CS-534: Packet Switch Architecture   Department of Computer Science
Spring 2007   © copyright: University of Crete, Greece

Exercise Set 3:
Memory Access Rate, Cut-through

Assigned: Wed. 28 Mar. 2007 (week 3) -- Due: Mon. 16 Apr. 2007 (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 the various SRAM technologies seen in class, in millions of accesses per second (Maccesses/s)? Consider the single-port and dual-port on-chip SRAM, the QDR (burst-of-2, burst-of-4), and the DDR (only burst-of-4 is available) off-chip SRAM examples seen in class.

(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 $64 \times 9 = 576$ 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?

(c) *--Optional Question--*
Look on the web to find newer, "available **now**" SRAM parts that offer a higher number of Maccesses/s than the best of (a). Look, for example, at companies like Micron, IDT, Cypress, IBM, Fujitsu, Hitachi, Samsung, etc. Do not spend more than 1 hour on this investigation.

3.2   DRAM Access Rate

Calculate the peak address rate for random, independent accesses for the dynamic RAM (DDR SDRAM) chip seen in class, in a similar manner 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.

What is the chip's peak data throughput (Gb/s) in this case? Assume all accesses are in the same direction (all reads, or all writes). Also, 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 READ or WRITE command, at which time the new burst starts right away, without any idle time on the data bus (I am not sure whether this is in fact possible, or whether the next ACTIVE command to the same bank implicitly terminates the previous burst, but let's assume for this question that it is possible).

How does this throughput change for alternating read-write accesses? Assume that we perform a read access of a certain burst size, followed by a write access of some appropriate burst size, followed by another read, etc, where the burst sizes are adjusted so as to not decrease the peak address rate that we started with.

3.3   PCI Bus in a First/Second Generation Router

A PCI bus can be 32- or 64-bit wide, and can use a 33 or 66 or 100 MHz clock. Assume that each data transfer over PCI takes a fixed 4-clock-cycle overhead (arbitration, turn-around, framing, addressing, etc), plus 1 additional clock cycle for each (32- or 64-bit) data word being transferred; thus, a 1-word transfer takes 5 clock cycles, while an 8-word-burst transfer takes 12 clock cycles.

(a) What is the PCI throughput, in Mbps, in the following 4 cases?

- (i) 32-bit 33 MHz PCI, transferring individual words all the time;
- (ii) 64-bit 100 MHz PCI, transferring individual words all the time;
- (iii) 32-bit 33 MHz PCI, transferring 64-byte bursts all the time;
- (iv) 64-bit 100 MHz PCI, transferring 64-byte bursts all the time.

(b) If a *first* generation switch *without* DMA is built around a PCI bus, and the system bottleneck is the PCI bus, what would be the maximum switch throughput (= aggregate incoming throughput = aggregate outgoing throughput), in Mbps, in the four cases of part (a)?

(c) Same question as (b), but for a first generation switch *with* DMA.

(d) Same question as (b), but for a *second* generation switch.

3.4   Cut-through when Port Rates differ

Consider a switch with some OC-12 links and some gigabit Ethernet links. The OC-12 (actually "OC-12c") links carry ATM traffic, with a peak throughput equal to what you calculated in exercise 1.2. The gigabit Ethernet links carry packets at the rate indicated in exercise 1.3. We wish the switch to provide cut-through, but, because of rate mismatches among its ports, cut-through transmission cannot always start "right away"; we wish to calculate the worst such required delay.

Consider the situation where a packet destined to a gigabit Ethernet output arrives through an OC-12 input. The maximum-sized Ethernet packet is 1518 bytes long, and, when arriving through an ATM port, it arrives segmented into 32 ATM cells (1518 bytes / 48 payload-bytes/cell = 31.63 cells). The first 1518 bytes in the (48-byte) payloads of the 32 cells are precisely the 1518 bytes that constitute the Ethernet packet. Assume that we are guaranteed that these 32 cells will always arrive back-to-back, without any idle cells in-between them (when ATM traffic is carried over SONET links, cells are always transmitted back-to-back, without any "spacing" between them other than the SONET overhead bytes; however, in general, not all cells need to be valid --idle cells can be freely injected into the SONET payload in the general case, but not in our case between same-packet cells).

How soon, in nanoseconds, after the arrival of the first payload byte of the first ATM cell of the above 32-cell train can we transmit the first byte of the 8-byte gigabit Ethernet preamble? Obviously, right after the 8-byte preamble is transmitted, the 1518 bytes of the Ethernet packet must be transmitted, back-to-back, at the gigabit Ethernet rate, without any "hiccups". Our switch circuits are able to transmit any given byte within 100 ns at the earliest, after that byte is received at an input port.