

3. Time Switching, Multi-Queue Memories, Shared Buffers, Output Queueing Family

3.1 TDM, Time Switching, Cut-Through

3.2 Wide Memories for High Thruput, Segm'tn Ovrhd

3.3 Multiple Queues within a Buffer Memory

3.4 Queueing for Multicast Traffic

3.5 Shared Buffering and the Output Q'ing Family

Manolis Katevenis

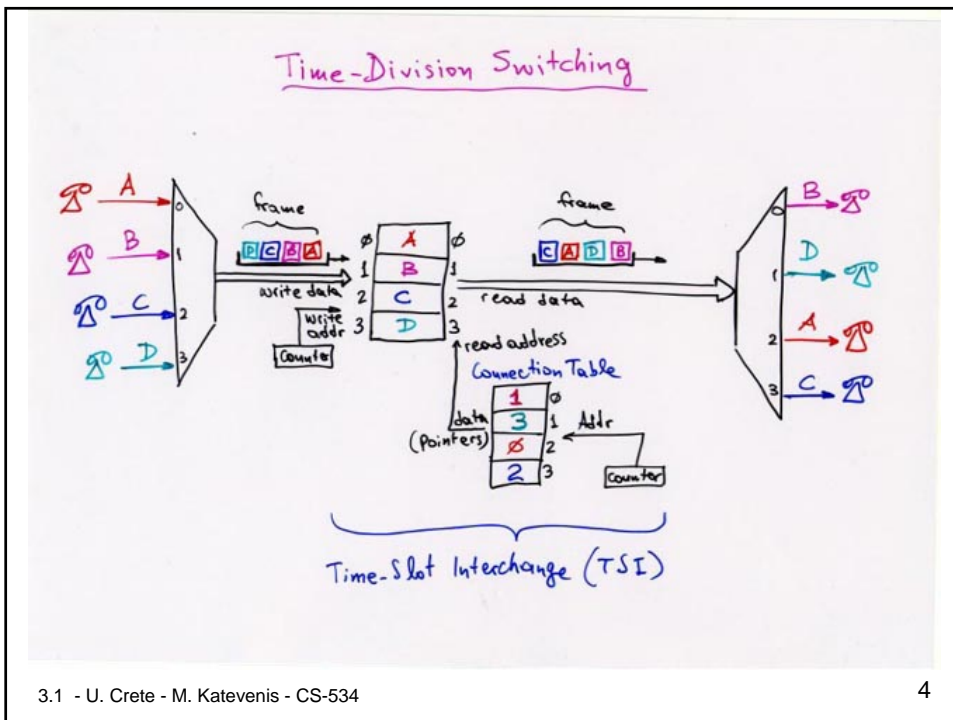
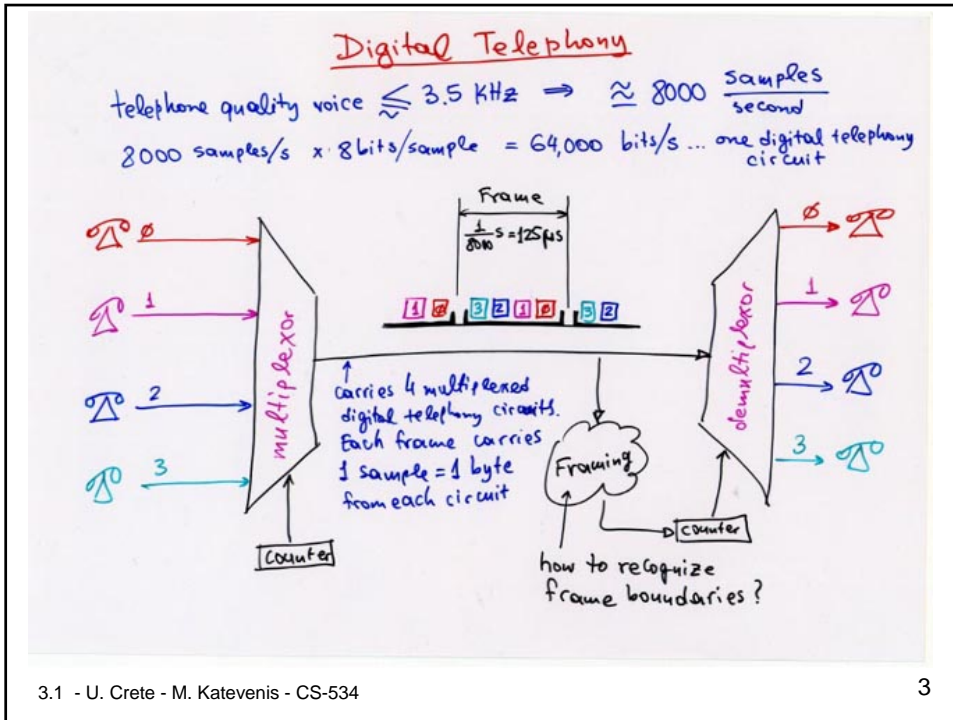
CS-534 – Univ. of Crete and FORTH, Greece

<http://archvlsi.ics.forth.gr/~kateveni/534>

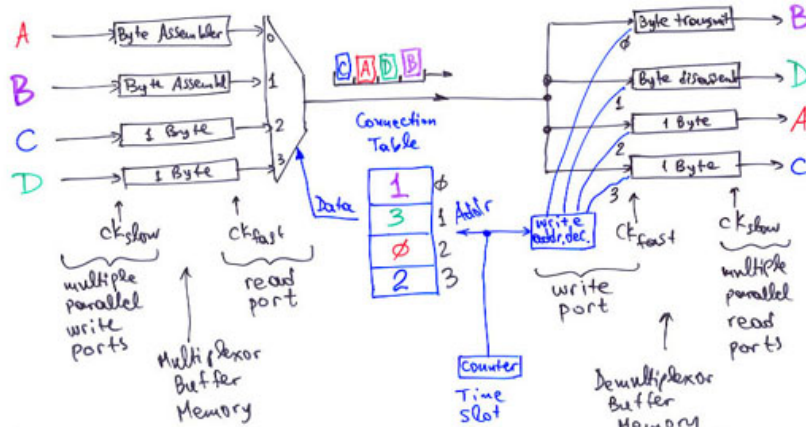
3.1 TDM, Time Switching, Cut-Through

Table of Contents:

- Digital Telephony, Time-Division Multiplexing (TDM)
- Time Switching, Time-Slot Interchanges (TSI)
- “First Generation” Switches
 - general-purpose computers, using a shared I/O bus
- “Second Generation” Switches
 - using DMA from input to output line cards
- Store-and-Forward versus Cut-Through

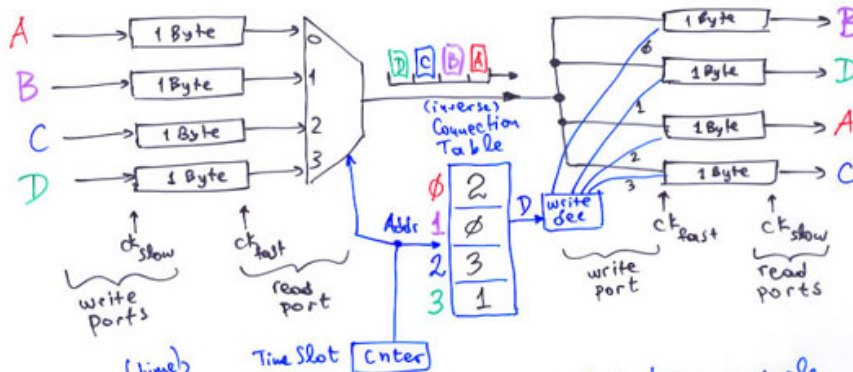


Time-Division Switching:
 When each external link carries a single connection, the time-slot interchange can be merged with the multiplexor (or with...



(Could also use TDM at bit granularity, with the connection table running at 8 times higher clock rate)

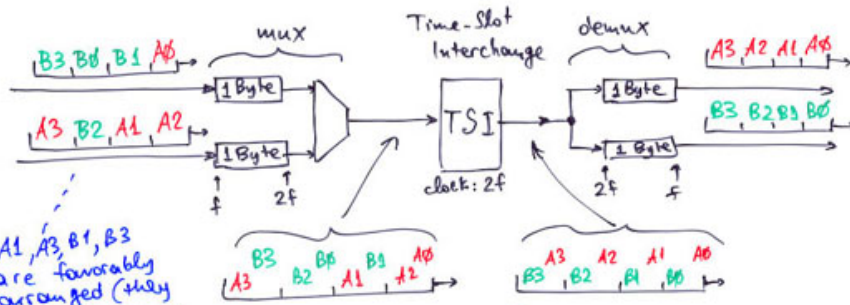
Time-Division Switching with external links carrying a single connection each:
 Alternatively, the time-slot interchange can be merged with the demultiplexor



Note: for switching to be possible, it is essential to have a single, shared line going from all inputs to all outputs. Every single quantum of information goes through that single common line. The throughput of that line = $\sum \text{inp. thrup.} = \sum \text{out. thrup.}$. In this case, $f_{\text{fastclock}} = n \cdot f_{\text{slowclock}}$ ($n = \# \text{ ext. lines}$).

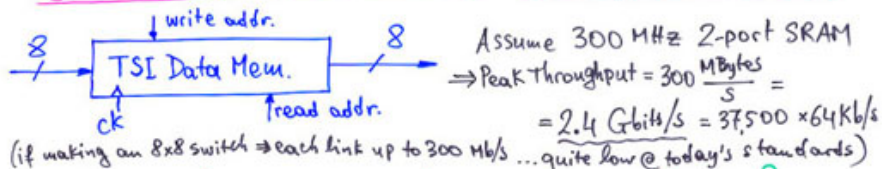
Time-Division Switching: more complex case:
multiple connections per external line

- Mux and Demux need less buffer memory than full frame
- Internal TSI needed with 1 full frame of buf. memory
- Internal TSI cannot be merged w. mux or demux
- Worst-case delay = 1 frame time, again.

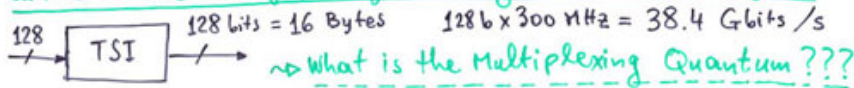


A1, A3, B1, B3 are favorably arranged (they arrive on correct time slot); however, not so for A0, A2, B0, B2, because A0, B0 arrive on same link \Rightarrow cannot be placed both in correct timeslot.
Discussion: Can I freely rearrange positions of connections inside inp. & output frames?

Byte-by-Byte Time Switching: Throughput Limit?



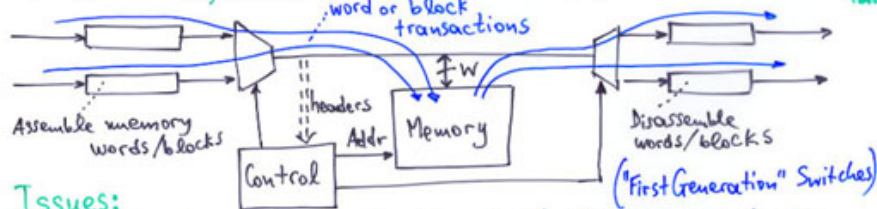
Can we Increase Throughput by Widening the TSI Memory?



- 16 Bytes belonging to a same 64Kbps channel?:
 - \rightarrow must wait 16 frames = $16 \times 125 \mu s = 2 \text{ ms}$ to collect all these bytes!
 - \rightarrow buffer size for collection = $2 \text{ ms} \times 38.4 \text{ Gb/s} = 76.8 \text{ Mbits}$
- 16 Bytes belonging to 16 "adjacent" 64Kbps channels?:
 - \rightarrow must switch all 16 channels together:
where one of them goes, all 16 of them must go!

Time Division Switching:
From Circuit Switching to Packet Switching

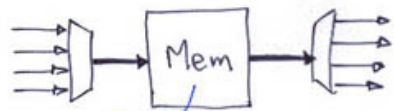
...from statically, off-line scheduled, fixed-throughput/channel to dynamically, demand-driven, on-line scheduled, variable-throughput/channel



Issues:

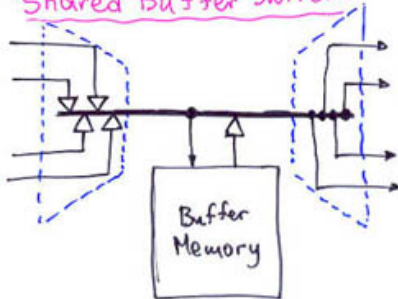
- Granularity of transfers In→Mem, Mem→Out (multiplexing quantum):
 fine grain... ⊕ small buffers in I/O ⊖ narrow mem ⇒ small throughput
 (narrow word/block): ⊕ small packets OK! ⊖ access rate/bus turn-around bound
- Control structure & operation:
 • where to store the words or blocks of each packet? contiguous? scattered?
 can I mix multiple packets in one block?
 • where to store the packets going to a certain output link?
 (or from a certain input link?) (or of a same QoS class?)

Time Switching:

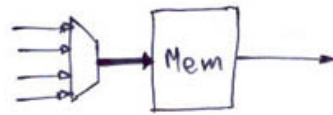


Where in this memory to store packets? (which address?)

"Shared Buffer Switch":



Simplification:
 $n \times 1$ Switch "Output Queuing"



Note: can build the general case out of this:

