

Data acquisition system for high speed atomic force microscopy

Georg E. Fantner,^{a)} Paul Hegarty, Johannes H. Kindt, Georg Schitter, Geraldo A. G. Cidade, and Paul K. Hansma

Department of Physics, University of California Santa Barbara, Santa Barbara, California 93106

(Received 24 August 2004; accepted 16 November 2004; published online 25 January 2005)

With the development of atomic force microscopes that allow higher scan speeds, the need for data acquisition systems (DAQ) that are capable of handling the increased amounts of data in real time arises. We have developed a low cost data acquisition and scan control system around a commercially available DAQ board in a WINDOWS environment. By minimizing the involvement of the processor in the data transfer using direct memory access, and generation of the scan signals synchronously with the data acquisition, we were able to record 30 frames per second with a pixel resolution of 150×150 pixels and 14 bit per channel. © 2005 American Institute of Physics.

[DOI: 10.1063/1.1850651]

With the improvement of mechanical atomic force microscopes (AFM) instrumentation,^{1–6} high speed cantilever technology,^{7–11} and improved feedback techniques,^{12–15} much higher scan rates are achievable than with commercial AFM systems, while still assuring precise tracking of the sample and the nanopositioner.^{16–18} As it is in the case of fast scanning tunneling microscopy (STM),¹⁹ the acquisition of the data generated by a fast scanning AFM soon exceeds the limit of typical commercial AFM/STM controllers. Modern digital components are capable of much higher transfer rates but require a significant amount of custom hardware and software development.^{20,21} We chose to implement a solution as close as possible to a commercially available general purpose data acquisition system (DAQ). Table I lists the necessary bandwidths for the three translation axes (and therefore the DAQ bandwidth to record these) for three different imaging settings of high speed AFM measurements.

Figure 1 shows the dataflow in a typical AFM setup. The solid lines are the data transfer lines we focus on in this work.

In order to process the data, four steps have to be achieved: (1) convert the analog signals of the AFM into digital data (A/D conversion); (2) transfer the data into the PC main memory; (3) correlate the digital data stream to line and frame sync timing; and (4) display and/or save the data.

To process the required amounts of data with a PC running a standard operating system (in our case, a WINDOWS environment), the data acquisition strategy has to be able to work within the limits provided by that operating system. In particular, the uncertainty of when the operating system allows time for data processing puts restraints on the amount of CPU processing that can be done after the data is transferred to main memory, in our case over the PCI (peripheral component interface) bus. To handle this uncertainty we minimize the required amount of CPU data processing by establishing a “coherent” data acquisition. To achieve data coherence the scan signals are timed to be synchronous to the A/D converters (see Fig. 2). The timing for the A/D and

D/A channels are both derived from the same central clock. The clock signal is synchronous to three divider counters (open arrows represent clock synchronization lines). Here the clock speed (in our case 40 MHz) is divided by an integer (power of 2) to provide the required sample pulses for the specific components. The A/D converter and the digital input are timed from the same divider. Data of both inputs is then combined to a 16 bit word that is transferred to main memory (the solid arrows represent data flow). Each word then includes one data point that is labeled with the line and frame sync information. Recording the line and frame sync information (generated by a “general purpose timer counter,” GPTC) is not strictly necessary since the data already arrives in an ordered form (see Fig. 3), however it is very useful for detecting overflows in the post transfer processing.

The line and pixel scan rates are calculated to ensure that each scan line has an equal and integral number of sample clock ticks. Meaning that the pixel sample clock needs to be synchronous with the line scan. Each pixel can then be uniquely assigned to a position in a specific line and frame [see Fig. 3]. The A/D channel sample rates are derived (by the programmable clock divider) from the same 40 MHz clock and therefore are synchronous to the D/A channels. The data can then be directly written to the PC main memory without any processor involvement.

As a computer platform for our DAQ System we use a PC (2.4 GHz Pentium 4, 512MB SDRAM) running under WINDOWS XP. The requirements on the DAQ card were the following:

TABLE I. Bandwidth requirements for high speed AFM.

	High	Intermediate	Medium
Pixel resolution	256×256 pixel	256×256 pixel	100×100 pixel
Frame rate	25 frames/s	10 frames/s	10 frames/s
Scan requirements			
Z direction	3.3 MHz	1.3 MHz	200 kHz
X direction ^a	6.4 kHz	2.5 kHz	1 kHz
Y direction ^a	12.5 Hz	5 Hz	5 Hz

^aReference 23.

^{a)}Electronic mail: fantner@physics.ucsb.edu

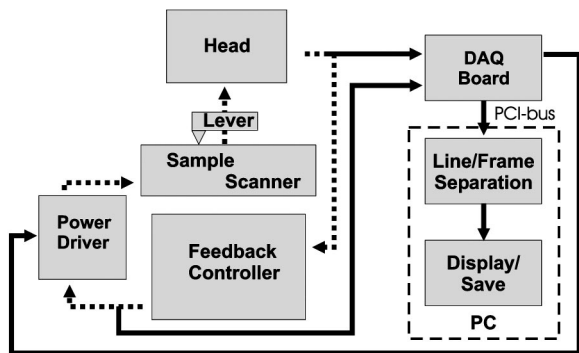


FIG. 1. Dataflow in a typical AFM setup. Solid lines show the data connections on which we focus.

- PCI Master-Mode DMA (direct memory access) to allow the transfer of data over the PCI interface directly to the main PC memory without processor involvement.
- Two 2 MS/s input channels with sufficient resolution (at least 14 bits) for data collection. In addition, it was desirable to have the ability to parse digitized data together with data from digital inputs. Data in each 16 bit data set consists then of 14 bits from a A/D converter and 2 bits from the simultaneously sampled digital inputs.
- Two 16 bits D/A channels used for generation of scan signals
- Two general purpose timer/counters (GPTC) for generating line and frame control information

One DAQ card that provided these resources was the Adlink DAQ 2002. The Adlink DAQ 2002 offers us several operation modes. We chose the continuous double buffered data acquisition mode because of the continuous nature of the data generated by the AFM. With “post trigger mode” selected, the A/D conversion is triggered by the first frame

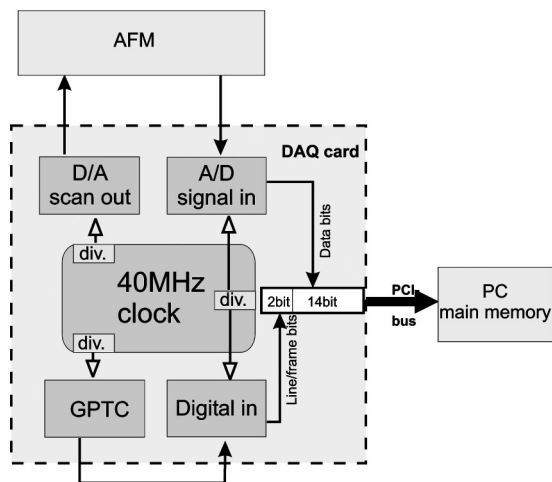


FIG. 2. Synchronization of the D/A, A/D converters, and digital input with data flow directions. The open arrows represent clock synchronization lines, the solid lines represent data flow. The clock signal goes synchronous to three dividers to provide the required sample pulses for the components. The A/D converter and the digital input are timed from the same divider. The data of both inputs is then combined to a 16 bit word (that represents the 14 bit signal data labeled with line and frame sync) that is transferred to main memory.

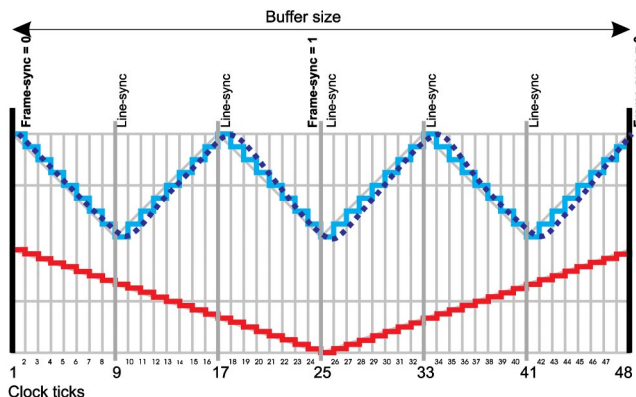


FIG. 3. Synchronization of pixel clock and line scan. Upper trace: fast scan axis; lower trace: slow scan axis. Each line has an equal and integral number of samples per line. Each frame has an integral number of lines. The dotted line in the fast scan signal represents the signal after lowpass filtering to avoid digitation.

sync. In this mode, the DAQ card waits for a trigger and then collects data until a stop command is sent.

We use the internal 40 MHz clock with programmable dividers to time the converters and “general purpose timer counters” (GPTC). In particular, the 24 bit register used for the A/D and digital input, allows us to divide the clock so we achieve a sampling rate (pixel clock) ranging from 2 Samples/s to 2 MSamples/s.

The input range of the A/D converter can be set from the control software (± 10 V, ± 5 V, ± 2.5 V) so that the resolution of the A/D conversion can be adjusted to the output

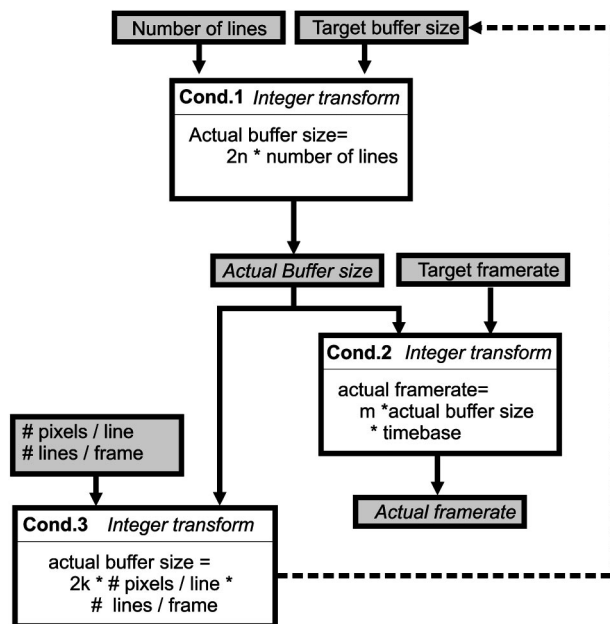


FIG. 4. Calculation of scan parameters to ensure synchronization. A fixed number of lines, a desired (target) buffer size, a desired (target) framerate, fixed number of pixels per line, and a fixed number of lines per frame are used as input parameters. The buffer size is then slightly adjusted to fit condition 1. The actual buffer size is used to calculate the actual framerate using condition 2. The outcome is checked for condition 3. Adjustments to the actual buffer size might be necessary (recursive process). The white blocks represent the conditions that need to be fulfilled. $n, m,$ and k are integer values which are calculated by the program. n and k ensure that the buffer size is an even number.

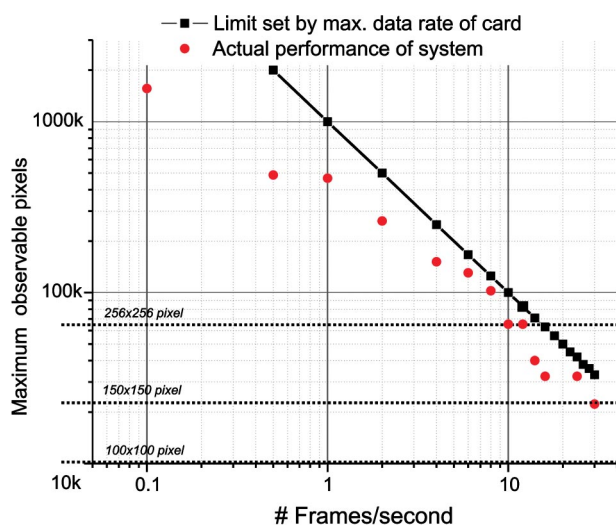


FIG. 5. Performance limits of the DAQ system. Square points represent the maximum recordable pixels at a given frame-rate set by the maximum DAQ rate of the card. Round points represent the maximum possible pixels with our implementation. One frame consists of either an upward trace or a downward trace of the slow scan axis. Both trace and retrace of the fast scan axis are recorded. For consecutive frames, this reduces the usable MSamples by a factor of 2.

range of the AFM with a maximum resolution of 0.15 mV per bit.

At the output of the D/A converter, a lowpass filter was used to smooth any digitation of the drive signals originating from the finite resolution of the D/A converter. If these steps (see Fig. 3) would not be smoothed, the step would be translated into an abrupt movement of the AFM scanner which would in turn excite the scanner at its resonances.¹⁸ The lowpass filter has to be dimensioned such that the digitation steps are filtered but the shape of the scan signal is preserved as much as possible. In order to maintain the strict correlation of scan generation and data recording, it is important that not too much phase shift is induced by the low pass filter. The dotted line in Fig. 3 gives the result of an appropriate filter. This is especially of concern if triangular scan signals are used. For sinusoidal scan signals, the lowpass filter can have a cutoff frequency just above the scan frequency.

For the post-transfer processing and DAQ card setup, we wrote a front end in LABVIEW 6.0 consisting of three applications: control panel, data recording, and data display (using an intensity graph for plotting the data).

The performance of the data acquisition system is determined by several factors. The full maximum data acquisition rate of the DAQ card could be continuously written directly to the main memory (2 MSamples/s). However, the possible combinations of frame rate and frame size are intrinsically restricted by the scheme of synchronizing the scan generation and the data acquisition to certain values (due to the necessary integer divisions). Figure 4 shows the steps to determine what combination of number of lines, buffer size, pixels per line and frame rate are possible. The variables n , k , and m are integer values.

Since almost the full 2 MSamples/s maximum data acquisition rate of the DAQ card could be transferred into the main memory, now the main speed restriction proved to be

the user defined post-transfer-data processing. In particular, the real time display of the data was processor intensive and became a limiting factor in our implementation using an intensity graph of LABVIEW.²² Depending on the display size of the intensity graph, the resources needed were different. Using lower level display methods (e.g., OpenGL or DirectX) could lessen this limitation.

Figure 5 shows the 2 MSamples/s limit compared to the actual performance that was achieved with our implementation while displaying the data in realtime. The deviation of the actual limit to the card limit originate mostly from the data post processing for the real time data display. In order to minimize this influence, we implemented different display modes: (a) line by line buildup for low scan-speeds and (b) frame at once display for high scan speeds. Choosing the right display option depending on the imaging speed, we were able to record data with up to 30 frames/s obtaining on average $62\% \pm 17\%$ of the theoretical card limit (see Fig. 5).

When only using the data acquisition part of this implementation (by direct writing of the data to the disc and displaying it off line after the experiment) almost the full 2 MSamples/s limit of the DAQ rate could be recorded. This DAQ system fulfills the requirements for the medium and intermediate goals for high speed AFM imaging described in Table I.

ACKNOWLEDGMENTS

The authors would like to thank Thomas Gutschmann, Jacqueline A Cutroni, and Marquesa M Finch for their assistance. This work was funded by the National Science Foundation under Award No. DMR-9988640 and MRL DMR00-80034, the National Institute of Health under Award No. GM65354, and NASA/URETI under Award No. NCC-1-02037 and DOC scholarship ÖAW and FWF J2395-N02 and Veeco/DI SB030071.

¹D. Knebel, M. Amrein, K. Voigt, and R. Reichelt, *Scanning* **19**, 264 (1997).

²M. B. Viani, T. E. Schaeffer, G. T. Palocz, L. I. Pietrasanta, B. L. Smith, J. B. Thompson, M. Richter, M. Rief, H. E. Gaub, K. W. Plaxco, A. N. Cleland, H. G. Hansma, and P. K. Hansma, *Rev. Sci. Instrum.* **70**, 4300 (1999).

³M. B. Viani, L. I. Pietrasanta, J. B. Thompson, A. Chand, I. C. Gebeshuber, J. H. Kindt, M. Richter, H. G. Hansma, and P. K. Hansma, *Nat. Struct. Biol.* **7**, 644 (2000).

⁴T. Ando, N. Kodera, E. Takai, D. Maruyama, K. Saito, and A. Toda, *Proc. Natl. Acad. Sci. U.S.A.* **98**, 12468 (2001).

⁵J. Kwon, J. Hong, Y.-S. Kim, D.-Y. Lee, K. Lee, S.-m. Lee, and S.-i. Park, *Rev. Sci. Instrum.* **74**, 4378 (2003).

⁶J. K. Kindt, G. E. Fantner, J. A. Cutroni, and P. K. Hansma, *Ultramicroscopy* **100**, 259 (2004).

⁷S. R. Manalis, S. C. Minne, and C. F. Quate, *Appl. Phys. Lett.* **68**, 871 (1996).

⁸D. A. Walters, B. L. Smith, A. M. Belcher, G. T. Palocz, G. D. Sucky, D. E. Morse, and P. K. Hansma, *Biophys. J.* **72**, 1425 (1997).

⁹G. T. Palocz, B. L. Smith, P. K. Hansma, and D. A. Walters, *Appl. Phys. Lett.* **73**, 1658 (1998).

¹⁰S. Hosaka, K. Etoh, A. Kikukawa, and H. Koyanagi, *J. Vac. Sci. Technol. B* **18**, 94 (2000).

¹¹B. Rogers, T. Sulchek, K. Murray, D. York, M. Jones, L. Manning, S. Malekos, B. Beneschcott, J. D. Adams, H. Cavazos, and S. C. Minne, *Rev. Sci. Instrum.* **74**, 4683 (2003).

¹²G. Schitter, P. Menold, H. F. Knapp, F. Allgöwer, and A. Stemmer, *Rev. Sci. Instrum.* **72**, 3320 (2001).

¹³S. Salapaka, A. Sebastian, J. P. Cleveland, and M. V. Salapaka, *Rev. Sci. Instrum.* **73**, 3232 (2002).

- ¹⁴G. Schitter, F. Allgower, and A. Stemmer, *Nanotechnology* **15**, 108 (2004).
- ¹⁵Q. Zou, K. K. Leang, E. Sadoun, M. J. Reed, and S. Devasia, *Asian J. Control* **6**, 164 (2004).
- ¹⁶D. Croft, S. Stilson, and S. Devasia, *Nanotechnology* **10**, 201 (1999).
- ¹⁷D. Croft, G. Shed, and S. Devasia, *ASME J. Dyn. Syst., Meas., Control* **123**, 35 (2001).
- ¹⁸G. Schitter and A. Stemmer, *IEEE Trans. Control Syst. Technol.* **12**, 449 (2004).
- ¹⁹L. Kuipers, M. S. Hoogeman, J. W. M. Frenken, and H. vanBeijeren, *Phys. Rev. B* **52**, 11387 (1995).
- ²⁰L. Kuipers, R. W. M. Loos, H. Neerings, J. Terhorstt, G. J. Ruwiel, A. P. deJongh, and J. W. M. Frenken, *Rev. Sci. Instrum.* **66**, 4557 (1995).
- ²¹A. D. L. Humphris, J. K. Hobbs, and M. J. Miles, *Appl. Phys. Lett.* **83**, 6 (2003).
- ²²Software available for download at <http://hansmalab.physics.ucsb.edu>
- ²³Due to the desired triangular scan signals, the bandwidth requirements on the positioner are one order of magnitude higher