/

p2q10
IAP

Year : 2000

Structured Hardware Design / IAP Extended Answer

Some networking researchers wish to investigate the behaviour of particular networking protocol when operating over long distance (high latency) links. To do this, they intend to build a network delay simulator, *which they will use to interconnect a pair of network nodes. Operating concurrently on each link direction, the device will receive the data stream, delay it, then transmit it onwards to the other node.*

The network uses a serial link operating at one gigabit per second, and the researchers require to be able to vary the delay such as to simulate links of between 1 and 5000 kilometers in length.

How much buffer memory is required to implement the device? *[3 marks]*

How can the data on the high-speed serial links be converted to a more manageable form? *[4marks]*

Outline a design for the delay simulator, and hence address the following points:

How many banks of memory does your design require, and what type(s) will be used?

How will the control logic function, and what technology will it be built using?

How will the delay inserted by the device be adjusted? *[13 marks]*

The high-speed input/outputs will be made more manageable by using serial to parallel converters (and vice versa), to create wider but lower clock speed buses. It is likely that if the network link is of a standard type (e.g. Gigabit Ethernet) then components to perform this conversion (along with clock recovery) will be readily available. They may be manufactured from GaAs, Bipolar or some other 'exotic' process in order to achieve the necessary speed (however, 1Gb/s is quite possible in modern CMOS processes). The serial interface between the transceiver and the serial/parallel converter is likely to use differential signalling to improve signal integrity.

Light travels at about 2E8m/s in fibre. 5000km is thus 25ms. 25ms at 125MB/s is 3.125MB. Since memory will be require in both directions we will need at least twice this.

Each link delay path will be implemented using a shift register built using an external memory component, accessed via two address counters (one writing received data into the memory, the other lagging behind reading out data to be transmitted). The delay introduced by the device will be set according to the lead the write address counter is given over the read address counter at initialisation time. A multiplexer will be used to select which of the two address registers is currently driving the address to the memory. The output of the serial to parallel converter will need to be fed through a tristate buffer before connecting to the memory's data bus, along with the parallel to serial converter.

Data is arriving at 1Gb/s = 125MB/s. For each link direction, we require to perform a read and write cycle to memory.

Try DRAM:

The read and write cycles will likely be to different DRAM pages, hence a full row access would be required, taking order 120ns. To achieve 250MB/s of R/W bandwidth per link direction would require a very wide bus.

A better approach would be to use burst mode accesses from the DRAM to read or write several words of data at a time, thus avoiding the overhead of a full row access. Some other small memory would then be required to FIFO buffer this data to match the rate with the serial/parallel

p 29/10
IAP

converters. This solution could be made to work well, but its probably easiest just to use SRAM: the quantity of memory required is relatively modest, and optimising the device's price is unlikely to be a major design goal, given the small build quantity.

SRAM:

SRAM with a 15ns access time is readily available. Treating each link direction separately, a 32 bit wide SRAM memory will be able to provide sufficient bandwidth. 512KB SRAM parts are available, so at least 7 will be required to make up the memory in each direction. One plausible solution would be to use eight 16x32KB parts. These could be connected as two 16 bit buses of four chips each. Both 16 bit bus halves would be accessed in parallel, and high order address bits used to select which pair of chips was enabled for the access.

The control logic will be replicated for each direction. The read/write counters will each need to be able to address a 4MB region with 32bit word granularity. The counters will thus need to be 20bits wide. The control logic could be implemented on an FPGA, though due to the relatively high clock rate careful design may be required. One approach would be to implement the counters using Johnson counters for speed. The only drawback of this is that it would make calculating the initialization values of the read/write counters tricky.