MODELING OF MOCAS PHASE II BATCH PROCESSING

DEPARTMENT OF DEFENSE

DEFENSE LOGISTICS AGENCY

Cameron Station,
Alexandria, Virginia 22304-6100

Operations Research and Economic Analysis Office

DECEMBER 1988

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Public Release; Unlimited Distribution
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Operations Research and Economic Analysis Office 6b OFFICE SYMBOL *(If applicable)* DLA-LO	7a. NAME OF MONITORING ORGANIZATION Defense Logistics Agency
6c. ADDRESS *(City, State, and ZIP Code)* Cameron Station Alexandria, VA 22304-6100	7b. ADDRESS *(City, State, and ZIP Code)* Cameron Station Alexandria, VA 22304-6100
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 8b. OFFICE SYMBOL *(If applicable)* DLA-L	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS *(City, State, and ZIP Code)* Cameron Station Alexandria, VA 22304-6100	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11 TITLE *(Include Security Classification)*

Modeling of MOCAS Phase II Batch Processing (U)

12. PERSONAL AUTHOR(S)

Maj Susan Alten, USAF and Lt Col Gordon M. Hodgson, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(Year, Month, Day)*	15. PAGE COUNT
Final	FROM Jan 87 TO Sep 87	1988 December	15

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
FIELD	GROUP	SUB-GROUP	Batch Processing, Computer Workload

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

The Defense Logistics Agency (DLA) Office of Telecommunications and Information Systems, Automated Information Systems Development and Control Division is responsible for determining the computer resources necessary to support the Mechanization of Contract Administration Services System (MOCAS), which is used by DLA contract administration activities for daily management of over 392,000 contracts valued at $290 billion. The recent Phase II implementation of MOCAS operates in two principal modes -- a daily on-line cycle, and a night batch cycle. With impending installation of Phase II at the larger Defense Contract Administration Services Regions, there was uncertainty as to whether the existing and planned computer resources would be sufficient to handle the workload. The primary purpose of this study was to develop a model to predict batch cycle run times at the larger DCASRs under MOCAS Phase II, and to confirm that the CPU and input/output devices would provide adequate batch processing service levels.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Eugene L. Round, Col, USAF	22b. TELEPHONE *(Include Area Code)* (202) 274-6715 22c. OFFICE SYMBOL DLA-LO

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

23. The Defense Logistics Agency (DLA) Office of Telecommunications and Information Systems, Automated Information Systems Development and Control Division is responsible for determining the computer resources necessary to support the Mechanization of Contract Administration Services System (MOCAS), which is used by DLA contract administration activities for daily management of over 392,000 contracts valued at $290 billion.

24. The recent Phase II implementation of MOCAS operates in two principal modes -- a daily on-line cycle, and a night batch cycle. With impending installation of Phase II at the larger Defense Contract Administration Services Regions, there was uncertainty as to whether the existing and planned computer resources would be sufficient to handle the workload.

25. The primary purpose of this study was to develop a model to predict batch cycle run times at the larger DCASRs under MOCAS Phase II, and to confirm that the CPU and input/output devices would provide adequate batch processing service levels.

Modeling of MOCAS Phase II Batch Processing

December 1988

Major Susan E. Alten, USAF
Lieutenant Colonel Gordon M. Hodgson, USAF
Operations Research and Economics Analysis Office
Headquarters Defense Logistics Agency
Cameron Station, Alexandria, Virginia

DEFENSE LOGISTICS AGENCY
HEADQUARTERS
CAMERON STATION
ALEXANDRIA, VIRGINIA 22304-6100

DLA-LO

December 1988

FOREWORD

The Defense Logistics Agency (DLA) Office of Telecommunications and Information Systems, Automated Information Systems Development and Control Division (DLA-ZS), is responsible for determining the computer resources necessary to support the Mechanization of Contract Administration Services System (MOCAS), which is used by DLA contract administration activities for daily management of over 392,000 contracts valued at $290 billion. The recent Phase II implementation of MOCAS operates in two principle modes -- a daily on-line cycle, and a night batch cycle. With the impending installation of Phase II at the larger Defense Contract Administration Services Regions, there was uncertainty as to whether the existing and planned computer resources would be sufficient to handle the workload. DLA-ZS requested the DLA Office of Operations Research and Economic Analysis to perform a study to: 1) identify the factors and volumes that influence MOCAS Phase II batch processing run time; 2) establish a model to predict batch processing times; and 3) determine the impact of certain computer housekeeping functions on the time required to complete a batch cycle.

The original scope of the project was changed because the housekeeping burden will be subsequently reduced by the acquisition of file maintenance software which will substantially lessen daily file backup activities, and because much of the data necessary to explore predictive relationships between MOCAS system resource consumption and daily activity by MOCAS personnel could not be made available. However, researchers were able to identify certain batch programs which consumed significantly more resources than others. Changes have been made to some of these programs, and data indicate improvements in program run times and disk access requirements. Additionally, upgrades in computer hardware have virtually eliminated concerns that the batch cycle times would intrude on daily on-line cycles. The data collected for this study demonstrate the improvements in performance.

ROGER C. ROY
Assistant Director
Policy and Plans

CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. Background

One of the Defense Logistics Agency's major missions is to administer contracts awarded by the Agency, Military Services, and certain other Federal agencies. The DLA contract administration mission is performed by nine Defense Contract Administration Services Regions (DCASRs) using the Mechanization of Contract Administration Services System (MOCAS). The system is used by DLA contract administration activities for daily management of over 392,000 contracts valued at $290 billion. The Defense Logistics Agency (DLA) Office of Telecommunications and Information Systems, Automated Information Systems Development and Control Division (DLA-ZS) is responsible for determining the computer resources necessary to support MOCAS.

The recent Phase II implementation of MOCAS operates in two modes -- a daily on-line cycle, and a night batch cycle. With the impending installation of Phase II at the larger DCASRs, there was uncertainty as to whether the existing and planned computer resources would be sufficient to handle the workload. Until recently, capacity planning in terms of sizing the MOCAS computer system was based on workload and service levels for the on-line cycle. DLA Systems Automation Center (DSAC) sizing analysis has been oriented toward the capacity of a computer's central processing unit (CPU) to provide response times in the 3-to-5 second range for the on-line process. The basis for this expression is the theory that the more on-line activity at a given time (i.e., the number of terminals logged on and active), the more CPU power was required. This theory has been proven in practice.

Traditionally, analysis of batch processing has shown a heavy demand for moving data in and out of the CPU, a function overlapped with CPU processing and critical to the length of batch processing times. Traditional solutions to speed up the batch cycle have been to distribute files across input/output devices and to get faster devices, or more of them. DSAC has examined the batch cycle to maximize processing concurrency of independent and nonsequential tasks. However, there was no information to indicate the adequacy of this approach for impending batch cycle workloads. Early experience at DCASRs Atlanta and Chicago indicated that the night batch cycle time was impinging on the daily on-line cycle and was beginning to affect productivity. A means was needed to determine the processing power required to handle this different type of computer workload.

Accordingly, DLA-ZS requested the DLA Operations Research and Economic Analysis Office to perform a study to: identify the factors and volumes that influence MOCAS Phase II batch processing run times; establish a model to predict batch processing times; and determine the impact of certain computer housekeeping functions on the time required to complete a batch cycle.

B. Purpose. As stated, the primary purpose of this study was to develop a model to predict batch cycle run times at the larger DCASRs under MOCAS Phase II, and to confirm that the CPU and input/output devices would provide adequate batch processing service levels.

C. Objectives. The objectives were to:

1. Identify the factors and volumes that influence MOCAS Phase II batch processing run times.

2. Establish a model that would predict batch processing times based on the factors and their volumes at the various DCASRs.

D. Scope.

The original scope of the project was to include:

- Housekeeping functions required to complete the batch cycle (e.g., file backups).

- Production jobs that are on the critical path of the MOCAS Phase II batch cycle. The critical path is identified as the sequence of jobs whose output provide mandatory input to other jobs in the stream or jobs that depend on previous job outputs and must be completed before the on-line session can begin.

During the study, the scope of the project was changed. Housekeeping functions were omitted because DLA-ZS awarded a contract for a file management system which greatly reduced the resources required to maintain up-to-date file backups. It also became apparent that the data to support development of a predictive model would not be available, since detailed statistics on daily activities that influenced run times could not be determined, and the influence of other systems operating simultaneously (e.g., Automated Pay, Cost and Personnel System (APCAPS)) could not be determined. Also, each DCASR had hardware that was sufficiently different as to make comparisons extremely difficult. While the original methodology was generally maintained, the expected output was changed to the identification of processes that required relatively more resources and the modification of those program methods that could be made more efficient. Subsequently, the data collected for this study helped verify that the changes had been effective.

II. CONCLUSIONS

Researchers were able to identify certain batch programs which consumed significantly more resources than others. Changes have been made to some of these programs, and data indicate improvements in program run times and disk access requirements. Additionally, upgrades in computer hardware have virtually eliminated concerns that the batch cycle times would intrude on daily on-line cycles, and the data collected for this study demonstrate the improvements in performance.

III. RECOMMENDATIONS

A. Application of Methodology. The methodology developed for this study proved useful. It allowed an analysis of the entire batch cycle, but focused attention early on specific resource consumers and allowed productive changes to be made. This approach could easily be implemented by

DSAC for other systems consisting of numerous batch jobs that have evolved and been carried over from earlier, less capable environments. DSAC should consider using this approach internally to identify and streamline large system batch cycles.

B. Collection of Data. The study objective of predicting batch cycle run times at larger DCASRs based on characteristics of DCASR activity and hardware could not be achieved due to lack of appropriate data. DLA-Z should consider whether the future ability to predict computer workload for given mixes of contracts managed would justify the routine collection of management data to produce an appropriate data base.

IV. BENEFITS

A. Direct CPU Time Benefits. The benefits derived from this study are diverse. Unfortunately, the data were not available to support the initial, primary purpose of developing predictive models for the larger DCASRs. However, three programs were identified for change. Changes made on two jobs showed improvements in CPU time consumed; the third change is scheduled for a later date. Many automated data processing (ADP) facilities assign a dollar value of CPU time for accounting purposes. A search for a Government facility using the Amdahl V8 (or equivalent), and using cost accounting procedures, revealed a cost for CPU time of $33.00 per minute during prime daytime processing, with a 50 percent discount for night batch processing (Department of Health and Human Services). Given a cost per minute of $16.50, program changes made so far have resulted in identifiable CPU time savings of $231.66 per week at Dallas, and $121.28 at Atlanta for the changes already made.

B. Peripheral Benefits

The data collected for this study have never been collected before. During the course of the study, the analysts were asked to address several questions that were related to the MOCAS batch cycle but not part of the study objectives. After DCASR Atlanta installed a more capable CPU and disk drives, analysts were able to provide batch cycle run time data to DCASR Atlanta to help determine that the third shift of computer operation (night shift) was no longer required, resulting in a manning reduction of about 40 manhours per day. Given an average operator's grade of GS-7, step 5, this reduction saved $2,584.00 per week at DCASR Atlanta.

When DCASR New York was nearing implementation of Phase II, analysts were asked to help determine if management actions to control workload timing could have an impact on peak CPU usage. This question was addressed under a consultation reported in DLA-LO Inter-Office Memorandum, 17 Apr 87, Management Survey, MOCAS Phase II Workload Balancing.

V. IMPLEMENTATION

Three program changes were identified for immediate implementation. Two of these programs have been changed. The results of these changes are assessed in the analysis portion of this report. The third job, UYCJDD05, is expected to be modified by DSAC during the next six months.

VI. METHODOLOGY

There were 115 jobs identified by DSAC in the overall job stream that makes up the batch cycle, exclusive of the housekeeping routines (see Appendix A). Not all jobs necessarily run every time or at every DCASR. Not all of these jobs are on the critical path of the cycle; that is, some jobs can run independently of the others and do not preclude operation of the on-line cycle. Therefore, those jobs off the critical path were excluded from further study. There were 55 jobs on the critical path.

DSAC prepared and implemented data collection programs to track job start and stop times for all jobs on the critical path. The data collection took place beginning in March and April 1987 at the five DCASRs then operating Phase II: Atlanta, Chicago, Cleveland, Dallas, and St Louis. After three months, these statistics were evaluated to identify which jobs needed to be examined in more detail. After consulting with DSAC-AB and DLA-ZS, a selection criteria of 10 minutes or more elapsed run time was accepted for determining which jobs required further review. There were eleven jobs that ran over ten minutes at one or more DCASRs and tended to stand out from the others in terms of total time and variability in run time.

The eleven jobs were reviewed in depth with DSAC programmers. The results of the review were used to determine whether more efficient practices could be employed in those programs. For those programs which could be improved, changes were identified and are in varied stages of implementation. Where changes have been made, additional statistics on CPU time and input/output counts were collected and assessed.

VII. ANALYSIS

A. Initial Analysis of Job Stream

Of the 11 jobs that ran over ten minutes in clock time at one or more DCASRs, five are Phase I jobs that have been modified to interface with the Phase II system: UYFCDD05, FYFADA05, UYFTDD10, FYCJDD05, and UNEXDD05.

UYFCDD05, UYFADA05, and UYFTDD10 will be replaced through Contract Payment and Reporting (CPR) redesign. For this reason, no intermediate changes to shorten run time are justified. Excess run time in Phase I holdover programs is usually due to the fact that these jobs run in a sequential mode, requiring both the reading and complete rewriting of all master files and some transaction files just to make minimal updates or changes. These jobs are, therefore, input/output (I/O) bound. Under the redesign concept, these programs will be replaced by programs utilizing data base design or Indexed Sequential Access Method (ISAM). This will eliminate the requirement to forward large master/transaction files, vastly reducing the I/O requirement. Also, a significant amount of the work being done by these programs will move to an on-line environment from the batch cycle.

UYCJDD05 has already been modified to operate using ISAM files for initial input. It is not scheduled for CPR redesign, but DSAC-ABA proposes another modification to shorten run time. Currently, the program chain reads all contracts that are in the Contract Completion Notice (CCN) chain. The vast majority of these contracts had no action during this cycle and will not need processing under this job. It may be possible to chain read only those contracts with a current Contract Processed Date, checking to see if the contract is on the CCN chain and then taking processing action if needed. Because the number of contracts processed on any given day is significantly less than the number of contracts in the CCN chain, the chain read should be shorter, thus shortening the job's run time. This change is expected to be made within the next six months.

UNEXDD05 is not a MOCAS program. It is an exit program under the control of DSAC-R. Modifications have been made to adapt it as much as possible to the Phase II system. It currently runs barely over ten minutes at DCASR Cleveland and under ten minutes at all the other Phase II DCASRs. Both DSAC-ABA and DSAC-RSC personnel doubt that further modification would be efficient.

UNMDDD05 cannot be shortened. A System Change Request reduced the number of abstracts produced by MOCAS, but inquiries are still being generated for the deleted abstracts. Abstracts are no longer produced for any modifications or for changes subsequent to the first on any contract, but these modifications and changes still generate an inquiry record which is later used by other subsystems, and further changes are not possible.

UNMDDD15 is an inquiry job that runs under ten minutes everywhere but DCASR Atlanta, where it runs over thirty minutes. DCASR-ATL-Z agrees that they use inquiries a great deal and they seem content with the run time. For purposes of information, however, DSAC-ABA examined the program and its operation in Atlanta. No problems were found.

UNCTTT92 was serially reading the MOD-V file, containing all contracts and modifications held in the MOCAS system at a given DCASR. The job deletes a few records created daily through normal sign-on procedures and other processing, providing those records have been on the data base for 90 days. The program was reading the entire MOD-V file, but this was changed. The program is now keyed to read only those records with a PIIN/SPIIN field containing the expression "ON LINE BACKLOG", so the program now only needs to read less than ten percent of the records, shortening the run time considerably.

UNJHDD05 is a data base purge program that deletes information -- a very time consuming process. It had been suggested that the deletes be done monthly rather than daily, but this would have created a very long job for the monthly cycle, which is already considered too long for convenience at some DCASRs. Instead, this job was changed to a weekly job that runs in conjunction with the weekend. It is thus less likely to impact any on-line activity.

UNJFDD05 is an update program that is already considered efficient. No changes are under consideration.

UNDGDD05 runs over ten minutes only at DCASR Atlanta. This is due, at least in part, to the addition there of Revised Delivery Forecast (RDF). This necessitated the addition of a new rechain program (DG81) to the original program (DG80). Two other DCASRs, Dallas and Chicago, also now have RDF, but DG81 is a separate job (UNDGDD10) at these sites. The total average time for both jobs at Dallas is 614 seconds, compared to 609 for the single job at

Atlanta. The Chicago total is 481 seconds. Neither job can be made more efficient.

B. Changing Environment. During the data collection for this study, hardware configurations at the DCASRs were in a constant state of change, contributing to the inability to develop predictive relationships. Between February and October 1987, DCASRs Atlanta, Dallas, Chicago and St. Louis upgraded to Amdahl V8s. During the same timeframe, DCASRs Dallas, Cleveland, Chicago, and St Louis upgraded disk drives to IBM 3380-compatibles. Other upgrades in I/O buffers and cache memory occurred at DCASRs Chicago and Cleveland.

C. Effectiveness of Clock Time as a Measure. In June 1987, DCASR Chicago was upgraded from an Amdahl V7C to an Amdahl V8, which had to function at reduced capability due to air conditioning limitations. The change in average job time for each job was observed. In most cases, jobs ran considerably faster, but in a few the average time was longer. Because CPU times would not be available until October 1987, no investigation of this occurrence was possible. Later, the Chicago V8 was further upgraded, and V7s at Atlanta and Dallas were replaced with V8s. In many cases the new average clock times were unexpectedly longer. It was determined that clock times are misleading, since they cannot account for the impact of other jobs running concurrently and sharing the CPU resource. Clock time may also include delays while a program waits for operators to change tapes.

D. Analysis of Changes.

1. Hardware Changes. The data distinctly illustrate the effect of the hardware changes. Figure 1 shows the change experienced in CPU time at DCASR Dallas for job UNCTDD92, described above under UNCTTT92. A perceptible drop in CPU time can be noted at Julian day 87304, following the installation of an Amdahl V8. Figure 2 shows a corresponding drop in CPU time per I/O. Figures 3 and 4 show similar behavior at Atlanta around Julian day 87297. Detailed data on all programs were not collected, which precluded assessment of the total improvements in CPU time.

2. Program Changes.

Two jobs have been changed to date as a result of this study. UNJHDD05, while not changed internally, has been rescheduled to run weekly, rather than daily. In looking at the change in CPU time and CPU time per I/O at Dallas before and after the change in schedule, the average CPU time for the weekly run was 2.06 minutes, or less than twice the 1.30 minutes for the daily runs made before the change. The CPU time per I/O was actually reduced for the longer running weekly job, falling from .000035 to .000027 CPU minutes per I/O. Therefore, DSAC was able to change a daily run to a weekly one and reduce the total CPU time consumed per week at DCASR Dallas from 6.50 minutes (1.3 minutes per day average times 5 days) to 2.06 minutes. Data were not available for a comparison at other DCASRs.

Figure 1

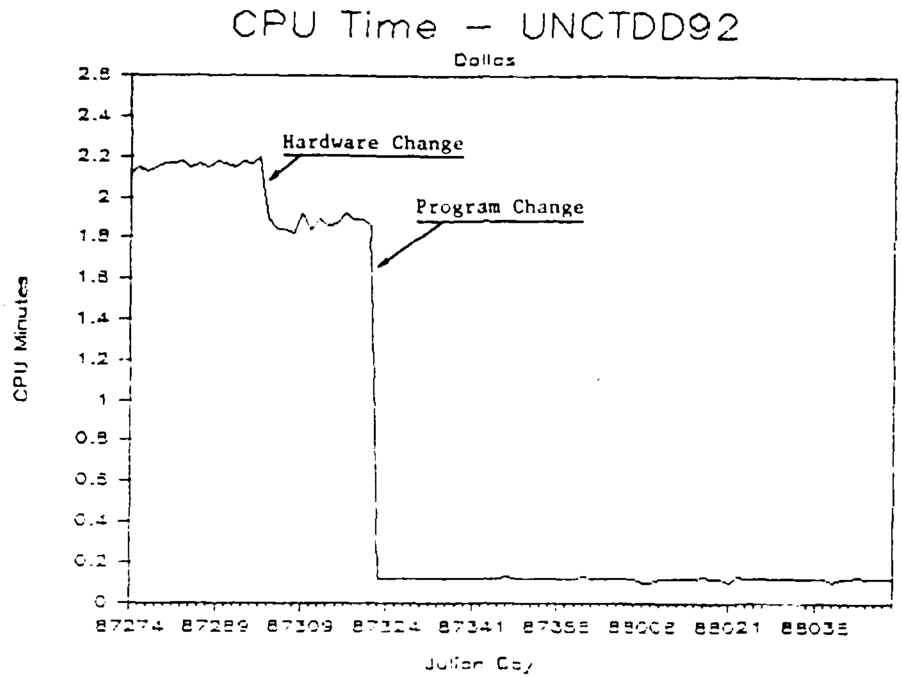


Figure 2

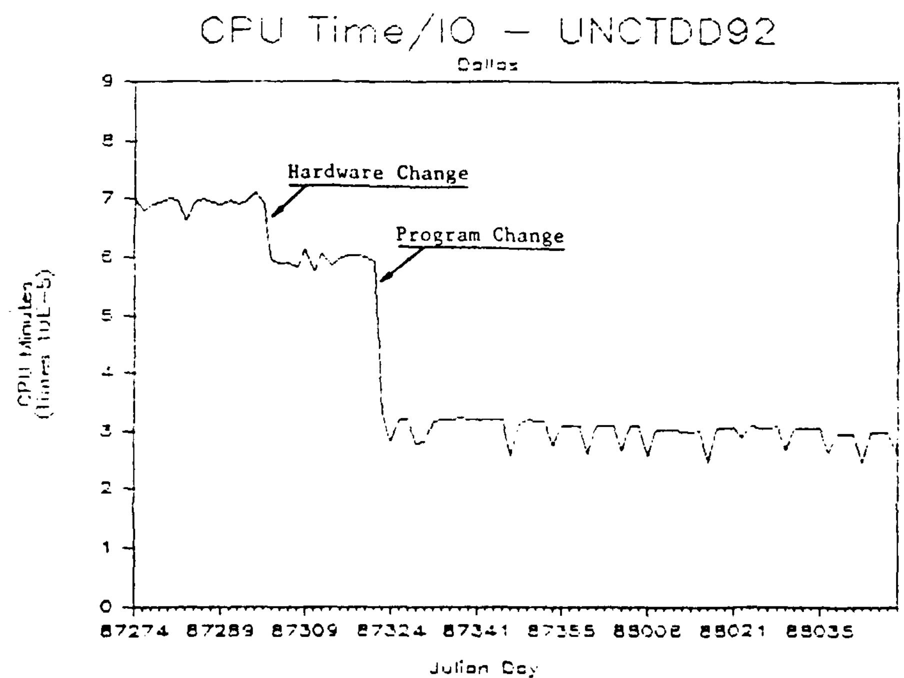


Figure 3

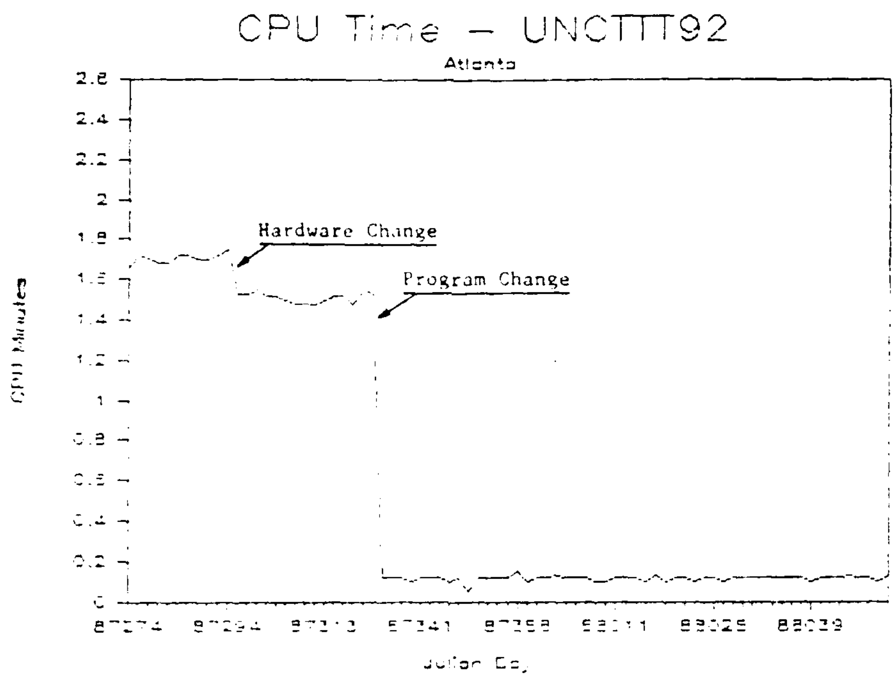
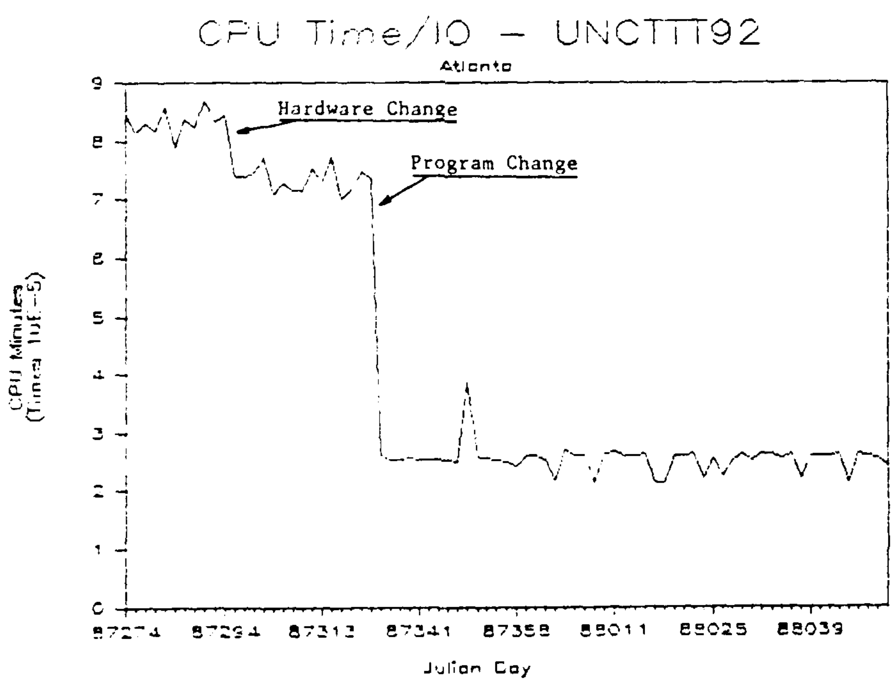


Figure 4



The other job already changed is UNCTTT92/UNCTDD92. Data collected on this job also reflect the CPU times and CPU time per I/O before and after the program change. Figures 1 through 4 illustrate the dramatic improvements in CPU time and CPU time per I/O incurred when the job was modified. Table 1 summarizes the CPU time, I/O count, and CPU time per I/O before and after the program change at Dallas and Atlanta.

Table 1

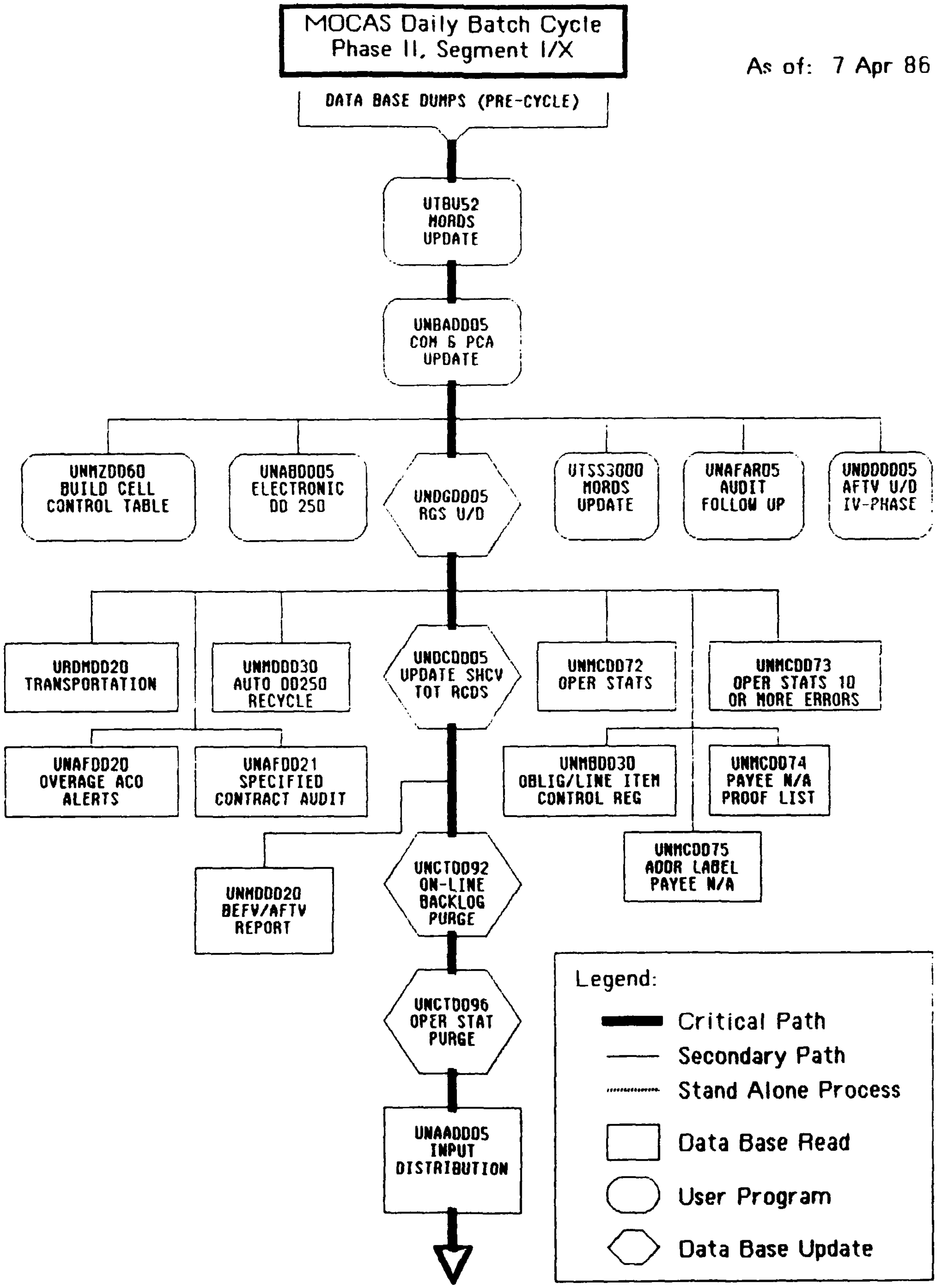
Comparison of Run Time Statistics for Job UNCTDD92

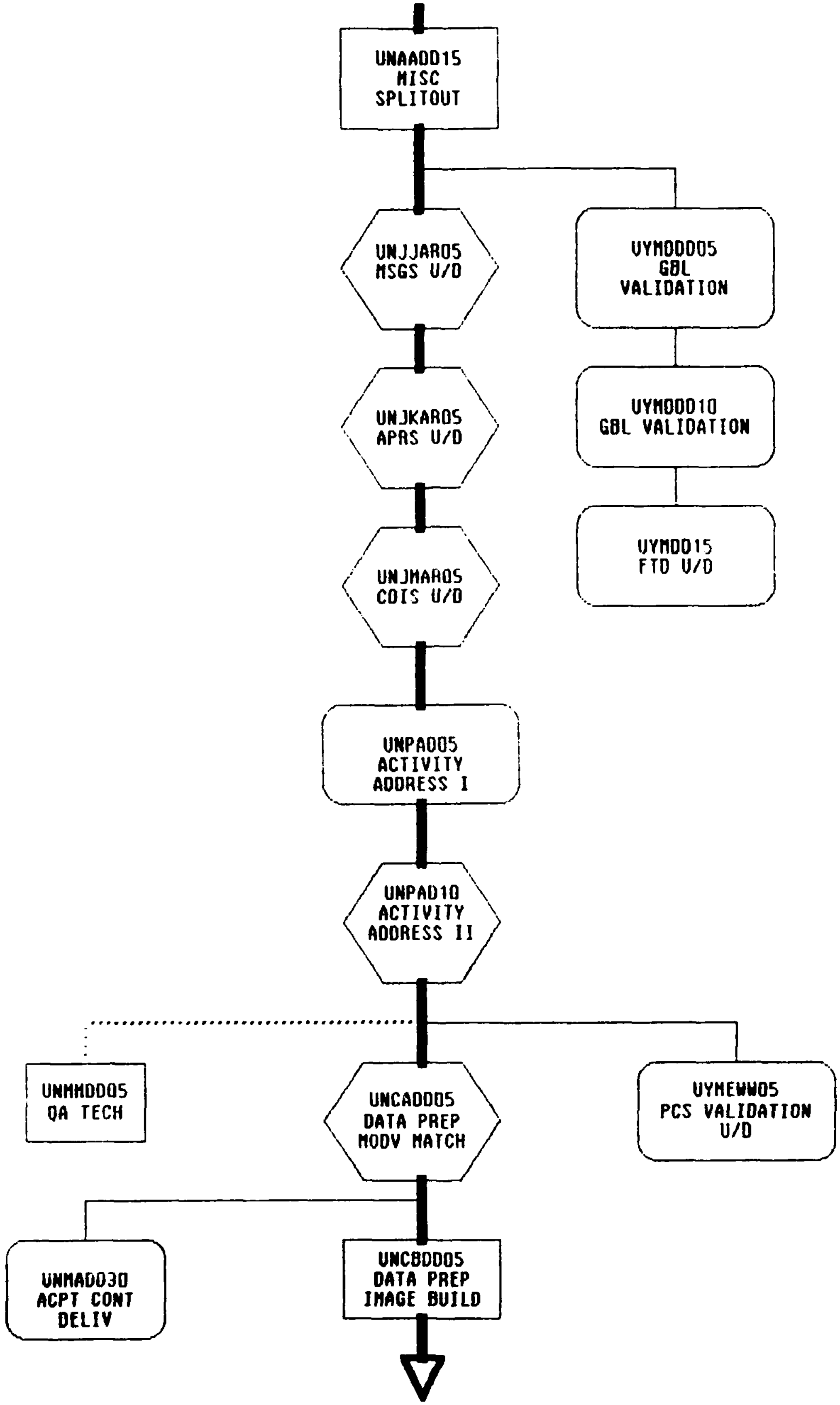
	Average CPU Time (min)	I/O Count	Average CPU Time per I/O (min)
Dallas			
Before Change	2.04	31,340	.000065
After Change	0.12	4,055	.000029
Atlanta			
Before Change	1.59	20,533	.000077
After Change	0.12	4,672	.000025

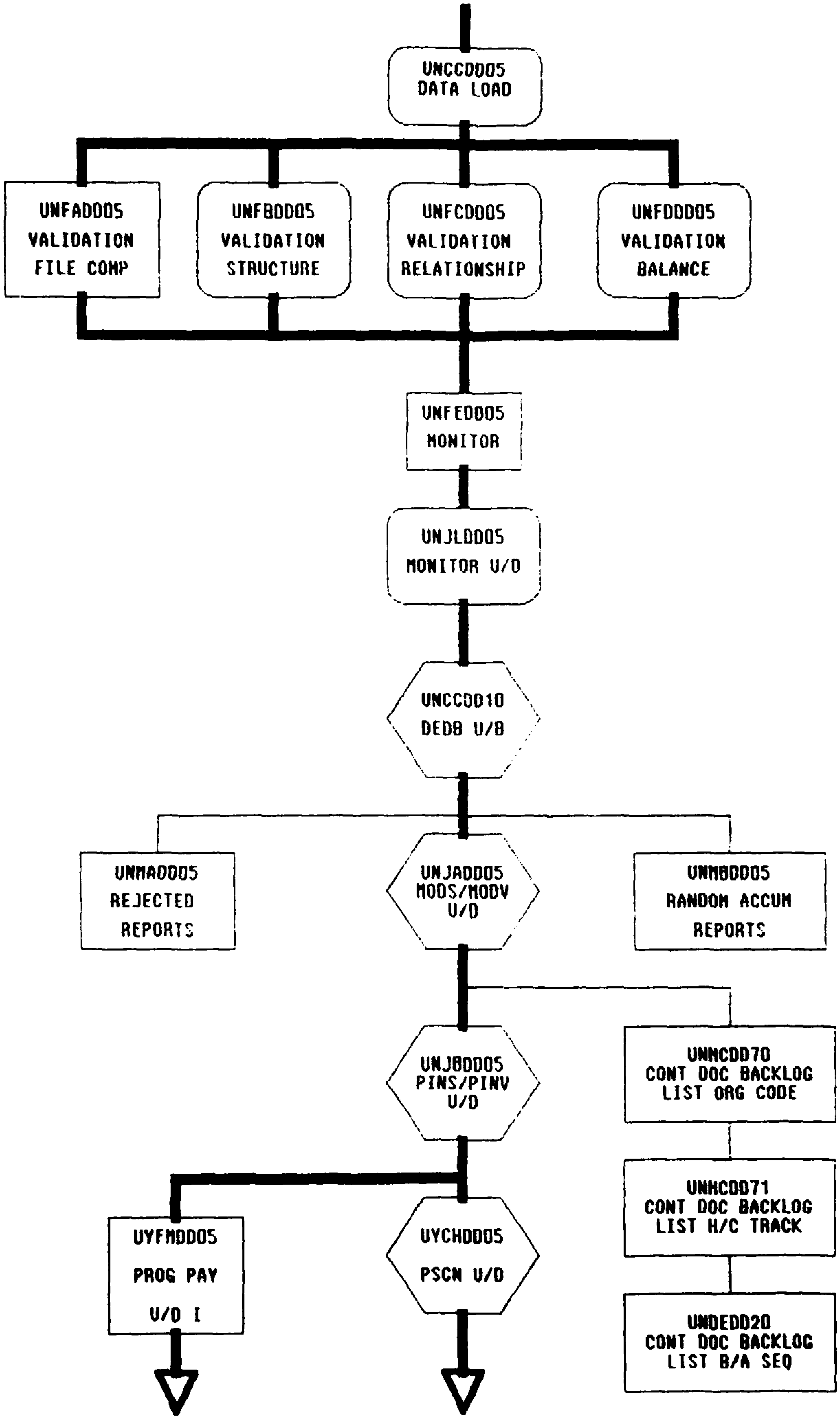
Given that the job was run five times a week, the average CPU time saved per week at Dallas would be 9.6 minutes for this job, and 7.35 minutes at Atlanta. The reduction in I/O count reflects the elimination of unnecessary disk write activity.

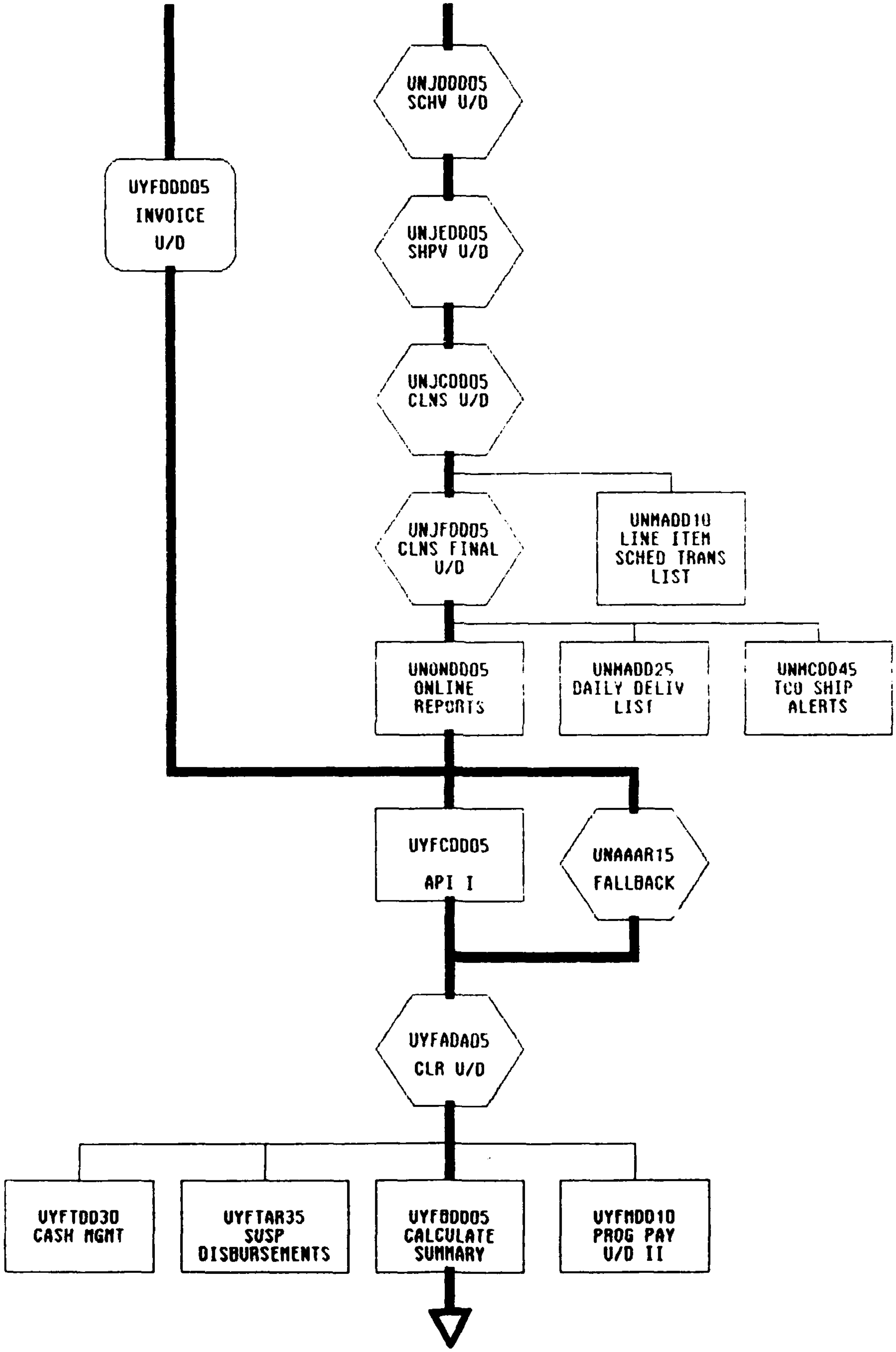
APPENDIX A

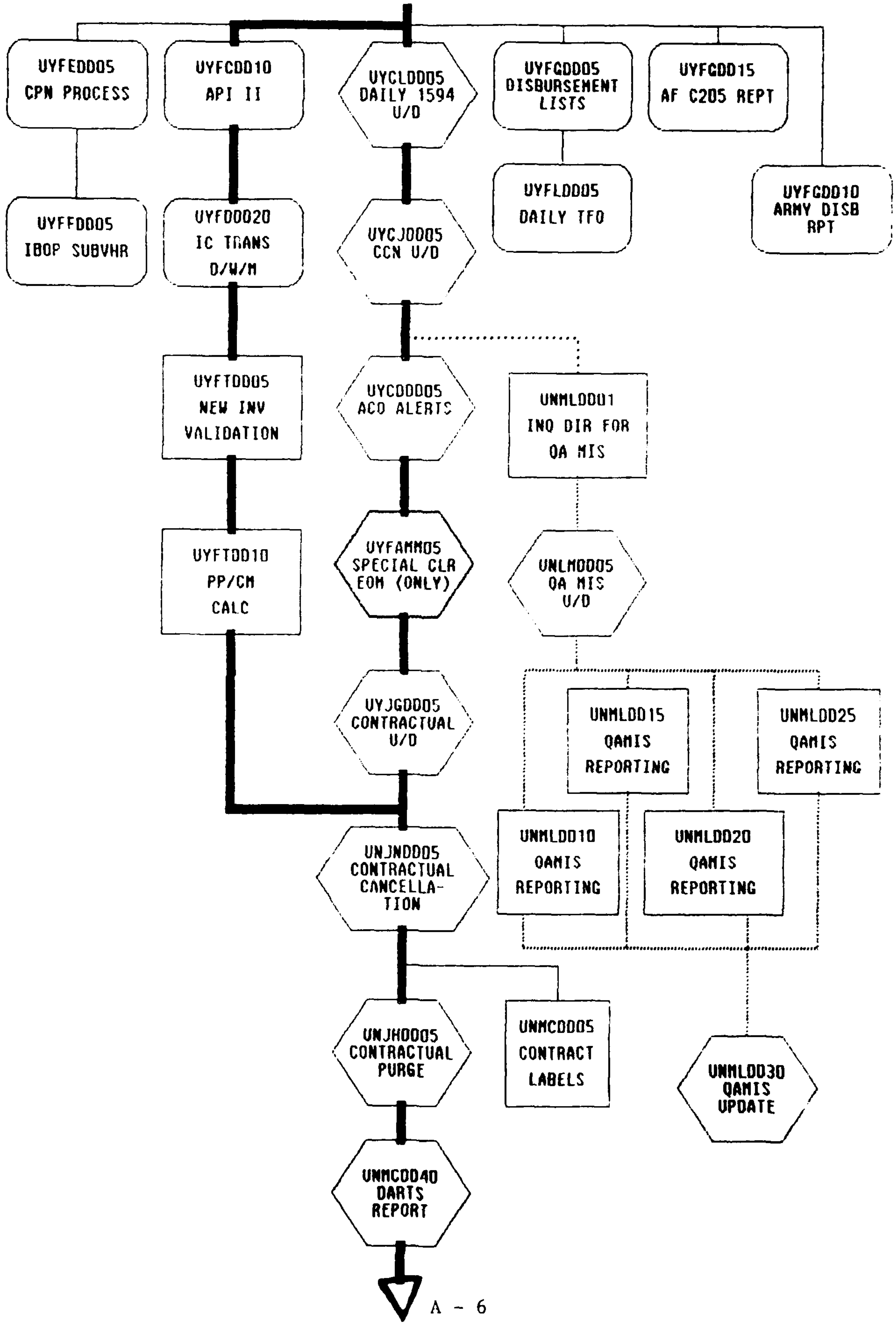
MOCAS Phase II Batch Job Names and Organization


MOCAS Daily Batch Cycle
Phase II, Segment I/X
As of: 7 Apr 86
DATA BASE DUMPS (PRE-CYCLE)
UTBU52 MORDS UPDATE
UNBADD05 COM & PCA UPDATE
UNMZDD60 BUILD CELL CONTROL TABLE
UNABDD05 ELECTRONIC DD 250
UNDGDD05 RGS U/D
UTSS3000 MORDS UPDATE
UNAFAR05 AUDIT FOLLOW UP
UNDDDD05 AFTV U/D IV-PHASE
URDMDD20 TRANSPORTATION
UNMDDD30 AUTO DD250 RECYCLE
UNDCDD05 UPDATE SHCV TOT RCDS
UNMCDD72 OPER STATS
UNMCDD73 OPER STATS 10 OR MORE ERRORS
UNAFDD20 OVERAGE ACO ALERTS
UNAFDD21 SPECIFIED CONTRACT AUDIT
UNMBDD30 OBLIG/LINE ITEM CONTROL REG
UNMCDD74 PAYEE N/A PROOF LIST
UNMCDD75 ADDR LABEL PAYEE N/A
UNMDDD20 BEFV/AFTV REPORT
UNCTDD92 ON-LINE BACKLOG PURGE
UNCTDD96 OPER STAT PURGE
UNAADD05 INPUT DISTRIBUTION
Legend:
Critical Path
Secondary Path
Stand Alone Process
Data Base Read
User Program
Data Base Update

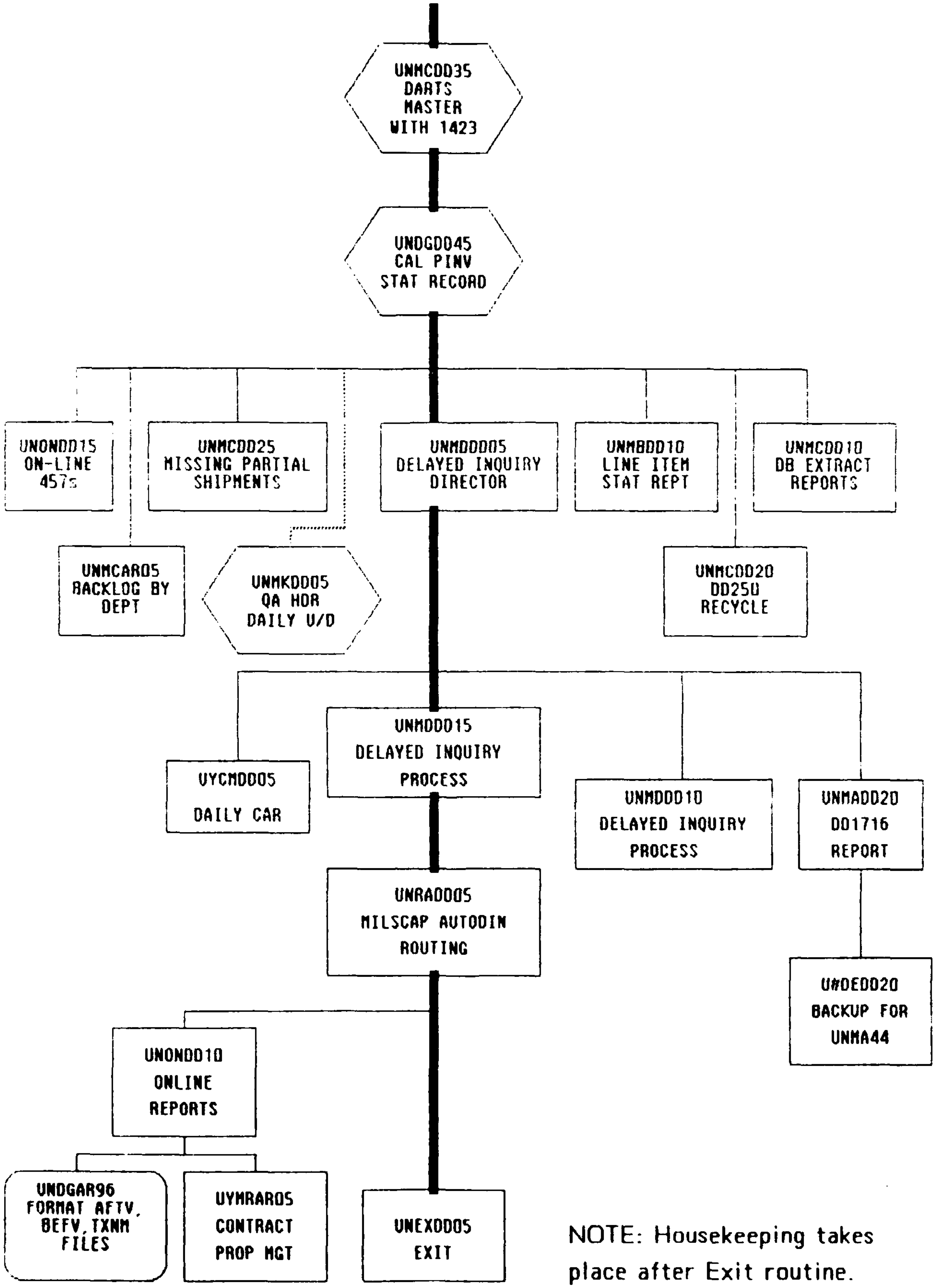

UNAADD15 MISC SPLITOUT
UNJJAR05 MSGS U/D
UYMDDD05 GBL VALIDATION
UNJKAR05 APRS U/D
UYMDDD10 GBL VALIDATION
UNJMAR05 COIS U/D
UYMDD15 FTD U/D
UNPAD05 ACTIVITY ADDRESS I
UNPAD10 ACTIVITY ADDRESS II
UNMMDD05 QA TECH
UNCADD05 DATA PREP MODV MATCH
UYMEWW05 PCS VALIDATION U/D
UNMAD030 ACPT CONT DELIV
UNCBDD05 DATA PREP IMAGE BUILD


UNCCDD05
DATA LOAD
UNFADD05
VALIDATION
FILE COMP
UNFBDD05
VALIDATION
STRUCTURE
UNFCDD05
VALIDATION
RELATIONSHIP
UNFDDD05
VALIDATION
BALANCE
UNFEDD05
MONITOR
UNJLDD05
MONITOR U/D
UNCCDD10
DEDB U/B
UNMADD05
REJECTED
REPORTS
UNJADD05
MODS/MODV
U/D
UNMBDD05
RANDOM ACCUM
REPORTS
UNJBDD05
PINS/PINV
U/D
UNMCDD70
CONT DOC BACKLOG
LIST ORG CODE
UNMCDD71
CONT DOC BACKLOG
LIST H/C TRACK
UYFMDD05
PROG PAY
U/D I
UYCHDD05
PSCN U/D
UNDEDD20
CONT DOC BACKLOG
LIST B/A SEQ


UNJDDD05
SCHV U/D
UYFDDD05
INVOICE
U/D
UNJEDD05
SHPV U/D
UNJCDD05
CLNS U/D
UNJFDD05
CLNS FINAL
U/D
UNMADD10
LINE ITEM
SCHED TRANS
LIST
UNONDD05
ONLINE
REPORTS
UNMADD25
DAILY DELIV
LIST
UNMCDD45
TCO SHIP
ALERTS
UYFCDD05
API I
UNAAAR15
FALLBACK
UYFADA05
CLR U/D
UYFTDD30
CASH MGMT
UYFTAR35
SUSP
DISBURSEMENTS
UYFBDD05
CALCULATE
SUMMARY
UYFMDD10
PROG PAY
U/D II



UYFEDD05 CPN PROCESS
UYFCDD10 API II
UYCLDD05 DAILY 1594 U/D
UYFGDD05 DISBURSEMENT LISTS
UYFGDD15 AF C205 REPT
UYFFDD05 IBOP SUBVHR
UYFDD020 IC TRANS D/W/M
UYCJDD05 CCN U/D
UYFLDD05 DAILY TFO
UYFGDD10 ARMY DISB RPT
UYFTDD05 NEW INV VALIDATION
UYCDDD05 ACO ALERTS
UNMLDD01 INQ DIR FOR QA MIS
UYFTDD10 PP/CM CALC
UYFAMM05 SPECIAL CLR EOM (ONLY)
UNLMDD05 QA MIS U/D
UYJGDD05 CONTRACTUAL U/D
UNMLDD15 QAMIS REPORTING
UNMLDD25 QAMIS REPORTING
UNJNDD05 CONTRACTUAL CANCELLA-TION
UNMLDD10 QAMIS REPORTING
UNMLDD20 QAMIS REPORTING
UNJHDD05 CONTRACTUAL PURGE
UNMCDD05 CONTRACT LABELS
UNMLDD30 QAMIS UPDATE
UNMCDD40 DARTS REPORT



NOTE: Housekeeping takes place after Exit routine.