arrives over a number of **Gigabit Ethernet** links (see also exercise 2.3). To calculate the peak packet rate of a Gigabit Ethernet link when carrying minimum sized IP packets, consider that:

- Peak packet rate is achieved over point-to-point links, where no collisions ever occur, and packets can be sent "back-to-back".
- Back-to-back packets over point-to-point Gigabit Ethernet links must be separated from each other by a 12-byte (minimum) "interframe gap".
- Each packet is preceded by an 8-byte "preamble" (for receiver clock synchronization --no other useful information is carried in that).
- The ethernet header is 14 bytes; in our case, no IP packet information is contained in this header, so it does **not** need to be stored in our buffer memory.
- The ethernet packet body contains the (one, single) IP packet. The ethernet packet body size must be at least 46 bytes (so that the total ethernet packet be at least 64 bytes, for collision detection purposes) and at most 1500 bytes. In our case, the 40-byte IP packet is padded to 46 bytes to satisfy the minimum ethernet packet body requirement.
- After its body, the ethernet packet finishes with a 4-byte CRC; this CRC contains no IP packet information, so it does **not** need to be stored in our buffer memory.

Find the peak packet rate of a Gigabit Ethernet link when carrying minimum sized IP packets. Based on this, calculate how many incoming Gigabit Ethernet links can be supported by the buffer memory of this exercise, for each SRAM technology. The incoming traffic from all links is multiplexed and written into our (single) buffer memory. Essentially, you are asked to divide the peak incoming packet rate of question (c) by the peak packet rate of one Gigabit Ethernet link; give the resulting number, ever if it is not an integer number. Is the aggregate nominal "throughput" of these links (number of links, times "1 Gbps" nominal each) higher or lower than the equivalent "SONET bit rate" in (c) (for each same technology)? Is this good or bad for the Gigabit Ethernet technology?

(f) *--Optional Question--*
Answer question (e) in the case of 68-Byte IP packets, as in question (d). As in (d), two segments per packet are needed, hence two (independent) buffer memory accesses per packet. As in question (e), assume Gigabit Ethernet links; one difference, here, is that no padding is needed in the ethernet packet body, since the 68-Byte IP packet size satisfies the 46 to 1500 byte ethernet packet body requirement.