

Chapter 3

The CFA Analytical Ultracentrifuge Architecture

Thomas M. Laue and J. Brett Austin

Abstract As part of the Open AUC Project, the CFA is the next generation of analytical ultracentrifuge. The design philosophy of the CFA, which is to encourage hardware and software innovation, has been published. However, the hardware and software that allow and encourage future development have not been described previously. Presented here is the CFA hardware and software architecture and the rationale for how this architecture was developed. Overall, both the hardware and software architecture is modular, allowing for updates and additions over time without the need for wholesale redevelopment. The common features needed by optical systems are contained in “stacks” of electronics for synchronizing the sources and detectors to the spinning rotor. Separate computer programs operate the stacks, the motion control, the centrifuge hardware, the experiment protocol, and each optical system. These programs communicate with one another to execute functions and to provide data and status information. Each program has a database associated with it to provide nonvolatile storage and inter-process communications. While the current implementation of the CFA uses one central computer to coordinate and operate all of the systems, the modular design includes provisions for using multiple computers should that be needed for a particular optical system.

Keywords Analytical ultracentrifuge • Ultracentrifuge instrumentation • Open AUC Project • Ultracentrifuge hardware description • Ultracentrifuge operating software description

3.1 Introduction

The Model E analytical ultracentrifuge (AUC) was built for 24 years, from 1948 to 1972. More recently, the XLA/I was introduced in 1990 and has been the only commercially available AUC for the past 25 years. This chronology suggests that it is time for the introduction of next generation of AUC. Recognizing that

T.M. Laue (✉) • J.B. Austin
Spin Analytical, Inc, 468 Portland Street, Berwick, ME 03901, USA
e-mail: Tom.Laue@Spinanalytical.com; Brett.Austin@Spinanalytical.com

the development of an AUC is costly and time consuming, and that AUC has a limited commercial market, the Open AUC Project began, in part, to encourage new generations of AUC hardware and software, with the Centrifugal Fluid Analyzer (CFA, Spin Analytical, Inc.) as the base platform (Cölfen et al. 2010).

Presented here is the rationale behind the CFA hardware and software design, along with the designs' architecture and implementation.

3.2 Hardware Description

3.2.1 Hardware Design Rationale

Serving as the foundation of the new AUC, the CFA hardware will allow new or improved optical systems to be added with minimum interference. At the same time, the hardware provides the common functions needed by any optical system, such as synchronizing data acquisition with the spinning rotor, motion control, and centrifuge operations. The CFA divides the data acquisition load across several interfaces (Table 3.1). Any given optical system may need to access all of the interfaces, and the interfaces must be able to handle requests from all of the optical systems.

In order to acquire data, each optical system has sets of electronic boards, referred to as stacks. These stacks allow the CFA to identify, control, and acquire data from the device. Some devices require more than one stack. A common example is when an optical system's source and detector are located in different places in the CFA. A typical stack contains separate, bussed boards for the following functions: power supply, master clock, synchronizer, digitizer, memory, and hardware control.

Table 3.1 CFA subsystems and services

System	Function	Notes
Parallel digital	Operate data acquisition stacks, high-speed acquisition	48-bit bus, address, command, and data transfer to stacks
Low-speed serial	Board identification, low-speed operations	
Motion control	Servo, stepper, linear motor, encoder operations	Each device runs in its own thread
Centrifuge hardware	Temperature, vacuum, motor control	Operate the centrifuge hardware
Experiment	Directs the CFA operations	Common to all optical systems, rpm, temperature, etc.
Sednterp2	Sample/solvent descriptions	Calculations available that are useful to AUC analysis
Master daemon	External communications	TCP/IP socket

3.2.1.1 Data Acquisition Stacks

Each optical system will require hardware for synchronizing the spinning rotor to data acquisition. Synchronizing may be needed for the source or the detector, and the nature of the synchronizing may differ for different detector types. There are synchronizing needs that are common to all optical systems, as well as needs that are specific to a particular optical system. The common needs are a jitter-free rotor timing pulse and variable-speed clock to determine the period of one rotation (as counts of the clock), a means to determine the period of a fraction of a rotation, a means to adjust the latter count for propagation delays, and a way to produce a properly timed window when data acquisition should occur (Yphantis et al. 1984; Laue et al. 1984).

Different data rates and different signal stream types are needed to accommodate data acquisition from different optical systems, and the CFA provides two useful data buses, one a high-speed parallel digital bus and the other a lower-speed serial bus.

3.2.2 *Parallel Digital Bus*

One inescapable feature of AUC is that the spinning rotor provides the “heartbeat” for any optical detection. The rotor timing pulse is asynchronous with respect to any computer operations and unforgiving with respect to the timing of signal acquisition from any sample (Laue et al. 1983). Consequently, intervening data acquisition hardware must be used to: (1) synchronize acquisition with the spinning rotor, (2) digitize and buffer data as it comes from a detector, and (3) allow the computer to configure the acquisition hardware and download data. A parallel digital bus is used to operate the data acquisition hardware. This bus provides for addressing, commands, and data paths to and from the stacks.

3.2.3 *Serial Bus*

Many operations do not require a high-speed bus. The CFA includes a serial bus that is used to identify each board, including the board type, which optical system it is part of, and other information needed to allow optical systems to be swapped in and out of the CFA. The bus also is used to perform other low-speed functions, such as selecting optical filters.

3.2.4 Motion Control Bus

Optical systems often require motion control, and monitoring motion control systems can consume substantial computer resources. Consequently, a separate computer network operates the motion control bus. The CFA uses a commercial system capable of high-speed, high-precision, high-accuracy positioning and is compatible with a variety of motion control devices (position encoders, stepping motors, linear motors, servo motors). The motion control system operates asynchronously from the main computer, while having the capability to interrupt the main computer with information about any of several events (limit reached, move done, error condition). Calibration information for converting encoder units to instrument units (e.g., radial position or wavelength) is stored in a database (below). Depending on the servo, encoder, and leadscrew, micron-level positioning may be achieved. Motion may be initiated on several axes simultaneously and motion status read for each axis during movement. The completion of movement generates a software interrupt.

3.2.5 Power Distribution

Power for motion control and device operation is available. For each stack, voltages for digital (5 and 3.3 V) and for analog (± 12 V) circuits are generated locally to minimize electrical noise and cross talk.

3.2.6 Stack Electronics

The synchronizing functions have been divided up and the electronics for each function put on separate, stackable boards. This architecture has the advantage of isolating the different optical systems from one another, thus reducing the chance that noise from one system will affect the others. The architecture also allows for new or upgraded boards to be incorporated in the stacks without requiring changes to existing boards. In addition to the power supply, each stack may contain the following boards.

3.2.6.1 Master Clock

Each stack may contain a master clock board. This board provides a high-frequency signal used to synchronize the rotor timing pulse (RTP) with other operations (Fig. 3.1). This signal is divided to produce two lower-frequency synchronizing clocks (Laue et al. 1984). Which of the two synchronizing clocks is used depends

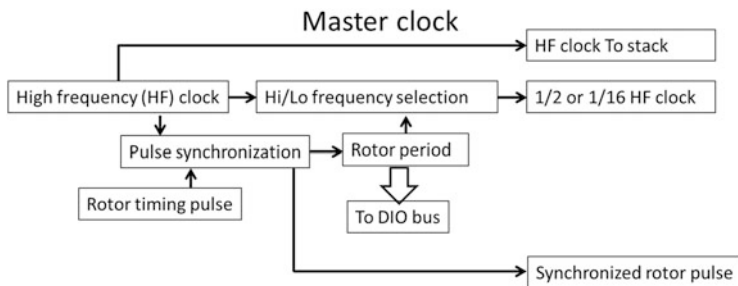


Fig. 3.1 Master clock block diagram. The master clock provides a high-frequency clock (HF) that is divided either by one-half or one-sixteenth for synchronizing purposes. The HF appears on the internal stack bus and is used to de-glitch asynchronous external signals for use by the stack. Which of the two lower-frequency clocks is used by the synchronizing circuitry depends on the rotor speed. Clock selection is made automatically to guarantee a minimum of 4000 counts per revolution while not exceeding 65,000 counts per revolution. There hysteresis built in to the clock selection circuit to prevent the clock speed from toggling rapidly between the higher and lower rates during acceleration or deceleration

on the rotor speed; above ~ 5000 rpm the higher-frequency clock is used, and below 5000 rpm the lower-frequency clock is used. The switch to higher clock rates occurs at a higher rotor speed than the switch back to the lower clock rate. This hysteresis prevents rapid switching between the two clock rates during acceleration or deceleration. The clock rates and rotor speed switch points were selected to guarantee that at least 4000 clock pulses will be provided per revolution while not exceeding 64,000 clock pulses per revolution. In this manner, 16-bit counters may be used for all synchronizing operations (Yphantis et al. 1984). In order to prevent “glitches” between the rotor timing pulse (RTP) and the clock, the undivided clock signal is used to synchronize the rotor timing pulse (SRTP) with the master clock (Laue et al. 1984). In so doing, there is an inherent uncertainty of up to 125 ns between the edges of the RTP and the SRTP. This uncertainty is less than the jitter specification ($\pm 0.1^\circ$) for the synchronizing circuit. The leading edge of the SRTP is used for all subsequent timing functions.

This board makes accessible the rotor period, which may be used to calculate the rotor speed to a fraction of an rpm. The SRTP operates a divide by two circuits to produce an ODD/EVEN signal. The ODD/EVEN signal is used to toggle between two banks of timers on the synchronizer board (below) and may be stored along with digitized data as a surrogate for the RTP.

3.2.6.2 Synchronizer

The synchronizer board produces the timing signal needed to time light source or detector operations with the spinning rotor (Fig. 3.2). Two sets of timers are used: for one revolution, timer 1 provides the synchronized signal while timer 2 is

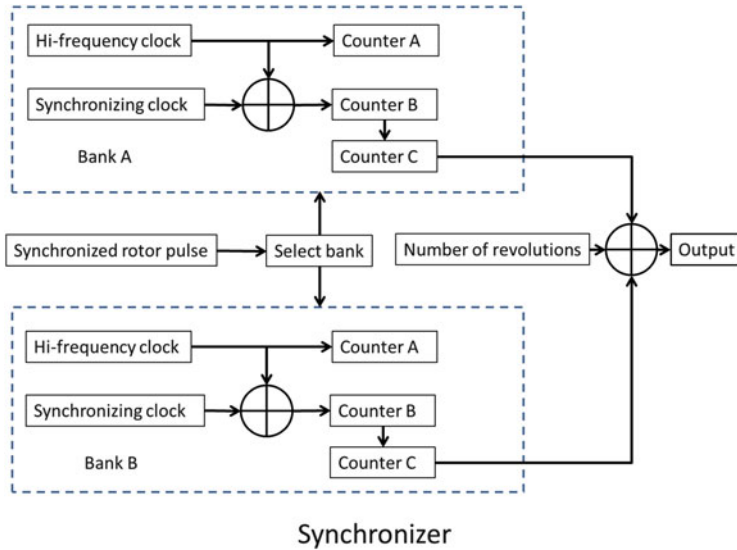


Fig. 3.2 Synchronizer block diagram. Two banks of three counters each are used. While bank one is in a “preparation” mode where the counts are loaded and adjusted for propagation delays, the other bank is generating a delay time and duration time to operate the source or detector. The circuitry can be configured to provide a single pulse per rotor revolution or to provide an equally spaced series of pulses each revolution. The synchronizer may be configured to operate for a fixed number of revolutions, or it may be allowed to run continuously

prepared for operation. On the next revolution, timer 2 provides the synchronized signal while timer 1 is prepared. This arrangement provides a correctly timed pulse each revolution, even when the timers are being updated by software.

Each timer consists of three separate 16-bit counters, A, B, and C. The counters initially are loaded with the desired count and then decrement until they reach zero counts. The A counter provides an “offset” count that compensates for propagation delays (Yphantis et al. 1984; Laue et al. 1984, and below). Counter B contains the count corresponding to the desired fraction of the rotor period (e.g., if the rotor period is 6000 counts, and you want to trigger an event at $\frac{1}{4}$ of the rotation, the B counter would start with 1500 counts). Counter C contains the period of the event in terms of a fraction of a revolution. While both the B and C counters count a fraction of the rotor period using the synchronizing clock, the A is active for a time equivalent to the propagation delays in an optical system and is used to adjust the count in the B counter as described previously (Laue et al. 1984; Yphantis et al. 1984). The operation of the A counter and consequent adjustment of the B counter occurs during the preparation phase of a timer (Yphantis et al. 1984).

The synchronizer may be operated in three modes. First, the synchronizer may be enabled a given number of rotor revolutions. This mode is used when data are collected for a fixed number of revolutions (NREV), and the signals from the samples are sorted out in software (Laue et al. 2006). Second, counter B provides the

delay (in a fraction of a revolution) needed to start an event, and counter C provides the duration of the event (in a fraction of a revolution). This mode is used to operate the pulsed laser light source for interference optical system and the pulsed Xe lamp for the absorbance optics. Third, counter B provides the time between events during one revolution, and counter C provides the duration of each event. In this mode, several pulses are provided by counter C each rotor revolution. This mode may be used to collect data from several samples at a time, for example, when using a turbidity or light-scattering detector (Mächtle 1999). The latter two modes may be used in conjunction with the NREV counter so that data collection occurs only for a fixed number of revolutions.

3.2.6.3 Analog to Digital Converter (ADC)

An optical system stack may include an analog to digital converter with up to 16-bit resolution to convert analog signals to digital values. The ADC may be operated by the signal coming from the synchronizer board, or an external signal may be supplied. Data from the ADC board is put on the stack's internal data bus, along with a pulse indicating that a valid digitized value (VDV) is available. The leading edge of the VDV is used to store the data in memory, and the trailing edge is used to increment the memory address.

3.2.6.4 Memory

Each memory card in a stack holds 1 Mbyte of 16-bit data. The data may be accessed randomly. Up to four memory cards may be connected in series, so that when one is full, the next board is activated. Data are stored sequentially from the internal data bus. The memory address may be set or read from software, and the data at an address may be read from software. A typical data acquisition sequence would be: (1) set the initial memory address to zero; (2) initiate data acquisition; (3) when acquisition is done, re-zero the address; and (4) read data and increment the address (one software command will do this). The software should not change the memory address during data acquisition.

3.2.6.5 Source/Detector Control

A control board unique to each stack handles the specific signals needed to operate a light source or detector and provides a way for the source/detector pair to coordinate their operations. For example, array detectors need to "free run" in order to minimize any dark current background signal. In operation, the source must wait to fire until the array detector has completed a scan and halted. Thus, a typical operating sequence for this type of detector is: (1) the software issues an "acquire" command; (2) the array detector completes a scan and halts further scanning, and the detector

stack issues a “ready” command to the light source stack; (3) the light source stack then provides properly synchronized bursts of light (the number of bursts depends on the source intensity and detector sensitivity, which must be determined separately); (4) when the light bursts are completed, the source stack signals the detector stack to resume scanning and to save the data from the next scan; and (5) the detector stack signals that data are available.

3.3 Software Description

3.3.1 Software Design Rationale

The CFA software is modular. Separate computer programs run each of the subsystems listed in Table 3.1, and each program is considered a service. The details needed to operate a particular subsystem are contained within its service. By using separate programs to operate the services, it will be possible to update the software for a particular service in response to hardware changes without requiring the complete replacement of the other services.

Likewise, separate computer programs operate each optical system (OptSys). Multiple optical systems may be running simultaneously (OptSys1, OptSys2, etc.), with some sharing a single optical track (e.g., schlieren and interference systems). Using this design, the details of operation, user interface, and data handling for each optical system are contained in its OptSys program and may be updated independently of the other optical systems. All of the programs broadcast a “heartbeat” once a second to let the other programs know they are still functioning. Likewise, every second each system broadcasts a status structure that provides information (e.g., busy, queue lengths, etc.) to the other programs.

3.3.1.1 Inter-process Communications

The OptSys programs send requests and commands to the services via inter-process communications (IPC). Similarly, the services send status updates and data to the OptSys programs by IPC. Because processing some commands may take a relatively long time, all services and OptSys programs have their own internal message queues. When a command is received, it is immediately placed on the internal queue, unblocking the calling program. Messages are processed from the internal queue in a first-in-first-out manner. The internal queue system does provide for the possibility of checking priorities on messages, however thus far this has not been needed. After a command has been processed, a reply, containing both the original command and any requested data, is sent to the calling program. The reply message and data are placed on the internal queue of the calling program for processing. The messaging system, then, operates in full-duplex mode.

Every message contains an address structure containing the sender ID, receiver ID, the time when sent, the command, a unique message number, whether the sender expects a receipt acknowledgment message prior to command execution, and, in a reply, whether the command was executed successfully. Each of the services has a unique set of commands associated with it. Some of these commands are described below.

3.3.2 Parallel Digital

This system operates the high-speed digital input/output system. The structure of a command to this system consists of a stack number, board number, command, and value. The first two determine which board will receive the command. The “value” is an optional 16-bit number that will appear on the data output lines when the command is executed. In addition to the stack, board, command, and value, an optional “delay” value may be provided. When used, the delay value causes the service to suspend operations for the number of milliseconds prior to handling another command. Ordinarily, the delay is not used, but is provided if some source/detector board commands need extra processing time.

In addition to sending single commands, a calling program may send a sequence of up to 10 commands in a single message. Because DIO operations are rapid, this capability is convenient and efficient. For example, the synchronizer board may be set to the angles needed for a particular sample and an “acquire data” command issued. Once the acquire data command has been issued, an acknowledgment is sent to the calling program indicating that the acquisition process is ongoing. The calling program may then issue status requests to the stacks to determine whether the acquisition process is complete and a command to retrieve the data after acquisition.

3.3.3 Motion Control

The commands to the motion control subsystem include ones to initialize the system and to command the axis to move. Each axis has an entry in a database maintained by the motion control system that identifies each axis and has the parameters needed to convert an encoder reading into a meaningful value (e.g., to cm from the center of the rotor or nm for a monochromator).

3.3.4 Centrifuge Hardware

The centrifuge hardware includes the vacuum system, the rotor speed control (rpm, acceleration, and deceleration), and the chamber temperature control. Because its

status is important to all of the other service and OptSys programs, the centrifuge hardware service broadcasts a status structure every second and records the information in a file kept with the experimental data. The status structure includes the rotor speed, temperature, vacuum, time, ω^2t , and status flags.

3.3.5 Experiment

The experiment subsystem provides information that is common to all experimental protocols including the user, the method, the rotor setup, and the samples. The user information includes a log-in name, a password, the last data path storage path, the last method and rotor setup used, and whether the user is an administrator. An administrator can set up or modify user accounts and determine whether or not logging in is required. At the start of an experiment, the user is provided the option to save the data in a location other than the default location.

The method includes the information for operating the machine. Method information includes a description, whether to wait for temperature, how long to wait after the temperature is reached before beginning the protocol, and whether to shut the machine off after the protocol is done. Each method also has one or more steps that include the rotor speed, temperature, acceleration and deceleration, duration, and flag indicating the completion of the step. The duration may be a time or it may be the number of scans required of each optical system. The number of scans is used as the default by the optical systems and may be overridden locally by them. Completion of a step is signaled when all of the optical systems indicate that they have completed all of the scans at that rotor speed and temperature.

The rotor setup includes information on the rotor and the type of cell in each rotor hole. The cell information includes a description of the cell type, the radial positions of the top and bottom of each channel, as well as the angular offset and width of each channel. This information is needed by the optical systems for data acquisition. Tables are included for keeping an inventory of rotors and cells. This inventory allows their use to be logged and may be used to associate data with a specific cell.

The samples include solvent and solute information for each channel. Solvent and solute information are saved in Sednterp, making it easier to perform data interpretation. The concentration of each solute is kept for each channel.

3.3.6 Hardware Inventory

The CFA uses a low-speed serial bus and serial number firmware to create an inventory of what optical systems are present and how they are configured. The hardware inventory subsystem allows optical systems to be swapped in and out of the CFA.

3.3.7 *Master Daemon*

The master daemon is a service used to: (1) monitor the operating status of the services, (2) monitor the operating status of the optical systems, and (3) provide external communications. This service has access to all of the databases (below) and provides status updates to any program connected to its server socket. Updates are provided every second on the status of all of the other services. These status updates include whether a service exists and is: (1) initialized, (2) busy, (3) ready to receive commands, (4) writing to its database, (5) operating in simulation mode, and (6) has encountered a critical error. In addition, the message count for the input, output, and pending queues is provided. Also, the machine status (above) is broadcast every second. Data to or from tables are handled as SQL statements. Data from tables are returned as either SQL or XML strings.

3.4 Databases

3.4.1 *Database Rationale*

Databases provide a structured means to store data that may be accessed from several programs concurrently. The CFA operating system uses SQLite databases (www.sqlite.org). SQLite is open source and widely used. There is no requirement for a separate database server application, simplifying program distribution and operations. There are SQLite bindings for over 30 programming languages and dozens of operating systems. While SQLite provides thread-safe read access (i.e., several programs may access a database at once), it is not thread-safe for writing. To prevent potential conflicts when more than one service wishes to access a database during a write, a simple write-lock system is used that consists of creating a file for the lifetime of the write process. Any blocked process must wait for the disappearance of this file before accessing the database. Because each service and each optical system has its own database, and since writing to that database is made only by that service, conflicts do not occur. The write-lock is provided in cases where an optical system is saving data to its database and an analysis program wishes to read from that database. The databases used by the CFA operating system are presented in Table 3.2.

In addition to the CFA operating system databases, each optical system will have two types of databases associated with it. The first database will hold the information needed to operate the hardware for that optical system. The second database will hold the data from a particular experiment for that optical system. For example, the multiwavelength absorbance (MWA) optical system uses a hardware database that contains the information presented in Table 3.3. A new stand-alone data database is created for each experiment. In this way, a data database will provide an analysis programs with a complete record of an experiment.

Table 3.2 CFA operating system databases

Database	User	Tables	Notes
Processes	All services and optical systems	Paths	Folder locations
		Processes	Identifying information for inter-process communications
Hardware	Machine operations and all optical systems	Axes	Motion control
		Instrument	Service information
		Inventory	Stack and board information, offset, and speed dependence parameters
		Optical tracks	
Experiment	All optical systems	User	Information used by all optical systems
		Method and steps	
		Setup	
		Samples	
		Status	
Error	All services and optical systems	Errors	Log of all errors encountered, which service or system and when
Utility	All services and optical systems	Units	Various units, length, concentration, etc.
		Min and max	

Table 3.3 MWA hardware database

Table	Contents
Channels	Detailed angle information for each channel
Finetune	Rotor speed-dependent offset values
GainMeta	Statistics for gain scans by wavelength
GainScans	Intensity scan at each wavelength taken during gain setting
Gains	Gain and averaging used at each wavelength
Instrument	Identification and service interval information
Monochromator	Information about the monochromator
NolightScans	Intensity scans acquired using the gain settings, but without light
OpticalTrack	Angles and speed dependence parameters

Acknowledgments The authors wish to acknowledge the support of the AUC community during the development of the CFA. Particular appreciation is expressed to Bo Demeler and Walter Stafford, who have contributed to this work. The tireless work of Anthony Cipolla and Paul Vachon at Spin Analytical must be recognized. They not only have helped contribute to the design process, but they have built, assembled, and integrated components. We wish to dedicate this work to our

predecessors: Ed Pickels, Howard Schachman, David Yphantis, Ken Van Holde, David Teller, and so many others who have spent time developing and improving AUC hardware over the years.

References

- Cölfen H, Laue TM, Wohlleben W et al (2010) The open AUC project. *Eur Biophys J* 39(3):347–359
- Laue TM, Domanik RA, Yphantis DA (1983) Rapid precision interferometry for the analytical ultra-centrifuge 1. A laser controller based on a phase-lock-loop circuit. *Anal Biochem* 131(1):220–231
- Laue TM, Yphantis DA, Rhodes DG (1984) Rapid precision interferometry for the analytical ultracentrifuge. III. Determination of period of rotation, frequency of rotation and elapsed time. *Anal Biochem* 143(1):103–112
- Laue TM, Austin JB, Rau DA (2006) A light intensity measurement system for the analytical ultracentrifuge. *Progr Colloid Polym Sci* 131:1–8
- Mächtle W (1999) High-resolution, submicron particle size distribution analysis using gravitational-sweep sedimentation. *Biophys J* 76:1080–1091
- Yphantis DA, Laue TM, Anderson IA (1984) Rapid precision interferometry for the analytical ultracentrifuge. II. A laser controller based on a rate-multiplying circuit. *Anal Biochem* 143(1):95–102