# Appendix A

# Computer-Experiment Interfacing in the Active Feedback Experiment

The active-feedback experiment incorporates several computers and auxiliary boards for timing control of the overall experiment, data acquisition, and feedback implementation. This Appendix describes the hardware and software used to accomplish these tasks as of the end of April, 2003. The summary presented here will be of interest chiefly to those involved in future versions of the same experiment. Computers in the lab are referred to by name. Several specific programs or software-setting files are also named; these are archived separately on hard drive and CD-ROM within our research group.

## A.1 Overall Experiment Timing

The timing of the overall experiment, including the MOT cycle and data acquisition trigger, is described in Section 6.1.4. This master timing is set by a Labview program outputting timing signals from a National Instruments AT-AO-10 analog output board. The board takes a 1-kHz external update trigger which defines its clock speed. Then it updates outputs on up to ten channels every clock cycle, as instructed by the Labview routine. The computer Qolinux1 (at least seven years old right now) is more or less dedicated to running this timing program. The current Labview routine is Transits_Oct24_2002.vi, which lives in the Qolinux folder C:\User\Kevin\.

The channel 0 output of this program controls the upper MOT gradient field. It goes to the external voltage control of a Kepco ATE6-10M power supply which drives the upper MOT main coils.

The channel 1 output controls the trapping light intensity for both MOT's. It goes to the video input of an Isomet 233A acousto-optic modulator driver. Thus it controls the RF power going to the single-pass AOM in the trapping beam path.

The channel 2 output gives a trigger signal for a camera; this channel is irrelevant in the actual experiment and is only used for MOT and cooling diagnostics.

The channel 3 output gives a trigger signal for the framegrabber, a Data Translation board which acquires an image from a camera and transfers it to the computer Cqed. This channel is only used for MOT and cooling diagnostics.

The channel 4 output controls the lower MOT gradient field. It goes to an op-amp (OP27) configured for a gain of 2, and thence to the voltage control of a Kepco ATE15-50M power supply which drives the lower MOT main coils.

The channel 5 output controls the MOT trapping light detuning, allowing it to be changed during sub-Doppler cooling. This output goes to the tuning voltage ($V_T$) input of an Isomet D323B acousto-optic deflector driver. Thus it controls the frequency of the RF signal going to the double-pass AOM in the trapping beam path.

The channel 6 output sends a trigger to the Gage board for data acquisition.

The channel 7 output is used only for locking the physics cavity to the probe beam itself. This channel goes to a Stanford (SR560) preamplifier blanking input, and can be used to blank the physics cavity lock during an atom transit.

### A.1.1 Framegrabber Operation

The framegrabber board mentioned above is a Data Translation DT3152 board run with modified Data Translation Open Layers software on the computer Cqed. The framegrabber board accepts an external trigger. The program we use to acquire images is acq11hst.exe, which allows acquisition of one frame with or without triggering, or of either 60 or 180 consecutive frames on triggers. The relevant software development package is a out-of-date version of DT Open Layers; the directory D:\Dtol\ on

Cqed contains the code for acq11hst and the auxiliary libraries it requires.

## A.2 Data Acquisition

The cavity transmission data is sent to the computer in the form of a video output signal from an Agilent spectrum analyzer, just as described in Chapter 3. This signal goes to a Gage data acquisition (a/d) board (CompuScope 1450) on the computer Bigg. Data is displayed on the screen and saved using the GageScope 3.0 software package. Data traces are saved as GageScope signal files (*.sig* file extension). The current program for acquiring this data is the software's default data acquisition program; it does *not* perform a conversion from spectrum analyzer output voltage to photon number. This conversion and calibration is summarized in MathCAD routines on the computer Cqed, in the directory C:\Winmcad\. The relevant routines are PhotDBM.mcd, Beat.mcd, and Beatlin.mcd.

A second quantity, $\dot{\rho}_{est}$, must be saved as well since it forms the basis for feedback figures of merit (Chapter 5). This signal is generated on the FPGA board, as described below.

## A.3 Triggering and Feedback

The triggering and feedback, unlike the master timing of the experiment, are controlled by the FPGA board (GVA-290a) which is programmed from the computer Bigg. The FPGA board itself has four analog input channels, so the spectrum analyzer video out signal (already with 100kHz bandwidth) is sent directly to the board. Four on-board output channels can be used. The relevant output for triggering and feedback is a control voltage sent to an AOM in the probe path for switching of the probe intensity. The feedback algorithm also generates $\dot{\rho}_{est}(t)$ for each trapped atom; a record of this quantity must be saved as it is the basis for our feedback figure of merit as presented in Chapter 5. The FPGA program is largely the creation of Kevin Birnbaum, and is still very much an evolving entity. Here I give a high-level tour of its functions.

The board digitizes the input transmission signal at a depth of 12 bits. The digitized signal is compared to a programmed reference level for initial triggering to trap an atom. The transmission signal, currently binned into only 256 discrete values, is sent to a lookup table which converts transmission to $\rho_{est}$ for an atom trapped with given detunings and drive strength. The $\rho_{est}$ record is used in the slope-estimation FIR filter of Chapter 5 to generate $\dot{\rho}_{est}$. This quantity goes to an output of the FPGA board so it can be sent to the Gage board and saved.

Switching conditions, i.e., limit-crossings of $\dot{\rho}_{est}$, are determined by comparing $\dot{\rho}_{est}$ to programmed reference values. Times of these limit-crossings are noted and the delayed-feedback algorithm of Chapter 5 is implemented. Switching from one drive level to another involves: (1) switching the output control voltage going to the AOM and (2) switching lookup tables for the transmission to $\rho_{est}$ conversion on the FPGA board itself.