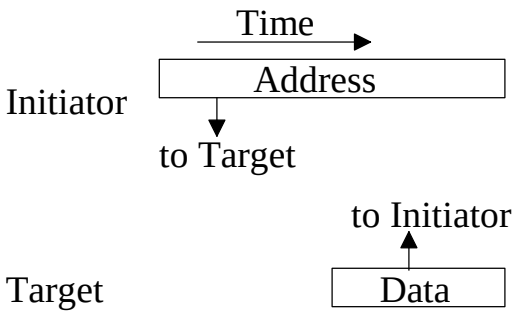
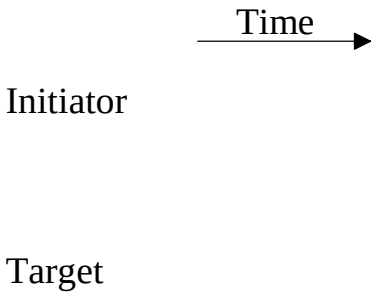
Computer Systems Test 2

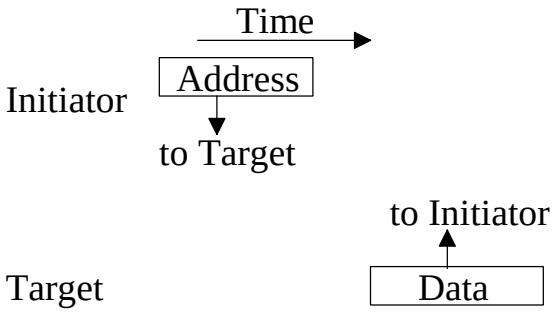
Question 1. (5 points) A READ operation on a bus with dedicated (non-multiplexed) address and data lines has a timing diagram of



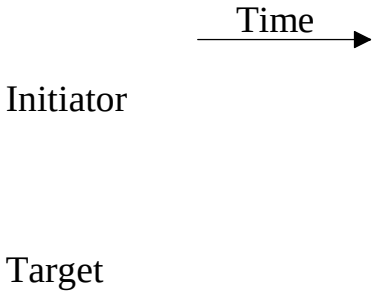
Draw the timing diagram of a WRITE operation on a bus with dedicated address and data lines.



Question 2. (5 points) A READ operation on a bus with time-multiplexed address and data lines has a timing diagram of

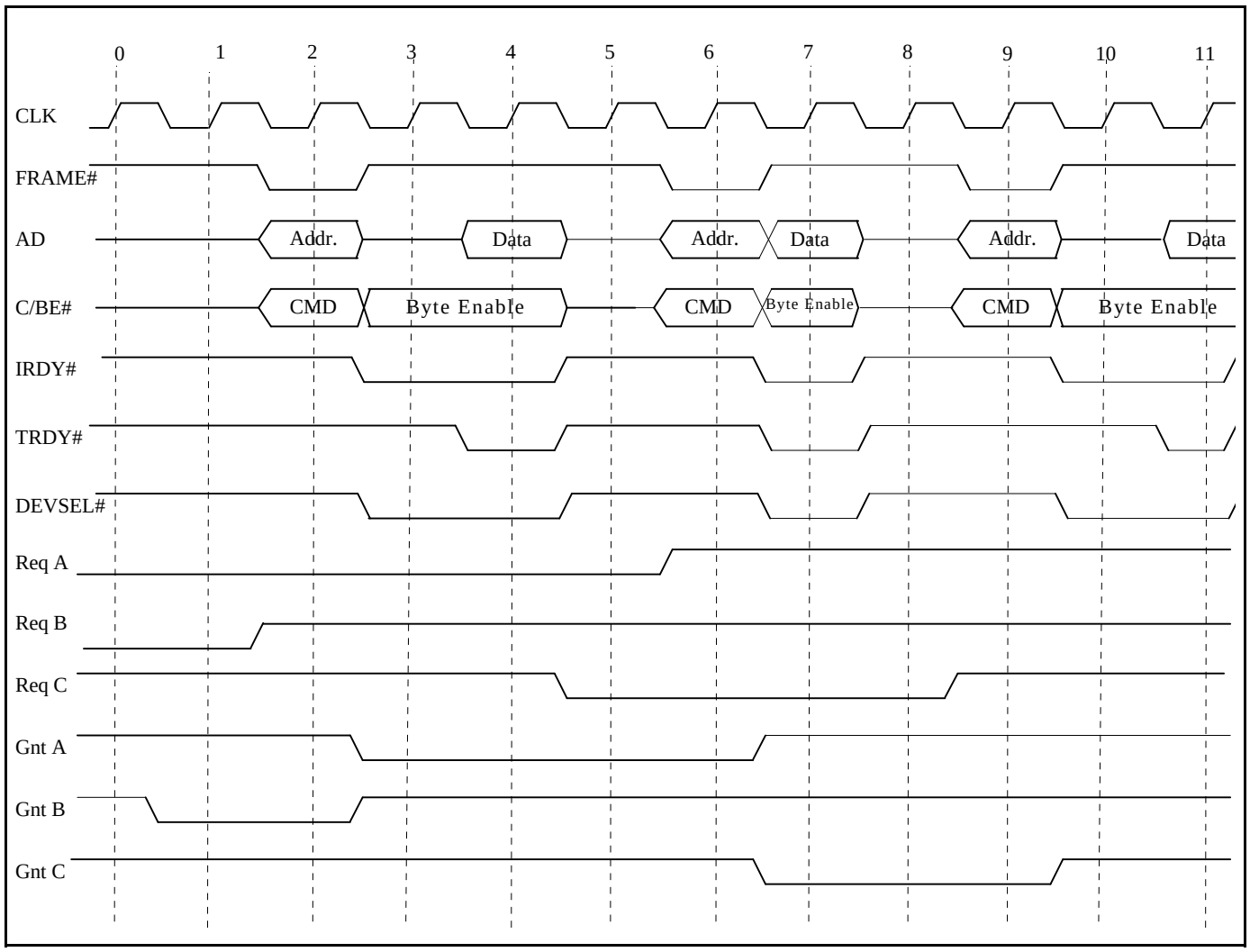


Draw the timing diagram of a WRITE operation on a bus with time-multiplexed address and data lines.



Question 3. (5 points) Why is the READ operation's timing lengthened more than the WRITE operation when using time-multiplexed address and data lines on a bus?

Question 4. (20 points) Consider the following PCI timing diagram.



a) At what clock cycles is data read off the AD wires?

b) Who controls (i.e., puts signals on) the "Req B" wire?

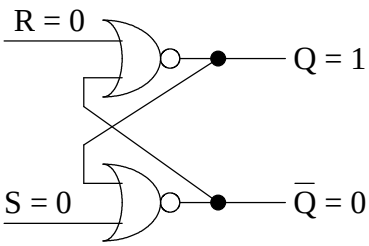
c) Who controls (i.e., puts signals on) the "Gnt B" wire?

d) For the second bus transaction, which device asserted the FRAME wire at clock cycle 5.5?

e) For the second bus transaction, how does the device know when it is time to assert the FRAME wire?

f) Use the above diagram to explain how the PCI protocol pipelines the arbitration for using the bus next with the current bus transfer.

Question 5. (10 points) The SR-latch pictured is remembering a "1". If R goes to a 1 value, what happens to the output on Q and $\overline{Q}$? (Justify your answer)



Question 6. (30 points) We looked at the comparison between implementing large memories using (i) a register-file implementation, and (ii) a square-array-of-bits implementation with a row and column decoder.
a) Explain why the square-array-of-bits implementation scales better for large size memories.

b) List one advantage of the register-file implementation over the square-array-of-bits implementation. (Explain why it is a necessary feature for a register file)

c) Explain how the square-array-of-bits implementation aids advanced memories (such as the EDRAM and SDRAM) when supplying a "burst" of localized memory accesses in a reduced amount of time.

d) In the memory hierarchy (registers, L1 cache, L2 cache, main memory, ...), why is it important for the main memory to be able to supply a "burst" of localized memory accesses in a reduced amount of time?

e) For a square-array-of-bits implementation with 2^{21} words, why would it be better to have a 11-bit row decoder and 10-bit column decoder instead of a 10-bit row decoder and 11-bit column decoder?

Question 7. (20 points) Suppose we have 30-bit memory addresses, a byte-addressable memory, and a 256 KB (2^{18} bytes) cache with 16 (2^4) bytes per line.

a) How many total lines are in the cache?

b) If the cache is direct-mapped, how many cache lines could a specific memory block be mapped to?

c) If the cache is direct-mapped, what would be the format (tag bits, cache line bits, block offset bits) of the address? (Clearly indicate the number of bits in each)

d) If the cache is 8-way set associative, how many cache lines could a specific memory block be mapped to?

e) If the cache is 8-way set associative, how many sets would there be?

f) If the cache is 8-way set associative, what would be the format of the address?

g) Why would a fully-associative cache of the above size be impractical?

Question 8. (5 points) Why is a set-associative cache more flexible than a direct-mapped cache with respect to what is in the cache at the same time?