

*Idaho
National
Engineering
Laboratory*

INEL-96/0056
Revision 2

RECEIVED
OCT 07 1996
OSTI

***SWEP Assay System
Version 2.0
Software Requirements
Specification***

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED 

MASTER

June 1996

LOCKHEED MARTIN 

SWEPP Assay System Version 2.0

Software Requirements Specification

S. D. Matthews
L. V. East
E. S. Marwil
J. J. Ferguson

Published June 1996

Idaho National Engineering Laboratory
Lockheed Martin Idaho Technologies Company
Idaho Falls, Idaho 83415

Prepared for the
U.S. Department of Energy
Office of Environmental Restoration and Waste Management
Under DOE Idaho Operations Office
Contract DE-AC07-76ID01570

DISCLAIMER

**Portions of this document may be illegible
in electronic image products. Images are
produced from the best available original
document.**

DISCLAIMER

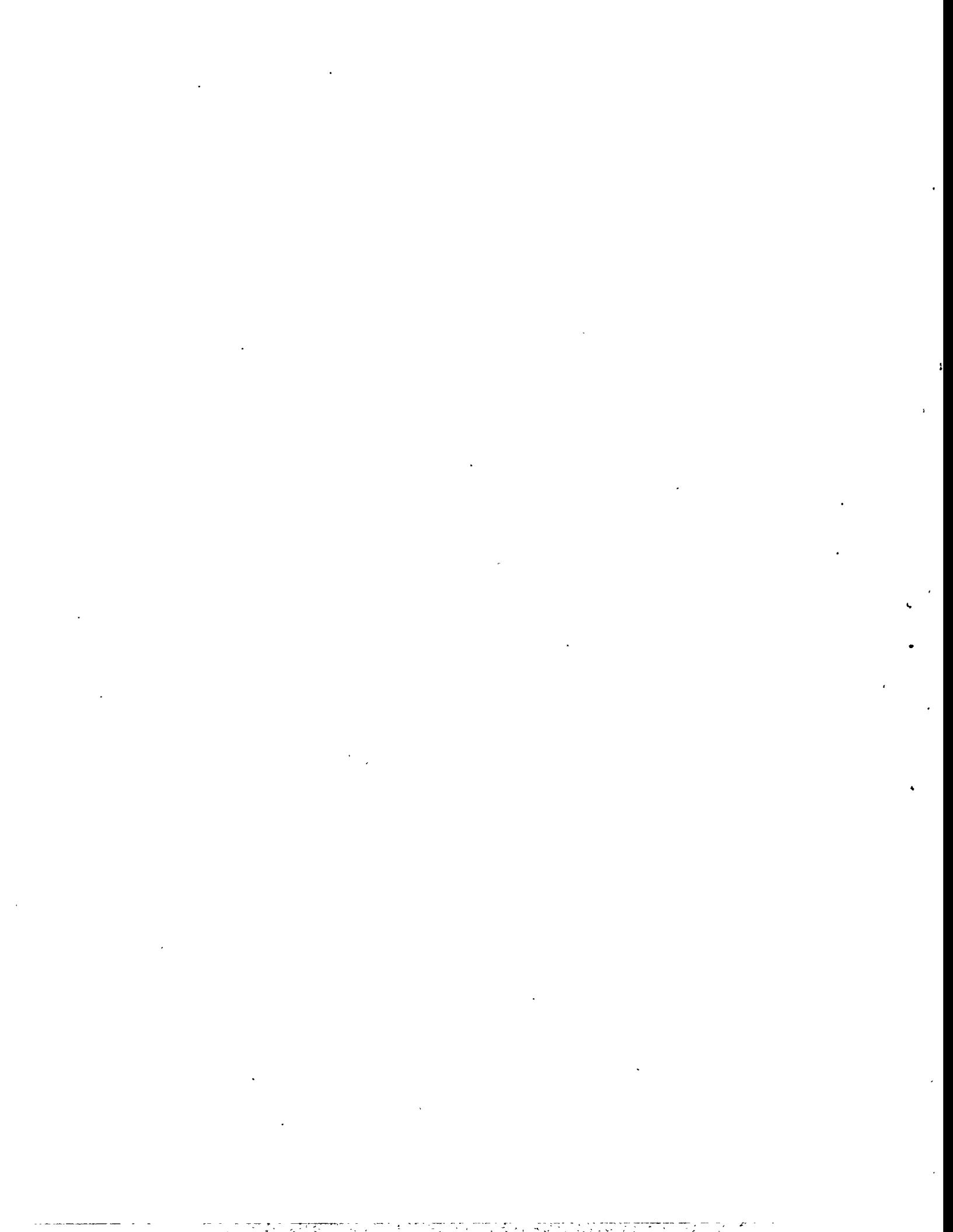
This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Acknowledgments

The comments and suggestions provided by the RWMC/SWEP Operations Personnel under the direction of G. K. Tedford during the development and testing of this software are gratefully acknowledged. Extensive testing and evaluation of the software during development of Version 2.0 was provided by G. W. Twedell. D. E. Evans provided invaluable assistance in editing this document.

Approvals

Approval signatures for this document are contained in DAR-EO-RS-2795 on file in the Document Control Office of the INEL Radioactive Waste Management Complex. *[Signature]*



Contents

Acknowledgments	ii
Approvals	ii
1. INTRODUCTION.....	1
1.1 Purpose.....	2
1.2 Scope.....	2
1.3 Document Overview.....	2
1.4 Definitions, Acronyms, and Abbreviations	2
1.5 References.....	3
2. GENERAL DESCRIPTION.....	5
2.1 Perspective.....	5
2.1.1 System Description.....	5
2.1.2 Non-Destructive Measurement Principle	6
2.1.3 Passive Data Acquisition Technique.....	7
2.1.4 Active Data Acquisition Technique	8
2.1.5 SAS Response Interpretation	9
2.2 Product Functions	10
2.2.1 User Interface	11
2.2.2 Data Acquisition	11
2.2.3 Data Reduction.....	12
2.2.4 Remote and Local Mode.....	12
2.2.5 DMS Interface.....	12
2.2.6 SGRS Interface.....	13
2.3 User Characteristics.....	13
2.4 General Constraints.....	13
2.4.1 Regulatory Policies	13
2.4.2 Hardware Environment	13
2.5 Assumptions and Dependencies	15
3. SPECIFIC REQUIREMENTS.....	17
3.1 Functional Requirements	17
3.1.1 Software Model.....	17

3.1.2 User Functional Requirements.....	27
3.1.3 Data Acquisition Functional Requirements.....	30
3.1.4 Data Reduction Functional Requirements.....	62
3.1.5 DMS Interface Functional Requirements.....	101
3.2 Performance Requirements	108
3.2.1 User Interface	108
3.2.2 CAMAC Interface	108
3.2.3 DMS Interface.....	108
3.3 Design Constraints	108
3.3.1 NEUT2 Functionality.....	108
3.3.2 Hardware Limitations	108
3.4 Attributes.....	109
3.4.1 Quality.....	109
3.4.2 Security	109
3.4.3 Portability	109
3.5 External Interface Requirements.....	109
3.5.1 User Interfaces.....	109
3.5.2 Report Interface.....	109
3.5.3 Hardware Interfaces.....	110
3.5.4 Software Interfaces.....	110
3.5.5 Communication Interfaces.....	110
3.6 Other Requirements.....	110

List of Figures

1.	Drum assay chamber	6
2.	Block diagram of system components.....	12
3.	Assay system object model, level 1.....	18
4.	Assay system object model, level 1.1.....	19
5.	Assay system object model, level 1.2.....	19
6.	Assay system object model, level 1.3.....	20
7.	Assay system data flow diagram, context level.	21
8.	Assay system data flow diagram, level 0.....	22
9.	Assay system data flow diagram, level 1.....	23
10.	Assay system data flow diagram, level 1.1.....	24
11.	Assay system data flow diagram, level 1.3.....	25
12.	Assay system data flow diagram, level 2.....	26
13.	User state transition diagram.....	27
14.	Data acquisition state transition diagram.	31
15.	DMS state transition diagram.....	102

List of Tables

1.	CAMAC crate module configuration.....	14
2.	Default chamber attribute values.	33
3.	Default chi-square parameters.....	35
4.	CAMAC crate port configuration, drum chamber, active acquisition.	37
5.	CAMAC crate port configuration, drum chamber, passive acquisition.	38
6.	CAMAC crate port configuration, crate chamber, passive acquisition.....	39
7.	CAMAC crate port configuration, crate chamber, active acquisition.....	40
8.	Mass selection criteria.....	66
9.	Moderator index.	67
10.	Self absorption correction factor coefficients.....	67
11.	Equation parameter, active mass calculation.....	71
12.	Equation parameter, passive mass calculation.....	77

SWEPP Assay System Version 2.0

Software Requirements Specification

1. INTRODUCTION

The Idaho National Engineering Laboratory (INEL) Stored Waste Examination Pilot Plant (SWEPP) operations staff use nondestructive analysis methods to characterize the radiological contents of contact-handled radioactive waste containers. Containers of waste from Rocky Flats Environmental Technology Site and other DOE sites are currently stored at SWEPP. Before these containers can be shipped to the Waste Isolation Pilot Plant (WIPP), SWEPP must verify compliance with storage, shipping, and disposal requirements. This program has been in operation since 1985 at the INEL Radioactive Waste Management Complex (RWMC).

One part of the SWEPP program measures neutron emissions from the containers and estimates the mass of plutonium and other transuranic (TRU) isotopes present. A Passive/Active Neutron (PAN) assay system developed at the Los Alamos National Laboratory [1]¹ is used to perform these measurements. A computer program named NEUT2 was originally used to perform the data acquisition and reduction functions for the neutron measurements. This program was originally developed at Los Alamos and extensively modified by a commercial vendor of PAN systems and by personnel at the INEL. NEUT2 uses the analysis methodology outlined in [1], but no formal documentation exists on the program itself. The SWEPP Assay System (SAS) computer program replaced the NEUT2 program in early 1994. The SAS software was developed using an "object model" approach and is documented in accordance with ANSI/IEEE standards. The new program incorporates the basic analysis algorithms found in NEUT2. Additional functionality and improvements include a graphical user interface, the ability to change analysis parameters without program code modification, an "object model" design approach and other features for improved flexibility and maintainability.

This document supersedes the original SAS software requirements specification (SRS) [2]. The original requirements that were used to develop the initial software release (Version 1.0) are presented with corrections, along with new requirements for its first major upgrade (designated as Version 2.0). Companion documents include a software design description [3], a software verification and validation plan [4], a software test plan and report [5], and a software configuration management plan [6]. Development of this technology is expected to continue after the Version 2.0 release.

Version 1.0 of the SAS software was developed by staff at the INEL Scientific Computing unit under the technical direction of the Nuclear and Radiation Physics unit. Version 2.0 is under development by staff from the INEL Nuclear Physics and Radiation Chemistry Department, with documentation and testing assistance provided by personnel from the Software and Electronics Department and the Engineering Analysis Department. Funding for this work has been provided through the INEL Transuranic (TRU) Waste Programs Department.

¹ References cited in this document are listed in Section 1.5.

1.1 Purpose

The primary purpose of this specification is to provide a technical basis for SAS software design, development and verification and validation. This specification covers the SAS interfaces and is guided by the requirements specification format and content guideline suggested by ANSI/IEEE software engineering standards [8], [9].

The intended audiences for this specification are the SAS software developers and verification and validation analysts. In addition, this SRS provides a basis for understanding between the developers and the users of the SAS software.

1.2 Scope

This document specifies the requirements for the SAS software as installed at the INEL and was written to comply with the RWMC quality requirements for software documentation and configuration management [26].

1.3 Document Overview

Section 2 of this SRS presents a general description of the SAS software and its high-level requirements. Section 3 follows with specific requirements depicted by object diagrams, state transition diagrams, data flow diagrams, and process specifications. Quality requirements are specified in Section 3.4.1.

Analysis notations for specific requirements are those commonly associated with object-oriented analysis [10]. Objects are characterized with attributes (variables) and methods (functions). A computer aided software engineering tool, MacAnalyst [11], was used to automate the initial object designs.

1.4 Definitions, Acronyms, and Abbreviations

Reference [9] defines general terms, acronyms, and abbreviations typical to an SRS and to the life cycle approach to software development. The following are specific to this document:

CAMAC	Computer aided measurement and control [12].
DMS	Data management system.
GPIB	General Purpose Interface Bus [13].
INEL	Idaho National Engineering Laboratory.
NEUT2	A previously existing computer program which performed the data acquisition and reduction for the neutron measurements. The computer program specified in this document replaces the NEUT2 program.

object-oriented	A software analysis methodology that decomposes a problem into objects encapsulating all the functions and attributes of that system entity.
PAN	Passive/Active Neutron assay system.
RWMC	Radioactive Waste Management Complex.
SAS	SWEPP Assay System, the data acquisition system with which the computer program specified in this document will be used.
SGRS	SWEPP Gamma-Ray Spectrometer [14].
SRS	Software requirements specification.
state transition	Software model of an object's behavior as a function of time.
SWEPP	Stored Waste Examination Pilot Plant.
TRAMPAC	TRUPACT-II Authorized Methods for Payload Control.
TRU	Transuranic waste.
TRUPACT	Transuranic package transporter.
WGPu	Low burn-up ("weapons grade") plutonium.
WIPP	Waste Isolation Pilot Plant.

1.5 References

All references cited in this document are listed below.

- [1] J. T. Caldwell, et. al., *The Los Alamos Second-Generation System for Passive and Active Neutron Assays of Drum-Size Containers*, LA-10774-MS, Los Alamos National Laboratory, September 1986.
- [2] S. D. Matthew, G. K. Becker, E. S. Marwil and G. V. Miller, *SWEPP Assay System Software Requirements Specification*, EGG-RAAM-10423, EG&G Idaho, Inc., September 1993.
- [3] L. V. East and E. S. Marwil, *SWEPP Assay System Version 2.0 Software Design Description*, INEL-96/0057, Lockheed Idaho Technologies Co., April 1996.
- [4] S. D. Matthews, *SWEPP Nondestructive Assay Software Verification and Validation Plan*, INEL-96/0008, Lockheed Idaho Technologies Co., March 1996.
- [5] J. J. Ferguson and T. K. Overlin, *SWEPP Assay System Version 2.0 Software Test Plan and Report*, INEL-96/0058, Lockheed Idaho Technologies Co., April 1996.
- [6] J. J. Ferguson, K. D. Boyd and E. S. Marwil, *SWEPP Assay System Version 2.0 Software Configuration Management Plan*, INEL-96/0007, Lockheed Idaho Technologies Co., March 1996.
- [7] *Quality Assurance Management Manual*, PDD-1, Lockheed Idaho Technologies Co. November 1995.

- [8] ANSI/IEEE Std. 830-1984, *IEEE Guide to Software Requirements Specifications*, Institute of Electrical and Electronic Engineers, Inc.
- [9] ANSI/IEEE Std. 610.1a-1990, *IEEE Standard Glossary of Software Engineering Terminology*, Institute of Electrical and Electronic Engineers, Inc.
- [10] James Rumbaugh et al., Object-Oriented Modeling and Design, Prentice-Hall, New York, 1991.
- [11] "MacAnalyst, Release 3.2," obtained from Excel Software, Marshalltown, Iowa.
- [12] ANSI/IEEE Std. 583-1975, *IEEE Standard Modular Instrumentation and Digital Interface System (CAMAC)*, Institute of Electrical and Electronic Engineers, Inc.
- [13] ANSI/IEEE Std. 488.2-1987, *IEEE Standard Digital Interface for Programmable Instruments*, Institute of Electrical and Electronic Engineers, Inc.
- [14] T. E. Curry, *SWEPP Gamma-Ray Spectrometer System Software Requirements Specification*, EG&G Idaho, Inc. internal document ES-51544, September 1994.
- [15] D. Reilly, et. al., Passive Nondestructive Assay of Nuclear Materials, Sect. 16.4.2, NUREG/CR-5550, Washington, DC, March 1991.
- [16] "PathWorks Version 4.1" obtained from Digital Equipment Corporation, Maynard, Massachusetts.
- [17] National Instruments Corp., Austin, Texas.
- [18] "Reference Manual," Model 217 Gated Clock Generator with Option 1, Jorway Corporation, Westbury, NY, 1984.
- [19] "Operator's Manual," CAMAC Model 2323A Dual Gate and Delay Generator, LeCroy, Chestnut Ridge, NY, 1987.
- [20] "Operator's Manual," CAMAC Model 2551, 12-Channel, 100 MHz Scaler, LeCroy, Chestnut Ridge, NY, 1983.
- [21] "Technical Reference Manual," Model QS-450 100 MHz Quad Scaler, DSP Technology, Inc., Fremont, CA.
- [22] "Operator's Manual," Model 8901A CAMAC to GPIB Interface, LeCroy, Chestnut Ridge, NY, 1987.
- [23] H. W. Coleman and W. G. Steele, Experimentation and Uncertainty Analysis for Engineers, John Wiley & Sons 1989.
- [24] J. R. Taylor, An Introduction to Error Analysis, Oxford University Press 1982.
- [25] G. K. Becker, "SWEPP Non-Destructive Assay (NDA) Methodology for Waste Enriched Uranium," EG&G Idaho EDF-RWMC-657, September 1993.
- [26] "RWMC Quality Assurance Project Plan (QAPjP)," INEL-94/0084, 1994.
- [27] D. E. Kuden and C. C. Rauen, "RWMC Compliance Plan for Trupact-II Authorized Methods for Payload Control," EG&G Idaho WM-PD-88-012, October 1991.
- [28] "Quality Assurance Requirements for Nuclear Facilities", ASME NQA-1-1994.

2. GENERAL DESCRIPTION

This section describes the SAS software design concept, current capabilities, limitations, and user characteristics.

2.1 Perspective

SWEPP operations use nondestructive assay (NDA) measurements to quantify the masses of fissile and TRU nuclides, e.g., weapons grade plutonium (WGPU), in contact-handled waste containers. Such quantification is necessary to ensure compliance with storage, shipping, and disposal requirements. The primary NDA measurement system used for this quantification is the SWEPP Assay System (SAS), a passive-active neutron (PAN) assay system originally developed at the Los Alamos Scientific Laboratory [1]. In addition, a system utilizing high resolution gamma-ray spectroscopy to determine isotopic mass ratios has recently been put into operation at SWEPP. Results from this system, designated as the SGRS [14], are used by the SAS software to determine final isotopic mass values.

SAS measurements can be acquired from either of two assay chambers: the drum assay chamber and the crate assay chamber. The drum assay chamber, shown in Figure 1, is designed principally for 55-gallon drums. The crate assay chamber can process larger containers, typically large boxes and crates. The crate assay chamber has not been certified for SWEPP use and is not currently in use; the following discussion will therefore be specific to the drum assay chamber.

2.1.1 System Description

The SAS system will detect and quantify TRU material entrained in the waste matrix of a 55-gallon drum. The design concept is based on the moderation of fast fission neutrons to slow or thermal neutron energies to induce fissions in fissile isotopes and the detection of neutrons from the spontaneous fission of other TRU nuclides, primarily ^{240}Pu .

The drum chamber structure is fabricated from aluminum. Compared with other structural materials, such as carbon steel, aluminum has a relatively low neutron absorption cross-section. Graphite and polyethylene are the moderating materials incorporated into the structure. Polyethylene also shields the chamber from external background neutrons to prevent a response to neutron sources other than those in the container being measured. Borated polyethylene lines the internal surface of the polyethylene shield layer to absorb background neutrons moderated by the polyethylene shield.

Neutrons are detected with ^3He neutron detector assemblies mounted in the top, bottom, and four sides of the chamber. In each of these six walls are mounted two types of detector assemblies, referred to as the bare and shielded detectors. The bare detectors consist of 2-in. diameter, 2-atmosphere ^3He detectors imbedded in polyethylene. The shielded detectors are similar, except that the shielding consists of 0.5-in. polyethylene adjacent to the detectors. These assemblies are encased in a 0.031-in. carbon steel structural support which is wrapped in 0.062-in. of cadmium and 0.125-in. of boroflex. One of the primary purposes for incorporating two

different detectors is to measure the degree of moderation that occurs in the waste matrix, thereby accounting for the resultant efficiency variations of the system.

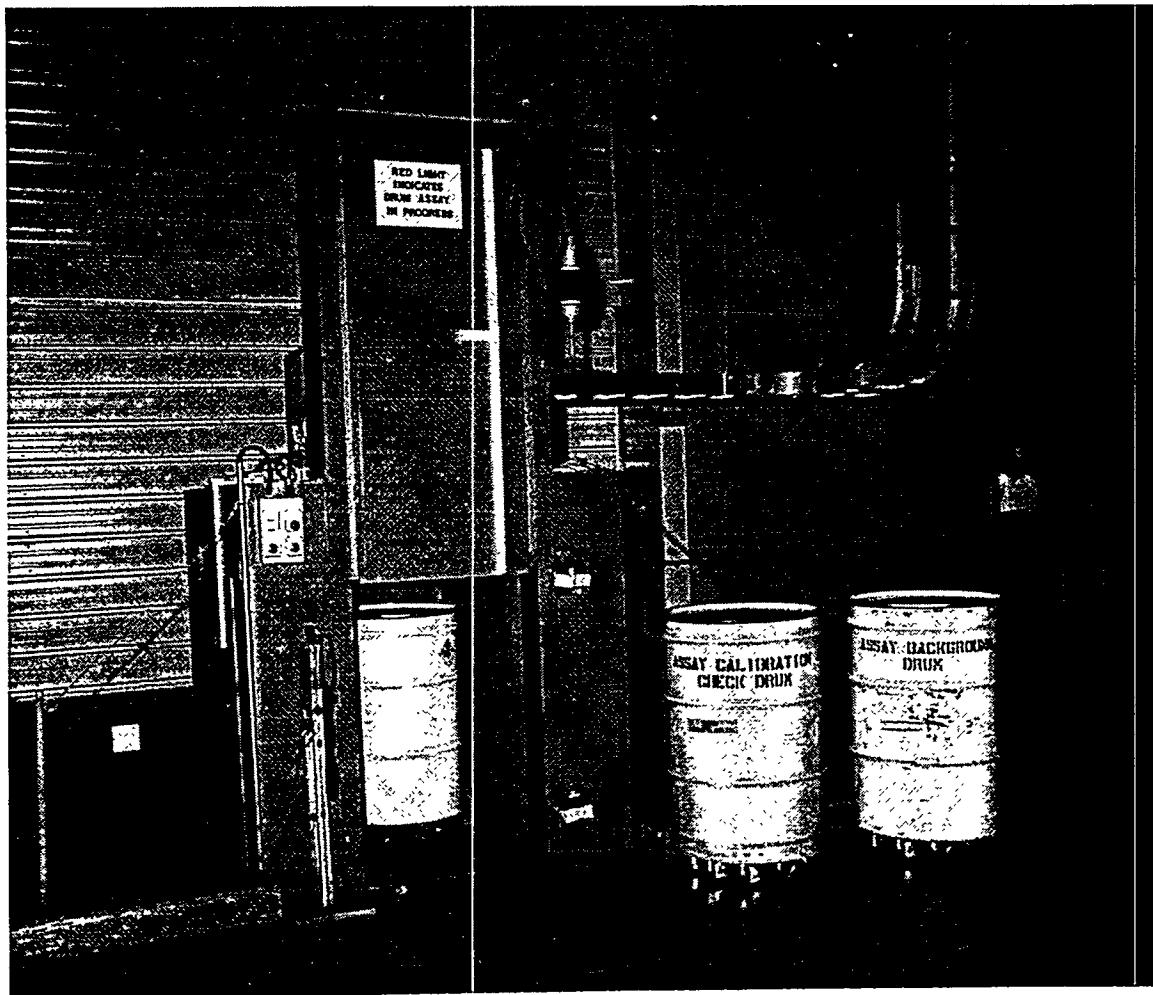


Figure 1. Drum assay chamber.

2.1.2 Nondestructive Measurement Principle

The SAS response to fission neutrons is obtained by passive and active measurements. Passive measurements quantify the spontaneous fission neutron signal of plutonium, which results primarily from ^{240}Pu . Active measurements are based on inducing fissions in fissile material contained in the waste, primarily ^{239}Pu and possibly ^{235}U , via an external neutron source and quantifying the resultant fission neutron signal. The active interrogation neutron source is based on the $^2\text{H}(\text{H}^3, \text{He}_2)\text{n}$ reaction produced by a Zetatron-MA-165 neutron generator mounted inside the detection chamber. The active mode provides more sensitive measurements. However, the fissile material mass cannot be determined by the active measurements for some waste forms because of complications arising from matrix and source interactions. For these waste forms, the WGpu

mass is quantified with the passive measurements of the ^{240}Pu fission neutron signal.² Results from the SGRS are used to confirm that the plutonium isotopic abundances are consistent with those expected from WGPu. SGRS results may also be combined with SAS measurements to quantify ^{241}Am and ^{235}U present in the waste containers. When SGRS data are not present, ^{241}Am content is estimated from the total neutron flux emitted from the waste package in conjunction with the ^{239}Pu or ^{240}Pu mass determined by the SAS.

The drum chamber acquires data from plutonium containing waste through the detectors incorporated into the chamber walls. The mechanism of neutron detection is based on the $^3\text{He}(\text{n},^3\text{H})\text{p}$ reaction. For slow neutrons the reaction Q-value is 764 keV, with 573 keV and 191 keV spatially distributed in opposite directions by the proton and ^3H reaction products, respectively. Within the detectors, the reaction products create gas ions and electrons, which are swept to their respective collecting electrode by an electric field. This produces a measurable signal, which represents a neutron detection event. The $^3\text{He}(\text{n},^3\text{H})\text{p}$ reaction is highest at thermal neutron energies and falls off at higher energies with a relation that is inversely proportional to the neutron energy. Thus, for better detection, the fission neutrons from the source must be moderated in order to maximize the probability of detection through the $^3\text{He}(\text{n},^3\text{H})\text{p}$ reaction. As indicated earlier, bulk quantities of graphite and polyethylene are used to provide the required neutron moderation.

2.1.3 Passive Data Acquisition Technique

Quantifying ^{240}Pu by passive neutron counting depends on the neutron emission characteristics of the ^{240}Pu isotope and all other neutron sources in a given waste container. If the only neutron source in the waste matrix was ^{240}Pu , the mass could be derived by quantifying the total number of neutrons detected and applying a correction factor that relates neutron counts to mass. Unfortunately, there are other neutron sources in the typical waste matrix. These additional or "interfering" neutron sources must be determined in order to calculate the fraction of neutrons that result from ^{240}Pu and thereby relate their detection rate to a mass of WGPu. The most significant interfering neutron source results from reactions between alpha emitting radionuclides, that is, ^{238}Pu , ^{239}Pu , and ^{241}Am , and light elements, such as beryllium, oxygen, boron, and fluorine. The neutrons produced by this reaction are called α, n (α, n) neutrons. The most common technique used to distinguish between ^{240}Pu neutrons and neutrons from other nonfission sources is coincidence counting. The rudiments of this technique are addressed below.

The spontaneous fission of ^{240}Pu results in the simultaneous release of, on the average, 2.16 neutrons per fission. Because fractions of neutrons are not possible, the fission events of significance are either 1, 2, 3, or 4 neutrons per fission, the most probable being 2 neutrons per fission. The fission neutrons are emitted simultaneously, which is valuable in quantifying the ^{240}Pu mass in the presence of an (α, n) neutron source. Because the interfering (α, n) neutrons result from a reaction that generates a single neutron, it is possible to separate the fission neutrons from

² Waste forms for which only passive (or active) assay results should be used must be determined by a detailed evaluation of the neutron moderation, absorption and (α, n) characteristics of the waste form.

the "single" neutrons on a time-correlated basis. Since the desired data are time-correlated, the events coincide in time. Therefore, the data acquisition technique is called coincidence counting.

The coincidence counting technique in the SAS is a simplified version of the updating one-shot method [15]. Briefly, this technique generates an electronic "gate" of a fixed width when the first neutron is detected and coincident neutrons are counted while the gate is open. Since the neutron detectors respond most efficiently to neutrons of slow or thermal energies, the gate width must be carefully determined as a function of the fission neutron energy and the moderating effect of the counting chamber. The coincidence counts include both real and accidental events. The real events result from actual fissions that emit time-correlated neutrons, and the accidental events result from nonfission and background single neutrons. The contribution from the accidental events must be subtracted to determine the true coincidence events from ^{240}Pu fissions. The net coincidence rate is then related to the fissile mass through a chamber calibration factor, as grams of WG Pu per net coincidence counts per second.

2.1.4 Active Data Acquisition Technique

Acquiring data using the active technique also relies on the time-correlated behavior of fission neutrons. Although the same phenomena are measured, the data are acquired somewhat differently. The active measurement is initiated by pulsing the Zetatron neutron generator mounted inside the chamber. The neutron generator produces approximately one million 14 MeV neutrons per pulse. (The neutron generator is typically pulsed 2,000 times at a rate of 50 pulses/second.) A small fraction of the generated neutrons are moderated to thermal neutron energies in the chamber and the waste matrix. This moderation results in an "interrogating" thermal neutron flux that induces some fission reactions in the fissile nuclides within the waste matrix. These fission neutrons can then be detected using the same ^3He detectors used for passive measurements (only the shielded detectors are used for reasons discussed below).

To quantify ^{239}Pu by active neutron counting, a gate is generated at a fixed time after the neutron generator pulse. The gate is applied to only the shielded detector set in order to discriminate against thermal neutrons resulting from the initial 14 MeV neutron pulse. This "early" gate is opened 700 microseconds after the pulse and remains open for 2,000 microseconds. The reason for this timing is twofold. First, the interrogation neutrons must moderate to thermal energies prior to inducing fissions, and second, the interrogation neutrons must also die away in the shielded detectors, which requires the delay before the gate opens. Thus, when the gate is opened, a thermal neutron flux should be present and fissions will be induced in the fissile material. The fission neutrons can be detected while the gate is open.

The neutron events detected during the early gate result from the induced fissions, spontaneous fissions, and (α, n) neutrons in the waste drum. Because the active measurement is calibrated in terms of induced fissions, the spontaneous and (α, n) neutrons are an unwanted background and must be subtracted. This is achieved by generating a second gate, referred to as the "late" gate, 5,700 microseconds after the neutron pulse and remains open for 10,000 microseconds. The induced fission neutrons will have died away by this time, thus events detected while this gate is open result from spontaneous fissions and (α, n) neutrons. The net induced fission rate is obtained by subtracting the background rate, determined by the late gate, from the

gross rate, determined by the early gate. This active net rate is related to fissile mass through a chamber calibration factor.

2.1.5 SAS Response Interpretation

In principle, the passive coincidence counting technique separates real fission coincidence events from accidental events. As indicated, this differentiation is achieved by determining the coincidence rate resulting from real and accidental events and subtracting the rate resulting from accidental events. The procedure works relatively well when these rates are significantly different and if the neutron count rate is less than the inverse of the coincidence gate width. It is therefore important to ensure that of the two passive detection subsystems, the one least affected by these limitations is selected in quantifying the fissile material mass and error.

As previously discussed, the two types of neutron detectors have their own time, energy, and efficiency response functions. The counts from the six banks of shielded detectors are summed together to give a shielded coincidence rate. The counts from the bare detectors are summed with the shielded total count to yield a "system" coincidence rate. Because the neutronic characteristics of the system and shielded detectors are significantly different, one or the other will provide more accurate measurement results and less uncertainty than the other, based on the neutron count rate and matrix and source configuration.

The coincidence gate widths for the system and shielded detectors are predetermined values based on the neutron die-away time characteristic of the shielded and bare detectors. This die-away time is the time required for the number of fission neutrons to decay to 1/e of their original number. Because the neutron die-away time for the system detectors is much longer than that of the shielded detectors, its fixed coincidence gate length will be correspondingly longer. The system detector gate width is currently set at 250 microseconds and the shielded detector gate width is 35 microseconds. The system detectors are more efficient because they include the bare detectors. However, the shielded detectors respond better than the bare detectors when neutron emission rates are high and when significant (α, n) interference neutrons are present. Therefore, based on various response parameter criteria, either the system or shielded response can be selected to determine the fissile mass in a given container.

An additional consideration for interpretation of the SAS response is the effects of the matrix and source configuration in the waste container. Several effects that can affect the passive measurements are neutron moderation and absorption. To account for moderation and absorption, initial calibrations were performed with a variety of matrixes, and functional dependencies were derived for the assay system response. These effects are measured by the shielded and system detector responses and the internal cavity (chamber) and barrel (container) flux monitors. Thus, the dependencies established between the waste matrix and the various detector responses allow corrections for neutron detection bias in the system.

A similar calibration process was used for the active mode. In this case, the internal cavity and barrel flux monitor responses are required because they yield data regarding the interrogation flux. Functional dependencies based on the ratio of the cavity to barrel flux monitors have been established through calibrations to account for interrogation flux absorption and moderation. This

ratio and the passive moderation dependency are used to correct the active mode shielded detector response.

The SAS is currently configured to acquire and reduce measurement data in response to WGPU and to determine plutonium mass and associated derived characteristics. These derived characteristics include thermal power and thermal power density, various activities and fissile gram equivalent (FGE) mass values. For source material with an isotopic composition significantly different from nominal WGPU, adjustments are required in both the data acquisition and the data reduction algorithm. The SGRS can be used to determine plutonium isotopic ratios, as well as the amounts of ^{235}U and ^{241}Am relative to ^{239}Pu in a waste drum. SGRS measurement results can therefore be used to correct and extend measurement results obtained from the SAS.

2.2 Product Functions

This section describes the functions required by the SAS software. This software is a functional replacement for the NEUT2 program which was based on the methodology outlined in Reference [1]. The following functions in NEUT2 have been replicated in the SAS software:

- Acquire data from either the drum or crate assay chambers.
- Reduce data, providing estimates of mass, thermal power, thermal power density, total alpha activity, and activity concentration.
- Operate in local and remote modes.
- Interface with the data management system (DMS).

The SAS software provides the following features which were not in the NEUT2 program:

- Corrections for known defects in NEUT2.
- Optional simulations of NEUT2 calculations, including known defects.
- All equation coefficients placed in ASCII files external to source code.
- Chi-square test can be performed to help verify proper hardware operation.
- CAMAC hardware configuration information in ASCII files external to source code.
- Ability to configure an arbitrary number of "assay chambers".
- Ability to modify the list of container types.
- Ability to modify the list of matrix types.
- User logon required; functions available determined by user access level.
- A graphical user interface (GUI).

Version 2.0 of the SAS software provides the following additional functionality:

- Integration of mass ratio results from the SGRS into SAS calculations.
- Reporting of individual plutonium isotopic masses.
- Implementation of ^{235}U corrections to active assay results using SGRS results and reporting ^{235}U mass based on SGRS results.

- Use SGRS results, if available, to determine ^{241}Am mass.
- Ability to reanalyze saved data using new calibration, matrix or SGRS data.
- Calculation of TRU activity, total activity, plutonium equivalent Curies and fissile gram equivalent (FGE).
- Assay results and other information needed during data reviews added to data files.
- Additional “data integrity” checks added to minimize the possibility of performing assays when the system hardware is not properly functioning.

The major elements of the system are the user, the CAMAC crate, the waste containers, and the links to external systems (DMS and SGRS). The user interface is described in Section 2.2.1, the data acquisition and reduction are described in Sections 2.2.2 and 2.2.3. The remote and local modes, including the DMS interface, are described in Sections 2.2.4 and 2.2.5. An interface to the SGRS system has been added in Version 2.0; this interface is described in Section 2.2.6.

2.2.1 User Interface

The SAS software user interface will be a graphical user interface utilizing MS Windows Version 3.1 (or later) operating in “386 Enhanced Mode”. The functions available to each user will depend on the registered access level of that user.

2.2.2 Data Acquisition

The SAS acquires active and passive data from either the drum assay chamber or the crate assay chamber. The system collects raw neutron counts from specified detectors within an assay chamber. For the active data, the system collects counts from the flux monitor, barrel flux monitor, and shielded detectors. For the passive assay, the system collects counts from the flux monitor, barrel flux monitor, the shielded detectors, and bare detectors. Coincidence counts are collected from shielded and system detectors using corresponding short and long coincidence gates.

The events counted by the assay system detectors are acquired using CAMAC technology. Analog signals from the detectors are processed through a preamplifier and an amplifier. These signals are then input to CAMAC scaler modules that record the events. Data are retrieved from the CAMAC modules through the CAMAC dataway (or data bus) under the control of a GPIB module connected to an IBM PC compatible computer. The SGRS computer is interrogated for the presence of isotopic mass ratio information; if this information is present, it is used by SAS to determine individual isotopic masses. Once the data are accumulated into PC memory, data reduction processes are performed. Figure 2 depicts the system components.

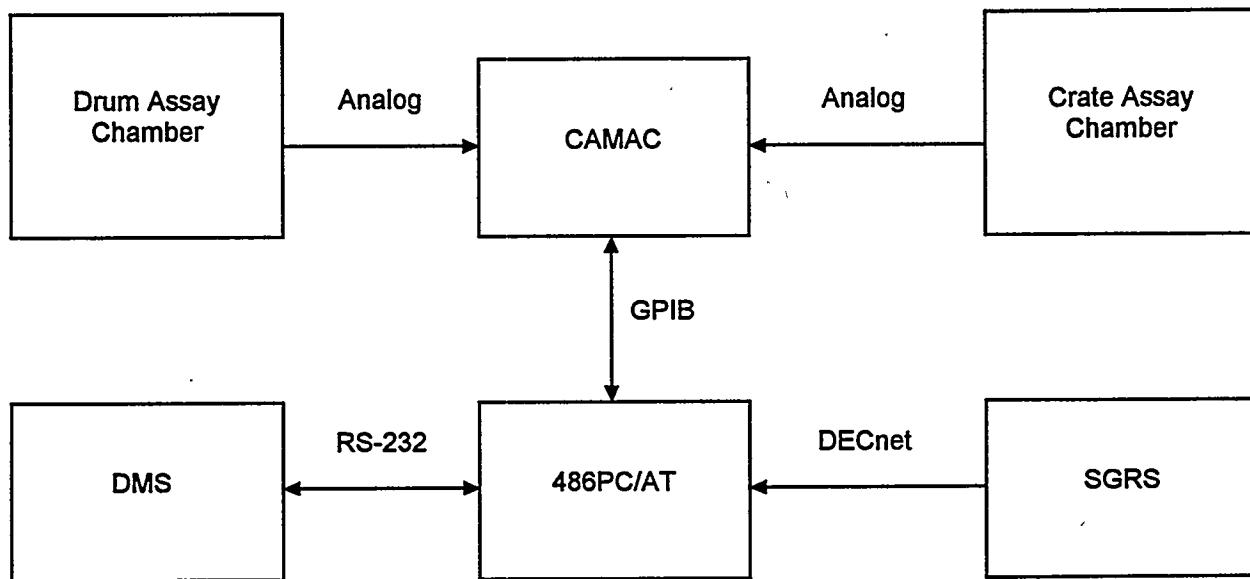


Figure 2. Block diagram of system components.

2.2.3 Data Reduction

The SAS uses the raw counts to "assay" the content, that is, to estimate the masses of plutonium and americium. These mass estimates are used to compute values for thermal power, thermal power density, total alpha activity, activity concentration, etc.

2.2.4 Remote and Local Mode

The SAS software can be operated independently of the DMS computer. This "local mode" is used for passive background data acquisition and calibration of the system. SAS is normally operated in remote mode, receiving processing requests from the DMS computer.

2.2.5 DMS Interface

The data management system (DMS) is a VAX minicomputer running the Ultrix operating system. An operator uses a terminal at the assay station to select a processing function and enter requested data.

The DMS computer communicates with the assay system PC via an RS-232 connection. The DMS computer sends requests to SAS either (a) to compute an assay on newly acquired data or (b) recompute an assay from previously acquired data. SAS acknowledges receipt of the request, performs the requested function, and sends the results to DMS.

2.2.6 SGRS Interface

The SWEPP Gamma-Ray Spectroscopy System (SGRS) incorporates a VAX workstation running the VMS operating system for data acquisition, analysis and storage. This system is operated independently from the SAS. A DECnet link (using Ethernet as the transport mechanism) between the SGRS computer and SAS PC and special software [16] allows the PC to access a directory on the VAX disk (VMS logical device name PAN\$RATIOS) as if it were a disk attached to the PC. After the SGRS completes its data analysis on a waste container, it stores the measured mass ratio results along with other parameters of interest for data verification and validation in a file on the PAN\$RATIOS logical device which can then be read by the SAS computer. The file name is the six character barcode of the waste container with the extension .GAM. Multiple SGRS analyses of the same drum are stored in files with the same name but with increasing version number designations. Since the PC operating system has no file version capability, only the most recent file version is made accessible to the PC.

2.3 User Characteristics

SAS has three levels of users: (1) Operator, (2) Physicist, and (3) Administrator. An Operator can perform normal assay operations. A Physicist can also perform all Operator functions. In addition, a Physicist can modify calibration data for each assay chamber, define/change attributes for the containers and matrix types that can be processed, and reanalyze stored data using new calibration, matrix, SGRS or background data. The Administrator can add and delete SAS users and print/purge log files, but cannot perform any Operator or Physicist functions.

The data tables and/or files that can be modified by users with Administrator or Physicist access levels will be under configuration management and special RWMC procedures must be followed before they can be modified to insure system integrity.

2.4 General Constraints

2.4.1 Regulatory Policies

SAS provides the assay results required to verify that the contact-handled waste containers to be shipped to the WIPP satisfy TRUPACT-II Authorized Methods for Payload Control (TRAMPAC) requirements [27].

2.4.2 Hardware Environment

The SAS program must execute in the current hardware environment. An Intel 486 class IBM PC compatible computer with 16 MB of memory running MS DOS V6.0 and MS Windows 3.1 equipped with a 500 MB disk, 3.5 and 5.25 in. floppy disk drives, mouse, VGA color monitor, and LaserJet III compatible printer is currently being used for data acquisition control and analysis. An RS-232 serial port is required for connection to the DMS computer, and a PathWorks compatible Ethernet interface is required for communications with the SGRS, and a

National Instruments GPIB interface with supporting software [17] is required for communications with the SWEPP CAMAC data acquisition system.

The SAS software should be usable on any 386 or higher class PC with at least 8 MB of memory and capable of supporting MS Windows 3.1. A floppy disk drive and a LaserJet compatible printer are required. A least 10 MB of free disk space should be available for data storage.

The CAMAC [12] crate, modules, and controller currently used at SWEPP for data acquisition and control shall continue to be used. The SAS program must access signal data from the CAMAC crate using the GPIB [13] controller currently available. The SAS software can be operated independently of the CAMAC hardware, that is, without data acquisition capabilities, to reanalyze stored data.

The current configuration of the SWEPP CAMAC crate by station number, module type, and module name is provided in Table 1. References [18] through [22] describe specific capabilities of each module type.

Table 1. CAMAC crate module configuration.

Station	Module type	Module name
01-02	2323	LeCroy 2323A Gate Generator
03-04	2323	LeCroy 2323A Gate Generator
05	2551	LeCroy 2551 Scaler
06	2551	LeCroy 2551 Scaler
07-08	2323	LeCroy 2323A Gate Generator
09-10	2323	LeCroy 2323A Gate Generator
11	2551	LeCroy 2551 Scaler
12	2551	LeCroy 2551 Scaler
13	450	DSP 450 Gated Quad Scaler
14-15	2323	LeCroy 2323A Gate Generator
16-17	2323	LeCroy 2323A Gate Generator
18	2551	LeCroy 2551 Scaler
19	2551	LeCroy 2551 Scaler
20	450	DSP 450 Gated Quad Scaler
21	217	Jorway 217 Gated Clock Generator
22	202	Jorway 202 Dataway Display
23	2551	LeCroy 2551 Scaler
24-25	8901	LeCroy 8901A GPIB Interface

2.5 Assumptions and Dependencies

This section lists miscellaneous assumptions and dependencies that impact the requirements listed in this document.

The INEL TRU Waste Programs Department in conjunction with the RWMC/SWEPP Programs Department will provide resources for acceptance testing of the SAS software. The TRU Waste Programs Department is also responsible for the mathematical models that have been or can be validated with the physical process being modeled. The software developers are not responsible for the validity of the mathematical model.³

³ Development and support of Version 2.0 of the SAS software has been taken over by the INEL Nuclear Physics and Radiation Chemistry Department. Personnel from this department are responsible for supplying the TRU Waste Programs Department with recommendations for SAS system improvements, including the mathematical algorithms used for data analysis.

Left blank for pagination purposes.

3. SPECIFIC REQUIREMENTS

3.1 Functional Requirements

This section describes the detailed functional requirements for the SAS software. It includes the user, the data acquisition, the data reduction, and the external interface requirements.

The requirements were initially derived using object-oriented analysis. Thus, the requirements for each of these areas are based on the objects identified for that set of functions in the SAS program. The user requirements subsection contains the requirements for the user object. The active signals, assay system, CAMAC crate, chamber, and passive signals objects constitute the data acquisition requirements. The matrix object contains the data reduction requirements. Finally, the DMS object encapsulates the external interface requirements. The SGRS interface, added in Version 2.0, is not encapsulated within an object. This interface is provided transparently to the SAS software through the DOS file system and PathWorks software. (As far as the SAS program is concerned, it simply reads the SGRS data from a file.) Incorporation of SGRS data into the SAS data analysis is accomplished within the matrix object.

Section 3.1.1 provides diagrams for the objects and the data reduction process as a part of the software model. Subsequent sections list the object attributes (variables) and the object methods (functions). Where appropriate, state transition diagrams are provided within an object specification.

3.1.1 Software Model

This section shows the software requirements model through a series of diagrams. The first view is by objects, showing the relationship of the objects in the system. The second is by function, using a data flow diagram to show the data reduction process.

3.1.1.1 Object Model

Provided in the following sections are the functional requirements for the SAS software. The object models shown in Figures 3 through 5 define the high level SAS requirements and provide a model of the relationships within the SAS software. The notation used is from Rumbaugh [10]. Some objects shown are for informational purposes only and do not contain any functional requirements.

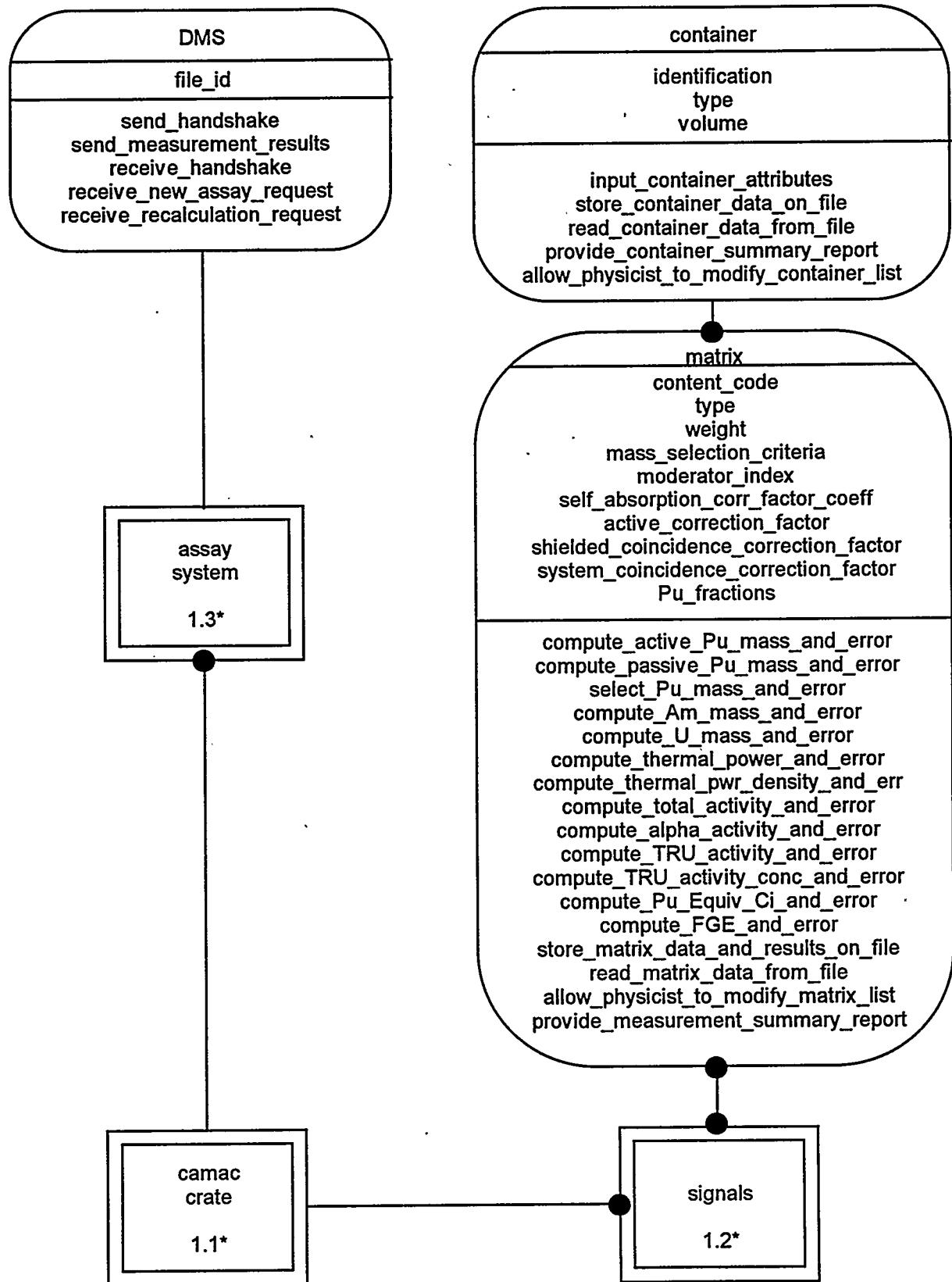


Figure 3. Assay system object model, level 1.

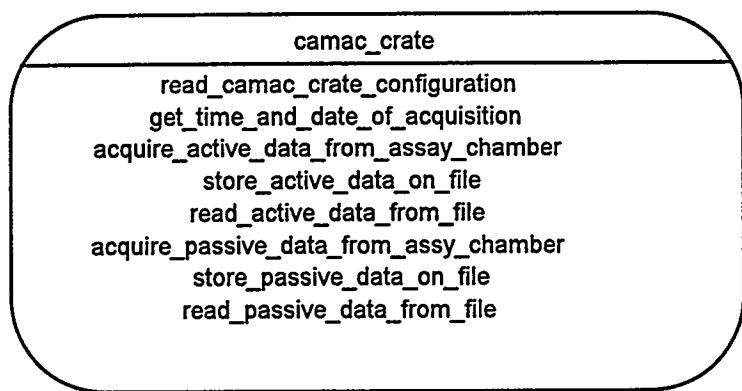


Figure 4. Assay system object model, level 1.1.

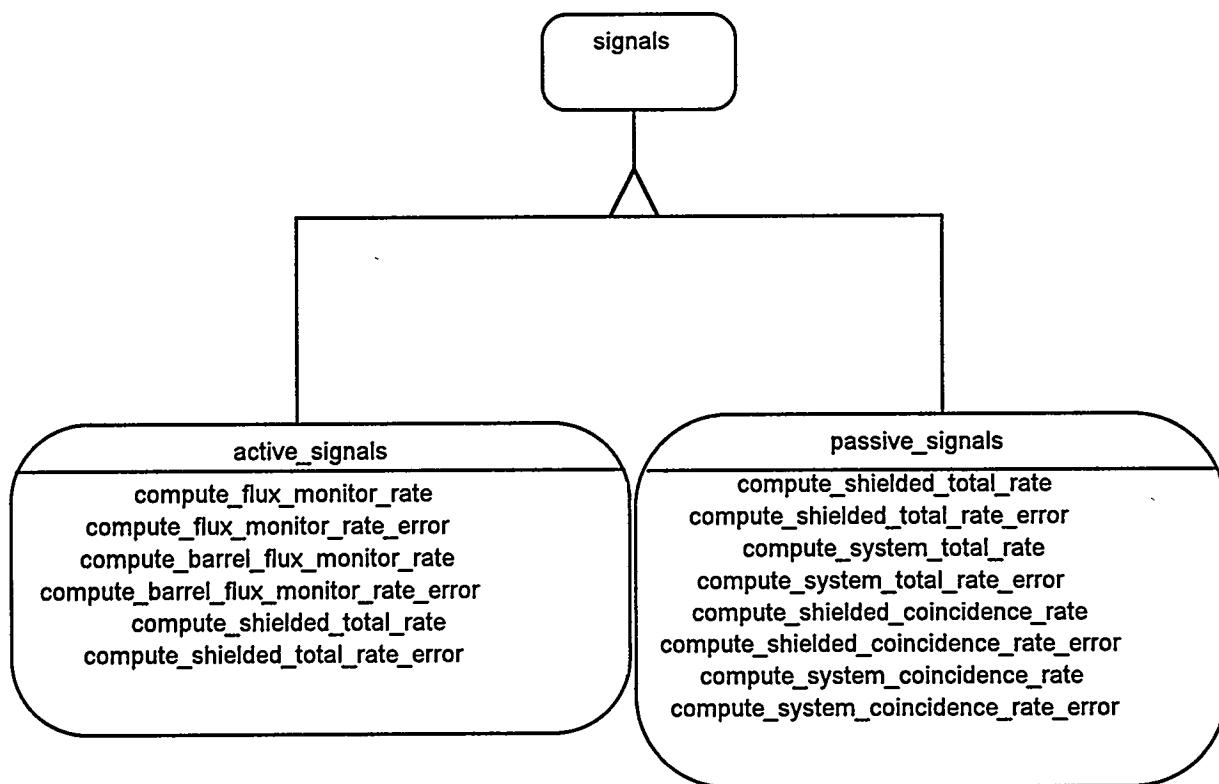


Figure 5. Assay system object model, level 1.2.

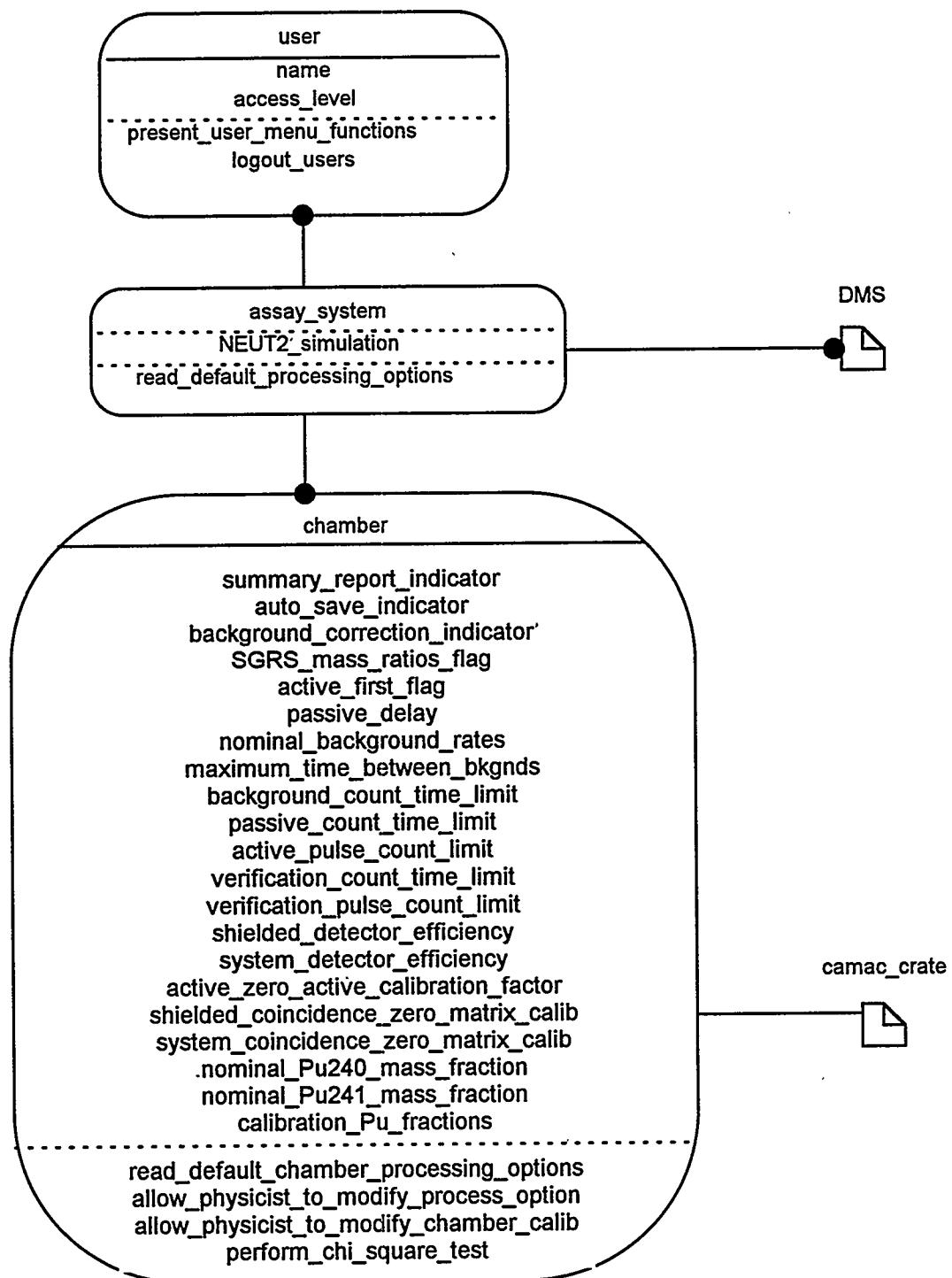


Figure 6. Assay system object model, level 1.3.

3.1.1.2 Functional Model

The data reduction process is best modeled by a traditional structured analysis or data flow model depicting the processing steps or functions. Figures 7 through 12 show the functional model for the data reduction.

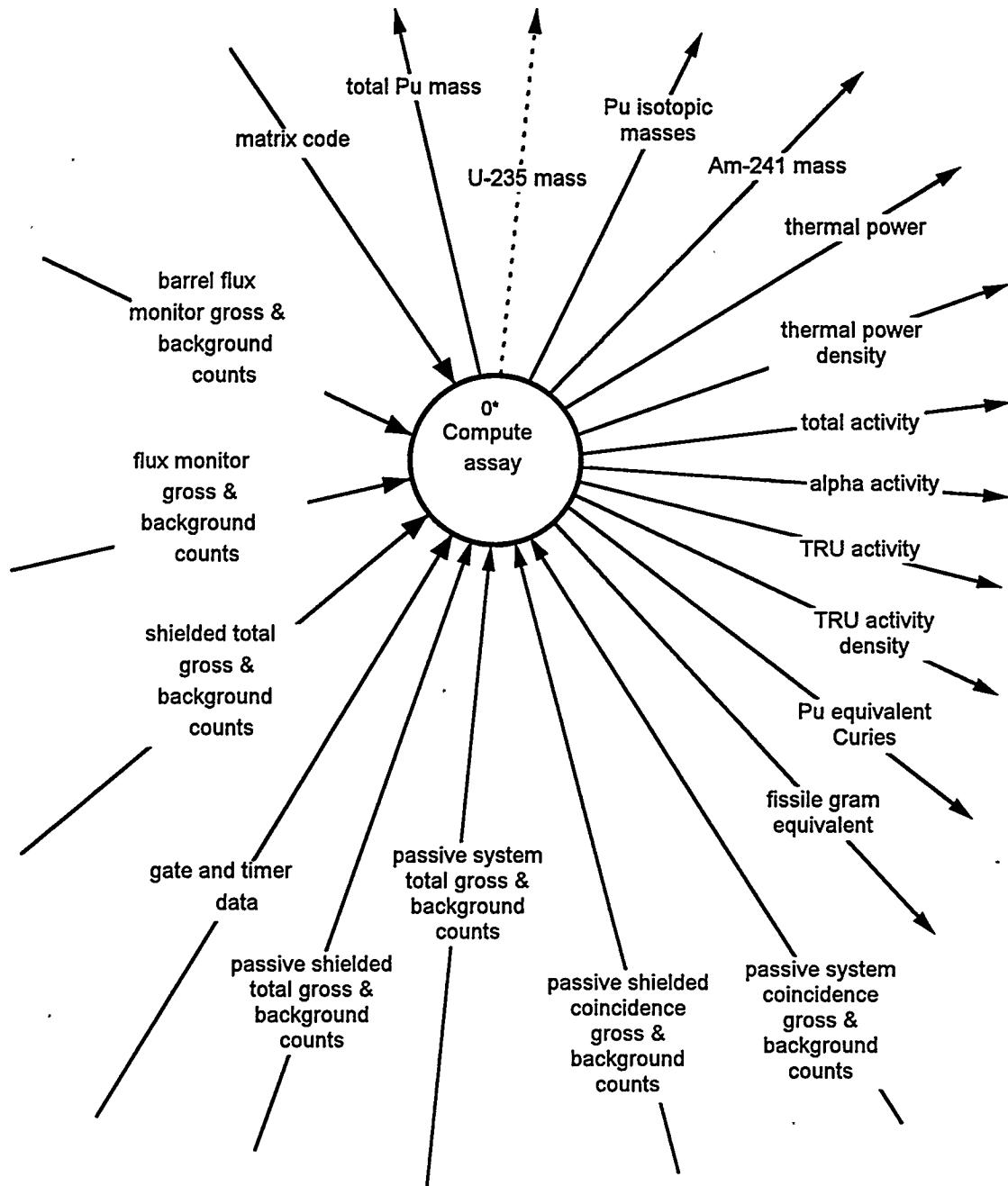


Figure 7. Assay system data flow diagram, context level.

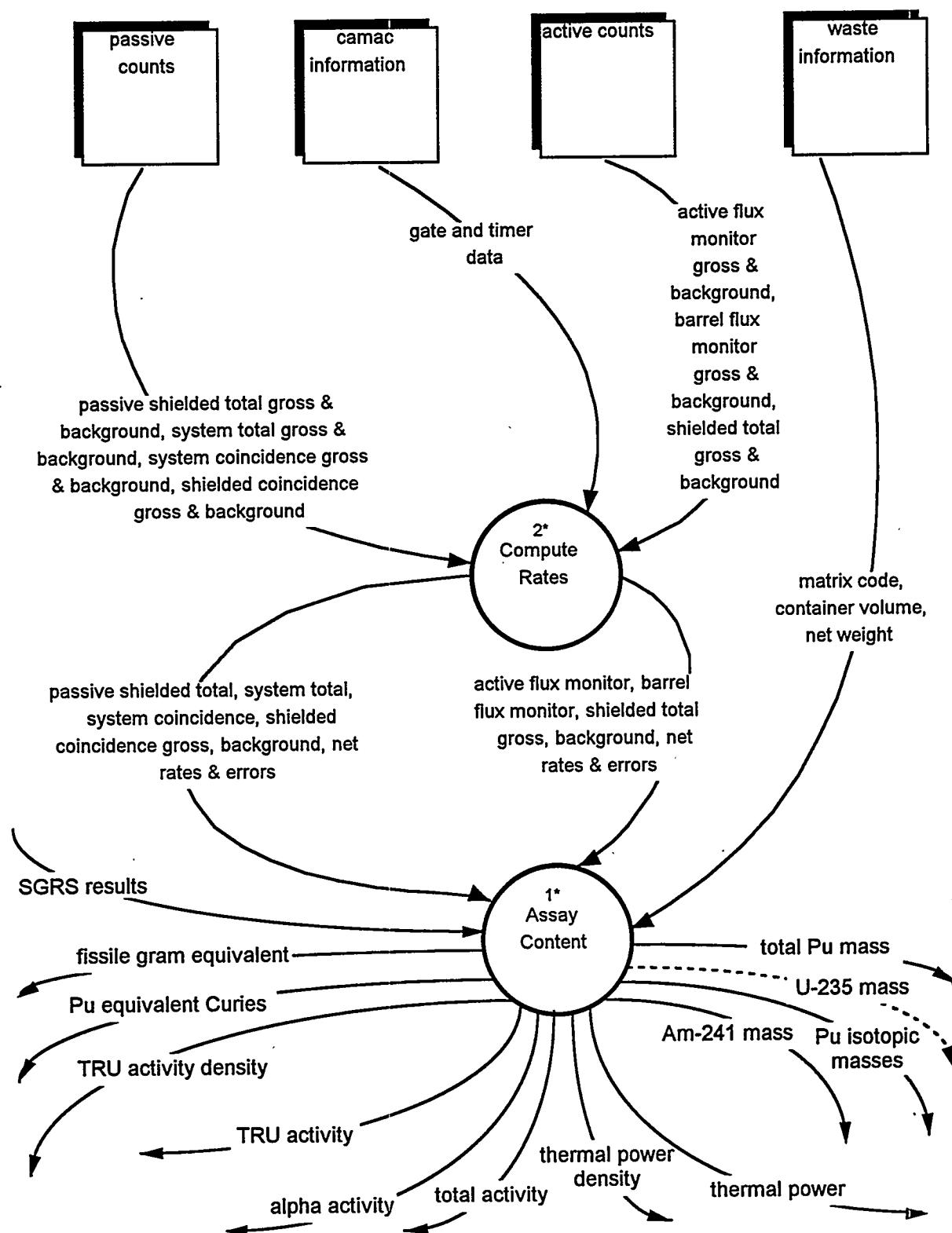


Figure 8. Assay system data flow diagram, level 0.

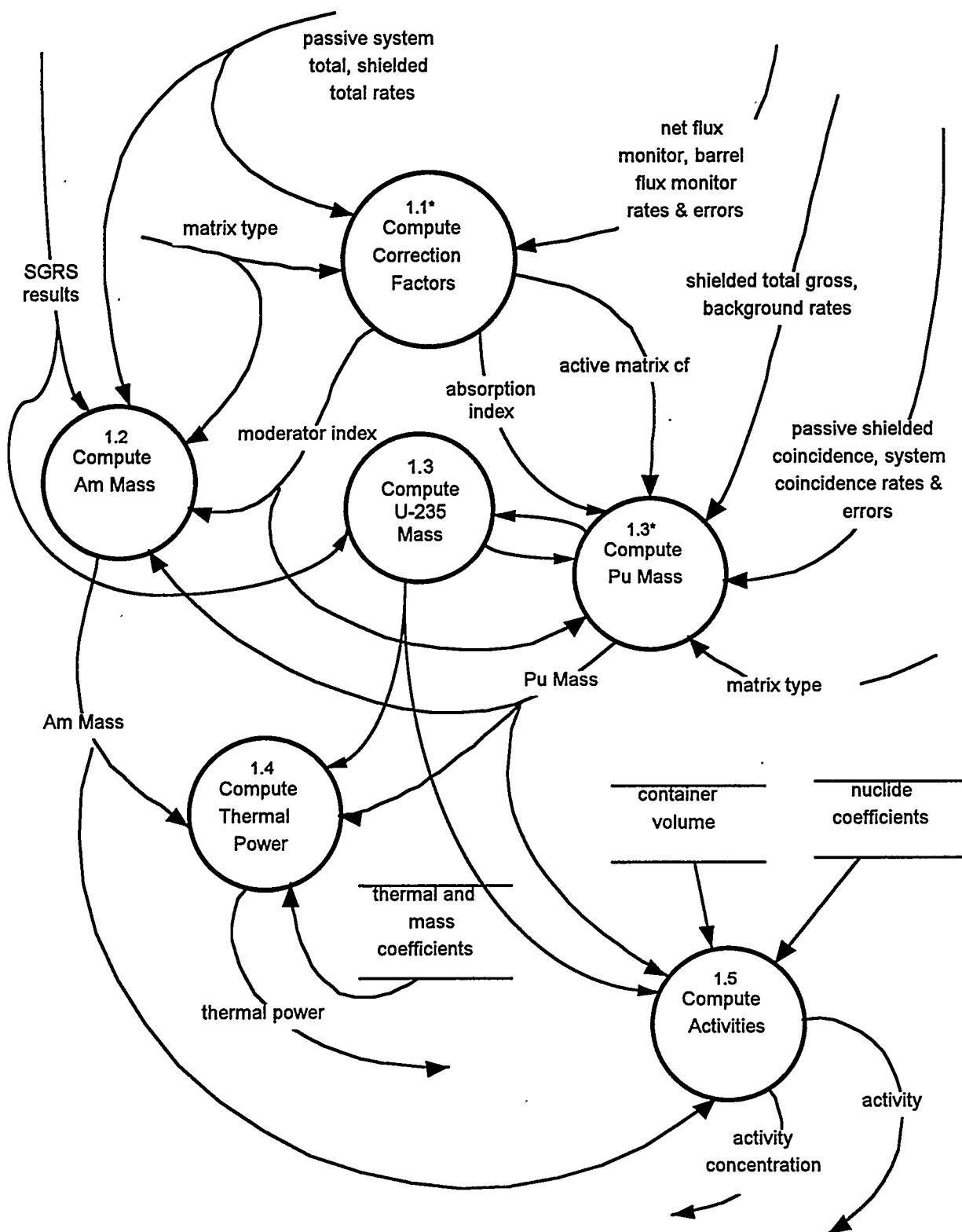


Figure 9. Assay system data flow diagram, level 1.

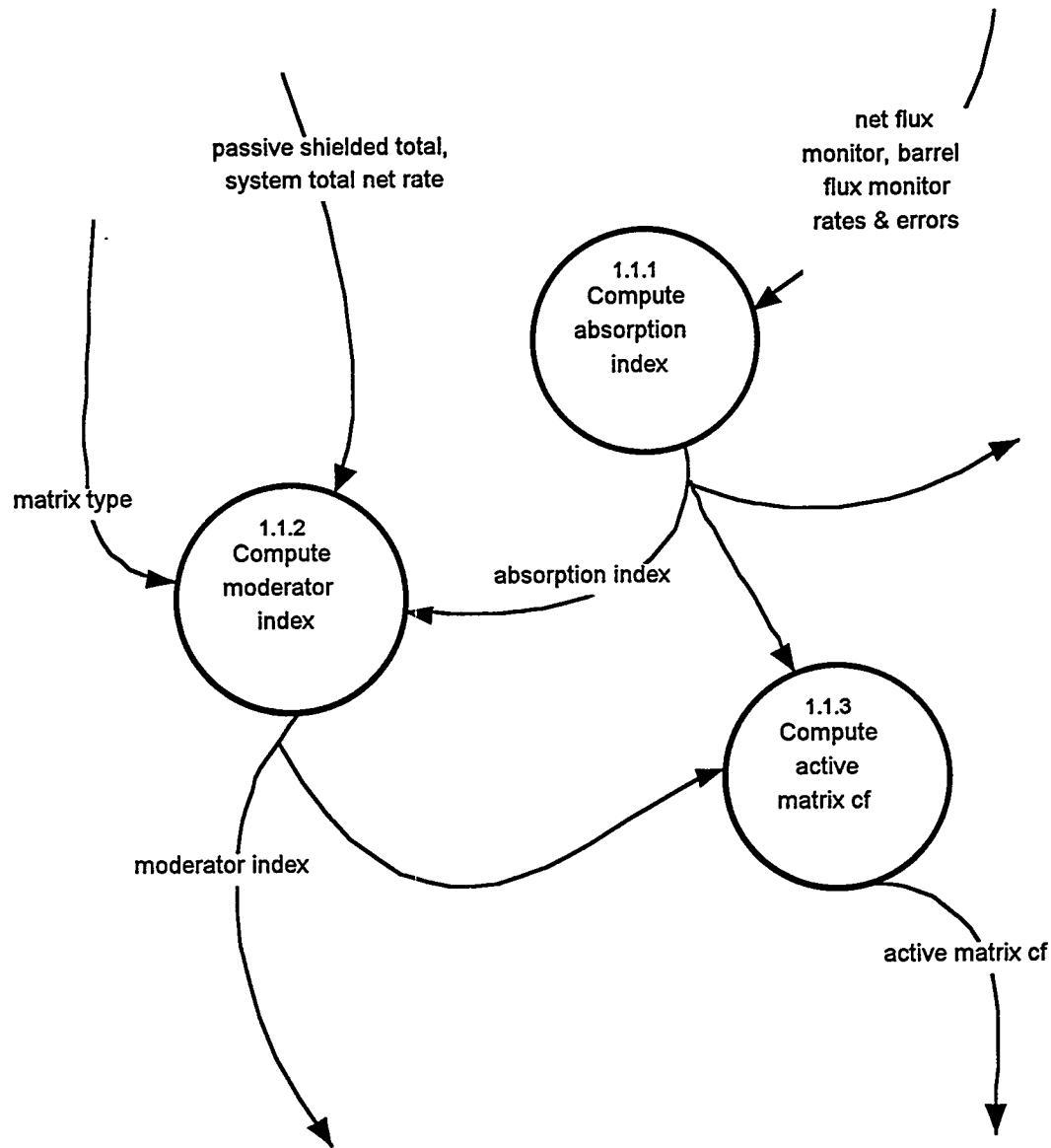


Figure 10. Assay system data flow diagram, level 1.1.

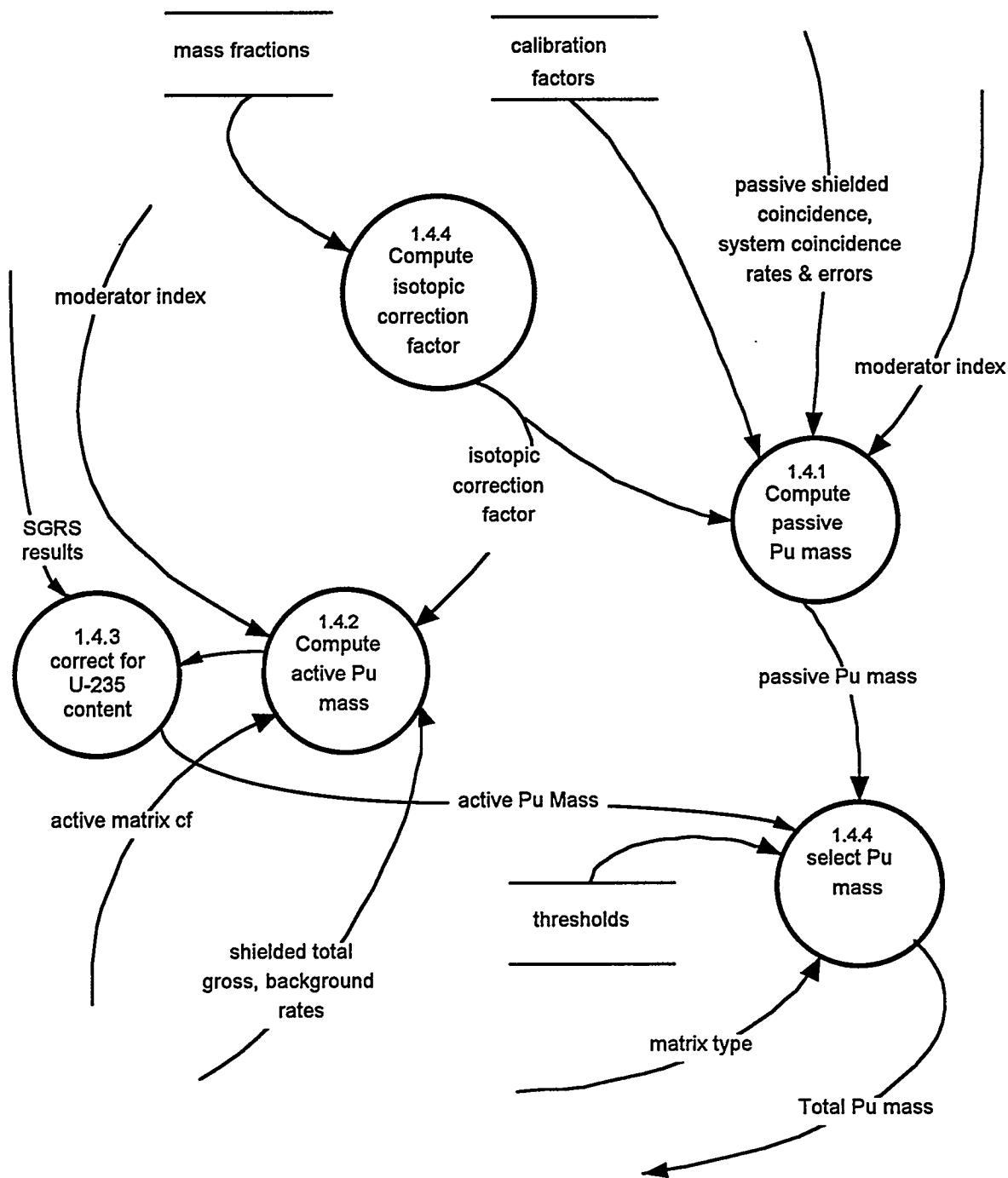


Figure 11. Assay system data flow diagram, level 1.3.

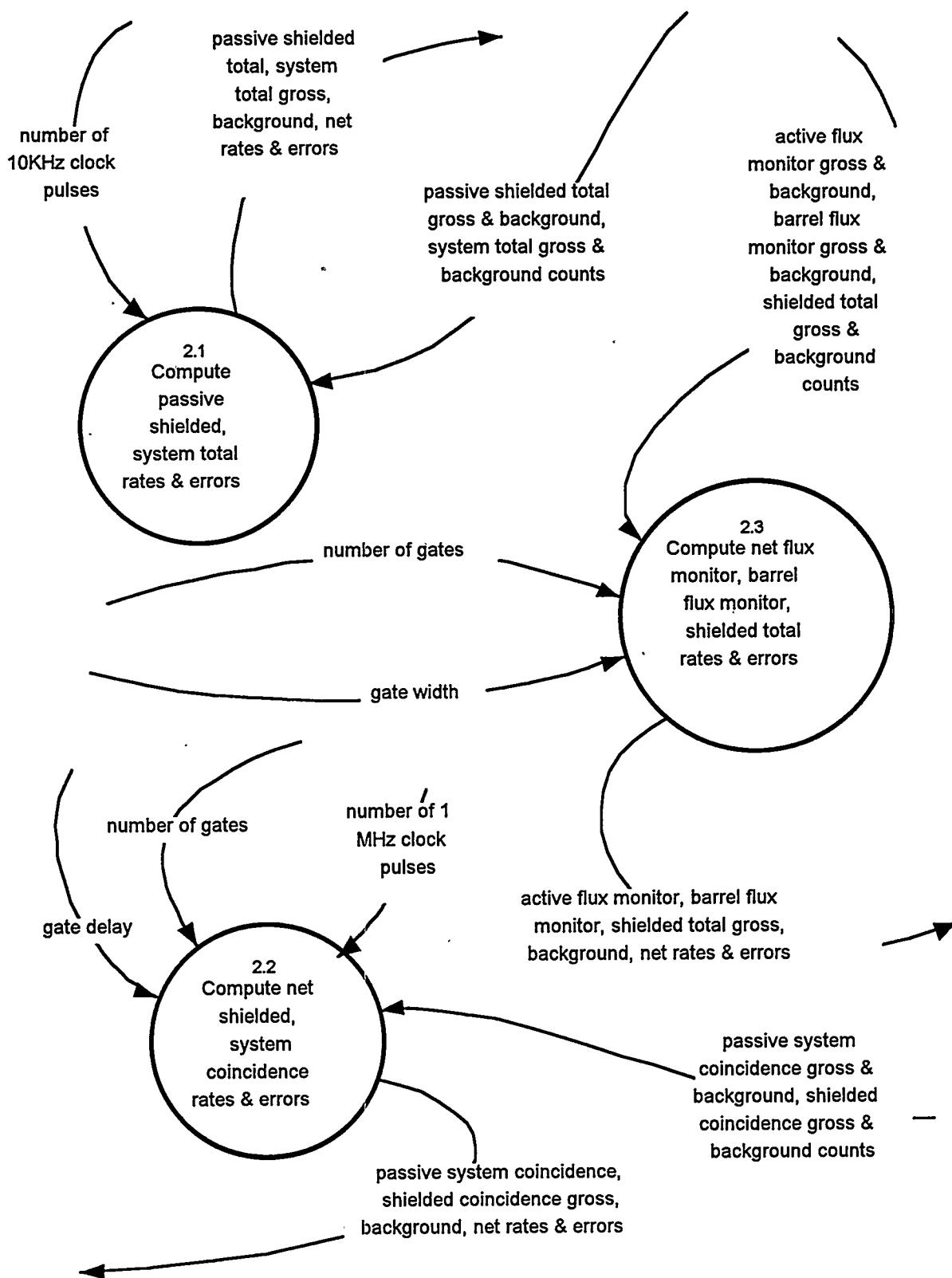


Figure 12. Assay system data flow diagram, level 2.

3.1.2 User Functional Requirements

This section provides the functional requirements for the user interface to the SAS.

3.1.2.1 State Transition Diagrams

The state transitions for the user model are shown in Figure 13.

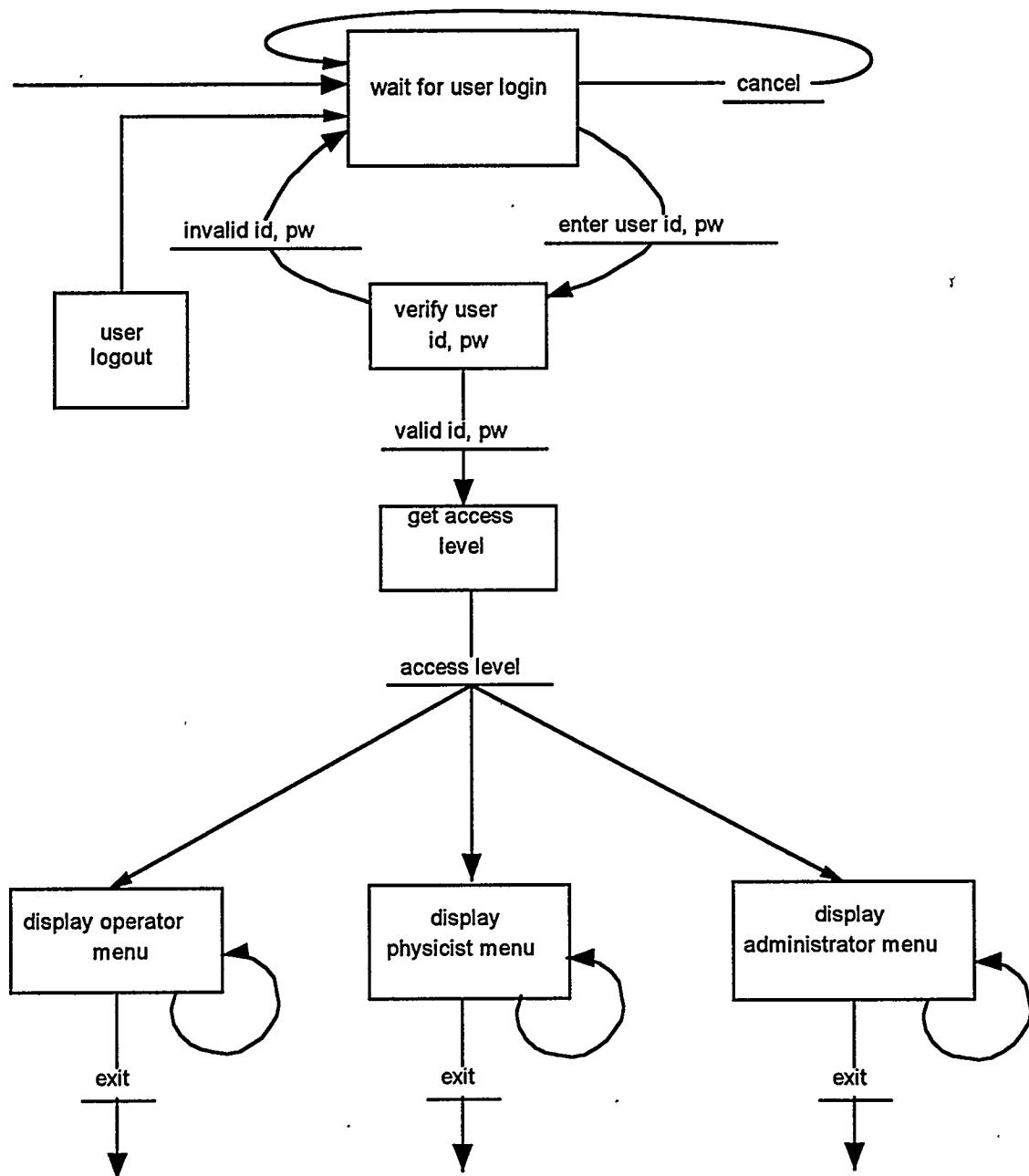


Figure 13. User state transition diagram.

3.1.2.2 User Requirements

A. User Attributes

- name
- access level {Operator, Physicist, Administrator}

M. User Methods

M.1 Login users

Login to the SAS system for any user will require both a user ID and a password. The password is not a secure password, and is a device for administrative control (user identification) only. The assay system PC itself is not password protected and is not a "secure" system. Physical access to the system is controlled by the access control procedures in effect at the RWMC/SWEPP facility.

Input

- user ID
- password

Process

The user starts the SAS program in Microsoft Windows by double clicking on the SAS icon. The assay system starts up and provides a login dialog for the user to enter a user ID and password. The system validates the login request against the assay system user list.

Output

- access level

M.2 Present user menu functions

Input

- access level

Process

- provide menu functions for the access level

Output

Operator menu functions:

- logout
- acquire passive background data
- perform chi-square test
- perform new assay
 - full assay -- active and passive data, data analysis

- acquire active data only
- acquire passive data only
- recalculate assay from previously saved data and parameters
- enter remote communication mode with DMS
- enable or disable automatic summary report
- enable or disable automatic data save⁽¹⁾
- modify processing parameters
- exit

⁽¹⁾ Added in Version 2.0.

Physicist menu functions:

- logout
- acquire passive background data
- perform chi-square test
- perform new assay
 - full assay -- active and passive data, data analysis
 - acquire active data only
 - acquire passive data only
- recalculate assay
 - from previously saved data and parameters
 - from previously saved data and current parameters and/or current background⁽¹⁾
- enter remote communication mode with DMS⁽¹⁾
- enable or disable automatic summary report
- enable or disable automatic data save⁽¹⁾
- enable or disable background correction
- modify processing parameters
- modify chamber calibration parameters
- modify matrix list
- modify container list
- exit

⁽¹⁾ Added in Version 2.0.

Administrator menu functions:

- logout
- modify user list
- print/purge user and error logs
- exit

M.3 Logout users

Input

- logout menu selection

Process

Log the user out of the assay system. Provide an initial menu to permit another user to login.

Output

Initial menu:

- login
- exit

M.4 Allow Administrator to modify the user list

Input

- user ID
- user name
- password
- access level

Process

Set new values.

Output

- user ID
- user name
- password
- access level

3.1.3 Data Acquisition Functional Requirements

This section specifies the data variables, i.e., attributes, and the functional requirements, i.e., methods, required for acquiring the assay data from the containers.

3.1.3.1 State Transition Diagram

The state transitions for the data acquisition are shown in Figure 14.

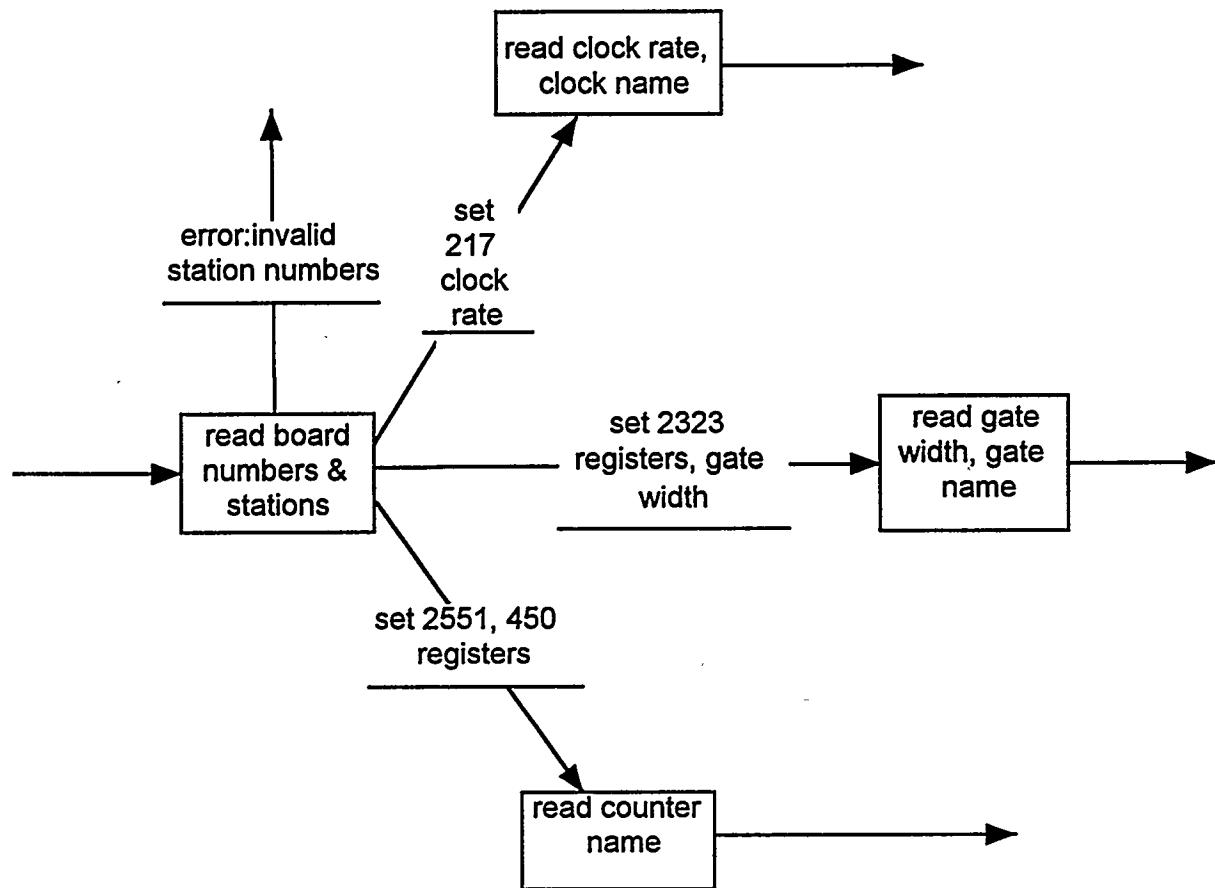


Figure 14. Data acquisition state transition diagram.

3.1.3.2 Assay System Requirements

A. Assay System Attributes

- NEUT2 simulation indicator {ON, OFF}
- location of system root directory
- file and device names
- assay chamber names

M. Assay System Methods

M.1 Read default processing options for assay system

Input

- NEUT2 simulation indicator {ON, OFF}
- location of system root directory

- file and device names
- assay chamber names

Process

Read the attribute values from file and retain in memory.

Output

None.

3.1.3.3 Assay Chamber Requirements

A. Chamber Attributes

- summary report indicator {ON, OFF}
- automatic save indicator {ON, OFF}⁽¹⁾
- background correction indicator {ON, OFF}
- use SGRS Pu mass ratios {YES, NO}⁽¹⁾
- perform active before passive {YES, NO}⁽¹⁾
- delay time before performing passive after active⁽¹⁾
- nominal shielded coincidence background rate⁽¹⁾
- nominal system coincidence background rate⁽¹⁾
- maximum time between background runs⁽¹⁾
- background count time limit
- passive count time limit
- active pulse count limit
- verification count time limit
- verification pulse count limit
- shielded detector efficiency
- system detector efficiency
- active zero active calibration factor
- shielded coincidence zero matrix calibration factor
- system coincidence zero matrix calibration factor
- nominal Pu-240 fraction⁽¹⁾
- nominal Pu-241 fraction⁽¹⁾
- calibration fraction Pu-238
- calibration fraction Pu-239
- calibration fraction Pu-240
- calibration fraction Pu-241
- calibration fraction Pu-242

⁽¹⁾ Added in Version 2.0.

M. Chamber Methods

M.1 Read default chamber processing options

Input

Table 2 provides the default input values for each of the initially configured SAS assay chambers. In the current INEL operations, only the Drum Chamber is used and the Crate Chamber configuration shall be disabled. However, it must be possible to enable the Crate Chamber at a later time without making any modifications to the program's source code.

Table 2. Default chamber attribute values.

Attribute	Drum Chamber	Crate Chamber
Summary report	ON	ON
Automatic save ⁽¹⁾	OFF	OFF
Background correction	ON	ON
Use SGRS mass ratios ⁽¹⁾	YES	YES
Perform active before passive ⁽¹⁾	YES	YES
Passive delay time (seconds) ⁽¹⁾	15	15
Nominal shielded coincidence background rate ⁽¹⁾	(3)	(3)
Nominal system coincidence background rate ⁽¹⁾	(3)	(3)
Maximum time between background runs ⁽¹⁾	(3)	(3)
Background count time limit (seconds) ⁽²⁾	600	600
Passive count time limit (seconds)	200	200
Active pulse count limit	2000	5000
Verification count time limit (seconds)	200	200
Verification pulse count limit	2000	2000
Shielded detector efficiency	0.029	0.029
System detector efficiency	0.1265	0.1265
Active zero matrix calibration factor	4.35	18.5
Shielded coincidence zero matrix calibration factor	28.1	84.0
System coincidence zero matrix calibration factor	1.28	2.92
Nominal Pu-240 fraction	(3)	(3)
Nominal Pu-241 fraction	(3)	(3)
Calibration fraction Pu-238	0.00012	0.00012
Calibration fraction Pu-239	0.93826	0.93826
Calibration fraction Pu-240	0.0582	0.0582
Calibration fraction Pu-241	0.0034	0.0034
Calibration fraction Pu-242	0.00024	0.00024

⁽¹⁾ Added in Version 2.0.

⁽²⁾ Changed in Version 2.0.

⁽³⁾ Checks using this parameter disabled if no value supplied.

Process

Read the values for the attributes from file and retain in memory.

Output

None.

M.2 Allow an Operator or Physicist to modify the processing options

Input

- summary report indicator
- automatic data save indicator
- background correction indicator⁽¹⁾
- background count time limit
- passive count time limit
- active pulse count limit
- verification count time limit
- verification pulse count limit

⁽¹⁾ Can only be changed by users with Physicist access level.

Process

Set new values.

Output

- summary report indicator
- automatic data save indicator
- background correction indicator
- background count time limit
- passive count time limit
- active pulse count limit
- verification count time limit
- verification pulse count limit

M.3 Allow Physicist to modify the chamber calibration

Input

- shielded detector efficiency
- system detector efficiency
- active zero matrix calibration factor
- shielded coincidence zero matrix calibration factor
- system coincidence zero matrix calibration factor
- calibration fraction Pu-238

- calibration fraction Pu-239
- calibration fraction Pu-240
- calibration fraction Pu-241
- calibration fraction Pu-242

Process

Set new values.

Output

- shielded detector efficiency
- system detector efficiency
- active zero matrix calibration factor
- shielded coincidence zero matrix calibration factor
- system coincidence zero matrix calibration factor
- calibration fraction Pu-238
- calibration fraction Pu-239
- calibration fraction Pu-240
- calibration fraction Pu-241
- calibration fraction Pu-242

M.4 Perform chi-square test

Input

Table 3 provides the input chi-square parameter values for each assay chamber. It must be possible to modify these values without modifying the program code.⁴

Table 3. Default Chi-square parameters.

Parameter	Drum Chamber	Crate Chamber
Sample size	20	20
Passive count time limit (seconds)	10	10
Confidence interval	0.99	0.99
Minimum acceptance value	6.844	6.844
Maximum acceptance value	38.582	38.582

⁴ Appropriate configuration management controls must be followed, however.

Process

1. Read Chi-square processing parameter from file.
2. Perform sample-size consecutive passive data acquisitions with the specified passive count time limit.
3. Compute the chi-square statistics for the shielded total:

$$average = \frac{\sum_{i=1}^{sample_size} shielded_total_i}{sample_size}$$

$$shielded_total_chi_square = \frac{\sum_{i=1}^{sample_size} (average - shielded_total_i)^2}{average}$$

$$shielded_total_acceptance = \\ shielded_total_chi_square \in (minimum_acceptance, maximum_acceptance)$$

4. Compute the chi-square statistic for the system total.

$$average = \frac{\sum_{i=1}^{sample_size} system_total_i}{sample_size}$$

$$system_total_chi_square = \frac{\sum_{i=1}^{sample_size} (average - system_total_i)^2}{average}$$

$$system_total_acceptance = \\ system_total_chi_square \in (minimum_acceptance, maximum_acceptance)$$

Output

- shielded total chi-square
- system total chi-square
- shielded total acceptance
- system total acceptance

3.1.3.4 CAMAC Crate Requirements**A. CAMAC Crate Attributes**

The CAMAC crate is identified by a number between 0 and 31 on the GPIB bus. The AT-GPIB controller is configured by default to recognize only the first 16 devices, with 0 reserved for the controller itself.

M. CAMAC Crate Methods**M.1 Read CAMAC crate configuration from file***Input*

Tables 4 through 7 list the CAMAC port, or channel, configuration at the installation of Version 1.0 of the SAS (this configuration remains unchanged for Version 2.0). It must be possible to establish other configurations without modifying the program code.

Table 4. CAMAC port configuration, drum chamber, active acquisition.

Module		Port		
Station	Type	Num.	Name	Value
01	2323	0	Early Gate Delay	700 μ s
		1	Early Gate Width	2,000 μ s
03	2323	0	Late Gate Delay	5,700 μ s
		1	Late Gate Width	10,000 μ s
05	2551	0	Early Gate Shielded Total	
		1	Early Gate Flux Monitor	
		2	Early Gate Barrel Flux Monitor	
06	2551	0	Late Gate Shielded Total	
		1	Late Gate Flux Monitor	
		2	Late Gate Barrel Flux Monitor	
11	2551	6	Active Pulse	

Table 5. CAMAC port configuration, drum chamber, passive acquisition.

Module		Port		
Station	Type	Num.	Name	Value
07	2323	0	Short Gate Delay	6 μ s
		1	Short Gate Width	35 μ s
09	2323	0	Long Gate Delay	3 μ s
		1	Long Gate Width	250 μ s
11	2551	0	Shielded Total	
		1	System Total	
		2	Long Gate	
		3	Short Gate	
		4	10 kHz Clock	
		5	1 kHz Clock	
		10	Flux Monitor	
		11	Barrel Flux Monitor	
		0	Bare Door	
		1	Bare Right	
		2	Bare Back	
12	2551	3	Bare Left	
		4	Bare Top	
		5	Bare Bottom	
		6	Shielded Door	
		7	Shielded Right	
		8	Shielded Back	
		9	Shielded Left	
		10	Shielded Top	
		11	Shielded Bottom	
13	450	0	Short Gate Shielded Total	
		1	Short Gate 1 MHz Clock	
		2	Long Gate System Total	
		3	Long Gate 1 MHz Clock	
21	217	1	1 MHz Clock Rate	1,000,000/s
		3	10 kHz Clock Rate	10,000/s
		4	1 kHz Clock Rate	1,000/s

Table 6. CAMAC port configuration, crate chamber, passive acquisition.

Module		Port		
Station	Type	Num.	Name	Value
14	2323	0	Short Gate Delay	6 μ s
		1	Short Gate Width	35 μ s
16	2323	0	Long Gate Delay	3 μ s
		1	Long Gate Width	250 μ s
18	2551	0	Shielded Total	
		1	System Total	
		2	Long Gate	
		3	Short Gate	
		4	10 kHz Clock	
		5	1 kHz Clock	
		10	Flux Monitor	
		11	Barrel Flux Monitor	
		0	Bare Door	
		1	Bare Right	
		2	Bare Back	
		3	Bare Left	
19	2551	4	Bare Top	
		5	Bare Bottom	
		6	Shielded Door	
		7	Shielded Right	
		8	Shielded Back	
		9	Shielded Left	
		10	Shielded Top	
		11	Shielded Bottom	
20	450	0	Short Gate Shielded Total	
		1	Short Gate 1 MHz Clock	
		2	Long Gate System Total	
		3	Long Gate 1 MHz Clock	
21	217	1	1 MHz Clock Rate	1,000,000/s
		3	10 kHz Clock Rate	10,000/s
		4	1 kHz Clock Rate	1,000/s

Table 7. CAMAC port configuration, crate chamber, active acquisition.

Module		Port		
Station	Type	Num.	Name	Value
01	2323	0	Early Gate Delay	700 μ s
		1	Early Gate Width	2,000 μ s
03	2323	0	Late Gate Delay	5,700 μ s
		1	Late Gate Width	10,000 μ s
05	2551	3	Early Gate Shielded Total	
		4	Early Gate Flux Monitor	
		5	Early Gate Barrel Flux Monitor	
06	2551	3	Late Gate Shielded Total	
		4	Late Gate Flux Monitor	
		5	Late Gate Barrel Flux Monitor	
18	2551	6	Active Pulse	

Process

Read the station number, module type, port numbers, and port names for each module in the current hardware configuration as specified in an ASCII file. In addition, read the port values for the gate generators and the clock generators. Separate files shall be used for the passive and active acquisition modules, and for each configured chamber.

Output

- CAMAC module and port configuration stored in memory

M.2 Get time and date of data acquisition***Input***

None.

Process

- get CPU clock time and date of the data acquisition

Output

- time and date of the data acquisition

M.3 Acquire active data from assay chamber

Each pulse from the neutron generator triggers the early gate and the late gate. The count of the pulses, and hence the count for each of the gates, is the active pulse count. The early gate corresponds to the gross counts while the late gate corresponds to the background counts.

Input

- active CAMAC crate configuration (M.1)
- active pulse count limit (3.1.3.3 A)

Process

1. Program the gate generators as specified by the CAMAC configuration.
2. Request the user initiate neutron pulse generator operation.⁵
3. Poll the active pulse scaler port until the active pulse count limit is reached or the user cancels the operation.
4. If active count limit is reached, accumulate counts from the CAMAC scaler ports.

Output

- gross shielded total counts
- gross flux monitor counts
- gross barrel flux monitor counts
- background shielded total counts
- background flux monitor counts
- background barrel flux monitor counts
- active pulses counts

M.4 Store active data on file

Note: These data are only stored if requested by the user after all data have been acquired and analyzed. In Version 2.0, data will be automatically stored at the end of an assay if "automatic data save" is enabled.

Input

- time and date of the data acquisition
- CAMAC crate configuration for active signals (M.1)
- gross shielded total counts
- gross flux monitor counts
- gross barrel flux monitor counts
- background shielded total counts
- background flux monitor counts
- background barrel flux monitor counts
- active pulses counts

Process

Write inputs to a file.

⁵ Current INEL safety requirements prohibit automatic initiation of the neutron generator operation.

Output

A file containing:

- time and date of the data acquisition
- CAMAC crate configuration for active signals (M.1)
- gross shielded total counts
- gross flux monitor counts
- gross barrel flux monitor counts
- background shielded total counts
- background flux monitor counts
- background barrel flux monitor counts
- active pulses counts

M.5 Read active data from file

During a recalculation, data are obtained from a file previously saved rather than from the CAMAC crate.

Input

- time and date of the data acquisition
- CAMAC crate configuration for active signals (M.1)
- gross shielded total counts
- gross flux monitor counts
- gross barrel flux monitor counts
- background shielded total counts
- background flux monitor counts
- background barrel flux monitor counts
- active pulses counts

Process

Read the data from the file and retain the configuration and values in memory.

Output

None.

M.6 Acquire passive data from assay chamber

The short gate corresponds to the shielded counts while the long gate corresponds to the system counts.

Input

- passive CAMAC crate configuration (M.1)
- passive count time limit (3.1.3.3 A)

Process

1. Program the gate generators as specified by the CAMAC crate configuration.
2. Clear the scalers and start counting events.
3. Poll the 10 kHz clock scaler port until the passive count time limit is reached or the user cancels the operation.
4. If passive count time limit is reached or exceeded, accumulate counts from the CAMAC scaler ports.

Output

- shielded total counts
- system total counts
- long gate counts
- short gate counts
- 10 kHz clock counts
- 1 kHz clock counts
- flux monitor counts
- barrel flux monitor counts
- bare door counts
- bare right counts
- bare back counts
- bare left counts
- bare top counts
- bare bottom counts
- shielded door counts
- shielded right counts
- shielded back counts
- shielded left counts
- shielded top counts
- shielded bottom counts
- shielded coincidence gate total counts
- shielded coincidence 1 MHz clock counts
- system coincidence gate total counts
- system coincidence 1 MHz clock counts

M.7 Store passive data on file

Note: These data are only stored if requested by the user after all data have been acquired and analyzed. In Version 2.0, data will be automatically stored at the end of an assay if "automatic data save" is enabled.

Input

- time and date of the data acquisition
- CAMAC crate configuration for passive signals (M.1)
- shielded total counts
- system total counts
- long gate counts
- short gate counts
- 10 kHz clock counts
- 1 kHz clock counts
- flux monitor counts
- barrel flux monitor counts
- bare door counts
- bare right counts
- bare back counts
- bare left counts
- bare top counts
- bare bottom counts
- shielded door counts
- shielded right counts
- shielded back counts
- shielded left counts
- shielded top counts
- shielded bottom counts
- shielded coincidence gate total counts
- shielded coincidence 1 MHz clock counts
- system coincidence gate total counts
- system coincidence 1 MHz clock counts

Process

Write inputs to a file.

Output

A file containing:

- time and date of the data acquisition
- CAMAC crate configuration for passive signals (M.1)
- shielded total counts
- system total counts
- long gate counts
- short gate counts
- 10 kHz clock counts
- 1 kHz clock counts
- flux monitor counts
- barrel flux monitor counts
- bare door counts
- bare right counts
- bare back counts
- bare left counts
- bare top counts
- bare bottom counts
- shielded door counts
- shielded right counts
- shielded back counts
- shielded left counts
- shielded top counts
- shielded bottom counts
- shielded coincidence gate total counts
- shielded coincidence 1 MHz clock counts
- system coincidence gate total counts
- system coincidence 1 MHz clock counts

M.8 Read passive data from file

During a recalculation, data are obtained from a file previously saved rather than from the CAMAC crate.

Input

- time and date of the data acquisition
- CAMAC crate configuration for passive signals (M.1)
- shielded total counts
- system total counts

- long gate counts
- short gate counts
- 10 kHz clock counts
- 1 kHz clock counts
- flux monitor counts
- barrel flux monitor counts
- bare door counts
- bare right counts
- bare back counts
- bare left counts
- bare top counts
- bare bottom counts
- shielded door counts
- shielded right counts
- shielded back counts
- shielded left counts
- shielded top counts
- shielded bottom counts
- shielded coincidence gate total counts
- shielded coincidence 1 MHz clock counts
- system coincidence gate total counts
- system coincidence 1 MHz clock counts

Process

Read the data from the file and retain the configuration and values in memory.

Output

None.

3.1.3.5 Active Signals Requirements

A. Active Signals Attributes

None.

M. Active Signals Methods

See Table 4 and 7 for inputs associated with the “active” CAMAC crate configuration.

M.1 Compute flux monitor gross, background, and net rates***Input***

- number of early gates read from the station and channel specified in the crate configuration, e.g., station 11, channel 6
- number of late gates read from the station and channel specified in the crate configuration, e.g., station 11, channel 6
- early gate flux monitor count read from the station and channel specified in the crate configuration, e.g., station 5, channel 1
- late gate flux monitor count read from the station and channel specified in the crate configuration, e.g., station 6, channel 1
- early gate width read from the station and channel specified in the crate configuration, e.g., station 1, channel 1
- late gate width read from the station and channel specified in the crate configuration, e.g., station 3, channel 1

Process

1. Compute gross flux monitor rate.

$$\text{early_gate_count_time} = \text{number_of_early_gates} * \text{early_gate_width}$$

$$\text{gross_flux_monitor_rate} = \frac{\text{early_gate_flux_monitor_count}}{\text{early_gate_count_time}}$$

2. Compute background flux monitor rate.

$$\text{late_gate_count_time} = \text{number_of_late_gates} * \text{late_gate_width}$$

$$\text{background_flux_monitor_rate} = \frac{\text{late_gate_flux_monitor_count}}{\text{late_gate_count_time}}$$

3. Compute net flux monitor rate.

$$\text{flux_monitor_rate} = \text{gross_flux_monitor_rate} - \text{background_flux_monitor_rate}$$

Output

- gross flux monitor rate in counts per second
- background flux monitor rate in counts per second
- flux monitor rate in counts per second

M.2 Compute flux monitor gross, background, and net rate errors***Input***

- number of early gates read from the station and channel specified in the crate configuration, e.g., station 11, channel 6
- number of late gates read from the station and channel specified in the crate configuration, e.g., station 11, channel 6

- early gate flux monitor count read from the station and channel specified in the crate configuration, e.g., station 5, channel 1
- late gate flux monitor count read from the station and channel specified in the crate configuration, e.g., station 6, channel 1
- early gate width read from the station and channel specified in the crate configuration, e.g., station 1, channel 1
- late gate width read from the station and channel specified in the crate configuration, e.g., station 3, channel 1

Process

1. Compute the error in the gross flux monitor rate.

$$\text{early_gate_count_time} = \text{number_of_early_gates} * \text{early_gate_width}$$

$$\text{gross_flux_monitor_rate_error} = \frac{\sqrt{\text{early_gate_flux_monitor_count}}}{\text{early_gate_count_time}}$$

2. Compute the error in the background flux monitor rate.

$$\text{late_gate_count_time} = \text{number_of_late_gates} * \text{late_gate_width}$$

$$\text{background_flux_monitor_rate_error} = \frac{\sqrt{\text{late_gate_flux_monitor_count}}}{\text{late_gate_count_time}}$$

3. Compute the error in the net flux monitor rate.

$$\text{flux_monitor_rate_error} =$$

$$\sqrt{\text{gross_flux_monitor_rate_error}^2 + \text{background_flux_monitor_rate_error}^2}$$

Output

- gross flux monitor rate error in counts per second
- background flux monitor rate error in counts per second
- flux monitor rate error in counts per second

M.3 Compute barrel flux monitor gross, background, and net rates

Input

- number of early gates read from the station and channel specified in the crate configuration, e.g., station 11, channel 6
- number of late gates read from the station and channel specified in the crate configuration, e.g., station 11, channel 6
- early gate barrel flux monitor count read from the station and channel specified in the crate configuration, e.g., station 5, channel 2
- late gate barrel flux monitor count read from the station and channel specified in the crate configuration, e.g., station 6, channel 2

- early gate width read from the station and channel specified in the crate configuration, e.g., station 1, channel 1
- late gate width read from the station and channel specified in the crate configuration, e.g., station 3, channel 1

Process

1. Compute gross barrel flux monitor rate.

$$\text{early_gate_count_time} = \text{number_of_early_gates} * \text{early_gate_width}$$

$$\text{gross_barrel_flux_monitor_rate} = \frac{\text{early_gate_barrel_flux_monitor_count}}{\text{early_gate_count_time}}$$

2. Compute background barrel flux monitor rate.

$$\text{late_gate_count_time} = \text{number_of_late_gates} * \text{late_gate_width}$$

$$\text{background_barrel_flux_monitor_rate} = \frac{\text{late_gate_barrel_flux_monitor_count}}{\text{late_gate_count_time}}$$

3. Compute net barrel flux monitor rate.

$$\text{barrel_flux_monitor_rate} =$$

$$\text{gross_barrel_flux_monitor_rate} - \text{background_barrel_flux_monitor_rate}$$

Output

- gross barrel flux monitor rate in counts per second
- background barrel flux monitor rate in counts per second
- barrel flux monitor rate in counts per second

M.4 Compute barrel flux monitor gross, background, and net rate errors

Input

- number of early gates read from the station and channel specified in the crate configuration, e.g., station 11, channel 6
- number of late gates read from the station and channel specified in the crate configuration, e.g., station 11, channel 6
- early gate barrel flux monitor count read from the station and channel specified in the crate configuration, e.g., station 5, channel 2
- late gate barrel flux monitor count read from the station and channel specified in the crate configuration, e.g., station 6, channel 2
- early gate width read from the station and channel specified in the crate configuration, e.g., station 1, channel 1
- late gate width read from the station and channel specified in the crate configuration, e.g., station 3, channel 1

Process

1. Compute the error in the gross barrel flux monitor rate.

$$\text{early_gate_count_time} = \text{number_of_early_gates} * \text{early_gate_width}$$

$$\text{gross_barrel_flux_monitor_rate_error} = \frac{\sqrt{\text{early_gate_barrel_flux_monitor_count}}}{\text{early_gate_count_time}}$$

2. Compute the error in the background barrel flux monitor rate.

$$\text{late_gate_count_time} = \text{number_of_late_gates} * \text{late_gate_width}$$

$$\text{background_barrel_flux_monitor_rate_error} = \frac{\sqrt{\text{late_gate_barrel_flux_monitor_count}}}{\text{late_gate_count_time}}$$

3. Compute the error in the net barrel flux monitor rate.

$$\text{barrel_flux_monitor_rate_error} =$$

$$\sqrt{\text{gross_barrel_flux_monitor_rate_error}^2 + \text{background_barrel_flux_monitor_rate_error}^2}$$

Output

- gross barrel flux monitor rate error in counts per second
- background barrel flux monitor rate error in counts per second
- barrel flux monitor rate error in counts per second

M.5 Compute shielded total gross, background, and net rates***Input***

- number of early gates read from the station and channel specified in the crate configuration, e.g., station 11, channel 6
- number of late gates read from the station and channel specified in the crate configuration, e.g., station 11, channel 6
- early gate barrel flux monitor count read from the station and channel specified in the crate configuration, e.g., station 5, channel 0
- late gate barrel flux monitor count read from the station and channel specified in the crate configuration, e.g., station 6, channel 0
- early gate width read from the station and channel specified in the crate configuration, e.g., station 1, channel 1
- late gate width read from the station and channel specified in the crate configuration, e.g., station 3, channel 1

Process

1. Compute gross shielded total rate.

$$\text{early_gate_count_time} = \text{number_of_early_gates} * \text{early_gate_width}$$

$$\text{gross_shielded_total_rate} = \frac{\text{early_gate_shielded_total_count}}{\text{early_gate_count_time}}$$

2. Compute background shielded total rate.

$$\text{late_gate_count_time} = \text{number_of_late_gates} * \text{late_gate_width}$$

$$\text{background_shielded_total_rate} = \frac{\text{late_gate_shielded_total_count}}{\text{late_gate_count_time}}$$

3. Compute net shielded total rate.

$$\text{shielded_total_rate} = \text{gross_shielded_total_rate} - \text{background_shielded_total_rate}$$

Output

- gross shielded total rate in counts per second
- background shielded total rate in counts per second
- shielded total rate in counts per second

M.6 Compute shielded total gross, background, and net rate errors**Input**

- number of early gates read from the station and channel specified in the crate configuration, e.g., station 11, channel 6
- number of late gates read from the station and channel specified in the crate configuration, e.g., station 11, channel 6
- early gate barrel flux monitor count read from the station and channel specified in the crate configuration, e.g., station 5, channel 0
- late gate barrel flux monitor count read from the station and channel specified in the crate configuration, e.g., station 6, channel 0
- early gate width read from the station and channel specified in the crate configuration, e.g., station 1, channel 1
- late gate width read from the station and channel specified in the crate configuration, e.g., station 3, channel 1

Process

1. Compute the error in the gross shielded total rate.

$$\text{early_gate_count_time} = \text{number_of_early_gates} * \text{early_gate_width}$$

$$\text{gross_shielded_total_rate_error} = \frac{\sqrt{\text{early_gate_shielded_total_count}}}{\text{early_gate_count_time}}$$

2. Compute the error in the background shielded total rate.

$$\text{late_gate_count_time} = \text{number_of_late_gates} * \text{late_gate_width}$$

$$\text{background_shielded_total_rate_error} = \frac{\sqrt{\text{late_gate_shielded_total_count}}}{\text{late_gate_count_time}}$$

3. Compute the error in the net shielded total rate.

shielded_total_rate_error =

$$\sqrt{gross_shielded_total_rate_error^2 + background_shielded_total_rate_error^2}$$

Output

- gross shielded total rate error in counts per second
- background shielded total rate error in counts per second
- shielded total rate error in counts per second

3.1.3.6 Passive Signals Requirements

The passive signals include the results from both gross and background measurements. These measurements are made at different times. The background measurements are obtained with an empty container in the assay chamber, while the gross measurements are obtained with a contact-handled waste container in the assay chamber. With the current configuration, these signals are processed through the same modules and channels of the CAMAC crate.

A. Passive Signals Attributes

None.

M. Passive Signals Methods

See Tables 5 and 6 for inputs associated with the “passive” CAMAC crate configuration.

M.1 Compute shielded total rate

Input

- 10 kHz clock pulse rate read from the station and channel specified in the crate configuration file, e.g., station 21, channel 3
- number of 10 kHz clock pulses read from the station and channel specified in the crate configuration file, e.g., station 11, channel 4
- shielded total count read from the station and channel specified in the crate configuration file, e.g., station 11, channel 0
- number of background 10 kHz clock pulses read from the station and channel specified in the crate configuration file, e.g., station 11, channel 4
- background shielded total count read from the station and channel specified in the crate configuration file, e.g., station 11, channel 0
- background correction indicator (3.1.3.3 A)

Process

1. Compute the count time and gross shielded total rate.

$$\text{count_time} = \frac{\text{number_of_10kHz_clock_pulses}}{10\text{kHz_clock_pulse_rate}}$$

$$\text{gross_shielded_total_rate} = \frac{\text{gross_shielded_total_count}}{\text{count_time}}$$

2. Compute the background count time and the background shielded total rate.

$$\text{background_count_time} = \frac{\text{number_of_background_10kHz_clock_pulses}}{10\text{kHz_clock_pulse_rate}}$$

$$\text{background_shielded_total_rate} = \frac{\text{background_shielded_total_count}}{\text{background_count_time}}$$

3. Compute the net shielded total rate.

IF *background_correction* THEN

$$\text{shielded_total_rate} = \text{gross_shielded_total_rate} - \text{background_shielded_total_rate}$$

ELSE

$$\text{shielded_total_rate} = \text{gross_shielded_total_rate}$$

Output

- count time in seconds
- background count time in seconds
- gross shielded total rate in counts per second
- background shielded total rate in counts per second
- shielded total rate in counts per second

M.2 Compute shielded total rate error

Input

- shielded total count read from the station and channel specified in the configuration file, e.g., station 11, channel 0
- background shielded total count read from the station and channel specified in the configuration file, e.g., station 11, channel 0
- count time (M.1)
- background count time (M.1)
- background correction indicator (3.1.3.3 A)

Process

1. Compute the gross shielded total rate error in counts per second.

$$\text{gross_shielded_total_rate_error} = \frac{\sqrt{\text{gross_shielded_total_count}}}{\text{count_time}}$$

2. Compute the background shielded total rate error in counts per second.

$$\text{background_shielded_total_rate_error} = \frac{\sqrt{\text{background_shielded_total_count}}}{\text{background_count_time}}$$

3. Compute the net shielded total rate error in counts per second.

IF *background_correction* THEN

• *shielded_total_rate_error* =

$$\sqrt{\text{gross_shielded_total_rate_error}^2 + \text{background_shielded_total_rate_error}^2}$$

ELSE

• *shielded_total_rate_error* = *gross_shielded_total_rate_error*

Output

- gross shielded total rate error in counts per second
- background shielded total rate error in counts per second
- shielded total rate error in counts per second

M.3 Compute system total rate

Input

- 10 kHz clock pulse rate read from the station and channel specified in the configuration file, e.g., station 21, channel 3
- number of 10 kHz clock pulses read from the station and channel specified in the configuration file, e.g., station 11, channel 4
- system total count read from the station and channel specified in the configuration file, e.g., station 11, channel 1
- number of background 10 kHz clock pulses read from the station and channel specified in the configuration file, e.g., station 11, channel 4
- background system total count read from the station and channel specified in the configuration file, e.g., station 11, channel 1
- background correction indicator (3.1.3.3 A)

Process

1. Compute the count time and the gross system total rate.

$$\text{count_time} = \frac{\text{number_of_10kHz_clock_pulses}}{10\text{kHz_clock_pulse_rate}}$$

$$\text{gross_system_total_rate} = \frac{\text{gross_system_total_count}}{\text{count_time}}$$

2. Compute the background count time and the background system total rate.

$$\text{background_count_time} = \frac{\text{number_of_background_10kHz_clock_pulses}}{\text{background_10kHz_clock_pulse_rate}}$$

$$\text{background_system_total_rate} = \frac{\text{background_system_total_count}}{\text{background_count_time}}$$

3. Compute the net system total rate.

IF *background_correction* THEN

$$\text{system_total_rate} = \text{gross_system_total_rate} - \text{background_system_total_rate}$$

ELSE

$$\text{system_total_rate} = \text{gross_system_total_rate}$$

Output

- count time in seconds
- background count time in seconds
- gross system total rate in counts per second
- background system total rate in counts per second
- system total rate in counts per second

M.4 Compute gross, background, and net system total rate error

Input

- system total count read from the station and channel specified in the configuration file, e.g., station 11, channel 2
- background system total count read from the station and channel specified in the configuration file, e.g., station 11, channel 2
- count time (M.3)
- background count time (M.3)
- background correction indicator (3.1.3.3 A)

Process

1. Compute the gross system total rate error in counts per second.

$$\text{gross_system_total_rate_error} = \frac{\sqrt{\text{gross_system_total_count}}}{\text{count_time}}$$

2. Compute the background system total rate error in counts per second.

$$\text{background_system_total_rate_error} = \frac{\sqrt{\text{background_system_total_count}}}{\text{background_count_time}}$$

3. Compute the net system total rate error in counts per second.

IF *background_correction* THEN

$$\text{system_total_rate_error} =$$

$$\sqrt{\text{gross_system_total_rate_error}^2 + \text{background_system_total_rate_error}^2}$$

ELSE

$$\text{system_total_rate_error} = \text{gross_system_total_rate_error}$$

Output

- gross system total rate error in counts per second
- background system total rate error in counts per second
- system total rate error in counts per second

M.5 Compute shielded coincidence rate

Input

- 1 MHz clock pulse rate read from the station and channel specified in the configuration file, e.g., station 21, channel 1
- gross shielded coincidence gate delay time read from the station and channel specified in the configuration file, e.g., station 7, channel 0
- number of gross shielded coincidence gates read from the station and channel specified in the configuration file, e.g., station 11, channel 3
- gross shielded coincidence counts read from the station and channel specified in the configuration file, e.g., station 13, channel 0
- number of gross shielded coincidence 1 MHz clock pulses read from the station and channel specified in the configuration file, e.g., station 13, channel 1
- background shielded coincidence gate delay time read from the station and channel specified in the configuration file, e.g., station 7, channel 0
- number of background shielded coincidence gates read from the station and channel specified in the configuration file, e.g., station 11, channel 3
- background shielded coincidence counts read from the station and channel specified in the configuration file, e.g., station 13, channel 0
- number of background shielded coincidence 1 MHz clock pulses read from the station and channel specified in the configuration file, e.g., station 13, channel 1
- count time (M.1)
- gross shielded total rate (M.1)
- background count time (M.1)
- background shielded total rate (M.1)
- background correction indicator (3.1.3.3 A)
- NEUT2 simulation indicator (3.1.3.2 A)

Process

1. Compute the gross shielded coincidence rate.

```

gate_time =  $\frac{\text{gross_shielded_coincidence\_1MHz\_clock\_pulse\_counts}}{\text{1MHz\_clock\_pulse\_rate}}$ 
IF neut2_simulation THEN
  live_time = count_time - gate_time - gate_delay
ELSE
  live_time = count_time - gate_time - number_of_gross_shielded_coincidence_gates*gate_delay
gross_shielded_coincidence_rate =
   $\left( \frac{\text{gross_shielded_coincidence_counts}}{\text{gate_time}} - \text{gross_shielded_total_rate} \right) * \frac{\text{gate_time}}{\text{live_time}}$ 

```

2. Compute the background shielded coincidence rate.

```

gate_time =  $\frac{\text{background_shielded_coincidence\_1MHz\_clock\_pulse\_counts}}{\text{1MHz\_clock\_pulse\_rate}}$ 
IF neut2_simulation THEN
  live_time = background_count_time - gate_time - gate_delay
ELSE
  live_time = background_count_time -
    gate_time - number_of_background_shielded_coincidence_gates*gate_delay
background_shielded_coincidence_rate =
   $\left( \frac{\text{background_shielded_coincidence_counts}}{\text{gate_time}} - \text{background_shielded_total_rate} \right) * \frac{\text{gate_time}}{\text{live_time}}$ 

```

3. Compute the net shielded coincidence rate.

```

IF background_correction THEN
  shielded_coincidence_rate =
    gross_shielded_coincidence_rate - background_shielded_coincidence_rate
ELSE
  shielded_coincidence_rate = gross_shielded_coincidence_rate

```

Output

- shielded coincidence rate in counts per second

M.6 Compute shielded coincidence rate error

Input

- 1 MHz clock pulse rate read from the station and channel specified in the configuration file, e.g., station 21, channel 1
- gross shielded coincidence gate delay time read from the station and channel specified in the configuration file, e.g., station 7, channel 0

- number of gross shielded coincidence gates read from the station and channel specified in the configuration, e.g., station 11, channel 3
- gross shielded coincidence counts read from the station and channel specified in the configuration, e.g., station 13, channel 0
- number of gross shielded coincidence 1 MHz clock pulses read from the station and channel specified in the configuration file, e.g., station 13, channel 1
- background shielded coincidence gate delay time read from the station and channel specified in the configuration file, e.g., station 7, channel 0
- number of background shielded coincidence gates read from the station and channel specified in the configuration file, e.g., station 11, channel 3
- background shielded coincidence counts read from the station and channel specified in the configuration file, e.g., station 13, channel 0
- number of background shielded coincidence 1 MHz clock pulses read from the station and channel specified in the configuration file, e.g., station 13, channel 1
- count time (M.1)
- gross shielded total rate error (M.2)
- background count time (M.1)
- background shielded total rate error (M.1)
- background correction indicator (3.1.3.3 A)
- NEUT2 simulation indicator (3.1.3.2 A)

Process

1. Compute the gross shielded coincidence rate error.

$$\text{gate_time} = \frac{\text{gross_shielded_coincidence_1MHz_clock_pulse_counts}}{\text{1MHz_clock_pulse_rate}}$$

IF *neut2_simulation* THEN

$$\text{live_time} = \text{count_time} - \text{gate_time} - \text{gate_delay}$$

ELSE

$$\text{live_time} = \text{count_time} - \text{gate_time} - \text{number_of_gross_shielded_coincidence_gates} * \text{gate_delay}$$

$$\text{gross_shielded_coincidence_rate_error} = \frac{\text{gate_time} * \sqrt{\frac{\text{gross_shielded_coincidence_counts}}{\text{gate_time}^2} + \text{gross_shielded_total_rate_error}^2}}{\text{live_time}}$$

2. Compute the background shielded coincidence rate error.

```

gate_time =  $\frac{\text{background_shielded_coincidence\_1MHz\_clock\_pulse\_counts}}{\text{1MHz\_clock\_pulse\_rate}}$ 

IF neut2_simulation THEN
  live_time = background_count_time - gate_time - gate_delay
ELSE
  live_time = background_count_time - gate_time -
    number_of_background_shielded_coincidence_gates*gate_delay
background_shielded_coincidence_rate_error =

$$\frac{\text{gate_time} * \sqrt{\frac{\text{background_shielded_coincidence_counts}}{\text{gate_time}^2} + \text{background_shielded_total_rate_error}^2}}{\text{live_time}}$$


```

3. Compute the net shielded coincidence rate error.

```

IF background_correction THEN
  shielded_coincidence_rate_error =

$$\sqrt{\text{gross_shielded_coincidence_rate_error}^2 + \text{background_shielded_coincidence_rate_error}^2}$$

ELSE
  shielded_coincidence_rate_error = gross_shielded_coincidence_rate_error

```

Output

- shielded coincidence rate error in counts per second

M.7 Compute system coincidence rate

Input

- 1 MHz clock pulse rate read from the station and channel specified in the configuration file, e.g., station 21, channel 1
- gross system coincidence gate delay time read from the station and channel specified in the configuration file, e.g., station 9, channel 0
- number of gross system coincidence gates read from the station and channel specified in the configuration, e.g., station 11, channel 2
- number of gross system coincidence counts read from the station and channel specified in the configuration, e.g., station 13, channel 2
- number of system coincidence 1 MHz clock pulses read from the station and channel specified in the configuration file, e.g., station 13, channel 3
- number of background system coincidence gate delay time read from the station and channel specified in the configuration file, e.g., station 7, channel 0
- number of background system coincidence gates read from the station and channel specified in the configuration file, e.g., station 11, channel 3
- background system coincidence counts read from the station and channel specified in the configuration file, e.g., station 13, channel 0

- number of background system coincidence 1 MHz clock pulses read from the station and channel specified in the configuration file, e.g., station 13, channel 1
- count time (M.3)
- gross system total rate (M.3)
- background count time (M.3)
- background system total rate (M.3)
- background correction indicator (3.1.3.3 A)
- NEUT2 simulation indicator (3.1.3.2 A)

Process

1. Compute the gross system coincidence rate.

$$\text{gate_time} = \frac{\text{gross_system_coincidence_1MHz_clock_pulse_counts}}{\text{1MHz_clock_pulse_rate}}$$

IF *neut2_simulation* THEN

$$\text{live_time} = \text{count_time} - \text{gate_time} - \text{gate_delay}$$

ELSE

$$\text{live_time} = \text{count_time} - \text{gate_time} - \text{number_of_gross_system_coincidence_gates} * \text{gate_delay}$$

gross_system_coincidence_rate =

$$\left(\frac{\text{gross_system_coincidence_counts}}{\text{gate_time}} - \text{gross_system_total_rate} \right) * \frac{\text{gate_time}}{\text{live_time}}$$

2. Compute the background system coincidence rate.

$$\text{gate_time} = \frac{\text{background_gross_system_coincidence_1MHz_clock_pulse_counts}}{\text{1MHz_clock_pulse_rate}}$$

IF *neut2_simulation* THEN

$$\text{live_time} = \text{background_count_time} - \text{gate_time} - \text{gate_delay}$$

ELSE

$$\text{live_time} = \text{background_count_time} -$$

$$\text{gate_time} - \text{number_of_background_system_coincidence_gates} * \text{gate_delay}$$

background_system_coincidence_rate =

$$\left(\frac{\text{background_system_coincidence_counts}}{\text{gate_time}} - \text{gross_system_total_rate} \right) * \frac{\text{gate_time}}{\text{live_time}}$$

3. Compute the net system coincidence rate.

IF *background_correction* THEN

$$\text{system_coincidence_rate} = \text{gross_system_coincidence_rate} - \text{background_system_coincidence_rate}$$

ELSE

$$\text{system_coincidence_rate} = \text{gross_system_coincidence_rate}$$

Output

- system coincidence rate in counts per second

M.8 Compute system coincidence rate error*Input*

- 1 MHz clock pulse rate read from the station and channel specified in the configuration file, e.g., station 21, channel 1
- gross system coincidence gate delay time read from the station and channel specified in the configuration file, e.g., station 7, channel 0
- number of gross system coincidence gates read from the station and channel specified in the configuration, e.g., station 11, channel 3
- number of gross system coincidence counts read from the station and channel specified in the configuration, e.g., station 13, channel 0
- number of system coincidence 1 MHz clock pulses read from the station and channel specified in the configuration file, e.g., station 13, channel 1
- background system coincidence gate delay time read from the station and channel specified in the configuration file, e.g., station 7, channel 0
- number of background system coincidence gates read from the station and channel specified in the configuration file, e.g., station 11, channel 3
- number of background system coincidence counts read from the station and channel specified in the configuration file, e.g., station 13, channel 0
- number of background system coincidence 1 MHz clock pulses read from the station and channel specified in the configuration file, e.g., station 13, channel 1
- count time (M.3)
- gross system total rate error (M.4)
- background count time (M.3)
- background system total rate error (M.4)
- background correction indicator (3.1.3.3 A)
- NEUT2 simulation indicator (3.1.3.2 A)

Process

1. Compute the gross system coincidence rate error.

$$\text{gate_time} = \frac{\text{gross_system_coincidence_1MHz_clock_pulse_counts}}{\text{1MHz_clock_pulse_rate}}$$

IF *neut2_simulation* THEN

$$\text{live_time} = \text{count_time} - \text{gate_time} - \text{gate_delay}$$

ELSE

$$\text{live_time} = \text{count_time} - \text{gate_time} - \text{number_of_gross_system_coincidence_gates} * \text{gate_delay}$$

$$\text{gross_system_coincidence_rate_error} = \frac{\text{gate_time} * \sqrt{\frac{\text{gross_system_coincidence_counts}}{\text{gate_time}^2} + \text{gross_system_total_rate_error}^2}}{\text{live_time}}$$

2. Compute the background system coincidence rate error.

$$\text{gate_time} = \frac{\text{background_system_coincidence_1MHz_clock_pulse_counts}}{\text{1MHz_clock_pulse_rate}}$$

IF *neut2_simulation* THEN

$$\text{live_time} = \text{background_count_time} - \text{gate_time} - \text{gate_delay}$$

ELSE

$$\text{live_time} = \text{background_count_time} - \text{gate_time} -$$

$$\text{number_of_background_system_coincidence_gates} * \text{gate_delay}$$

$$\text{background_system_coincidence_rate_error} = \frac{\text{gate_time} * \sqrt{\frac{\text{background_system_coincidence_counts}}{\text{gate_time}^2} + \text{background_system_total_rate_error}^2}}{\text{live_time}}$$

3. Compute the net system coincidence rate error.

IF *background_correction* THEN

$$\text{system_coincidence_rate_error} =$$

$$\sqrt{\text{gross_system_coincidence_rate_error}^2 + \text{background_system_coincidence_rate_error}^2}$$

ELSE

$$\text{system_coincidence_rate_error} = \text{gross_system_coincidence_rate_error}$$

Output

- system coincidence rate error in counts per second

3.1.4 Data Reduction Functional Requirements

This section specifies all the requirements for reducing the raw signal assay data to measurement estimates of the container contents.

3.1.4.1 State Transition Diagrams

None.

3.1.4.2 Container Requirements

A. Container Attributes

- identification, an external identifier such as the waste generator ID or bar-code number
- type, a description of the container, e.g., 55 gallon drum
- volume, in liters (L)

M. Container Methods

M.1 Input container attributes

Input

- identification
- type

Process

1. Provide dialog for user to enter identification and select type from container list.
- 1a. Version 2.0: Use the file name entered by the user as the default ID.
2. Set the container volume corresponding to the type selected.

Output

- identification
- type
- volume

M.2 Store container data on file

Note: These data are only stored if requested by the user after all data have been acquired and analyzed. In Version 2.0, data will be automatically stored at the end of an assay if "automatic data save" is enabled.

Input

- identification
- type
- volume

Process

Write the inputs to a file.

Output

A file containing:

- identification
- type

- volume

M.3 Read container data from file

Input

- identification
- type
- volume

Process

Read the data from file and retain the attributes in memory.

Output

None

M.4 Provide container summary report

Note: This information is to be included in general assay summary report that can be requested by the user at the end of an assay. If the “automatic summary report” indicator is set, the summary report will be generated automatically at the end of an assay.

Input

- identification
- type
- volume

Process

Write inputs to report.

Output

report with following information:

- Identifier: identification
- Type: type
- Volume: volume in liter (L)

M.5 Allow Physicist to modify the container list

Input

- type
- volume

Process

Set new values.

Output

- type
- volume

3.1.4.3 Matrix Requirements**A. Matrix Attributes**

- content code, a number indicating type of material
- content description, e.g., dry combustibles
- matrix type; one of { nonsludge, sludge, Am-sludge}
- net weight, in kilograms (kg)
- mass selection criterion; one of {active, passive, default}
- moderator index (override calculated value)
- self absorption correction factor coefficient c_0
- self absorption correction factor coefficient c_1
- self absorption correction factor coefficient c_2
- self absorption correction factor coefficient c_3
- self absorption correction factor coefficient div
- active correction factor (override calculated value)
- shielded coincidence correction factor (override calculated value)
- system coincidence correction factor (override calculated value)
- fraction Pu-238
- fraction Pu-239
- fraction Pu-240
- fraction Pu-241
- fraction Pu-242

The current mass selection criteria are provided in Table 8. The "matrix type" designator is of importance only for those matrices categorized as "Am sludge"; currently only Content Code 001 is so categorized. A moderator index value is supplied only for certain "sludge" type matrices. These moderator index values are provided in Table 9. The self absorption correction factor coefficients are provided in Table 10. No correction factors are currently supplied; calculated values will be used for all current matrices (methods M.5 and M.7, this section). All matrix attributes, with the exception of the plutonium mass fractions, shall be stored in a file, sorted by content code. The plutonium mass fraction values shall be obtained from a file keyed to the content code,⁶ or from a list of default values provided by Physics technical support.

⁶ Files containing the plutonium mass fraction values are generated by the SGRS.

Table 8. Mass selection criteria.

Content code	Mass selection method		
	Active	Passive	Default
0			x
1	x		
2	x		
3	x		
4	x		
7	x		
90		x	
300		x	
301		x	
302		x	
330		x	
335		x	
336		x	
337		x	
338		x	
339		x	
371		x	
374		x	
432		x	
440		x	
442		x	
463		x	
464		x	
480		x	
481		x	
490		x	
900		x	
960		x	
970		x	
990		x	
all other			x

Table 9. Moderator index values.

content code	moderator index
1	0.64
2	0.56
3	0.456
4	0.51
7	0.714

Table 10. Self absorption correction factor coefficients.

Content code	Self absorption correction factor coefficients				
	c_0	c_1	c_2	c_3	div
0	1.0	0.015	0.0	0.0	1.0
1	1.0	0.0	0.0	0.0	1.0
2	1.0	0.0	0.0	0.0	1.0
3	1.0	0.0	0.0	0.0	1.0
4	1.0	0.0	0.0	0.0	1.0
7	1.0	0.0	0.0	0.0	1.0
292	1.0	0.0358	0.0	0.0	1.0
320	2.5	0.05	0.0	0.0	1.0
339	2.6	0.0	0.0	0.0	1.0
376	2.54	0.0	0.0	0.0	1.0
393	2.82	0.0	0.0	0.0	1.0
409	10.8	0.0	0.0	0.0	0.0
411	13.0	0.0	0.0	0.0	0.0
480	2.9	0.0	0.0	0.0	1.0
all other	1.0	0.015	0.0	0.0	1.0

M. Matrix Methods

M.1 Input matrix attributes

Input

- content code
- net weight

Process

1. Provide dialog for user to select content code from matrix list.

1a. Version 2.0: Default to last content code used.

2. Obtain net weight (kg) from user.
3. Set the matrix content code to that selected and store weight.

Output

- content code
- net weight

M.2 Store matrix data on file

Input

- content code
- content description
- matrix type
- mass selection criterion
- moderator index
- self absorption correction factor coefficient c_0
- self absorption correction factor coefficient c_1
- self absorption correction factor coefficient c_2
- self absorption correction factor coefficient c_3
- self absorption correction factor coefficient div
- active correction factor
- shielded coincidence correction factor
- system coincidence correction factor
- net weight
- fraction Pu-238
- fraction Pu-239
- fraction Pu-240
- fraction Pu-241
- fraction Pu-242

Process

Write inputs to a file.

Output

A file containing:

- content code
- content description
- matrix type
- mass selection criterion

- moderator index
- self absorption correction factor coefficient c_0
- self absorption correction factor coefficient c_1
- self absorption correction factor coefficient c_2
- self absorption correction factor coefficient c_3
- self absorption correction factor coefficient div
- active correction factor
- shielded coincidence correction factor
- system coincidence correction factor
- net weight
- fraction of Pu-238
- fraction of Pu-239
- fraction of Pu-240
- fraction of Pu-241
- fraction of Pu-242

M.3 Read matrix data from file

Input

- content code
- content description
- matrix type
- mass selection criterion
- moderator index
- self absorption correction factor coefficient c_0
- self absorption correction factor coefficient c_1
- self absorption correction factor coefficient c_2
- self absorption correction factor coefficient c_3
- self absorption correction factor coefficient div
- active correction factor
- shielded coincidence correction factor
- system coincidence correction factor
- net weight
- fraction of Pu-238
- fraction of Pu-239
- fraction of Pu-240
- fraction of Pu-241
- fraction of Pu-242

Process

Read the data from the file and retains the configuration and values in memory.

Output

None.

M.4 Allow Physicist to modify the matrix list

Input

- content code
- content description
- matrix type
- mass selection criterion
- moderator index
- self absorption correction factor coefficient c_0
- self absorption correction factor coefficient c_1
- self absorption correction factor coefficient c_2
- self absorption correction factor coefficient c_3
- self absorption correction factor coefficient div
- active correction factor
- shielded coincidence correction factor
- system coincidence correction factor

Process

Set new values.

Output

- content code
- content description
- matrix type
- mass selection criterion
- moderator index
- self absorption correction factor coefficient c_0
- self absorption correction factor coefficient c_1
- self absorption correction factor coefficient c_2
- self absorption correction factor coefficient c_3
- self absorption correction factor coefficient div
- active correction factor
- shielded coincidence correction factor
- system coincidence correction factor

M.5 Compute active Pu mass***Input***

- content code
- fraction Pu-239
- calibration fraction Pu-239 (3.1.3.3 A)
- U-235/Pu-239 mass ratio⁽¹⁾
- Pu-239 effective fission cross-section = 690.0 (b)⁽¹⁾
- Pu-239 neutrons per fission = 2.89⁽¹⁾
- U-235 effective fission cross-section = 501.0 (b)⁽¹⁾
- U-235 neutrons per fission = 2.42⁽¹⁾
- active zero matrix calibration factor (3.1.3.3 A)
- active net flux monitor rate (3.1.3.5 M.1)
- active net barrel flux monitor rate (3.1.3.5 M.3)
- active gross shielded total rate (3.1.3.5 M.5)
- active background shielded total rate (3.1.3.5 M.5)
- passive net shielded total rate (3.1.3.6 M.1)
- passive net system total rate (3.1.3.6 M.3)
- NEUT2 simulation indicator (3.1.3.2 A)

⁽¹⁾ Added in Version 2.0.

Table 11. Equation parameters, active mass calculation.

Equation parameter	Drum chamber	Crate chamber
Absorber index default	1.0	0.18
Absorber index minimum	1.0	0.1
Moderator index a_0	1.0	3.3
Moderator index a_1	0.226	0.235
Moderator index a_2	1.04	1.0
Moderator index a_3	0.2924	-0.048
Moderator index minimum	0.0	0.0
Moderator index maximum	0.75	0.70
ai threshold	2.72	2.72
Active matrix absorber coefficient ac_0	0.5404	0.5404
Active matrix absorber coefficient ac_1	0.612	0.612
mi threshold	0.4	0.4
Active matrix moderator coefficient mc_0	0.483	0.483
Active matrix moderator coefficient mc_1	1.817	1.817
Active mass ic_0	0.00359	0.00359
Active mass ic_1	0.00139	0.00139
Active background scale factor	0.977	1.0

Process

1. Compute the absorption index.

$$absorption_index = \begin{cases} \text{IF } active_barrel_flux_monitor_rate = 0.0 \text{ THEN} \\ \quad absorber_index_default \\ \text{ELSE} \\ \quad \max\left(absorber_index_min, \frac{active_flux_monitor_rate}{active_barrel_flux_monitor_rate}\right) \end{cases}$$

2. If moderator index value read from the matrix attributes file = 0.0, compute the moderator index; otherwise use the supplied value.

$$moderator_index = \begin{cases} \text{IF } passive_shielded_total_rate \leq 10.0 \text{ THEN} \\ \quad 0.15 \\ \text{ELSE} \\ \quad \text{IF } passive_system_total_rate \leq 0.0 \text{ THEN} \\ \quad \quad mi_min \\ \text{ELSE} \\ \quad \quad \min\left(mi_max, \max\left(mi_min, mi\right)\right) \end{cases}$$

where

$$mi = a_0 * \left[1.0 - \left(\frac{passive_shielded_total_rate}{passive_system_total_rate} \right) * \frac{1.0}{a_1} \right] * (a_2 + a_3 * \ln(absorption_index))$$

3. Compute the active matrix correction factor.

$$active_matrix_cf = \begin{cases} \text{IF } neut2_simulation \text{ THEN} \\ \quad absorption_cf * moderator_cf \\ \text{ELSE} \\ \quad \text{IF } no \text{ externally supplied matrix cf } \text{ THEN} \\ \quad \quad absorption_cf * moderator_cf \\ \text{ELSE} \\ \quad \quad externally_supplied_matrix_cf \end{cases}$$

where

$$absorption_cf = \begin{cases} \text{IF } absorption_index \leq ai_threshold \text{ THEN} \\ \quad 1.0 \\ \text{ELSE} \\ \quad ac_0 * (absorption_index)^{ac_1} \end{cases}$$

and

$$moderator_cf = \begin{cases} \text{IF } moderator_index \leq mi_threshold \text{ THEN} \\ \quad 1.0 \\ \text{ELSE} \\ \quad mc_0 * \exp(mc_1 * moderator_index) \end{cases}$$

4. Compute the interrogation background.

$$interrogation_background = \frac{ic_0 + ic_1 * \ln(absorption_index)}{\left[1.0 + \frac{1.0}{(1.0 - moderator_index)} \right]}$$

5. Compute the active shielded total adjusted rate.

$$active_shielded_total_adjusted_rate = active_gross_shielded_total_rate - \\ active_background_scale_factor * active_background_shielded_total_rate$$

6. Compute the isotopic correction factor.

$$isotopic_cf = \begin{cases} \text{IF } fraction_Pu_239 < .01 \text{ or } fraction_Pu_239 > .99 \text{ THEN} \\ \quad 1.0 \\ \text{ELSE} \\ \quad \text{IF } neut2_simulation \text{ THEN} \\ \quad \quad \left(\frac{0.9425}{fraction_Pu_239} \right) \\ \quad \text{ELSE} \\ \quad \quad \left(\frac{calibration_fraction_Pu_239}{fraction_Pu_239} \right) \end{cases}$$

7. Compute Pu mass, corrected by the zero matrix calibration factor, the isotopic correction factor, and the active matrix correction factor.

$$Pu_mass = \left(\frac{active_shielded_total_adjusted_rate}{active_flux_monitor_rate} - interrogation_background \right) * \\ active_zero_matrix_calibration_factor * isotopic_cf * active_matrix_cf$$

8. Compute the self absorption correction factor based on the content code and the coefficients in Table 10.

$$self_absorption_cf = \frac{(c_0 + c_1 * Pu_mass)}{denom} + c_2 * \exp(c_3 * Pu_mass)$$

where

$$denom = \begin{cases} \text{IF } div = 0.0 \text{ THEN} \\ \quad active_matrix_cf \\ \text{ELSE} \\ \quad 1.0 \end{cases}$$

9. Compute U-235 correction factor as described in Reference [25].

IF $U235_mass_ratio > 0.0$ AND NOT $neut2_simulation$ THEN

$$U235_cf = 1.0 + \frac{U235_cross_section}{Pu239_cross_section} * \frac{U235_N_fission}{Pu239_N_fission} * U235_mass_ratio$$

ELSE

$$U235_cf = 1.0$$

$$Pu_mass = \frac{Pu_mass}{U235_cf}$$

10. Compute the Pu mass.

```

active_Pu_mass = { IF neut2_simulation THEN
                   Pu_mass*self_absorption_cf
               ELSE
                   IF internally_computed_active_matrix_cf THEN
                       Pu_mass*self_absorption_cf
                   ELSE
                       Pu_mass

```

Output

- moderator index
- active shielded total adjusted rate
- active matrix correction factor
- U-235 correction factor
- active Pu mass in grams

M.6 Calculate active Pu mass error*Input*

- active Pu mass (M.5)
- active matrix correction factor (M.5)
- active mass error coefficient $ec_0 = 0.003$
- active mass error coefficient $ec_1 = 0.05$
- active mass error coefficient $ec_2 = 0.05$
- active shielded total adjusted rate (M.5)
- fraction Pu-239 and error
- U-235/Pu-239 mass ratio error
- calibration fraction Pu-239 and error (3.1.3.3 A)
- active zero matrix calibration factor and error (3.1.3.3 A)
- active flux monitor rate and error (3.1.3.5 M.1, M.2)
- active gross shielded total rate and error (3.1.3.5 M.5, M.6)
- active background shielded total rate and error (3.1.3.5 M.5, M.6)
- NEUT2 simulation indicator (3.1.3.2 A)

Process

If NEUT2 simulation, compute the active Pu mass error as follows:

$$active_Pu_mass_error = \text{abs}(ec_0 + (ec_1 + ec_2 * active_matrix_cf) * active_Pu_mass)$$

Otherwise, use the error propagation for multiplication and division to compute the active Pu mass error. The error propagation formulae are as follows (Reference [23], Chapter 3 and Reference [24], Chapter 3):

$$(x \pm \Delta x) * (y \pm \Delta y) = \left(x * y \pm \sqrt{y^2 * \Delta x^2 + x^2 * \Delta y^2} \right)$$

$$(x \pm \Delta x) / (y \pm \Delta y) = \left[\frac{x}{y} \pm \left(\frac{\sqrt{y^2 * \Delta x^2 + x^2 * \Delta y^2}}{y^2} \right) \right]$$

In many of the equations in this and following sections, an "error" function is shown indicating that standard error propagation rules are to be used. This is done rather than presenting the complete error propagation equations in order to conserve space.

1. Compute the active shielded total adjusted rate error.

$$\text{active_shielded_total_adjusted_error} = \left[\text{active_gross_shielded_total_error}^2 + (\text{active_background_scale_factor} * \text{active_background_shielded_total_error})^2 \right]^{0.5}$$

2. Compute the isotopic correction factor error.

$$\text{isotopic_cf_error} = \begin{cases} \text{IF } \text{fraction_Pu_239} < .01 \text{ or } \text{fraction_Pu_239} > .99 \text{ THEN} \\ \quad 0.0 \\ \text{ELSE} \\ \quad \text{error} \left(\frac{\text{cal_fraction_Pu_239} \pm \text{cal_fraction_Pu_239_error}}{\text{fraction_Pu_239} \pm \text{fraction_Pu_239_error}} \right) \end{cases}$$

3. Compute active Pu mass error, corrected by the zero matrix calibration factor, the isotopic correction factor, and the active matrix correction factor.

$$\text{active_Pu_mass_error} = \text{error} \left(\begin{array}{l} ((\text{mass} \pm \text{mass_error}) * \\ \left(\text{active_zero_matrix_calibration_factor} \pm \right. \\ \left. \text{active_zero_matrix_calibration_factor_error} \right) * \\ ((\text{isotopic_cf} \pm \text{isotopic_cf_error}) * \\ (\text{active_matrix_cf} \pm \text{active_matrix_cf_error})) \end{array} \right)$$

where

$$\text{mass} = \left(\frac{\text{active_shielded_total_adjusted_rate}}{\text{active_flux_monitor_rate}} - \text{interrogation_background} \right),$$

$$\text{mass_error} = \text{error} \left(\frac{\frac{\text{active_shielded_total_adjusted_rate} \pm \text{active_shielded_total_adjusted_error}}{\text{active_flux_monitor_rate} \pm \text{active_flux_monitor_error}} - \text{interrogation_background}}{\text{interrogation_background}} \right)$$

4. If the active matrix correction factor is not externally supplied, apply the internally computed self absorption correction factor to the Pu mass error.

*active_Pu_mass_error = active_Pu_mass_error*self_absorption_cf*

5. If U-235 is present and not using NEUT2 simulation, propagate the U-235/Pu-239 mass ratio error using standard error propagation rules [23, 24] and the example in Reference [25].

Output

- active Pu mass error in grams

M.7 Calculate passive Pu mass

Input

- shielded coincidence rate and error (3.1.3.6 M.5, M.6)
- system coincidence rate and error (3.1.3.6 M.7, M.8)
- moderator index (M.5)
- calibration fraction Pu-240 (3.1.3.3 A)
- shielded coincidence zero matrix calibration factor (3.1.3.3 A)
- system coincidence zero matrix calibration factor (3.1.3.3 A)
- NEUT2 simulation indicator (3.1.3.2 A)

Table 12. Equation parameters, passive mass calculation.

Equation parameter	Drum chamber	Crate chamber
Shielded coincidence coefficient hc_0	0.8092	0.8092
Shielded coincidence coefficient hc_1	0.2337	0.2337
System coincidence coefficient yc_0	0.5967	0.5967
System coincidence coefficient yc_1	0.4187	0.4187

Process

1. Compute the isotopic correction factor.

$$isotopic_cf = \begin{cases} \text{IF } fraction_Pu_240 < .01 \text{ or } fraction_Pu_240 > .99 \text{ THEN} \\ \quad 1.0 \\ \text{ELSE} \\ \quad \text{IF } neut2_simulation \text{ THEN} \\ \quad \quad \left(\frac{.0575}{fraction_Pu_240} \right) \\ \quad \text{ELSE} \\ \quad \quad \left(\frac{cal_fraction_Pu_240}{fraction_Pu_240} \right) \end{cases}$$

2. Compute the shielded coincidence correction factor.

$$shielded_coincidence_cf = \begin{cases} \text{IF } neut2_simulation \text{ THEN} \\ \quad \left(\frac{hc_0}{1.0 - moderator_index} + hc_1 \right)^2 \\ \text{ELSE} \\ \quad \text{IF } externally_supplied_shielded_coincidence_cf = 0.0 \text{ THEN} \\ \quad \quad \left(\frac{hc_0}{1.0 - moderator_index} + hc_1 \right)^2 \\ \text{ELSE} \\ \quad \quad externally_supplied_shielded_coincidence_cf \end{cases}$$

3. Compute the system coincidence correction factor.

$$system_coincidence_cf = \begin{cases} \text{IF } neut2_simulation \text{ THEN} \\ \quad \left(\frac{yc_0}{1.0 - moderator_index} + yc_1 \right)^2 \\ \text{ELSE} \\ \quad \text{IF } externally_supplied_system_coincidence_cf = 0.0 \text{ THEN} \\ \quad \quad \left(\frac{yc_0}{1.0 - moderator_index} + yc_1 \right)^2 \\ \text{ELSE} \\ \quad \quad externally_supplied_shielded_coincidence_cf \end{cases}$$

4. Compute the passive Pu mass.

```

IF neut2_simulation THEN
  factor = 1.0
ELSE
  factor = shielded_coincidence_rate
IF  $\left| \frac{\text{shielded\_coincidence\_rate\_error}}{\text{shielded\_coincidence\_rate}} \right| \leq \left| \frac{\text{system\_coincidence\_rate\_error}}{\text{system\_coincidence\_rate}} \right|$  AND factor > 0.0 THEN
  passive_Pu_mass = shielded_coincidence_zero_matrix_calibration_factor *
    . shielded_coincidence_cf * isotopic_cf * shielded_coincidence_rate
ELSE
  passive_Pu_mass = system_coincidence_zero_matrix_calibration_factor *
    system_coincidence_cf * isotopic_cf * system_coincidence_rate

```

Note: Test for $\text{shielded_coincidence_rate} > 0.0$ added in Version 2.0.

Output

- isotopic correction factor
- passive Pu mass in grams

M.8 Calculate passive Pu mass error

Input

- shielded coincidence rate and error (3.1.3.6 M.5, M.6)
- system coincidence rate and error (3.1.3.6 M.7, M.8)
- moderator index (M.5)
- calibration fraction Pu-240 (3.1.3.3 A)
- shielded coincidence zero matrix calibration factor (3.1.3.3 A)
- system coincidence zero matrix calibration factor (3.1.3.3 A)
- shielded coincidence correction factor
- system coincidence correction factor
- isotopic correction factor (M.7)
- NEUT2 simulation indicator (3.1.3.2 A)

Process

If NEUT2 simulation,

IF $\left(\frac{\text{shielded_coincidence_rate_error}}{\text{shielded_coincidence_rate}} \right) \leq \left(\frac{\text{system_coincidence_rate_error}}{\text{system_coincidence_rate}} \right)$ THEN
 $\text{passive_Pu_mass_error} = \text{shielded_coincidence_zero_matrix_calibration_factor} * \text{shielded_coincidence_cf} * \text{shielded_coincidence_rate_error}$
ELSE
 $\text{passive_Pu_mass_error} = \text{system_coincidence_zero_matrix_calibration_factor} * \text{system_coincidence_cf} * \text{system_coincidence_rate_error}$

Otherwise, use the error propagation for multiplication and division to compute the passive Pu mass error [23, 24].

1. Compute the isotopic correction factor error.

$$\text{isotopic_cf_error} = \begin{cases} \text{IF } \text{fraction_Pu_240} < .01 \text{ or } \text{fraction_Pu_240} > .99 \text{ THEN} \\ \quad 1.0 \\ \text{ELSE} \\ \quad \text{error} \left(\frac{\text{cal_fraction_Pu_240} \pm \text{cal_fraction_Pu_240_error}}{\text{fraction_Pu_240} \pm \text{fraction_Pu_240_error}} \right) \end{cases}$$

2. Compute the passive Pu mass error.

```

IF neut2_simulation THEN
  factor = 1.0
ELSE
  factor = shielded_coincidence_rate
  IF  $\left( \frac{\text{shielded\_coincidence\_rate\_error}}{\text{shielded\_coincidence\_rate}} \right) \leq \left( \frac{\text{system\_coincidence\_rate\_error}}{\text{system\_coincidence\_rate}} \right)$  AND factor > 0.0 THEN
    
$$\left( \begin{array}{l} \text{shielded\_coincidence\_zero\_matrix\_calibration\_factor} \\ \pm \text{shielded\_coincidence\_zero\_matrix\_calibration\_factor\_error} \end{array} \right) * \left( \begin{array}{l} \text{shielded\_coincidence\_cf} \pm \text{shielded\_coincidence\_cf\_error} \\ \text{shielded\_coincidence\_rate} \pm \text{shielded\_coincidence\_rate\_error} \end{array} \right) *$$

  passive_Pu_mass_error = error IF neut2_simulation THEN
    1.0 (1)
  ELSE
    (isotopic_cf ± isotopic_cf_error)
  ELSE
    
$$\left( \begin{array}{l} \text{system\_coincidence\_zero\_matrix\_calibration\_factor} \\ \pm \text{system\_coincidence\_zero\_matrix\_calibration\_factor\_error} \end{array} \right) * \left( \begin{array}{l} \text{system\_coincidence\_cf} \pm \text{system\_coincidence\_cf\_error} \\ \text{system\_coincidence\_rate} \pm \text{system\_coincidence\_rate\_error} \end{array} \right) *$$

  passive_Pu_mass_error = error IF neut2_simulation THEN
    1.0 (1)
  ELSE
    (isotopic_cf ± isotopic_cf_error)
  

```

(1) This reflects a known error in NEUT2.

Output

- passive Pu mass error in grams

M.9 Select either the active or passive Pu mass***Input***

- content code
- mass selection criterion

- active Pu mass in grams (M.5)
- passive Pu mass in grams (M.7)
- passive Pu mass error in grams (M.8)
- mass threshold
- NEUT2 simulation indicator (3.1.3.2 A)

Process

```

IF neut2_simulation THEN
  Pu_mass =
    { IF mass_selection_criteria = passive THEN
      passive_Pu_mass
    ELSE IF active_Pu_mass ≤ 20.0 THEN
      active_Pu_mass
    ELSE IF content_code ∈ {1,2,3,4,7} THEN
      active_Pu_mass
    ELSE IF passive_Pu_mass_error < 40.0 OR passive_Pu_mass < 200.0 THEN
      passive_Pu_mass
    ELSE
      active_Pu_mass
  }
ELSE
  Pu_mass =
    { IF mass_selection_criteria = active THEN
      active_Pu_mass
    ELSE IF mass_selection_criteria = passive THEN
      passive_Pu_mass
    ELSE IF active_Pu_mass < mass_threshold AND flux_rates_ok THEN (1)
      active_Pu_mass
    ELSE IF passive_Pu_mass < 0.0 OR bad_passive_data THEN (2)
      active_Pu_mass
    ELSE
      passive_Pu_mass
  }

```

⁽¹⁾ Changed in Version 2.0.

⁽²⁾ Added in Version 2.0.

Output

- Pu mass in grams

M.10 Select either the active or passive Pu mass error***Input***

- content code
- mass selection criterion
- active Pu mass in grams (M.5)
- active Pu mass error in grams (M.6)
- passive Pu mass in grams (M.7)
- passive Pu mass error in grams (M.8)
- mass threshold = 10.0
- NEUT2 simulation indicator (3.1.3.2 A)

Process

```

IF neut2_simulation THEN
  Pu_mass_error = {
    IF mass_selection_criteria = passive THEN
      passive_Pu_mass_error
    ELSE IF active_Pu_mass ≤ 20.0 THEN
      active_Pu_mass_error
    ELSE IF content_code ∈ {1,2,3,4,7} THEN
      active_Pu_mass_error
    ELSE IF passive_Pu_mass_error < 40.0 OR passive_Pu_mass < 200.0 THEN
      passive_Pu_mass_error
    ELSE
      active_Pu_mass_error
  }
ELSE
  Pu_mass_error = {
    IF mass_selection_criteria = active THEN
      active_Pu_mass_error
    ELSE IF mass_selection_criteria = passive THEN
      passive_Pu_mass_error
    ELSE IF active_Pu_mass < mass_threshold AND flux_rates_ok THEN(1)
      active_Pu_mass_error
    ELSE IF passive_Pu_mass < 0.0 OR bad_passive_data THEN(2)
      active_Pu_mass_error
    ELSE
      passive_Pu_mass_error
  }

```

(1) Changed in Version 2.0.

(2) Added in Version 2.0.

Output

- Pu mass error in grams

M.11 Calculate plutonium isotopic masses

This is a new requirement added in Version 2.0. The isotopic mass for Pu-242 is not currently required.

Input

- active isotopic correction factor (M.5)
- passive isotopic correction factor (M.7)
- Pu mass (M.9)
- *if* NEUT2 simulation
 - Pu-239 calibration fraction = 0.9425
 - Pu-240 calibration fraction = 0.0575
- *else*
 - Pu-239 calibration fraction
 - Pu-240 calibration fraction
- plutonium fractions

Process

IF *Pu_mass_from_active* THEN

$$Pu239_mass = \frac{Pu_mass * Pu239_cal_mass_fraction}{active_isotopic_cf}$$

$$Pu[i]_mass = \sum_{i=238}^{241, \neq 239} Pu_mass * Pu[i]_mass_fraction$$

ELSE

$$Pu240_mass = \frac{Pu_mass * Pu240_cal_mass_fraction}{passive_isotopic_cf}$$

$$Pu[i]_mass = \sum_{i=238}^{241, \neq 240} Pu_mass * Pu[i]_mass_fraction$$

Output

- Pu-238 mass
- Pu-239 mass
- Pu-240 mass

- Pu-241 mass

M.12 Calculate plutonium isotopic mass errors

This requirement added in Version 2.0.

Input

- active isotopic correction factor (M.5)
- passive isotopic correction factor (M.7)
- Pu mass error (M.10)
- *if* NEUT2 simulation:
 - Pu-239 calibration fraction = 0.9425
 - Pu-240 calibration fraction = 0.0575
- *else*
 - Pu-239 calibration fraction
 - Pu-240 calibration fraction
- plutonium fractions

Process

Calculate one standard deviation errors for the plutonium isotopic mass values using standard error propagation techniques described in References [23] and [24].

Output

- Pu-238 mass error
- Pu-239 mass error
- Pu-240 mass error
- Pu-241 mass error

M.13 Calculate U-235 mass

Input

- Pu-239 mass (M.11)
- U-235/Pu-239 mass ratio

Process

This process is not performed in NEUT2 simulation.

Compute U-235 mass.

$$U235_mass = Pu239_mass * U235_mass_ratio$$

Output

- U-235 mass

M.14 Calculate U-235 mass error*Input*

- Pu-239 mass (M.11) and error (M.12)
- U-235/Pu-239 mass ratio and error
- U-235 correction factor (M.5)

Process

This process is not performed in NEUT2 simulation.

1. Compute U-235 mass error from Pu-239 mass error and mass ratio error.

$$U235_mass_error = \text{error}([Pu239_mass \pm Pu239_mass_error] * [U235_ratio \pm U235_ratio_error])$$

2. If Pu-239 mass was determined from active assay, apply correlation correction [25].

IF *Pu_mass_from_active* THEN

$$\left. \begin{cases} correlation_cf = 2.0 * (Pu239_mass)^2 * \frac{1.0 - U235_cf}{U235_cf} \\ U235_mass_error = \sqrt{(U235_mass_error)^2 + correlation_cf * (U235_mass_ratio_error)^2} \end{cases} \right\}$$

M.15 Calculate the Am-241 mass*Input*

- content code
- passive shielded total rate (3.1.3.6 M.1)
- passive system total rate (3.1.3.6 M.3)
- moderator index (M.5)
- active Pu mass (M.5)
- Pu mass (M.9)
- Am sludge mass coefficient $sc_0 = 1.12$
- Am sludge mass coefficient $sc_1 = 76.0$
- Am sludge mass coefficient $sc_2 = 2.93E-4$
- shielded detector efficiency (3.1.3.3 A)
- specific activity Am 241 = 3.47
- NEUT2 simulation indicator (3.1.3.2 A)
- if not NEUT2 simulation, then⁽¹⁾
 - Am-241/U-235 mass ratio
 - Am-241/Pu-239 mass ratio

⁽¹⁾ Added in Version 2.0; mass ratio data obtained from SGRS.

Process

If no SGRS Am-241 mass ratio data are available or if NEUT2 simulation, then estimate the Am-241 mass from SAS measurement data.

1. Compute Am mass from decay of Pu 241. In Version 2.0, the "decayed" value of the Pu-241 mass fraction is used, rather than the undecayed value. Therefore, when NEUT2 simulation is not in effect, the "grown" mass of ^{241}Am is set equal to the ^{241}Pu mass.

$$grown = \begin{cases} \text{IF } neut2_simulation \text{ THEN} \\ \frac{Pu_mass * 0.0034}{2.0} \\ \text{ELSE} \\ Pu_mass * fraction_Pu_241 \end{cases}$$

2. Calculate Am mass from the measured Pu mass and "neutron excess".

$$mass = \frac{numer}{denom}$$

where

$$numer = \frac{passive_shielded_total_rate * count_rate_cf}{sheilded_detector_efficiency} - Pu_mass * fraction_Pu_240 * 1000.0 * 5.83$$

with

$$count_rate_cf = \frac{1.0}{1.0 - moderator_index}$$

and

$$denom = \begin{cases} \text{IF } neut2_simulation \text{ THEN} \\ 3.23 * 1000.0 * 10.0 \\ \text{ELSE} \\ Am241_specific_activity * 1000.0 * 20.0 \end{cases}$$

3. Select Am mass.

```

IF Am_bearing_sludge THEN
  Am_mass = Am_sludge_mass
ELSE
  IF mass < 0.0 THEN
    Am_mass = grown
  ELSE
    Am_mass = grown + mass

```

where .

$$\text{Am_sludge_mass} = (sc_0 * \text{passive_system_total_rate} - sc_1 * \text{active_Pu_mass}) * sc_2$$

If not simulating NEUT2 and Am-241/Pu-241 and/or Am-241/U-235 mass ratio data are available from the SGRS, then calculate the Am-241 mass as follows (new for Version 2.0):

```

IF (U235 / Pu239)_ratio ≥ 1.0 AND (Am241 / U235)_ratio > 0.0 THEN
  Am_mass = (Am241 / U235)_ratio * U235_mass
ELSE IF (Am241 / Pu239)_ratio > 0.0 AND (Am241 / U235)_ratio ≤ 0.0 THEN
  Am_mass = (Am241 / Pu239)_ratio * Pu239_mass
ELSE IF (Am241 / Pu239)_mass ≤ 0.0 AND (Am241 / U235)_ratio > 0.0 THEN
  Am_mass = (Am241 / U235)_ratio * U235_mass
ELSE IF (Am241 / Pu239)_ratio > 0.0 AND (Am241 / U235)_ratio > 0.0 THEN
  the mass ratio resulting in a positive mass value with the smallest relative error shall be used
ELSE
  Am241_mass cannot be determined from mass ratio data

```

Output

Am-241 mass in grams

M.16 Calculate the Am mass error

Input

- content code
- passive shielded total rate error (3.1.3.6 M.2)
- passive system total rate error (3.1.3.6 M.4)
- moderator index (M.5)
- active Pu mass error (M.6)
- Pu mass error (M.10)
- Am sludge mass coefficient $sc_0 = 1.12$
- Am sludge mass coefficient $sc_1 = 76.0$

- Am sludge mass coefficient $sc_2 = 2.93E-4$
- Am sludge mass coefficient $sc_3 = 0.30$
- shielded detector efficiency (3.1.3.3 A)
- specific activity Am 241 = 3.47
- NEUT2 simulation indicator (3.1.3.2 A)
- *if not NEUT2 simulation, then*⁽¹⁾
 - Am-241/U-235 mass ratio
 - Am-241/Pu-239 mass ratio

⁽¹⁾ Added in Version 2.0; mass ratio data obtained from SGRS.

Process

If not simulating NEUT2 and Am-241/Pu-241 and/or Am-241/U-235 mass ratio data are available from the SGRS, then propagate the error from the mass ratio and isotopic mass used to calculate the Am-241 mass.

If not simulating NEUT2 or no mass ratio values available, calculate the Am-241 mass error as follows:

1. Compute error in Am mass from decay of Pu 241.

$$grown_error = \begin{cases} \text{IF } neut2_simulation \text{ THEN} \\ \quad \frac{Pu_mass_error * 0.00034}{2.0}^{(1)} \\ \text{ELSE} \\ \quad Pu_mass_error * fraction_Pu_241 \end{cases}$$

⁽¹⁾ This reflects a known error in NEUT2; value should be 0.0034.

2. Calculate error in Am mass from measured Pu mass.

$$\text{mass_error} = \frac{\sqrt{\text{term1}^2 + \text{term2}^2}}{\text{denom}}$$

where

$$\text{term1} = \text{error} \left(\frac{\text{passive_shielded_total_rate} * \text{count_rate_cr}}{\text{shielded_detector_efficiency}} \right)$$

with

$$\text{count_rate_cr} = \frac{1.0}{1.0 - \text{moderator_index}}$$

and

$$\text{term2} = \text{error} (\text{Pu_mass} * \text{fraction_Pu_240} * 1000.0 * 5.83)$$

and

$$\text{denom} = \begin{cases} \text{IF neut2_simulation THEN} \\ \quad 3.23 * 1000.0 * 10.0 \\ \text{ELSE} \\ \quad \text{Am241_specific_activity} * 1000.0 * 20.0 \end{cases}$$

3. Select Am mass error.

$$\text{Am_mass_error} = \begin{cases} \text{IF Am_bearing_sludge THEN} \\ \quad \text{Am_mass_error} = \text{Am_sludge_mass_error} \\ \text{ELSE} \\ \quad \text{IF mass} < 0.0 \text{ THEN} \\ \quad \quad \text{Am_mass_error} = \text{error} (\text{grown} \pm \text{grown_error}) \\ \text{ELSE} \\ \quad \quad \text{Am_mass_error} = \text{error} ([\text{grown} \pm \text{grown_error}] + [\text{mass} \pm \text{mass_error}]) \end{cases}$$

where

$$\text{Am_sludge_mass_error} =$$

$$\begin{cases} \text{IF neut2_simulation THEN} \\ \quad \text{sc}_3 * \text{Am_sludge_mass} \\ \text{ELSE} \\ \quad \text{sc}_2 * \sqrt{\text{sc}_0^2 * \text{passive_system_total_rate_error}^2 + \text{sc}_1^2 * \text{active_Pu_mass_error}^2} \end{cases}$$

Output

- Am-241 mass error in grams

M.17 Calculate thermal power*Input*

- Pu mass (M.9)

If NEUT2 simulation (3.1.3.2 A),

- fraction Pu-238 = .00012
- fraction Pu-239 = .93826
- fraction Pu-240 = .0582
- fraction Pu-241 = .0034
- fraction Pu-242 = .00024

else

- fraction Pu-238
- fraction Pu-239
- fraction Pu-240
- fraction Pu-241
- fraction Pu-242
- thermal power Pu-238 = 5.73E-01
- thermal power Pu-239 = 1.95E-03
- thermal power Pu-240 = 7.16E-03
- thermal power Pu-241 = 3.31E-03
- thermal power Pu-242 = 1.17E-04
- Am-241 mass (M.7)
- thermal power Am-241 = 1.16E-01

Process

Compute thermal power.

$$\text{thermal_power} = \text{Pu_thermal_power} + \text{Am_thermal_power}$$

where

$$\text{Pu_thermal_power} = \text{Pu_mass} * \sum_{i=\text{Pu_238}}^{\text{Pu_242}} \text{fraction}_i * \text{thermal_power}_i$$

and

$$\text{Am_thermal_power} = \text{Am_mass} * \text{Am_241_thermal_power}$$

Output

- thermal power in Watts

M.18 Calculate the thermal power error*Input*

- Pu mass error (M.10)

If NEUT2 simulation (3.1.3.2 A),

- fraction Pu-238 = .00012
- fraction Pu-239 = .93826
- fraction Pu-240 = .0582
- fraction Pu-241 = .0034
- fraction Pu-242 = .00024

else

- fraction Pu-238
- fraction Pu-239
- fraction Pu-240
- fraction Pu-241
- fraction Pu-242
- thermal power Pu-238 = 5.73E-01
- thermal power Pu-239 = 1.95E-03
- thermal power Pu-240 = 7.16E-03
- thermal power Pu-241 = 3.31E-03
- thermal power Pu-242 = 1.17E-04
- Am mass error (M.8)
- thermal power Am-241 = 1.16E-01

Process

Compute thermal power error.

$$\text{thermal_power_error} = \sqrt{\text{Pu_thermal_power_error}^2 + \text{Am_thermal_power_error}^2}$$

where

$$\text{Pu_thermal_power_error} = \text{error} \left(\text{Pu_mass} * \sum_{i=\text{Pu_238}}^{\text{Pu_242}} \text{Pu_mass_fraction_error}_i * \text{thermal_power}_i \right)$$

$$\text{Am_thermal_power_error} = \text{Am_mass_error} * \text{Am_241_thermal_power}$$

Output

- thermal power error in Watts

M.19 Calculate the thermal power density*Input*

- thermal power in Watts (M.17)

- normalization factor = 28.3 liter/ft³
- container volume (3.1.4.2 A)
- NEUT2 simulation indicator (3.1.3.2 A)

Process

Compute the thermal power density.

IF *neut2_simulation* THEN

$$\text{thermal_power_density} = \frac{\text{thermal_power}}{7.35}$$

ELSE

$$\text{thermal_power_density} = \frac{\text{thermal_power}}{\text{container_volume}} * \text{normalization_factor}$$

Output

- thermal power density in Watts/ft³

M.20 Calculate the thermal power density error

Input

- thermal power error in Watts (M.18)
- normalization factor = 28.3 liter/ft³
- container volume (3.1.4.2 A)
- NEUT2 simulation indicator (3.1.3.2 A)

Process

Compute the thermal power density error.

IF *neut2_simulation* THEN

$$\text{thermal_power_density_error} = \frac{\text{thermal_power_error}}{7.35}$$

ELSE

$$\text{thermal_power_density_error} = \frac{\text{thermal_power_error}}{\text{container_volume}} * \text{normalization_factor}$$

Output

- thermal power density error in Watts/ft³

M.21 Calculate activity values

In the initial SAS program version (as well as NEUT2), the total alpha activity due to the plutonium and americium determined to be present in a waste container was calculated. However, other activities are also required in order to satisfy various radioactive waste storage and transportation criteria. The needed activities are "Total Activity" (activity due to all decay modes,

not just alpha decay, of all nuclides determined to be present in "significant quantity"), "Alpha Activity" (corresponding to the original calculated quantity), "TRU Activity" (activity from the TRU components of the waste), and "Plutonium Equivalent Activity". All required activities will be calculated in SAS Version 2.0. It must be possible to modify the parameter values used in these calculations without modifying the program code. The parameter values shown below (other than the mass fraction values used in NEUT2) were taken from Reference [27].

Input

- Pu mass (M.9)

If NEUT2 simulation (3.1.3.2 A),

- fraction Pu-238 = .00012
- fraction Pu-239 = .93826
- fraction Pu-240 = .0582
- fraction Pu-241 = .0034
- fraction Pu-242 = .00024

else

- fraction Pu-238
- fraction Pu-239
- fraction Pu-240
- fraction Pu-241
- fraction Pu-242
- alpha branching ratio Pu-238 = 1.0
- alpha branching ratio Pu-239 = 1.0
- alpha branching ratio Pu-240 = 1.0
- alpha branching ratio Pu-241 = 2.45E-05
- alpha branching ratio Pu-242 = 1.0
- specific activity Pu-238 = 1.73E+01
- specific activity Pu-239 = 6.29E-02
- specific activity Pu-240 = 2.3E-01
- specific activity Pu-241 = 1.04E+02
- specific activity Pu-242 = 3.97E-03
- equivalent Pu factor Pu-238 = 1.10
- equivalent Pu factor Pu-239 = 1.0
- equivalent Pu factor Pu-240 = 1.0
- equivalent Pu factor Pu-241 = 52.0
- equivalent Pu factor Pu-242 = 1.10
- U-235 mass (M.13)
- alpha branching ratio U-235 = 1.0
- specific activity U-235 = 2.19E-06

- equivalent Pu factor U-235 = 0.0
- Pu mass (M.9)
- Am-241 mass (M.11)
- alpha branching ratio Am-241 = 1.0
- specific activity Am-241 = 3.47
- equivalent Pu factor Am-241 = 1.0

Process

1. Compute the total activity.

$$\begin{aligned} \text{total_activity} = & \text{Pu_mass} * \sum_{i=\text{Pu_238}}^{\text{Pu_242}} \text{Pu_mass_fraction}_i * \text{specific_activity}_i \\ & + \text{U235_mass} * \text{U235_specific_activity} \\ & + \text{Am241_mass} * \text{Am241_specific_activity} \end{aligned}$$

2. Compute the alpha activity.

$$\begin{aligned} \text{alpha_activity} = & \text{Pu_mass} * \sum_{i=\text{Pu_238}}^{\text{Pu_242}} \text{Pu_mass_fraction}_i * \text{specific_activity}_i * \text{alpha_branching_ratio}_i \\ & + \text{U235_mass} * \text{U235_specific_activity} * \text{U235_alpha_branching_ratio} \\ & + \text{Am241_mass} * \text{Am241_specific_activity} * \text{Am241_alpha_branching_ratio} \end{aligned}$$

3. Compute the TRU activity.

$$\begin{aligned} \text{TRU_activity} = & \text{Pu_mass} * \sum_{i=\text{Pu_238}}^{\text{Pu_242}, \neq \text{Pu_241}} \text{Pu_mass_fraction}_i * \text{specific_activity}_i * \text{alpha_branching_ratio}_i \\ & + \text{Am241_mass} * \text{Am241_specific_activity} * \text{Am241_alpha_branching_ratio} \end{aligned}$$

4. Compute the plutonium equivalent activity.

$$\text{Pu_equivalent_activity} =$$

$$\begin{aligned} & \text{Pu_mass} * \sum_{i=\text{Pu_238}}^{\text{Pu_242}} \text{Pu_mass_fraction}_i * \text{specific_activity}_i * \text{alpha_branching_ratio}_i \\ & \quad / \text{equivalent_pu_factor}_i \\ & + \text{Am241_mass} * \text{Am241_specific_activity} * \text{Am241_alpha_branch} / \text{Am241_equiv_pu_factor} \\ & + \begin{cases} \text{IF } \text{U235_equiv_pu_factor} > 0.0 \text{ THEN} \\ \quad \text{U235_mass} * \text{U235_specific_activity} * \text{U235_alpha_branch} / \text{U235_equiv_pu_factor} \\ \text{ELSE} \\ \quad 0.0 \end{cases} \end{aligned}$$

Output

- total activity in Curies

- alpha activity in Curies
- TRU activity in Curies
- equivalent plutonium activity in Curies

M.22 Calculate the activity errors

Input

- Pu mass (M.9)

If NEUT2 simulation (3.1.3.2 A),

- fraction Pu-238 = .00012
- fraction Pu-239 = .93826
- fraction Pu-240 = .0582
- fraction Pu-241 = .0034
- fraction Pu-242 = .00024

else

- fraction Pu-238
- fraction Pu-239
- fraction Pu-240
- fraction Pu-241
- fraction Pu-242
- alpha branching ratio Pu-238 = 1.0
- alpha branching ratio Pu-239 = 1.0
- alpha branching ratio Pu-240 = 1.0
- alpha branching ratio Pu-241 = 2.45E-05
- alpha branching ratio Pu-242 = 1.0
- specific activity Pu-238 = 1.73E+01
- specific activity Pu-239 = 6.29E-02
- specific activity Pu-240 = 2.3E-01
- specific activity Pu-241 = 1.04E+02
- specific activity Pu-242 = 3.97E-03
- equivalent Pu factor Pu-238 = 1.10
- equivalent Pu factor Pu-239 = 1.0
- equivalent Pu factor Pu-240 = 1.0
- equivalent Pu factor Pu-241 = 52.0
- equivalent Pu factor Pu-242 = 1.10
- U-235 mass (M.13)
- alpha branching ratio U-235 = 1.0
- specific activity U-235 = 2.19E-06

- equivalent Pu factor U-235 = 0.0
- Pu mass (M.9)
- Am-241 mass (M.16)
- alpha branching ratio Am-241 = 1.0
- specific activity Am-241 = 3.47
- equivalent Pu factor Am-241 = 1.0

Process

1. Compute the total activity error.

total_activity_error =

$$\sqrt{Pu_total_activity_error^2 + U235_total_activity_error^2 + Am_total_activity_error^2}$$

where

$$Pu_total_activity_error = \text{error} \left(Pu_mass * \sum_{i=Pu_238}^{Pu_242} Pu_mass_fraction_error_i * specific_activity_i \right)$$

$$U235_total_activity_error = U235_mass_error * U235_specific_activity$$

$$Am_total_activity_error = Am241_mass_error * Am241_specific_activity$$

2. Compute the alpha activity error.

alpha_activity_error =

$$\sqrt{Pu_alpha_activity_error^2 + U235_alpha_activity_error^2 + Am_alpha_activity_error^2}$$

where

Pu_alpha_activity_error =

$$\text{error} \left(Pu_mass * \sum_{i=Pu_238}^{Pu_242} Pu_mass_fraction_i * specific_activity_i * alpha_branching_ratio_i \right)$$

$$U235_alpha_activity_error = U235_mass_error * U235_specific_activity * U235_alpha_branching_ratio$$

$$Am_alpha_activity_error = Am241_mass_error * Am241_specific_activity * Am241_branching_ratio$$

3. Compute the TRU activity error.

$$TRU_activity_error = \sqrt{Pu_TRU_activity_error^2 + Am_TRU_activity_error^2}$$

where

Pu_TRU_activity_error =

$$\text{error} \left(Pu_mass * \sum_{i=Pu_238}^{Pu_242, \neq Pu_241} Pu_mass_fraction_i * specific_activity_i * alpha_branching_ratio_i \right)$$

$$Am_TRU_activity_error = Am241_mass_error * Am241_specific_activity * Am241_branching_ratio$$

4. Compute the plutonium equivalent activity error.

Pu_equiv_activity_error =

$$\sqrt{Pu_Pu_equiv_activity_error^2 + U_Pu_equiv_activity_error^2 + Am_Pu_activity_error^2}$$

where

Pu_Pu_equiv_activity_error =

$$\text{error} \left(\sum_{i=Pu_238}^{Pu_242} Pu_mass * Pu_mass_fraction_i * specific_activity_i * alpha_branch_i / equivalent_pu_factor_i \right)$$

U_Pu_equiv_activity_error =

$$\begin{cases} \text{IF } U235_equiv_pu_factor > 0 \text{ THEN} \\ \quad U235_mass_error * U235_specific_activity * U235_alpha_branch / U235_equiv_pu_factor \\ \text{ELSE} \\ \quad 0.0 \end{cases}$$

$$\begin{aligned} Am_Pu_activity_error = & Am241_mass_error * Am241_specific_activity * Am241_alpha_branch \\ & / Am241_equiv_pu_factor \end{aligned}$$

Output

- total activity error in Curies
- alpha activity error in Curies
- TRU activity error in Curies
- plutonium equivalent activity error in Curies

M.23 Calculate the TRU activity concentration

In the initial software version, a quantity called the “total alpha activity concentration” was calculated; this was a direct carry-over from the NEUT2 program. The activity concentration of regulatory interest is the “TRU activity concentration”. This is the quantity to be calculated in SAS Version 2.0.

Input

- TRU activity (M.21)
- normalization factor = 1.0E+06 (nCi/Ci * kg/g)
- net waste matrix weight in kg

Process

Compute the activity concentration.

$$TRU_activity_concentration = \frac{TRU_activity}{net_matrix_weight} * normalization_factor$$

Output

- TRU activity concentration in nCi/g

M.24 Calculate the TRU activity concentration error

Input

- TRU activity error (M.22)
- normalization factor = 1.0E+06 (nCi/Ci * kg/g)
- net waste matrix weight in kg

Process

Compute the activity concentration error.

$$TRU_activity_concentration_error = \frac{TRU_activity_error}{net_matrix_weight} * normalization_factor$$

Output

- TRU activity concentration error in nCi/g

M.25 Calculate Fissile Gram Equivalent

This quantity was not calculated by NEUT2 and was added to SAS Version 2.0.

Input

- plutonium isotopic masses (M.11)
- Pu-238 fissile gram equivalent = 0.113
- Pu-239 fissile gram equivalent = 1.00
- Pu-240 fissile gram equivalent = 0.0225
- Pu-241 fissile gram equivalent = 2.25
- U-235 mass (M.14)
- U-235 fissile gram equivalent = 1.00
- Am-241 mass (M.15)
- Am-241 fissile gram equivalent = 0.0187

Process

$$fissile_gram_equivalent = Pu_mass * \sum_{i=238}^{241} Pu[i]_mass_fraction * Pu[i]_fissile_gram_equivlent \\ + U235_mass * U235_fissile_gram_equivalent \\ + Am241_mass * Am241_fissile_gram_equivalent$$

Output

- fissile gram equivalent in grams

M.26 Calculate Fissile Gram Equivalent error

This quantity was not calculated by NEUT2 and was added to SAS Version 2.0.

Input

- plutonium isotopic mass errors (M.12)
- Pu-238 fissile gram equivalent = 0.113
- Pu-239 fissile gram equivalent = 1.00
- Pu-240 fissile gram equivalent = 0.0225
- Pu-241 fissile gram equivalent = 2.25
- U-235 mass error (M.14)
- U-235 fissile gram equivalent = 1.00
- Am-241 mass error (M.16)
- Am-241 fissile gram equivalent = 0.0187

Process

$$\text{fissile_gram_equiv_error} = \text{error} \left(\begin{array}{l} \text{Pu_mass} * \sum (\text{Pu}[i]_{\text{mass_fraction}} * \text{Pu}[i]_{\text{fissile_gram_equivalent}})^2 \\ + (\text{U235_mass_error} * \text{U235_fissile_gram_equivalent})^2 \\ + (\text{Am241_mass_error} * \text{Am241_fissile_gram_equivalent})^2 \end{array} \right)$$

Output

- fissile gram equivalent error in grams

M.27 Provide a measurement summary report for the container*Input*

- Pu mass (M.9) and Pu mass error (M.10)
- individual plutonium isotopic mass values (M.11) and errors (M.12)
- U-235 mass (M.13) and U-235 mass error (M.14)
- Am-241 mass (M.15) and Am-241 mass error (M.16)
- thermal power (M.17) and thermal power error (M.18)
- thermal power density (M.19) and thermal power density error (M.20)
- activities (M.21) and activity errors (M.22)
- TRU activity concentration (M.23) and TRU activity concentration error (M.24)
- fissile gram equivalent (M.25) and fissile gram equivalent error (M.26)

Process

Write the inputs to a report. A “fixed point” format should be used for values within the range [.01, 999] and a “floating point” format (e.g., scientific notation) used for smaller and larger values. Numeric values and associated error values must not be truncated to less than the number of significant digits dictated by the error values, nor should an excessive number of digits be displayed (see Reference [24], Chapter 2 for guidance).

Notes shall be appended to the report to indicate:

- whether the total plutonium mass value is based on active or passive assay results
- source of the mass fraction values used to determine plutonium isotopic masses
- the method use to determine the Am-241 mass

Output⁷

Measurement Results:

- Total Pu mass: (999999 \pm 999999) g
- U-235 mass:⁸ (999999 \pm 999999) g
- Pu-238 mass: (999999 \pm 999999) g
- Pu-239 mass: (999999 \pm 999999) g
- Pu-240 mass: (999999 \pm 999999) g
- Pu-241 mass: (999999 \pm 999999) g
- Am-241 mass: (999999 \pm 999999) g
- Thermal power: (999999 \pm 999999) Watts
- Thermal power density: (999999 \pm 999999) Watts/ft**3
- Total activity: (999999 \pm 999999) Ci
- Alpha activity: (999999 \pm 999999) Ci
- TRU activity: (999999 \pm 999999) Ci
- TRU activity concentration: (999999 \pm 999999) nCi/g
- Pu Equivalent Curies: (999999 \pm 999999) Ci
- Fissile Gram Equivalent: (999999 \pm 999999) g

3.1.5 DMS Interface Functional Requirements

This section identifies the functional requirements necessary for the SAS to interface with the Data Management System (DMS) computer.

⁷ The format used to display a value and its uncertainty shall be dictated by the absolute magnitude of a value, and the number of significant digits dictated by the uncertainty in the value. The number formats shown are representative only.

⁸ Only if the presence of U-235 is indicated by SGRS measurement results.

3.1.5.1 State Transition Diagrams

The state transitions for DMS are shown in Figure 15.

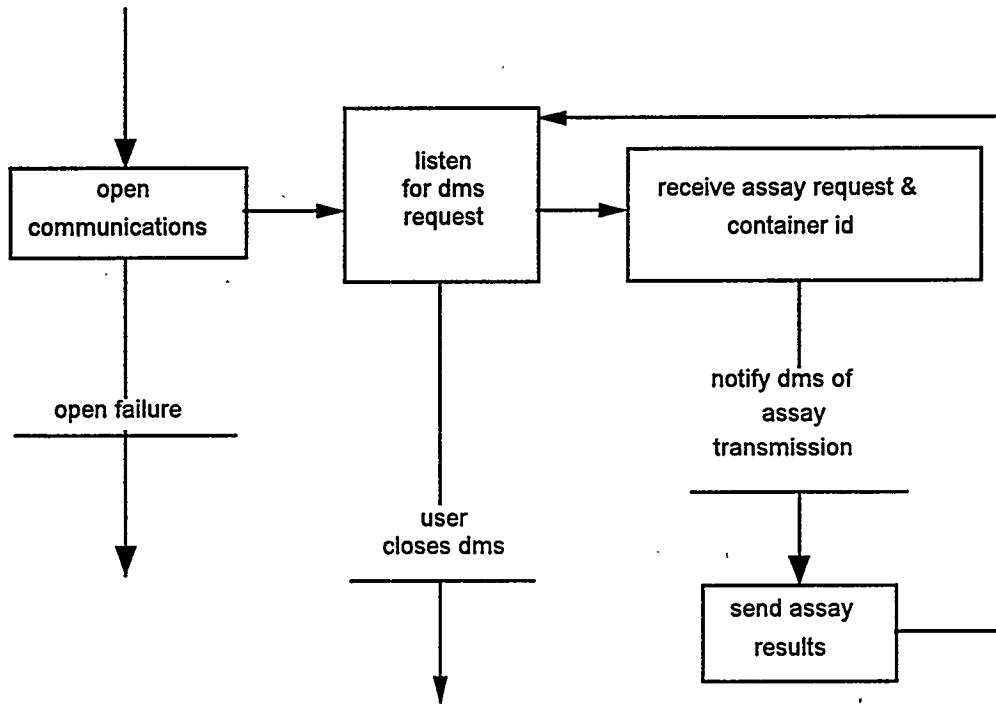


Figure 15. DMS state transition diagram.

3.1.5.2 DMS Requirements

A. DMS Attributes

- file identifier, a character string, e.g., six-character bar code

M. DMS Methods

M.1 Send "handshake" message to DMS

Input

None.

Process

Send handshake to DMS.

Output

- message code = 0
- OK code = 1

M.2 Send measurement results to DMS

The fields sent to and received from the DMS system were changed after the initial SAS software release. The following specifies the measurement results to be sent to the DMS by SAS Version 2.0.

Input

- file identifier, container bar code
- file extension (sequence number)
- SGRS file identification string
- date and time of SAS measurement
- date and time of SGRS measurement
- active Pu mass (3.1.4.3 M.5)
- active Pu mass error (3.1.4.3 M.6)
- passive Pu mass (3.1.4.3 M.7)
- passive Pu mass error (3.1.4.3 M.8)
- total Pu mass (3.1.4.3 M.9)
- total Pu mass error (3.1.4.3 M.10)
- fissile gram equivalent (3.1.4.3 M.25)
- fissile gram equivalent error (3.1.4.3 M.26)
- fissile gram equivalent total uncertainty⁽¹⁾
- alpha activity (3.1.4.3 M.21)
- alpha activity error (3.1.4.3 M.22)
- alpha activity total uncertainty⁽¹⁾
- TRU activity (3.1.5.3 M.21)
- TRU activity error (3.1.5.3 M.22)
- TRU activity total uncertainty⁽¹⁾
- thermal power (3.1.5.3 M.18)
- thermal power error (3.1.5.3 M.19)
- thermal power total uncertainty⁽¹⁾
- plutonium equivalent activity (3.1.5.3 M.21)
- plutonium equivalent activity error (3.1.5.3 M.22)
- plutonium equivalent activity total uncertainty⁽¹⁾
- Pu-238 mass (3.1.4.3 M.11)
- Pu-238 mass error (3.1.4.3 M.12)
- Pu-238 mass total uncertainty⁽¹⁾
- Pu-239 mass (3.1.4.3 M.11)
- Pu-239 mass error (3.1.4.3 M.12)
- Pu-239 mass total uncertainty⁽¹⁾

- Pu-240 mass (3.1.4.3 M.11)
- Pu-240 mass error (3.1.4.3 M.12)
- Pu-240 mass total uncertainty⁽¹⁾
- Pu-241 mass (3.1.4.3 M.11)
- Pu-241 mass error (3.1.4.3 M.12)
- Pu-241 mass total uncertainty⁽¹⁾
- Am-241 mass (3.1.4.3 M.15)
- Am-241 mass error (3.1.4.3 M.16)
- Am-241 mass total uncertainty⁽¹⁾
- U-235 mass (3.1.4.3 M.13)
- U-235 mass error (3.1.4.3 M.14)
- U-235 mass total uncertainty⁽¹⁾
- passive shielded total net rate (3.1.3.6 M.1)
- active flux monitor net rate (3.1.3.5 M.1)
- active barrel flux monitor net rate (3.1.3.5 M.3)
- active pulse count (number of neutron generator pulses)

⁽¹⁾ "Total uncertainty" is currently undefined.

Process

Send inputs to DMS.

Output

- message code = 3
- file identifier, container bar code
- file extension (sequence number)
- SGRS file identification string
- date and time of SAS measurement (MM/DD/YYYY:hh:mm:ss)
- date and time of SGRS measurement (MM/DD/YYYY: hh:mm:ss)⁽¹⁾
- active Pu mass
- active Pu mass error
- passive Pu mass
- passive Pu mass error
- total Pu mass
- total Pu mass error
- fissile gram equivalent
- fissile gram equivalent error
- fissile gram equivalent total uncertainty⁽²⁾
- alpha activity

- alpha activity error
- alpha activity total uncertainty⁽²⁾
- TRU activity
- TRU activity error
- TRU activity total uncertainty⁽²⁾
- thermal power
- thermal power error
- thermal power total uncertainty⁽²⁾
- plutonium equivalent activity
- plutonium equivalent activity error
- plutonium equivalent activity total uncertainty⁽²⁾
- Pu-238 mass
- Pu-238 mass error
- Pu-238 mass total uncertainty⁽²⁾
- Pu-239 mass
- Pu-239 mass error
- Pu-239 mass total uncertainty⁽²⁾
- Pu-240 mass
- Pu-240 mass error
- Pu-240 mass total uncertainty⁽²⁾
- Pu-241 mass
- Pu-241 mass error
- Pu-241 mass total uncertainty⁽²⁾
- Am-241 mass
- Am-241 mass error
- Am-241 mass total uncertainty⁽²⁾
- U-235 mass⁽³⁾
- U-235 mass error⁽³⁾
- U-235 mass total uncertainty⁽²⁾
- passive shielded total net rate
- active flux monitor net rate
- active barrel flux monitor net rate
- active pulse count (number of neutron generator pulses)

⁽¹⁾ If not available, send "01/01/0000:00:00:00".

⁽²⁾ Send "-1.00000E+00" to indicate that no value is available.

⁽³⁾ Send "-1.00000E+00" if U-235 mass was not determined.

M.3 Receive "handshake" message from DMS

Input

- message code = 0
- OK code = 1

Process

Receive handshake from DMS.

Output

None.

M.4 Receive new assay request from DMS

Input

- message code = 1
- container identification
- file identifier
- matrix content code
- chamber designation {D, C}
- matrix weight in kg (net)
- volume ID (used by operators to label archive disks)

Process

1. Send "handshake" to DMS, acknowledging receipt of request.
2. Enable automatic summary report. The measurement results are automatically printed following the data acquisition and data reduction.
3. Enable automatic background correction. Background correction is required when running the Assay System from DMS.
4. Acquire active and passive data from the assay chamber.
5. Compute the Pu mass, etc. as described in requirements 3.1.4.3 M.5 through M.26.
6. Send measurement results to DMS (M.2).

Output

None.

M.5 Receive recalculation request from DMS

Input

- message code = 2
- container identification
- file identifier
- matrix content code

- chamber designation {D, C}
- matrix weight in kg (net)
- volume ID (used by operators to label archive disks)
- file extension (sequence number)

Process

1. Send "handshake" to DMS, acknowledging receipt of request.
2. Enable automatic summary report. The measurement results are automatically printed following the data acquisition and data reduction.
3. Enable automatic background correction. Background correction is required when running the SAS from DMS.
4. Read active and passive data from specified file; read matrix, calibration and SGRS data from current files.
5. Allow Physicist to edit matrix parameters and/or specify that current background data be used.
6. Compute the Pu mass, etc., as described in requirements 3.1.4.3 M.5 through M.26.
7. Send measurement results to DMS (M.2).

Output

None.

3.2 Performance Requirements

3.2.1 User Interface

The user interface shall provide a visible status of requested functions or a response within three seconds.

3.2.2 CAMAC Interface

The CAMAC crate controller returns two flags in response to each requested function. One indicates whether the command was accepted, and the other is an optional response to certain commands. These flags are returned immediately as a part of the CAMAC cycle or execution of a command and response. The program must be capable of accepting these flags immediately from the crate controller.

3.2.3 DMS Interface

The SAS response to a DMS request for a new assay must occur within 3,000 seconds or the DMS software times out. The SAS response to a DMS request for a recalculation must occur within 150 seconds or the DMS software times out.

3.3 Design Constraints

3.3.1 NEUT2 Functionality

Retain the functionality of the NEUT2 main menu options ST, DD, RC, PN, PB, AN, AS, SS, and AD. Allow NEUT2 calculations to be simulated.

Retain separation of data acquisition and reduction.

3.3.2 Hardware Limitations

The SAS software shall be provided as a single executable program running under Microsoft Windows Version 3.1 on an Intel 80386 or higher platform.

3.4 Attributes

3.4.1 Quality

The SAS software will be developed as a Quality Level 2 system in compliance with RWMC quality requirements [26] and NQA-1 Subject 2.7 [28]. Additional guidance can be found in the Lockheed Idaho Technology Co.'s Quality Manual, Reference [7].

3.4.2 Security

The security level is unclassified sensitive.

3.4.3 Portability

The SAS software shall be implemented in a standard, portable language.

3.5 External Interface Requirements

3.5.1 User Interfaces

The SAS software will provide a graphical user interface (GUI), such as Microsoft Windows Version 3.1. This document does not provide specific requirements for the user interface; implementation details are left up to the software developers with guidance from SWEPP operations and physics support personnel.

3.5.2 Report Interface

The summary report shall be produced within one minute of the completion of an assay whenever the automatic summary report indicator is enabled. The report will identify the full path name of the file containing the signals data. At a minimum, the report will contain the following additional information:

CONTAINER

Identifier: container identification

Type: container type

Volume: 99.9 (L)

MATRIX

Content Code: 999

Type: matrix type

Net Weight: net weight of matrix 99.9 (kg)

Measurement Results:

Total Pu mass: (999999 \pm 999999) g

U-235 mass: (999999 \pm 999999) g

Pu-238 mass: (999999 \pm 999999) g

Pu-239 mass: (999999 \pm 999999) g
Pu-240 mass: (999999 \pm 999999) g
Pu-241 mass: (999999 \pm 999999) g
Am-241 mass: (999999 \pm 999999) g
Thermal power: (999999 \pm 999999) Watts
Thermal power density: (999999 \pm 999999) Watts/ft**3
Total activity: (999999 \pm 999999) Ci
Alpha activity: (999999 \pm 999999) Ci
TRU activity: (999999 \pm 999999) Ci
TRU activity concentration: (999999 \pm 999999) nCi/g
Pu Equivalent Curies: (999999 \pm 999999) Ci
Fissile Gram Equivalent: (999999 \pm 999999) g

Refer to the notes in 3.1.4.3 M.27 regarding number formats and significant figures.

3.5.3 Hardware Interfaces

The SAS software must interface with the SWEPP CAMAC crate [12] to obtain assay system counting data. The CAMAC crate is configured with clock, scaler, and gate generator modules for data acquisition and control. A crate controller module provides a standard GPIB [13] interface for control and data transfers between the CAMAC crate and an external device. The modules currently configured are listed in Table 1.

3.5.4 Software Interfaces

The SAS software must interface with the GPIB control software supplied by the GPIB PC interface board vendor [17].

3.5.5 Communication Interfaces

The SAS software must interface with the DMS computer to receive assay requests from the DMS and to transmit the assay results to DMS. The SAS computer is connected to DMS via an RS232 cable connection.

Communication with the SWEPP Gamma-Ray Spectrometer (SGRS) system is via data files; no direct interface is required.

3.6 Other Requirements

The SAS software must support multiple assay chambers and multiple CAMAC crates with arbitrary arrangements of supported modules. The CAMAC modules that must be supported are listed in Table 1. A CAMAC crate configuration file will be associated with each assay chamber; however, the same physical crate may be used for multiple chambers.