Chapter 5 – Data Buffering

"How to store and organize data in memory to facilitate processing of data.

Data Buffering Schemes

Assumptions for the data buffering schemes herein:

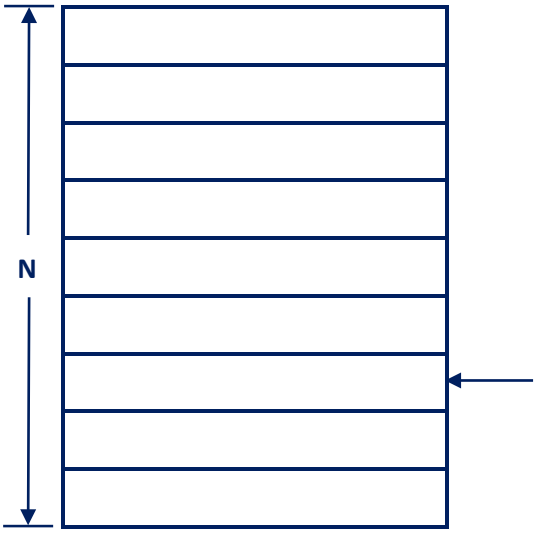
- There is a continuous stream of input data
- The data need to be processed in some way
- Often there is a resulting continuous stream of output data

Single Sample Buffer

e.g. Transfer function for high-voltage optical-based measurement:

$$\Delta\emptyset(x) = sin^{-1}\left\{\frac{1}{\propto}\left(\frac{2x}{P_0 G_1} - 1 - \propto \overrightarrow{E}\right)\right\}$$

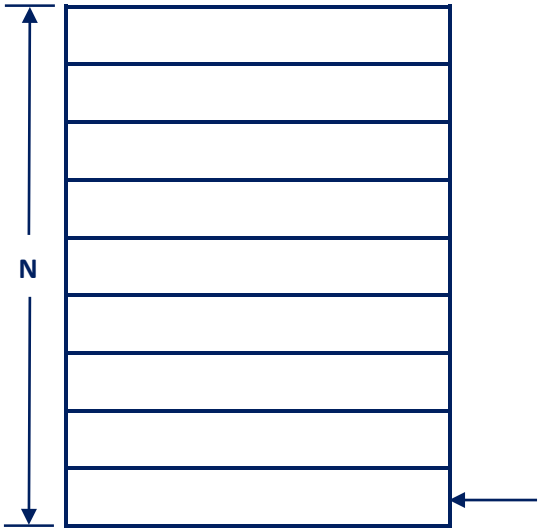
Circular Buffer e.g. Trace Buffer for Debugging – Store code sequence[1] in memory



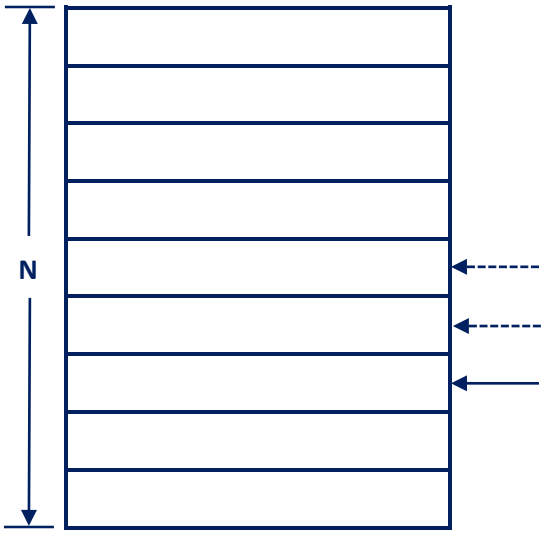
[1] Actually, program counter values are stored.

Circular Buffer vs Linear Buffer

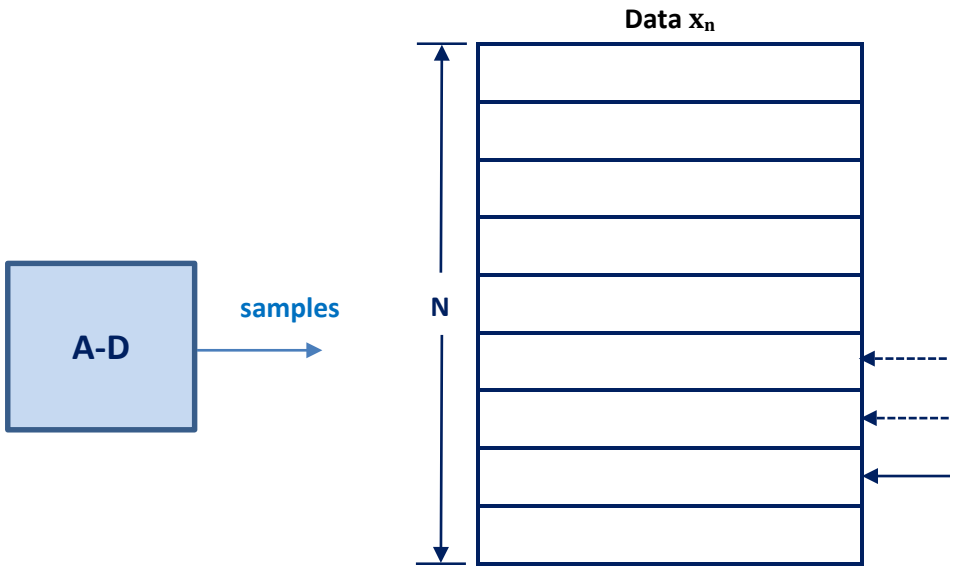
Linear



Circular



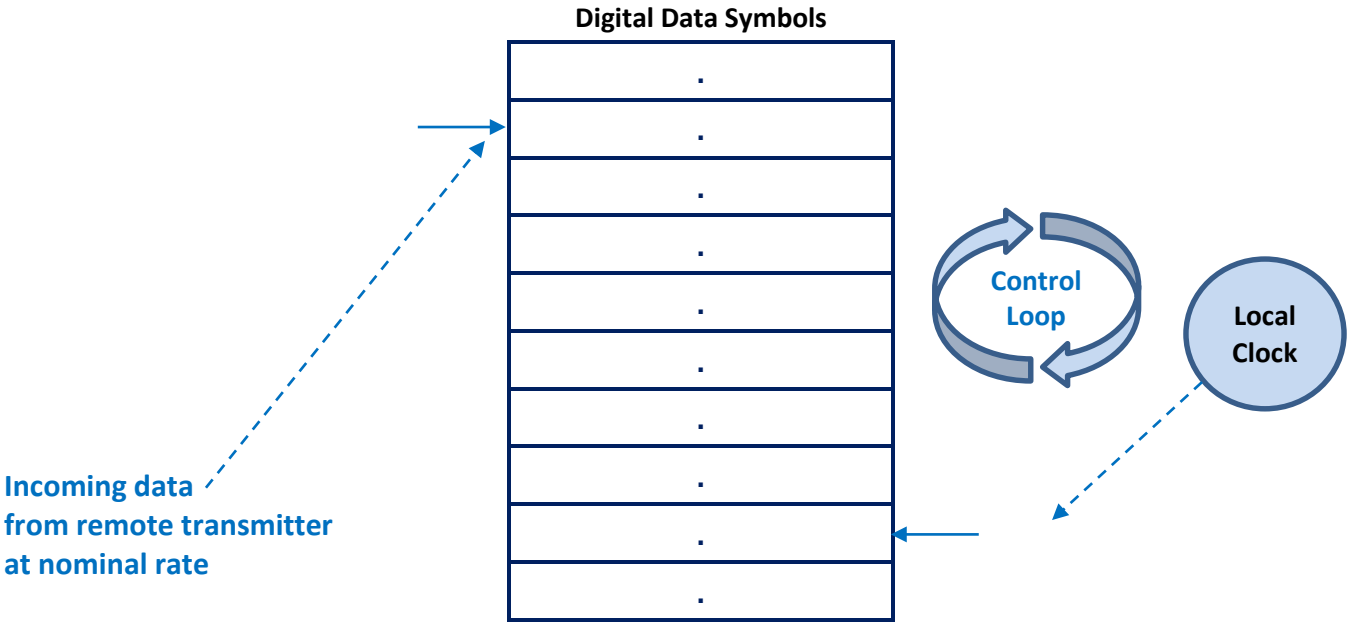
Circular Buffer e.g. FIR Filter Implemented in Time Domain



Do a dot product once every sample: $y_n = \sum_{i=0}^{N-1} c_i \cdot x_{n-i}$

(N coefficients c_i also stored in memory, but they don't change their values (typically).)

Circular Buffer e.g. Rate Synchronization for Telecom



By advancing or delaying the local clock based on the difference in pointers,

the receiver locks to the incoming data rate.

Ping-Pong Buffers for Blocks of Data e.g. FFT Processing