DDB-B23

CO-Res Operating Sys

ABSTRACT

CO-RESIDENT OPERATING SYSTEMS

TASK TEAM REPORT

April 18, 1973

The mission of the co-resident operating system task group was: "To define a cost effective mechamism within the new operating system to allow selected old operating systems to operate in their entirety permitting simultaneous execution with no modifications to the old operating system".

Four people were assigned to this task, two from NCR with experience on the NCR/Century and two from CDC experienced with CYBER 70 computer systems.

The scope of the problem was narrowed to a point where productive output could be obtained in the allotted time frame. It was decided the time could be spent most profitably in a detailed study of emulating the NCR Century computer systems on the IPL. This decision was heavily influenced by a desire to deal with specific problems rather than philosophical and inconclusive matters. It was also felt that the productivity of the team would be greater if it were kept small.

Having limited the problem in scope, various assumptions had to be made before further progress could be accomplished. Key in this area was the assumption that the problem was soluble. As a result, the team concentrated on how to effect the implementation, as opposed to if it was possible. A second key assumption was that all other teams would be successful. Having assumed this, it was not necessary to spend any time on definition of the host system except to equate it to the CYBER 70 system in terms of hardware and software capability.

The following computer systems were identified as candidates for emulation:

a. NCR Century I

b. NCR 315

c. CDC 3000L.

The CDC CYBER 70 computer systems were not considered to be candidates. The strategy for CYBER 70 systems was taken to be compatibility supported by conversion aids. It is felt that both the NCR 315 and CDC 3000L computer systems should be emulated, however, a detailed study of these two systems was impractical in the time frame. All three of these machines are architecturally similar and many of the problems to be solved emulating one of them apply to the others.

There are several problem areas that were covered in discussing implementation techniques for the century simulator. The term simulator is used deliberately to indicate that software is the major tool in the process, not hardware. One major area of concern was, which operating systems can and should be allowed in the simulation environment. The user interface to the B1 operating system is so broad that it does not make sense to consider interfacing simulation below the B1 operating system level. It appears that all the B series operating systems can run under simulation. The cases that cannot be handled are those which present solid time critical situations (e.g., MICR) where the operation fails when response is not received in a certain amount of time.

The main difference between SCOPE and B series operating systems seems to be in the handling of I/O devices. CDC's approach is logical right to the point of transfer of data. In the NCR systems, binding to devices is done much earlier -- partly in the source program, and partly at program load time. Many of the problems to be solved in the simulator design stem from this basic philosophical difference.

Problem areas discussed other than I/O were:

a. Privileged commands.

b. Supervisor calls.

c. Implicit configuration control.

d. Operator communications.

e. Idle loops.

f. Engineering and accounting logs.

g. Feature redundancy.

h. Remote Job Entry.

i. System initialization.

j. Real time considerations.

All of these points are discussed in detail in Section 5.2 of the report.

As stated previously it seems that all levels of B series operating systems can be simulated. The provision for simulating B4 in addition to B1 was considered. On the surface it appears that B4 need not be simulated since its primary function is to multiprogram. With multiple B1 jobs and a simulator being shared, through a shared code facility provided by the host OS, the differences between the two approaches are not obvious. However, provision for B4 simulation allows the least change for the user and vendor at a possibly substantial performance cost to the user. In the light of these considerations, arguments in support of both approaches can be made.

After analyzing the simulation problem from many angles it appears there are three different solutions that may be arrived at:

1. Partitioned Configuration.

Can utilize NCR standalone system software under simulation with minimum change.

Direct drive peripherals as NCR native devices.

Utilize "RJE type" job interchange between Host OS and simulated OS. Applies best to configurations having a small number of peripheral devices.

Allows customer and use to continue to use existing operation, administrative and usage protocols.

Least development expense.

Moderate relative performance (SWAG).

Least attractive to user because of disjoint system but least user conversion.

2. Host Imbedded Simulator Support Package (with direct driven peripherals) Total configuration control is maintained by Host OS.

Common I/O queues and input/output procedures.

Applies best to configurations having a moderate to large number of peripherals.

Most development expense.

Worst relative performance (SWAG)

Moderate attractiveness and user conversion.

3. Host Imbedded Simulator Support Package (with peripherals driven through host OS).

Total configuration control is maintained by Host OS.

Total control of I/O support is maintained by Host OS.

Applies best to configurations having a large number of peripherals and/or a large real memory capacity.

Moderate development expense.

Best relative performance (SWAG)

Most attrictive to users but requires greatest user conversion.

The peripheral support question seems to reduce to the following case analysis:

1. Simulator direct driving peripherals ⇉ C I devices / → C II devices

2. Simulator uses Host OS I/O services → C I devices / ⇉ IPL devices

(the double arrow indicates expected utilization density)

Case #1 is in fact the same case as that for NCR supporting its standalone customer base. The results of this NCR support in conjunction with the recommended partitioned configuration solution will be no additional effort for vendor or customer when moving to the IPL.

Case #2 will invalidate the partitioned configuration solution. It essentially requires that the Host OS and/or simulator become intimately aware of the inherent characteristics of the I/O structures and ptotocols for both CENTURY I and IPL systems.

It may be stated that in general this case complicates the process for both vendor and customer. The vendor must develop and maintain a more complicated simulator package that is in addition to the standard OS offerings. The customer must convert his data files from the CENTURY I device to the corresponding IPL device, (perhaps the most complex and difficult portion of any conversion effort).

Implementation costs of the simulator were estimated to be no more than $1 M, regardless of the approach, when all efforts are included. Maintenance costs will be $25 K yearly. The simulator would run at an execution rate approximately double that of a Century I - 300 and would leave at least 25% of the IPL processor free for other work.

CO-RESIDENT OPERATING SYSTEMS

FINAL REPORT

April 19, 1973

TABLE OF CONTENTS

1.0 Executive Summary

2.0 Introduction

3.0 Scope of Problem

- 3.1 Systems Considered
- 3.2 Selection Criteria
- 3.3 Final Selection

4.0 Assumptions

5.0 Analysis of Problem

- 5.1 Century I Operating Systems
- 5.2 General Simulator Design
- 5.3 An Alternative Approach
- 5.4 B1 Simulation Only versus B4 Simulation
- 5.5 Peripheral Support Issues

6.0 Cost and Performance Estimates

- 6.1 Cost to Produce Simulator
- 6.2 Performance of Simulator

1.0 EXECUTIVE SUMMARY

1.1 The mission of the co-resident operating systems task force was to deduce a mechanism whereby existing NCR and CDC computer systems could be emulated on the joint companies Integrated Product Line (IPL).

1.2 The following computer systems were identified as candidates for emulation:

a. NCR CENTURY I

b. NCR 315

c. CDC 3000L

The CDC CYBER 70 computer systems were not considered to be candidates. The strategy for these systems was taken to be compatibility supported by conversion aids.

1.3 The NCR/CENTURY series were analyzed in detail and an optimum solution involving partitioning of the hardware was devised. Implementation costs of the emulator were estimated to be $1 M with maintenance costs of $25K yearly. The emulator would run at an execution rate approximately double that of a CENTURY I/300, and would leave at least 25% of the IPL processor free for other work.

1.4 A more general solution utilizing true co-residency was analyzed in depth and shown to have similar costs but with a slightly degraded performance over the partitional technique. For this reason, it was not favored for implementation.

2.0 INTRODUCTION

2.1 The mission of the co-resident operating system task group was:

"To define a cost effective mechanism within the new operating system to allow selected old operating systems to operate in their entirety permitting simultaneous execution with no modifications to the old operating systems."

Four people were assigned to this task, two from NCR with experience on the NCR/CENTURY computer systems, and two from CDC experienced with CYBER 70 computer systems. The task was to be addressed over a six-week period.

2.2 It was apparent from the outset that the schedule was extremely tight, and with intervening meetings of the co-chairmen there were very few working days when the team could meet. For this reason, the first step taken was to narrow the problem down to a point where productive output could be obtained in the allotted timeframe. To this end, it was decided that the time could be spent most profitably in a detailed study of emulating the NCR/CENTURY computer systems on the IPL. This decision was heavily influenced by a desire to deal with specific problems rather than philosophical and inconclusive matters. It was also felt that the productivity of the team would be greater if it were kept small. All that remained was to make maximum use of the expertise of the team members.

2.3 It is not suggested that NCR/CENTURY computers are the only candidates for emulation on the IPL. In fact, it was felt that both NCR 315 and CDC 3000L computer systems should be emulated. However, a detailed study of all these systems was impractical in the time available. Fortunately, all of these machines are architecturally similar, single channel, single processor, programmable channel interrupt driven computers, and many of the problems to be solved emulating one of them apply to the others.

2.4 Having limited the problem in scope, various assumptions had to be made before any further progress could be made. Key in this area was the assumption that the problem was soluble. As a result, the team concentrated on how to effect an implementation, as opposed to if it was possible. Naturally, the former approach yields the latter answer, whether positive or negative.

A second key assumption was that all other teams would be successful. This assumption implicitly defined the host computer system. It was not necessary to detail this system in any more depth than equivalencing it to CYBER 70 systems in terms of hardware and software capabilities.

2.5 The bulk of this paper describes the implementation techniques for the simulator. The term simulator is used deliberately to indicate that software is the major tool in the process, not hardware. The most complete analysis covered co-resident systems, which was the mission. However, a slightly different approach for the NCR/CENTURY simulation was investigated and favored for that system. In that approach, the host computer system hardware is partitioned, yielding essentially a standalone CENTURY I emulator. This solution is not necessarily a general solution for all computer systems, but it does fit in well with the current development plans of the two companies. It is influenced greatly by peripheral configurations and the transition paths which may be taken between current NCR/CENTURY I peripherals and CENTURY II/IPL peripherals. For this reason particular attention was given to problems in that area and a separate section devoted to them.

3.0 SCOPE OF THE PROBLEM

3.1 Systems Considered

In the interests of limiting the scope of the problem to a manageable size, computers not manufactured by either CDC or NCR were not considered for emulation by the IPL. The computers and operating systems considered were:

3.1.1	CDC 3000L	- MSOS	
3.1.2	CDC 3111L	- MASTER	
3.1.3	CDC - CYBER 70	- SCOPE 3.4	Lower CYBER
3.1.4	CDC - CYBER 70	- KRONOS 2.1	Lower CYBER
3.1.5	CDC - CYBER 70	- SCOPE 2.0	Upper CYBER
3.1.6	NCR 315	- OS	
3.1.7	NCR 315 RMC	- OS	
3.1.8	NCR - CENTURY I	- B series as follows:	

OS / COMP.	B1	B2	B3	B4
50	-1	-1		
100	-1	-1		
101	-2/-2A	-2/-2A		
200	-2/-2A	-2/-2A	-2	
251	-3	-3	-3	-3
300	-3	-3	-3	-3

3.2 Selection Criteria

3.2.1 CDC 3000L - MSOS/MASTER

These computer systems were considered to be definite candidates for emulation on the IPL. However, it was decided not to address them specifically with the current committee. The NCR committee members could offer little expertise in the area, and the CDC committee members' experience was primarily on CDC-CYBER computers. It was felt that 3000L emulation was basically an internal CDC problem and increasing the team membership to address that problem would dilute the overall effort. In addition, the machine is architecturally similar to the NCR CENTURY computers with respect to emulation. By that, it is meant that both machines are single level memory, single processor computers.

3.2.2 CDC CYBER 70 - SCOPE 3.4/2.0 - KRONOS 2.1

These computer systems were not regarded as candidates for emulation. The current CDC strategy is not to emulate CYBER 70 systems on the IPL computers. The design objectives are for maximum compatibility with CYBER 70 and to ease the conversion problem that the user must face. The current CDC/NCR study effort has not uncovered any facts which would cause the strategy to be revised.

3.2.3 NCR 315/315 RMC OS

It was decided that these computer systems would not be addressed specifically by the committee. They are, nevertheless, candidates for emulation. The NCR CENTURY computers using the B1 operating system currently simulate the 315 machines, with hardware assistance for instruction emulation. 315 I/O simulation is performed in software under B1. Hence the problem has been solved, and further effort in this area at this time would not be utilizing resources at hand in an optimum fashion.

3.2.4 NCR CENTURY - B Series OS

These current NCR computer systems are prime candidates for emulation on the IPL, and it was decided that the team would devote most of its energies in this direction. The team members expertise is well matched to this exercise, hence the maximum benefit can be derived by drawing on this expertise.

3.3 Final Selection

In arriving at a single system for the emulation study, the committee recognized that much work would remain in other areas - notably 3000L. On the other hand, many of the techniques applied to CENTURY emulation will apply to 3000L emulation, and it is anticipated that many of the problem areas, if not the specific solutions, will be common to both systems. The team had a strong desire to deal with specific problems, in detail, and not to address the overall issue in a philosophical vein. For this reason, co-opting further members onto the team was seen as detrimental, and the problem was narrowed down to make optimal use of the limited time-frame and full utilize the talent of the team members. It must be remembered that CDC 3000L, NCR 315, and NCR CENTURY computer systems were all identified as candidates for emulation. The NCR CENTURY series will be considered in detail here.

4. ASSUMPTIONS

Since the IPL hardware and software, at best, are ill-defined at the present time, several assumptions must be made regarding the general environment. These assumptions are discussed below.

4.1 The objective of the exercise is total hardware emulation as opposed to the reproduction of selected operating system functions. All CENTURY operating system code and user application programs are to execute on the IPL without change.

Substantial impact - no examination was made into the possibility of directly supporting any level of NCR user jobs within the IPL OS (i.e., no simulation requirement).

4.2 The problem is assumed to be soluble in that similar problems have been solved successfully in the past.

Substantial impact - led to a detailed study of the specifics of the problem rather than the generalities of emulation/simulation.

4.3 The IPL operating system is assumed to be at least as powerful as CYBER 70 SCOPE 3.4 In addition, the logical constructs used in SCOPE 3.4 are assumed to exist in the new system in a similar form.

Minor impact - used as a point of reference for Host OS. capabilities.

4.4 It is assumed that the simulated system can obtain direct access to dedicated physical devices.

Substantial impact - the partitioned configuration solution was a direct fallout from this assumption. In an embedded solution the main effect is on the Host OS I/O devices and the extra development costs for these drivers.

4.5 The hardware instruction repertoire on the IPL is assumed to be roughly as powerful and flexible as that of CYBER 70 today. In this context the 6150, CYBER up CPU and CYBER 70 PPU or IPL CPU/PPU are assumed to be approximately equivalent processors.

Substantial impact - no study was made in the area of traditional "emulators". In particular, no dependence was made on the capabilities of the CPU supplying the processing power.

4.6 As a follow-on to the previous assumption, simulation as opposed to emulation is taken as the key area for investigation. Simulation being defined as imitation by software, emulation as imitation by hardware.

Substantial impact - no study was made in the area of traditional "emulators". In particular, no dependence was made on the capabilities of the CPU supplying the processing power.

4.7 It is assumed that NCR will implement a CENTURY I simulator on the 6150 for their standalone customer base.

Substantial impact - the partitioned configuration solution was a direct fallout of this assumption.

4.8 It is assumed that the 6150 is an integral component of IPL, being either a peripheral processor or a central processor, but, in any event, being present on all IPL systems.

Substantial impact - this allows all NCR standalone system software to be considered for use in a partitioned IPL configuration.

4.9 The CENTURY 300 and large 200 systems are considered the major candidates for simulation on the IPL. All other CENTURY series machines will be accommodated thru existing upward compatibility within the CENTURY I series.

4.9 Minor impact - reduced scope of problem.

5. ANALYSIS OF THE PROBLEM

5.1 CENTURY I Operating Systems

In order to simulate the CENTURY I operating systems, it is necessary to have a thorough understanding of the features offered by these systems, and the techniques used to implement them. A complete description of the B-series operating systems is out of place in this document; instead, their salient features will be noted and major differences in philosophy with CYBER systems will be pointed out.

5.1.1 B1 Operating System

This is a monoprogramming operating system which processes some basic control cards and assists the user with I/O organization. As such, it may best be regarded as a Job Supervisor or Job Monitor similar to the FORTRAN Monitor on the IBM 7090. The system uses 3-1/2K bytes of memory and requires a system disk for its operation. It is widely used on CENTURY I models 50, 100, 101, and 200. The interface between user job and the OS is well defined but is frequently violated. For the purposes of this report it will be taken as a given that total simulation of the B1 operating system will be provided.

5.1.2 B2 Operating System

B2 was developed to handle on-line, real-time devices -- typically terminals. A tasking capability was added in B2 although typically only one task runs under B2, servicing a number of terminals. There is a millisecond call-by-interrupt real-time clock running under B2 which is used to time-out users. There are no known time critical problems under B2. (Time critical is

5.1.2 defined to mean where failure to respond in a given time frame results in failure of the effort). At worst, slow simulation would lengthen the response time at the terminal, which could lead to operator/user frustration. However, such irritation does not preclude simulation, and for the balance of this report B2 simulation will be considered equivalent to B1 simulation.

5.1.3 B3 Operating System

B3 is a multiprogramming operating system which simultaneously executes a fixed number of jobs. Each job is under the control of a separate copy of B1 or B2. Memory is allocated in fixed segments, and the processor is shared by different programs, thereby giving better utilization during I/O processing. Device allocation and basic multiprogramming system used by the larger CENTURY I machines - the 200, 251, and 300.

5.1.4 B4 Operating System

This system is the logical extension to B3 and provides general memory management facilities. The B1 or B2 segments controlling jobs are no longer constrained to fixed memory areas. B4 is a large system (64K - 80K bytes) which offers unit record spooling, job accounting, and job control via a job specification language, in addition to other features not found in B3. It is a new system which is currently in use at only a few sites. However, in the IPL time-frame (1975-1976), it is anticipated that it will be the system favored by most CENTURY 251 and up customers, and will therefore become the major OS considered in this report.

5.1.5 B-Series Variants

Along with the four basic systems described above (B1-B4), there are numerous variants which take advantage of hardware features found only on the larger machines. These variants are cataloged in Section 3.1.8. Since the differences in these systems are hardware related, rather than software related, the impact on the proposed simulator is expected to be minimal. There is one variant not cataloged, namely the NCR time-sharing system which runs BASIC and performs many of the functions of the B2 operating system. It is not proposed to analyze this system in detail.

5.1.6 Differences Between CDC and NCR Systems

Perhaps the most important and most significant difference between SCOPE/KRONOS operating systems and the B-Series lies in the handling of I/O devices. CDC approaches the problem in a logical fashion. The user refers to logical file names, and logical device types, the operator refers to a logical device name. The binding of logical files to physical devices is a dynamic function which does not take place until actual data transfer occurs. The NCR systems are diametrically opposed to this position. Devices are referred to by their physical addresses, (actual channel, equipment and unit numbers are used), and binding to device types is performed early (in the source language) and is static. Many of the problems to be solved in the simulator design stem from this basic philosophical difference.

5.2 General Simulator Design

5.2.1 Users issue two types of instruction to the CENTURY I computer: normal commands and privileged commands. The normal commands may be simulated directly and require no special actions on the part of the host system. Privileged commands deal with I/O and control of the system, and, in general, require that some special action be taken by the host system or by the simulator. A list of these privileged commands follows:

INOUT	-	The only I/O command on CENTURY computers.
WAITI	-	Pause for operator action.
SWIN	-	With appropriate address field, reads eight console switches into a byte; otherwise used as a supervisor call.
LDMONR	-	Debug aid to breakpoint a program or place machine in step-mode.
LDBAR	-	Sets bounds registers.
IPON	-	Interrupt permit on.
IPOFF	-	Interrupt permit off.

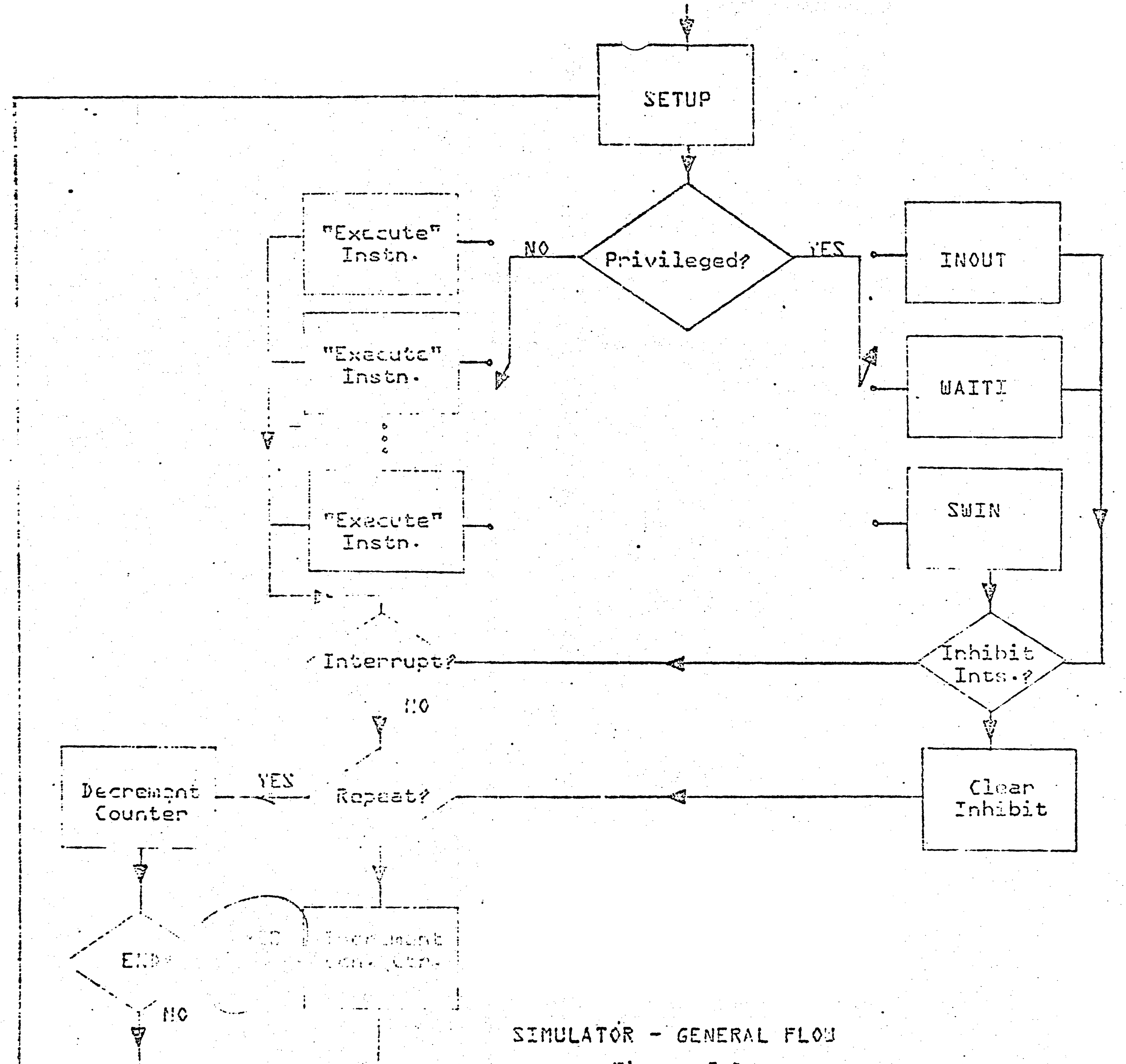
Logically then, instructions are divided into two categories by the simulator. Two other conditions must be treated by the simulator. The CENTURY I computer is interrupt driven and permits an instruction to be repeated a set number of times. Even if the host system is not interrupt driven, interrupts must be simulated for the CENTURY I environment to be totally reflected by the simulator.

5.2.2 A general block diagram of the simulator is shown in Figure 5.1. The box labelled "SETUP" simulates the instruction fetch of the CENTURY I and also simulates I/O interrupts. It is recognized that much remains hidden in this box, and the reasons for not breaking it down further are explained below. The privileged instructions requiring special action are shown on the right-hand side of the diagram and are discussed in some detail in the following paragraphs. It had been the intention of the team, having reached this step, to construct block diagrams for each CENTURY I hardware instruction. The diagrams would be broken down to the level where each box corresponded roughly to a 6150 instruction. It was hoped by this technique to establish approximate performance data for the simulator. Happily, this task was rendered unnecessary by the fact that NCR had already progressed beyond here to actual coding examples. Their resulting timing data has been utilized for cost/performance analyses discussed at a later stage. In addition to the basic instructions, NCR have also analyzed and coded the setup process which was described above.

5.2.3 Only the privileged instructions INOUT, WAITI, and SWIN are shown on the block diagram since the remainder can be handled without any interaction with the host system. The exception is LDMONR, which is strictly a debug aid. It was felt that the primary emphasis should be placed on the simulation of production codes, and certainly in the initial implementation debug aids could be omitted (N00Ped). There may also be marketing reasons for omitting the command from the simulator if it is desirable to have users make the transition from CENTURY I to IPL. Such philosophical reasons aside, there are no particular difficulties which were foreseen in simulating this command.

5.2.4 The INOUT Command

There are several problems associated with the INOUT command which are investigated in the following points.



SIMULATOR - GENERAL FLOW

Figure 5.1

5.2.4.1 Transformation of naming conventions

NCR identifies a device type by its physical address, that is, its trunk (T), position (P), and unit (U) numbers, or in CDC terminology its channel, equipment, and unit numbers. It is necessary for the simulator to make the translation from TPU to device type, and this can be done by constructing tables in the simulator as it is loaded. The B-series operating systems receive the same data via control cards (PAL cards) and the simulator input would parallel these entries. A suggested format of these entries in the simulator would be:

F L A G	T. P. U.	DEVIVE TYPE	o o o

The flag would indicate whether or not the INPUT being processed represented the first access to that particular device. The general flow for INOUT is therefore:

a. Locate PAF

b. Extract TPU and control word (CW)

c. If first access then REQUEST device from host operating system.

d. Translate TPU to lfn.

e. Translate request code.

f. Transform data/buffer formats.

g. Issue host I/O request.

h. Update I/O activity list for ENDOP test.

i. Insert S2 status.

5.2.4.2 Translation of TPU to lfn

Logical file names may be constructed by appending any alphanumeric character at the beginning of the TPU. For example, if the TPU was 6700, the lfn could be A6700.

5.2.4.3 Operator Messages

These will always be a problem in the simulated environment. The problem may be eased by establishing conventions for simulated jobs. For example, if a user issued messages to dismount one pack and mount another, then the operator would do exactly that - mounting the new pack on the same drive.

5.2.4.4 Alternative Solution

A more general solution to this entire problem area is to make physical device assignments, support NCR/

CENTURY peripherals directly, and dedicate a PPU to the simulator.

5.2.5 The SWIN Command

The general flow of the simulated SWIN command is shown in Figure 5.2. If the SWIN is a normal hardware instruction to read the console keys, then simulation will involve interaction with the host operator, or reading simulated keys held within the simulator field length. There may be a problem if NCR/CENTURY devices are not driven directly, since the SWIN command is used when calling on the operating system to share devices.

5.2.6 The WAITI Command

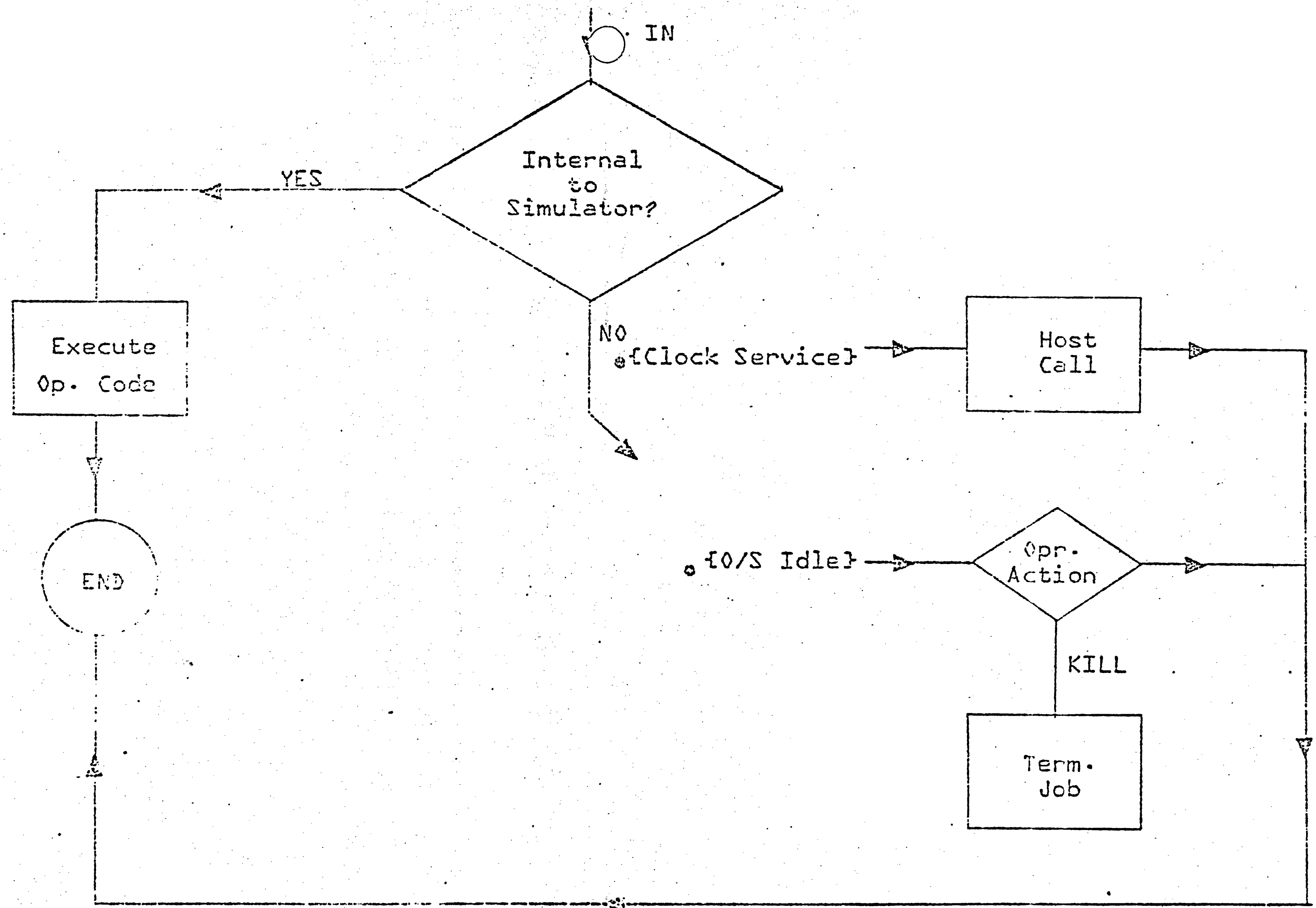
This command may be simulated directly by a call to the host to PAUSE, followed by operator action.

5.2.7 Other Problem Areas

Several other problem areas were identified by the team and solutions to them were proposed. Those problems and solutions are discussed below:

5.2.7.1 Data Path Transformation

File control procedures used on the CENTURY I computers are unique to them and are difficult to simulate. The simplest solution to this problem is to provide special PP codes that use the NCR/CENTURY



General SWIN Flow

Figure 5.2

file control procedures and tables.

5.2.7.2 Implicit Configuration Control

On the NCR/CENTURY computers, the configuration is controlled either by personal communication with the operator or by informal operator messages at the B1/2/3 level. B4 adds some explicit capability via control cards or by a formal protocol. Circumventing informal operator directives is a particularly difficult problem and three alternatives were postulated:

a. Design the simulator such that it is sensitive to message content. This solution was discarded by the team due to its EXTREME complexity.

b. Provide tools within the simulator for the opertor to determine the state of the simulated B-series environment. In addition, an operations protocol would have to be established as to how to use the tools provided.

c. Assuming that NCR/CENTURY devices are direct driven via a dedicated PPU, then the problem is automatically eliminated, and operator action would be identical to that seen today in the CENTURY I environment.

5.2.7.2 B4 Operator Displays

The general displays set up by B4 for the operator will need special actions by the simulator, particularly where touch plates are concerned. Normal device simulation through the host console should suffice to eliminate this problem. Touch keys are simulated by operator type-in and flag settings in the simulator.

5.2.7.4 Idle Loops

It is imperative that the host OS control any idling in the system. Idle loops in B4, for example, must relinquish control to the host immediately. There are two types of idle loops: transient (a job or jobs in the system but waiting for I/O maybe),and permanent - no more jobs to run. The simulated system must be modified to return control to the host whenever the idle loop is entered. In return, the host must give control back to the simulator on previously listed conditions being met. When the "no job" idle condition occurs, then the operator must be interrogated for subsequent action.

5.2.7.5 Engineering/Accounting Logs

The host system will maintain both engineering and accounting logs. These logs, or portions of them, may be duplicated by the simulated system, and accounting may become confused in the process. The two logs are addressed separately.

5.2.7.5.1 Engineering Logs

The host system is responsible for logging all activity on physical devices (including memory parity errors) and this information, in general, will not be passed on to the simulated system for duplication. Where the simulator is using direct drive NCR/CENTURY devices, error data will be handed back to the simulated operating system for logging there. The maintenance engineers, in this case, will have a choice of the source of data for their maintenance activities.

5.2.7.5.2 Accounting Logs

The accounting information will be kept in the simulated system, typically B4 since B1 does not perform any accounting functions. The difficulty is in absorbing the overhead due to simulation. The degree of difficulty in a rigorous solution to this problem by far outweighs the benefits derived by such an exercise, hence such a solution should not be attempted. In the B1, simulated environment, accounting will be accomplished by the host system, and the simulator overhead will be absorbed by the user automatically. This entire problem area is not regarded as serious, since few NCR/CENTURY customers are known to use accounting procedures.

5.2.7.6 Feature Redundancy

Many features will be duplicated by the host and simulated systems, for example:

- spooling
- accounting/engineering logs
- operator communication
- data transformation
- device error recovery
- job control language, etc.

It is desirable, to maximize performance, to minimize these redundant activities. Some of these problems are addressed separately. Spooling appears to be a major problem which will require special attention. A solution is to modify the simulated system (B4) to eliminate the feature from it. This particular form of solution has been avoided here by incurring the penalty of an additional disc to disc transfer to simulate the spooling function of B4. In such cases, trade-offs will have to be made between exact simulation and performance.

5.2.7.7 Remote Job Entry

This is a particularly difficult problem in the B4 simulated mode since the job being run by the host is the simulator. Jobs submitted to B4 by NCR/CENTURY users enter the host system as data to the simulator. It is difficult to arrange for this data to be submitted, without demand, from a remote site. The problem evaporates in the B1, environment, and consideration is given to such a simulation in a later section of this paper. The normal modus operandi of the NCR/CENTURY system is not RJE, hence the problem is not seen as a major one.

5.2.7.8 Writing on System Disc

What amounts to a common, read only disc in the host system is written on by individual B1 partitions.

Assuming that NCR/CENTURY devices are direct devices, there is no problem since in the B4 environment a solution has already been found; and in the B1 environment the host will only permit one B1 simulation to execute at one time per physical device. This will not require any special action on the part of the host if it is assumed that there are only sufficient devices for one simulation. Normal resource allocation algorithms will solve the problem. Where the NCR/CENTURY devices are simulated on the host disc, then multiple copies of the simulated system file will be made on the host disc, and users granted read, write, modify, and extend permission on the copied files.

5.2.7.9 System Initialization

In any simulated environment such as that proposed here, it is important to insure that the simulated computer can be initialized/deadstarted in an orderly fashion. The simulator will be loaded as a normal job in the host system, and its first task will be to process its control cards. It has already been discribed how it is necessary to describe the configuration to the simulator prior to execution of a simulated job. The simulator would next create the bootstrap code, normally supplied by hardware, then execute that code as a regular simulated sequence.

Since the system initialization process is usually a somewhat lengthy process, it is recommended that a checkpoint dump be made on completion of it and that subsequent usage commence from that point.

5.2.7.10 Real-Time Considerations

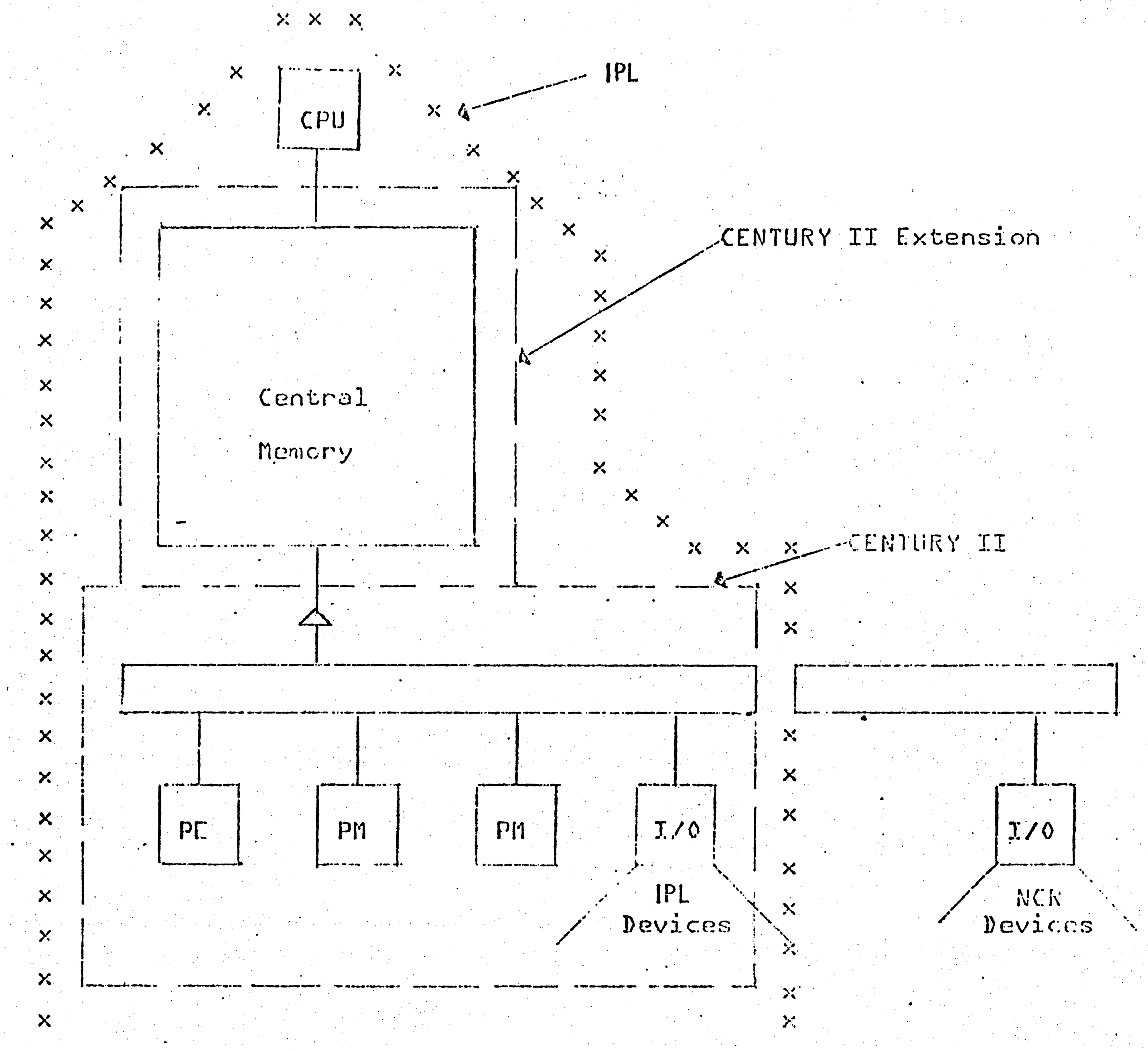
This problem is defined as the failure to respond to an external stimulus in a given time-frame causing the process to fail. A secondary issue is failure to respond to an external stimulus resulting in inefficiency or operator/user frustration. The primary case does not have a general solution. Either the simulator is fast enough to handle the required response time or it is not. If the simulator is not fast enough, then the application must be placed on the host computer. The only application or device on NCR/CENTURY equipment known to fall into this category is the MICR. On this device a check is read; and, while it is moving through the reader, computations are performed to route it to the correct output hopper. If these computations are not completed before the check reaches the output hoppers, then the check is automatically rejected by the device. All other time dependent situations on NCR computers appear to be of the secondary type of consideration.

5.3 An Alternative Approach

5.3.1 In the preceding pages the general problem of simulating one computer system on another has been addressed. The NCR/CENTURY series of computers has been investigated in detail, problem areas identified, and solutions proposed. It is postulated that the problems with that machine are not unique, but many of them, and their solutions, will apply to other computer systems such as CDC's 3000L series. For this reason, the problem has been addressed in a fairly general fashion, and no assumptions have been made regarding specific hardware configurations of the host computer. However, if probably configurations are considered together with current NCR development activities, then alternative approaches to the task in hand present themselves.

5.3.2 Configuration

The general, likely, configuration for the IPL is shown in Figure 5.3. A key element in this configuration is the NCR 6150 computer. Currently, NCR is engaged in a development program for a CENTURY II computer, namely the 6150, and CENTURY I simulators are under development for this computer. It is not difficult to establish a growth pattern for a CENTURY user from this configuration, and a viable pattern is shown in Figure 5.4. If this train of thought is pursued even further, then it is readily apparent that the entire task of simulation of the NCR/CENTURY computer can be achieved by a logical segmentation of the hardware. This logical separation of



Temporary Growth Configuration

Figure 5.3

a portion of the hardware by the host system will be termed "partitioning" in the remainder of this report.

5.3.3 The advantages of partitioning are numerous. Many of the problems already discussed disappear in this environment. The hardware growth pattern follows the general IPL approach. Duplicity of effort is eliminated or minimized. Against these advantages are the introduction of a difficult transition problem from the simulated to the real environment. In addition, some extra hardware may be required to drive both the NCR/CENTURY I/O devices and the IPL I/O devices. The transitional steps would be:

a. Run in a dedicated, simulated CENTURY I mode.

b. Alternate between the IPL mode and dedicated CENTURY I mode.

c. Run the IPL mode and CENTURY I mode concurrently.

d. Run in IPL native mode only.

or

e. Run an integrated IPL/CENTURY I system. By this is meant that the additional development steps would be required to:

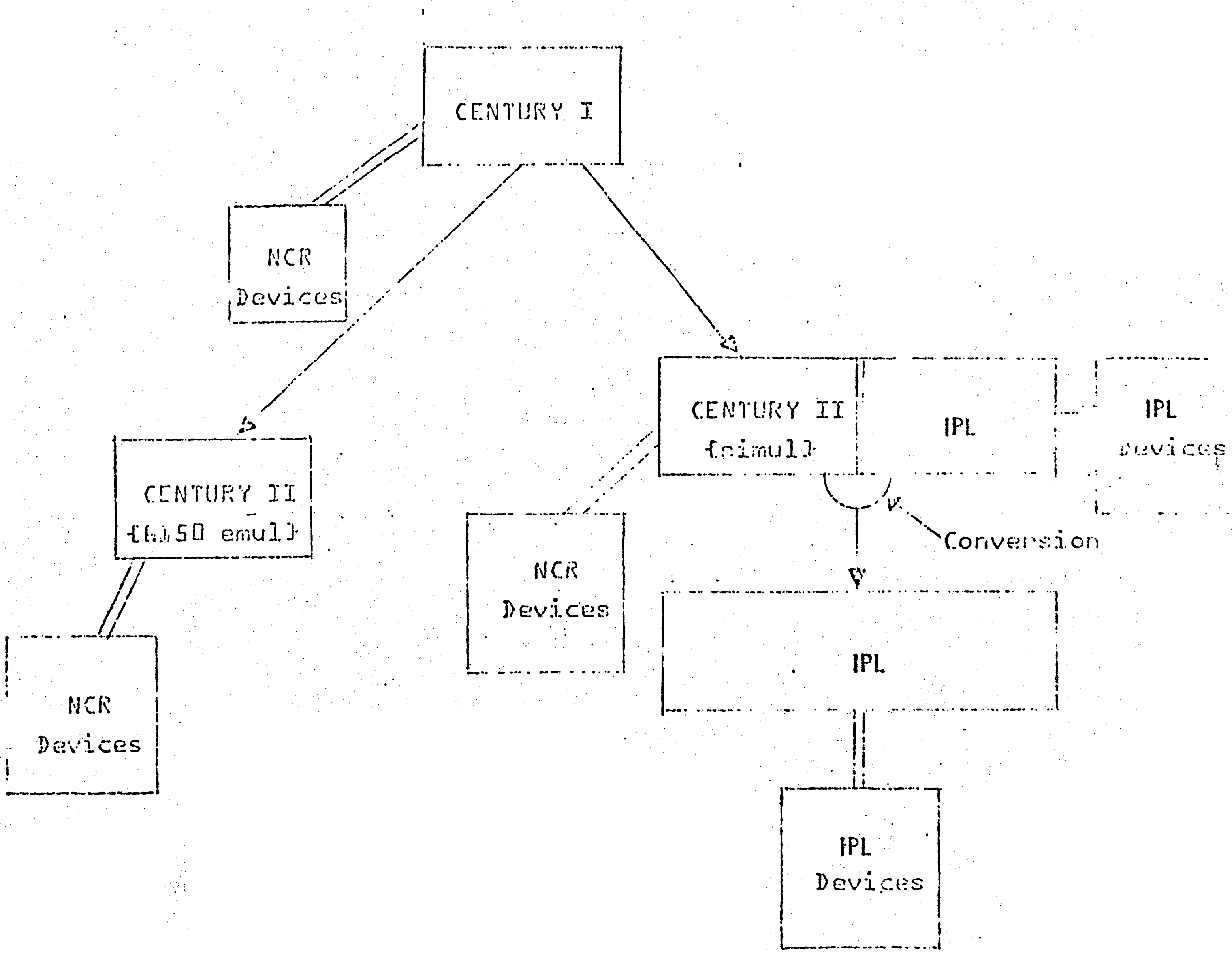
(1) Exchange jobs;

(2) Exchange job steps;

(3) Exchange entire files;

(4) Exchange partial files.

The exact course chosen here will probably be dictated by marketing strategies and other less technically oriented factors.



CENTURY I/IPL Growth Pattern

Figure 5.4

5.3.4 Given that the NCR 6150 is an integral part of the IPL, and given that NCR devices will be direct driven by the IPL, and given that NCR are pursuing the 6150 as a stand-alone CENTURY I emulator, then the partitioned approach is the favored approach. It is a solution unique to the CENTURY I problem and a similar approach for other computer systems does not follow automatically as a corollary to this argument.

5.4 Operating System Simulation Considerations

5.4.1 The overall problem of simulating the CENTURY I computer system may be addressed at several levels. The approach of providing like operating system functions and executing user code directly was considered but rejected due to the relative ease of alternate techniques which have substantially smaller effects on the host operating system and provide comprable service and performance to the customer and user. The possibility of simulating either the B1 operating system or the B4 operating system was considered also. On the surface there appears to be little gained by simulating B4, since its primary function is to multiprogram several jobs, each under the control of a B1 system. It can safely be assumed that the host system will be a multiprogramming system, in which case by simulating B1 it will provide many of the basic facilities supplied by B4, and thereby eliminate the need to simulate B4. However, a more exhaustive study of the problem reveals other factors which make the choice less obvious.

5.4.2 Advantages of Simulating B4 (In addition to B1)

a. Offers least change for the B4 user. In practice, probably no change need be made to a user job executing under a simulated B4.

b. It will be the least effort for the vendor. Assuming that B4 development on CENTURY I continues and overlaps the introduction of the IPL, then continuing changes to B4 will be accomodated automatically.

c. The host system or the simulator need not provide B4 features not provided by stand-alone B1.

5.4.3 Disadvantages of Simulating B4 (In addition to B1)

a. Several functions will be executed redundantly by B4 and the host system.

b. Changes will have to be made to B4. (Many small changes , no large changes identified)

c. The operator console is difficult (but not impossible) to simulate.

d. Performance will be degraded over a pure B1 simulation due to the additional code to be executed by the simulator.

5.4.4 Memory Requirements

When considering the pros and cons of simulating B4 versus B1, the amount of memory required, in addition to that of the user jobs must be factored in. For this purpose, the following size estimates were used:

a. Simulator - 50 - 80K Bytes (based on several simulated efforts on CDC CYBER 70)

b. B4 Operating System - 64 - 80K Bytes

c. B3 Operating System - 16K Bytes

d. B2 Operating System - 7K Bytes

e. B1 Operating System - 5K Bytes

This suggests that a single simulation of B4 is roughly equivalent to two versions of B1 being simulated simultaneously, remembering, of course, that each job under B4 has its own copy of B1. B4 is able to run several jobs simultaneously, hence memory utilization favors the B4 simulation. It is recognized that the memory requirements for the simulator have been exaggerated; however, that is offset by the resource allocation problem discussed previously, and the difficulties inherent in running multiple copies of a B1 simulator.

Note: Any shared code facility provided by the host OS will invalidate the above argument.

5.4.5 Summary

The provision for B4 simulation in addition to B1 simulation, allows the least change for user and vendor at a possible substantial performance cost to the user. All three points appear to be positive from the vendor's point of view and therefore it appears as a reasonable plan of action. (Performance cost to the user can be translated to additional hardware sales for the vendor).

5.5 Peripheral Support Issues

5.5.1 Transitional steps for CENTURY I customers

For purposes of evaluating the peripheral support issues involved in a customer moving through the CENTURY I, II product line the following progression paths have been postulated:

(i) CI H/W, CI S/W, CI PERIPHS. → (ii) a. CI H/W, CI S/W, CII PERIPHS. → (iii) a. CII H/W, CII S/W, CII PERIPHS. → (V)

OR (i) → (ii) b. CII H/W (=6150), CII S/W (=CI S/W), CI PERIPHS. → (iii) b. CII H/W, CII S/W, CI PERIPHS., CII PERIPHS. → (W)

from (iii) b. → (iv) CII H/W, CII S/W, CII PERIPHS. → (v) IPL H/W (8600÷6150), CII S/W, CII PERIPHS., IPL PERIPHS. → (vi) IPL H/W, IPL S/W, IPL PERIPHS.

from (iii) a. → (v)

Note: Any or all of the above steps may be skipped. (In fact some are redundant).

The distinction between CII peripherals and IPL peripherals may be non existant. It is herein defined to be at worst a different system utilization of exactly similar physical devices. (i.e., a software difference only.)

As per the peripherals the above transition network can be reduced to the following case analysis:

1 Simulator direct driving peripherals → CI devices ⇒ CII devices

2 Simulation uses host OS I/O devices → CI devices ⇒ IPL devices

(the double arrow indicates expected utilization density)

Case 1 is in fact the same case as that for NCR supporting its stand-alone customer base. The results of this NCR support in conjunction with the recommended partitioned configuration solution will be no additional effort for vendor or customer when moving to the IPL.

Case 2 will invalidate the parititoned configuration solution. It essentially requires that the host OS and/or simulator become intimately aware of the inherent characteristics of the I/O structures and protocols for both CENTURY I and IPL systems. The following sections indicate each major class of device type, special problem areas and some tentative solutions for e[illegible] It may be stated that in general this case complicates the process [illegible] both vendor and customer. The vendor must develop and maintain a more complicated simulator package that is in addition to the standard OS offerings. The customer must convert his data files from the CENTURY I device to the corresponding IPL device. (Perhaps the most complex and difficult portion of any conversion effort.)

5.5.2 Device Types

5.5.2.1 Disks – It is necessary in the simulator to map the physical addresses used by CENTURY I software into an address space provided by the host on a similar device. If a word addressable disk space is made available on the IPL, similar to that on CYBER-70 today, then there are few problems. In the general case it will be necessary to address the IPL device by byte. The transformation is then straightforward, and can be performed at execution time by the simulator.

– The user at some stage will face a conversion problem and a utility must be provided for that purpose. This utility could either execute from disk to disk or from disk to disk via magnetic tape.

5.5.2.2 Magnetic Tapes – National and international standards in this area make simulation a simple matter. All that is necessary is for the host to deliver a block of data to the simulator, or receive a block from the simulator on demand. Logical, record handling remains the responsiblity of the simulated system. The simulator must make appropriate character conversions if the new devices are incompatible with the old.

– As a general rule, error processing will be performed by the host, and only final status will be delivered to the simulator.

5.5.2.3 Card Readers/Punches - The only problem which must be solved is that of code conversion which can be readily handled by the simulator.

- For card punches which have stackers there is an additional problem. The stacker select codes will have to be translated by the simulator where equivalent devices are available. If there is no equivalent device then it really falls outside the scope of the discussion. However, the problem can still be solved quite easily by sending different stacker output to separate files.

5.5.2.4 Printers - Current 160 column line printers supported by NCR are not expected to be continued in the IPL and will not be considered.

- In the B4 environment spooling presents a problem. Spooling basically converts a parallel process into a serial process for a single printer. If spooling is removed from B4, then B4 will have to give assistance to the simulator to differentiate between the same TPU's emanating from different B1 segments. This problem is not unique to printers but applies to all unit record devices which are spooled by B4. The problem is non-serious in that if sufficient printers, either real or simulated are available, then B4 can handle all print files on-line.

Many simulated printers may be supplied via host spooled files.

- Different print trains are expected to impose a further data transformation on the simulator. Currently NCR does not support this facility, but will prior to the introduction of the IPL.

5.5.2.5 Paper-Tape - No particular problems are seen to exist with paper tape, and either a direct driven device or a spooled host service should suffice in much the same way as for card readers. Labels will present several difficulties since the host system will not be able to distinguish them from data and in the spooled (host) environment cannot label automatically. If trailer labels are utilized to carry reconciliatory data from the file then an insolvable problem exists.

5.5.2.6 Other Devices - The following devices were not addressed in detail.

(i)	CRAM	- No IPL equivalent
(ii)	MICR	- Already discussed elsewhere
(iii)	OCR	- Similar to a CR or CP
(iv)	Plotters	- No problems foreseen
(v)	Mohawk	- Normal MT problem
(vi)	Terminals	- Already discussed elsewhere

6.0 COST AND PERFORMANCE ESTIMATES

In order to devise a true cost/performance estimate, data being gathered by other teams (notably System Design) will have to be used. In this section, the cost to NCR/CDC to produce a simulator similar to that described here is calculated, along with the performance as compared to a CENTURY I/200, and in terms of the percentage of the resources required.

6.1 Cost to Produce Simulator

6.1.1 The cost of the simulator is directly calculable from the number of lines of code in the program. In deriving this key figure, heavy reliance was placed on past experience with similar programs. The data which was gathered is shown below.

	Simulator	Lines of Code
a.	1401 on CDC 6600	6000 (est)
b.	360/20 on CDC 6600	6000 (est)
c.	315 on NCR Century I	6000
d.	7600 on CDC 6600/7600	1800*
e.	8600 on CDC 6600	6000

* Excludes I/O simulation. There are 25K-30K lines of code covering this area and debug tools which cannot be broken apart readily.

On this basis, the Century I simulator described in this paper is assumed to consist of 6000 lines of assembly code.

6.1.2 Given that there are 21 working days per month, and that a programmer can generate ten lines of code per day, then the simulator can be written in 5 man-years. The figure of 10 lines of code per man day is high by industry standards but may be justified on two points: the resulting system is small, and a high level language will be used for implementation. Given the 5 man-year estimate and assuming a programmer cost of $50K per year:

Implementation cost = $250K

6.1.3 Integration and Validation costs are difficult to assess for such efforts. A cost estimate for CDC 3000L emulation on CYBER -70 has been set at $1.5M (this is viewed as an upper bound for Century simulation).

6.1.4 Since the code is small (6000 lines) the error rate will be low - perhaps no greater than 1%. Assuming such a rate, and assuming a programmer correction rate of 10 errors per month, then approximately 6 man months per year will be required for maintenance.

Maintenance costs = $25K per year.

6.2 Performance of Simulator

6.2.1 The two modes of simulation, imbedded and partitioned, discussed in Section 5 are considered in this performance analysis. As a basis for performance estimation, the simulation work conducted by J. N. Tomblin of NCR has been drawn on very heavily. In this work, the instruction set-up has been coded together with 38 of the simpler NCR/Century instructions. These have been used to run timing analyses through a Kendall mix. The results indicate that, given a 660 nsec memory 6150, then the simulator is roughly 1-1/3 times as fast as a Century I/300. (4.3 usec versus 6 usec). The design goal of this exercise is twice the Century I/300. This figure (2X) which is best case will be used arbitrarily in the following analysis.

6.2.2 Partitioned mode

Given that the Century I simulator can execute Century I code at twice the rate of a Century I/300, then 50% of the processor power on the IPL will be available to perform IPL work. If the IPL mode and simulator mode are run concurrently, and states are switched dynamically, and overhead when switching is 50%, then available IPL power is 25% of its full potential. The maximum power available is 50% of its total power when the overhead is zero. This is a worst case calculation since it is unlikely that typical jobs are CPU bound. All unused process time can go to IPL jobs.

6.2.3 Imbedded Mode

Performance in this mode is somewhat less favorable since additional overhead is incurred for the following reasons:

a. Data transformation;

b. Host interfacing;

c. Execution of redundant functions.

Consequently, it is estimated that the simulator will execute at a rate equal to 1-1/2 times that of a Century 1/300 In this case, the available processor power of the IPL will be approximately 15% of its full potential.

7.0 SUMMARY OF RECOMMENDED SOLUTIONS

7.1 Partitioned Configuration

- Can utilize NCR standalone system software as the simulator with minimum change.
- Direct drive peripherals as NCR native devices.
- Utilize "RJE type" job interchange between Host O.S. and simulated O.S.
- Applies best to configurations having a small number of peripheral devices.
- Allows customer and user to continue to use existing operation, administrative and usage protocols.
- Least development expense.
- Moderate relative performance (SWAG)
- Least attractive to user because of disjoint system but least user conversion.

7.2 Host Imbedded Simulator Support Package

(with direct driven peripherals)

- Total configuration control is maintained by Host O.S.
- Common I/O queues and input/output procedures.
- Applies best to configurations having a moderate to large number of peripherals.
- Most development expense.
- Worst relative performance (SWAG).
- Moderate attractiveness and user conversion.

7.3 Host Imbedded Simulator Support Package

(with peripherals driven thru Host O.S.)

- Take configuration control is maintained by Host O.S.
- Total control of I/O support is maintained by Host O.S.
- Applies best to configurations having a large number of peripherals and/or a large real memory capacity.
- Moderate development expense.
- Best relative performance (SWAG).
- Most attractive to users but requires greatest user conversion.