CERN 75-19
Laboratory I
Data Handling Division
22 December 1975

ORGANISATION EUROPÉENNE POUR LA RECHERCHE NUCLÉAIRE
CERN EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

INSTRUCTION TIMING FOR THE CDC 7600 COMPUTER

H. Lipps

G E N E V A

1975

CERN 75-19
Laboratory I
Data Handling Division
22 December 1975

ORGANISATION EUROPÉENNE POUR LA RECHERCHE NUCLÉAIRE
CERN EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

INSTRUCTION TIMING FOR THE CDC 7600 COMPUTER

H. Lipps

GENEVA

1975

CERN - Service d'Information scientifique - RD/93 - 1000 - decembre 1975
This report has been prepared using the text processing program CERNTEXT.

ABSTRACT

This report provides timing information for all instructions of the Control Data 7600 computer, except for instructions of type 01X, to enable the optimization of 7600 programs. The timing rules serve as background information for timing charts which are produced by a program (TIME76) of the CERN Program Library. The rules that co-ordinate the different sections of the CPU are stated in as much detail as is necessary to time the flow of instructions for a given sequence of code. Instruction fetch, instruction issue, and access to small core memory are treated at length, since details are not available from the computer manuals. Annotated timing charts are given for 24 examples, chosen to display the full range of timing considerations.

INTRODUCTION

This report addresses the programmer who wants to optimize his code for the Control Data 7600 computer, and who therefore requires detailed knowledge on how the 7600 executes its instructions in parallel. We assume that the reader is familiar with the 7600 to the extent that it is documented in the manuals available from the manufacturer. (See Control Data 7600 / Cyber 70 Model 76 Computer Systems Hardware Reference Manual, CDC Publication No. 60367200)

The 7600 is a synchronous computer with a clock period of 27.5 nanoseconds. All CPU registers retain their values throughout the clock period. Information moves during the clock period from the registers through transmission paths and static networks back to the registers. The static networks determine the register values for the following clock period. The contents of the registers then change simultaneously and instantaneously as the leading edge of the clock pulse arrives from the clock.

During any one clock period several instructions are normally in various stages of execution. Concurrency is achieved through use of nine independent, segmented functional units, use of an instruction stack which fetches instructions in advance, and use of independent memory banks. The concurrency is limited by sequencing controls which ensure that the outcome of the computation is the same as if the instructions of the program were executed in strict sequence. The steps taken by different sections of the CPU to perform their particular function, and the rules which coordinate the action of these sections will be stated below in as much detail as is necessary to time the flow of instructions for a given sequence of code. The instruction codes 01X will, however, be excluded from consideration. The timing rules which govern instruction fetch and instruction issue, together with the timing of SCM, are stated first. Then follow specific rules for each class of instructions. Finally, a number of examples are discussed which have been chosen to display the full range of timing considerations rather than meaningful or optimum code. Instruction fetch is covered at length, as it presents by far the most involved aspect of instruction timing for the 7600 and is not treated in the manuals. On first reading, it might be preferable to skip the chapter INSTRUCTION FETCH and approach the subject from the examples first.

The timing charts which accompany the examples have been printed by program TIME76 in accordance with the following conventions. Similar conventions apply for the timing charts which illustrate the main text.

IAS(12)

The full address of an instruction word is printed for the first clock period in which the instruction word is available from (rank 12 of) the instruction stack.

Address

The full address of the instruction word which is contained in the CIW is printed for the clock period in which the first instruction of this word issues.

COMPASS Mnemonic

Each instruction which is executed is printed for the clock period in which the instruction issues from the CIW.

Register Reservation

The reservation of an X-, A-, or B-register by a section of the CPU is shown by an appropriate letter. A dot indicates that the register is free. B0 is not shown as it is never reserved.

Clock

Consecutive clock periods are numbered 0,1,2,...; one line is printed per clock period.

IFA

As long as a request for an instruction word is held in the IFA register, the last two octal digits of its address are printed.

SAS

As long as a request for an SCM access is held in the SAS, the last two octal digits of its address are printed under the appropriate subheading (A, B, or C). Note that an entry under subheading B or C indicates an SAS backup condition.

Small Core Memory

As long as an SCM bank is busy executing a storage reference, the last octal digit of that bank is printed under the appropriate subheading.

INSTRUCTION SEQUENCING

The program establishes a strict sequence of instructions to be executed. The programmer may think of his instructions as being executed, one by one, in this order and assume that the result of any instruction is available to the next instruction when this second instruction begins execution.

Although the 7600 tends to execute several instructions in parallel, instruction overlap is limited by the condition that it must not alter the outcome of the computation or the flow of control. (The programmer of the 7600 will be aware of the precautions which he must take when he modifies instructions at execution time.)

THE CURRENT INSTRUCTION WORD REGISTER (CIW)

The 7600 determines the programmed sequence of instructions with the help of a 60-bit register, called the current instruction word register (CIW). Instruction words enter the CIW according to their programmed sequence.

Definition

> An instruction is being interpreted while it occupies the upper part of the CIW. (Bits 59 to 45 or bits 59 to 30, depending on the type of the instruction)

The 7600 interpretes instructions in exactly the sequence which is established by the program. In particular, the instructions in a given instruction word are interpreted from left to right. When interpretation of a 15-bit instruction ends and there are more instructions of the word which must be executed, then the contents of the CIW are shifted left 15 bits, so that the next instruction will be interpreted. When interpretation of a 30-bit instruction ends and there are more instructions of the word which must be executed, then the contents of the CIW are shifted left 30 bits, so that the next instruction will be interpreted. The shifts are end-off and with zero-fill from the right. They all execute in one clock period.

THE INSTRUCTION STACK

The instruction stack consists of the instruction word stack (IWS), the instruction address stack (IAS), and various flags and control registers. The IWS consists of twelve 60-bit registers; each one can hold an instruction word. The IAS consists of twelve 18-bit registers; each one can hold an instruction word address. The IWS and IAS registers are associated in pairs, and the pairs identified by "rank" - rank 1 to rank 12.

The IWS obtains its information, one word at a time, from SCM. As an instruction word arrives from memory it is entered into rank 12 of the IWS while its SCM address enters rank 12 of the IAS. During the same clock period the current contents of ranks 12, 11, ...,2 are all shifted one rank into ranks 11, 10,...,1 respectively. The information in rank 1 of the instruction stack is discarded.

Thus, rank 12 of the IWS always holds the last instruction word which has been entered from SCM while rank 1 holds the instruction word which has been longest in the stack. The addresses in the IAS need not be consecutive, and the same address may occasionally appear in more than one rank.

Execution of a return jump instruction or of any exchange jump voids the current contents of the instruction stack. (Zeroes are entered into all registers of the IAS.) Thereafter, the instruction stack gradually fills up again, starting with rank 12.

INSTRUCTION ISSUE

Definition

> An instruction is said to issue (from the CIW) during the last clock period of its interpretation; i.e. during the clock period in which it is overwritten in the CIW.

Definition

> An instruction word is said to issue (from the CIW) when the last instruction issues that must be executed in this word.

Whenever an instruction word issues from the CIW the address of the next instruction word is contained in the P-register. During the execution of linear code this will be the next sequential address; if the instruction which issues performs a jump, the P-register will hold the effective jump address.

As the instruction word issues, the P-register is compared concurrently with all twelve ranks of the IAS. Coincidence of P with rank n of the IAS causes the instruction word in rank n of the IWS to enter the CIW. The new instruction word enters the CIW in the same clock period in which the instruction word issues.

If there is no coincidence with any rank of the IAS a word of zeroes enters the CIW; the CIW will then be considered empty until it has obtained the next instruction word from the instruction stack. (See REFILL OF AN EMPTY CIW.)

REGISTER RESERVATION

X-REGISTER RESERVATION

The use of X-registers is controlled by eight X-register reservation flags, one per X-register. Any instruction which places a result into Xi sets the Xi reservation flag during the first clock period of instruction execution. The flag remains set during all further clock periods that are required for the execution of this instruction. It is cleared during the same clock period in which the result is placed into Xi.

Definition

> The Xi-register is said to be reserved (for an instruction) while the Xi reservation flag is set (as part of the execution of this instruction).

An instruction which uses an X-register as an operand or result register does not begin execution during a clock period in which the X-register is reserved. It follows that if an instruction places a result into an X-register, the X-register is always free during the first clock period of instruction execution, and also during the clock period which immediately follows the completion of the instruction.

Definition

> If an instruction does not begin execution because it uses an X-register which is currently reserved for some previous instruction, the delay is called an X-register conflict.

See Example III.

B-REGISTER RESERVATION

The use of B-registers is controlled by seven B-register reservation flags, one for every B-register other than BØ. Any instruction which places a result into Bi (i>Ø) sets the Bi reservation flag during the first clock period of instruction execution. The flag remains set during all further clock periods that are required for the execution of this instruction. It is cleared during the same clock period in which the result is placed into Bi.

Definition

> The Bi-register is said to be reserved (for an instruction) while the Bi reservation flag is set (as part of the execution of this instruction).

BØ is never reserved.

An instruction which uses a B-register as an operand or result register does not begin execution during a clock period in which the B-register is reserved. It follows that if an instruction places a result into a B-register, the B-register is always free during the first clock period of instruction execution, and also during the clock period which immediately follows the completion of the instruction.

Definition

> If an instruction does not begin execution because it uses a B-register which is currently reserved for some previous instruction, the delay is called a B-register conflict.

See Example IV.

A-REGISTER RESERVATION

The use of A-registers is controlled by eight A-register reservation flags, one per A-register. Any instruction which places a result into Ai (i.e. any increment instruction SAi ...) sets the Ai reservation flag during the first clock period of instruction execution. The flag remains set during the second clock period of instruction execution. It is cleared during the same clock period in which the result is placed into Ai.

Definition

> The Ai-register is said to be reserved (for an instruction) while the Ai reservation flag is set (as part of the execution of this instruction).

An instruction which uses an A-register as an operand or result register does not begin execution during a clock period in which the A-register is reserved. It follows that if an instruction places a result into an A-register, the A-register is always free during the first clock period of instruction execution, and also during the clock period which immediately follows the completion of the instruction.

Definition

> If an instruction does not begin execution because it uses an A-register which is currently reserved for some previous instruction, the delay is called an A-register conflict.

See Example V.

DESTINATION PATHS

X-REGISTER DESTINATION PATH

All eight X-registers share a common path over which they obtain new data, the X-register destination path. Only one of the X-registers can therefore receive a result during any given clock period, no matter from where the result arrives.

All instructions which place a result into an X-register reserve the X-register destination path for the one clock period in which the result is placed into the X-register. SCM destination control prevents all access to SCM bank address registers during clock period 14 of a divide instruction. An operand fetch instruction SAi (i = 1,2,...,5) will therefore never require the X-register destination path concurrently with the divide unit. All instructions other than operand fetch instructions which deliver a result to an X-register reserve the X-register destination path for the appropriate clock period during the clock period in which they issue.

Definition

> If an instruction which places a result into an X-register does not begin execution because the X-register destination path will not be free at the time when it will be required for the instruction, or if an address cannot be delivered to SCM because the X-register destination path will not be free at the time when it might be required for the delivery of an operand, the delay is called an Xd conflict.

See Example VI.

B-REGISTER DESTINATION PATH

All eight B-registers share a common path over which they obtain new data, the B-register destination path. Only one of the B-registers can therefore receive a result during any given clock period, no matter from where the result arrives.

All instructions which place a result into a B-register reserve the B-register destination path for the one clock period in which the result is placed into the B-register. The reservation is made for the appropriate clock period during the clock period of instruction issue.

No exception to this rule is made for B0.

Definition

> If an instruction which places a result into a B-register does not begin execution because the B-register destination path will not be free at the time when it will be required for the instruction, the delay is called a Bd conflict.

See Example VII.

SMALL CORE MEMORY (SCM)

Each SCM bank has 2K 60-bit words. Banks are interleaved, so that consecutive SCM addresses refer to different banks. A machine with 64K words of SCM has addresses n and n+32 in the same bank.

SCM WRITE OPERATION

Clock Period	Action
0	Address enters bank address register.
1	Begin read cycle.
2	
3	
4	End read cycle.
5	Data word enters write operand register.
6	Begin write cycle.
7	
8	
9	End write cycle.

SCM READ OPERATION

Clock Period	Action
0	Address enters bank address register.
1	Begin read cycle.
2	
3	
4	Storage word enters read operand register.
5	Storage word enters write operand register.
6	Begin write cycle.
7	
8	
9	End write cycle.

The storage word is always delivered from the read operand register to its destination during clock period 5 of the above sequence.

See Examples VIII and IX.

THE STORAGE ADDRESS STACK (SAS)

Requests for SCM access arise from the execution of instructions, interrupts, and I/O operations. Although they arise in different sections of the CPU, they are all channelled through the storage address stack. The SAS processes requests first-in-first-out.

At any one clock period SAS can hold up to three requests in its three ranks - A, B, and C. Each request consists of an SCM address and a four-bit tag which describes the function to be performed and, for read operations, the destination of the word which must be read.

Definition

If a memory request is stored in rank B or rank C of SAS, this is called an SAS backup condition.

Requests are entered into SAS under control of the SCM access control unit. The SCM access control unit can fill SAS at a maximum rate of one request per clock period, but delays may be caused by an SAS backup condition.

Requests leave the SAS for the appropriate SCM bank address registers under control of the SCM destination control unit. The SCM destination control unit can empty SAS at a maximum rate of one request per clock period. Delays will occur in case of SCM bank conflict or conflict with the divide unit.

The following rules apply to the flow of requests through SAS.

Rank B is never filled unless rank A holds a request. Any request in rank B remains there until rank A has become free.

Rank C is never filled unless rank B holds a request. Any request in rank C remains there until rank B has become free.

Rank A is never filled as long as ranks B or C hold a request.

While ranks B and C are free, a new request may enter rank A in the same clock period in which the previous request leaves rank A.

Consequently, as long as there is no delay by the SCM destination control unit all requests will pass through rank A even if one or more requests are made every clock period. If SAS holds more than one request in a given clock period, rank A holds the oldest and rank C the youngest request admitted to SAS.

SCM DESTINATION CONTROL UNIT

Issue of Request

The oldest SCM address in SAS is delivered to the appropriate SCM bank address register in CP 0 provided

The corresponding bank busy flag is clear during CP Ø.

No divide instruction was issued during CP -14.

Execution of Request

Following the delivery of an SCM address to an SCM bank address register in CP Ø

The bank busy flag of the SCM bank is set during CP 1 to CP 9.

The X-register destination path is reserved for CP 5 in case the SCM access results from an operand fetch instruction SAi (i=1,2,...,5).

SCM ACCESS CONTROL UNIT

A request for an SCM access and the appropriate SCM address itself are entered into SAS during CP Ø provided one of the following conditions holds.

If an increment instruction SAi (i>Ø) was issued during CP -1 then the effective address of this instruction is entered.

If no such instruction was issued during CP -1, and provided no SAS backup exists during CP Ø, then the first one of the following requests is entered into SAS during CP Ø.

If present, a request to store a return jump exit word. (See RETURN JUMP)

If present in the I/O section, a request to transfer an I/O word.

If present in the IFA register, a request to read an instruction word to the IWS. (See INSTRUCTION FETCH)

See Examples IX, X, and XI.

INSTRUCTION FETCH

The transfer of instruction words from SCM to the IWS is initiated by the instruction stack. In order to maintain an adequate flow of instructions to the CIW the instruction stack tries to anticipate which words will be wanted next by the CIW. Accordingly, it generates requests ahead of time.

The instruction stack may generate and hold up to two requests for instruction words during any one clock period. If two requests are held concurrently, they are always for consecutive locations of SCM; the first request is held in the instruction fetch address (IFA) register of the instruction stack while the second request is held in reserve. Whenever the instruction stack holds a single request, this request is always held in the IFA.

THE FETCH OPERATION

Requests held in the IFA are processed as follows.

Clock Period	Action
Ø	The request enters the IFA register.
1+m	The request moves from the IFA to SAS.
2+m+n	The request moves from SAS to the SCM bank.
7+m+n	The instruction word enters rank 12 of IWS.

The value of m (m .GE. Ø) is determined by the SCM access control unit. The value of n (n .GE. Ø) is determined by the SCM destination control unit. (See THE STORAGE ADDRESS STACK (SAS).) A further request may enter the IFA register during clock period 1+m.

One or two requests for instruction words are generated by the instruction stack under various sets of conditions. These may be outlined as follows.

A jump out of stack generates normally two requests; one for the location of the jump and a second one for the next location in store.

A "look-ahead condition" may trigger the instruction stack to request one or two consecutive words from store. The condition arises normally once while the last instruction of an instruction word is being interpreted. It depends on the instruction which is interpreted and will therefore be stated separately for each group of instructions. In addition, the look-ahead condition arises whenever the empty CIW obtains an instruction word. (See REFILL OF AN EMPTY CIW, below.)

The foregoing two procedures may, in exceptional circumstances, fail to supply an instruction word which is needed for the execution of the program. The instruction stack detects this when the flow of instruction words from SCM ceases and then requests the word for which the CIW has been waiting together with the next location in store.

The precise rules which govern the generation of requests for instruction words are stated below. They involve five different sets of conditions, including three cases of look-ahead. No two sets of the conditions will ever be satisfied concurrently. (See NOTE ON LOOK-AHEAD, below)

JUMP OUT OF STACK

The instruction stack requests two instruction words in CP 0 provided

A jump instruction (010, 02, 03i, 04, ..., 07) issues in CP 0 which effects a jump.

The effective address of the jump is not held in the IAS in CP 0.

The IFA holds no request for an instruction word in CP 0.

The two requests are for the effective address of the jump and for the next address in store.

See Examples XII, XIV, XV, and XVI.

INSTRUCTION LOOK-AHEAD. CASE 1

The instruction stack requests two instruction words in CP 0 provided

The look-ahead condition holds in CP 0.

Rank 12 of the IWS holds the next instruction word of the program in CP 0.

All instruction words requested previously have arrived in the IWS before CP 0.

If the IAS rank 12 holds address s in CP 0 then the two requests made are for locations s+1 and s+2.

INSTRUCTION LOOK-AHEAD. CASE 2

The instruction stack requests one instruction word in CP 0 provided

The look-ahead condition holds in CP 0.

Rank 12 of the IWS holds the next instruction word of the program in CP 0.

A single instruction word requested previously has not yet arrived in the IWS before CP 0.

If the IAS rank 12 holds address s in CP 0 then the request made is for location s+2.

INSTRUCTION LOOK-AHEAD. CASE 3

The instruction stack requests one instruction word in CP 0 provided

The look-ahead condition holds in CP 0.

Rank 11 of the IWS holds the next instruction word of the program in CP 0.

All instruction words requested previously have arrived in the IWS before CP 0.

If the IAS rank 12 holds address s in CP 0 then the request made is for location s+1.

See Example XVII.

RECOVERY FROM DEFEAT

The instruction stack requests two instruction words in CP 0 provided

The CIW is empty in CP 0.

The next instruction word of the program is not held by the IWS in CP 0.

All instruction words requested previously have arrived in the IWS before CP 0.

The two requests are for the next instruction word of the program and for the following word in store.

See Examples XIII and XVIII.

NOTE ON LOOK-AHEAD

Cases 1 and 3 of instruction look-ahead will never arise concurrently. Whenever rank 11 and rank 12 of the IAS hold the same address the request for the word in rank 12 must have been the first of either a jump out of stack or a recovery from defeat. But in either of these cases two words are requested at once, so that while the first word is in rank 12 the second word will still be on its way to the instruction stack. Thus, whenever rank 11 and rank 12 of the IAS hold the same address neither case 1 nor case 3 of instruction look-ahead will be satisfied.

REFILL OF AN EMPTY CIW

The CIW becomes empty in CP 0 provided

An instruction word issues in CP 0.

The instruction stack does not, during CP 0, hold the word required next for program execution.

The next instruction word will then enter the CIW in CP n where n is determined depending on the case.

The look-ahead condition will hold in the clock period in which the next instruction word enters the CIW.

Case 1

If the instruction which issues in CP 0 effects a jump then n is the smallest positive integer which satisfies all of the following conditions.

The instruction stack holds the word in CP n.

No SAS backup condition exists in CP n.

All instruction words which were requested prior to CP 0 have arrived in the IWS before CP n.

n .GE. 2.

See Examples XIX, XX, XXI, and XXII.

Case 2

If the next instruction word is reached through linear code then n is the smallest positive integer which satisfies the following conditions.

The instruction stack holds the word in CP n.

No SAS backup condition exists in CP n.

See Examples XXIII and XXIV.

PASS INSTRUCTION

gh	COMPASS Code	Clock Period	XXX ijk	AAA ijk	BBB ijk
46	NO	0			

Instruction Look-ahead

The look-ahead condition holds in the clock period of instruction interpretation, say CP 0, provided

The instruction occupies bits 14 to 0 of the instruction word.

Instruction Issue

The instruction issues in the first clock period in which it is interpreted. No delays can occur.

Instruction Execution

Instruction execution is complete with instruction issue.

JUMP INSTRUCTIONS

RETURN JUMP

gh	COMPASS Code		Clock Period	XXX ijk	AAA ijk	BBB ijk
			0			
			1			
			2+m			
			3+m+n			
010	RJ	K	4+m+n			

Instruction Look-ahead

The look-ahead condition never holds during the interpretation of this instruction.

Instruction Issue

The instruction issues in CP 4+m+n provided

Instruction interpretation begins with CP 0.

All instruction words requested previously have arrived at the IWS before CP 2+m.

No SAS backup condition exists in CP 3+m+n.

Instruction Execution

A request to store the return jump exit word enters the SAS in CP 3+m+n of instruction interpretation.

Instruction execution is complete with instruction issue. (See, however, INSTRUCTION FETCH for the time taken to load the next instruction word into the CIW.)

See Example IX.

UNCONDITIONAL JUMP

gh	COMPASS Code		Clock Period	XXX ijk	AAA ijk	BBB ijk
			0			.
			1			
02	JP	Bi+K	2			

Instruction Look-ahead

The look-ahead condition holds during the clock period of instruction issue, say CP 2, provided

The effective address of the jump is held in the IAS in CP 2.

Instruction Issue

The instruction issues in CP 2 provided

Bi is free in CP 0.

Instruction Execution

Instruction execution is complete with instruction issue. (See, however, INSTRUCTION FETCH for the time taken to load the next instruction word into the CIW.)

CONDITIONAL JUMPS TAKEN

gh	COMPASS Code		Clock Period	XXX ijk	AAA ijk	BBB ijk
			0	.		
			1			
030	ZR	Xj,K	2			
			0	.		
			1			
031	NZ	Xj,K	2			
			0	.		
			1			
032	PL	Xj,K	2			
			0	.		
			1			
033	MI	Xj,K	2			
			0	.		
			1			
034	IR	Xj,K	2			
			0	.		
			1			
035	OR	Xj,K	2			
			0	.		
			1			
036	DF	Xj,K	2			
			0	.		
			1			
037	ID	Xj,K	2			

			0	..
			1	
04	EQ	Bi,Bj,K	2	
			0	..
			1	
05	NE	Bi,Bj,K	2	
			0	..
			1	
06	GE	Bi,Bj,K	2	
			0	..
			1	
07	GT	Bi,Bj,K	2	

Instruction Look-ahead

The look-ahead condition holds during the clock period of instruction issue, say CP 2, provided

The address of the jump is held in the IAS in CP 2.

Instruction Issue

The instructions issue in CP 2 provided

Xj is free in CP 0. (h = 3)

Bi and Bj are free in CP 0. (h > 3)

Instruction Execution

Instruction execution is complete with instruction issue. (See, however, INSTRUCTION FETCH for the time taken to load the next instruction word into the CIW.)

CONDITIONAL JUMPS NOT TAKEN

gh	COMPASS Code		Clock Period	XXX ijk	AAA ijk	BBB ijk
			0	.		
030	ZR	Xj,K	1			
			0	.		
031	NZ	Xj,K	1			
			0	.		
032	PL	Xj,K	1			
			0	.		
033	MI	Xj,K	1			
			0	.		
034	IR	Xj,K	1			
			0	.		
035	OR	Xj,K	1			
			0	.		
036	DF	Xj,K	1			
			0	.		
037	ID	Xj,K	1			
			0			..
04	EQ	Bi,Bj,K	1			
			0			..
05	NE	Bi,Bj,K	1			
			0			..
06	GE	Bi,Bj,K	1			

07	GT	Bi,Bj,K	Ø 1	..

Instruction Look-ahead

The look-ahead condition holds during the clock period of instruction issue provided

The instruction occupies bits 29 to Ø of the instruction word.

Instruction Issue

The instructions issue in CP 1 provided

Xj is free in CP Ø. (h = 3)

Bi and Bj are free in CP Ø. (h > 3)

Instruction Execution

Instruction execution is complete with instruction issue.

POPULATION COUNT UNIT

gh	COMPASS Code		Clock Period	XXX AAA BBB ijk ijk ijk
47	CXi	Xk	Ø 1	. . P

Instruction Look-ahead

The look-ahead condition holds in any clock period of instruction interpretation, say CP Ø, provided

The instruction occupies bits 14 to Ø of the instruction word.

The X-register destination path has not been reserved for CP 1 by a previous instruction.

No SAS backup condition exists during CP Ø.

Instruction Issue

The instruction issues in CP Ø provided

Xi and Xk are free during CP Ø.

The X-register destination path has not been reserved for CP 1 by a previous instruction.

No SAS backup condition exists during CP Ø.

Instruction Execution

Following instruction issue in CP Ø

Xi will be reserved during CP 1.

The X-register destination path will be reserved for CP 1.

LONG ADD UNIT

gh	COMPASS Code		Clock Period	XXX AAA BBB ijk ijk ijk
36	IXi	Xj+Xk	0	...
			1	L
37	IXi	Xj-Xk	0	...
			1	L

Instruction Look-ahead

The look-ahead condition holds in any clock period of instruction interpretation, say CP 0, provided

The instruction occupies bits 14 to 0 of the instruction word.

The X-register destination path has not been reserved for CP 1 by a previous instruction.

No SAS backup condition exists during CP 0.

Instruction Issue

The instruction issues in CP 0 provided

Xi, Xj, and Xk are free during CP 0.

The X-register destination path has not been reserved for CP 1 by a previous instruction.

No SAS backup condition exists during CP 0.

Instruction Execution

Following instruction issue in CP 0

Xi will be reserved during CP 1.

The X-register destination path will be reserved for CP 1.

BOOLEAN UNIT

gh	COMPASS Code		Clock Period	XXX AAA BBB ijk ijk ijk
10	BXi	Xj	0	..
			1	B
11	BXi	Xj*Xk	0	...
			1	B
12	BXi	Xj+Xk	0	...
			1	B
13	BXi	Xj-Xk	0	...
			1	B
14	BXi	-Xk	0	. .
			1	B
15	BXi	-Xk*Xj	0	...
			1	B

16	BXi	-Xk+Xj	0 1	... B	
17	BXi	-Xk-Xj	0 1	... B	
26	UXi	Bj,Xk	0 1	. . B	. (j>0)
26	UXi	Xk	0 1	. . B	(j=0)
27	PXi	Bj,Xk	0 1	. . B	.

Instruction Look-ahead

The look-ahead condition holds in any clock period o instruction interpretation, say CP 0, provided

The instruction occupies bits 14 to 0 of the instruction word.

The X-register destination path has not been reserved for CP 1 by a previous instruction.

The B-register destination path has not been reserved for CP 1 by a previous instruction (gh = 26).

No SAS backup condition exists during CP 0.

Instruction Issue

The instructions issue in CP 0 provided

Xi and Xj are free during CP 0. (gh = 10)

Xi, Xj, and Xk are free during CP 0. (gh = 11, 12, 13, 15, 16, 17)

Xi and Xk are free during CP 0. (gh = 14, 26, 27)

Bj is free during CP 0. (gh = 26, 27)

The X-register destination path has not been reserved for CP 1 by a previous instruction.

The B-register destination path has not been reserved for CP 1 by a previous instruction (gh = 26).

No SAS backup condition exists during CP 0.

Instruction Execution

Following instruction issue in CP 0

Xi will be reserved during CP 1.

Bj will be reserved during CP 1. (gh = 26, j > 0)

The X-register destination path will be reserved for CP 1.

The B-register destination path will be reserved for CP 1. (gh = 26)

INCREMENT UNIT

gh	COMPASS Code		Clock Period	XXX ijk	AAA ijk	BBB ijk	
50	SAi	Aj+K	0	.	..		
			1	F	I		
			2	F			
			7+m	F			(m .GE. 0)
51	SAi	Bj+K	0	.	.	.	
			1	F	I		
			2	F			
			7+m	F			(m .GE. 0)
52	SAi	Xj+K	0	..	.		
			1	F	I		
			2	F			
			7+m	F			(m .GE. 0)
53	SAi	Xj+Bk	0	..	.	.	
			1	F	I		
			2	F			
			7+m	F			(m .GE. 0)
54	SAi	Aj+Bk	0	.	..	.	
			1	F	I		
			2	F			
			7+m	F			(m .GE. 0)
55	SAi	Aj-Bk	0	.	..	.	
			1	F	I		
			2	F			
			7+m	F			(m .GE. 0)
56	SAi	Bj+Bk	0	.	.	..	
			1	F	I		
			2	F			
			7+m	F			(m .GE. 0)
57	SAi	Bj-Bk	0	.	.	..	
			1	F	I		
			2	F			
			7+m	F			(m .GE. 0)
60	SBi	Aj+K	0		.	.	
			1			I	
61	SBi	Bj+K	0			..	
			1			I	

62	SBi	Xj+K	Ø		.		.	
			1				I	
63	SBi	Xj+Bk	Ø		.		. .	
			1				I	
64	SBi	Aj+Bk	Ø			.	. .	
			1				I	
65	SBi	Aj-Bk	Ø			.	. .	
			1				I	
66	SBi	Bj+Bk	Ø				...	
			1				I	
67	SBi	Bj-Bk	Ø				...	
			1				I	
70	SXi	Aj+K	Ø		.	.		
			1	I				
71	SXi	Bj+K	Ø		.		.	
			1	I				
72	SXi	Xj+K	Ø		..			
			1	I				
73	SXi	Xj+Bk	Ø		..		.	
			1	I				
74	SXi	Aj+Bk	Ø		.	.	.	
			1	I				
75	SXi	Aj-Bk	Ø		.	.	.	
			1	I				
76	SXi	Bj+Bk	Ø		.		. .	
			1	I				
77	SXi	Bj-Bk	Ø		.		. .	
			1	I				

Instruction Look-ahead

The look-ahead condition holds in any clock period of instruction interpretation, say CP Ø, provided

The instruction occupies bits 29 to Ø of the instruction word. (h = Ø, 1, 2)

The instruction occupies bits 14 to Ø of the instruction word. (h > 2)

The B-register destination path has not been reserved for CP 1 by a previous instruction. (g = 6)

The X-register destination path has not been reserved for CP 1 by a previous instruction. (g = 7)

No SAS backup condition exists during CP Ø.

Instruction Issue

The instructions issue in CP Ø provided

Ai is free during CP Ø. (g = 5)

Bi is free during CP Ø. (g = 6)

Xi is free during CP Ø. (g = 5, i > Ø)

Xi is free during CP Ø. (g = 7)

Aj is free during CP Ø. (h = Ø, 4, 5)

Bj is free during CP Ø. (h = 1, 6, 7)

Xj is free during CP Ø. (h = 2, 3)

Bk is free during CP Ø. (h > 2)

The B-register destination path has not been reserved for CP 1 by a previous instruction. (g = 6)

The X-register destination path has not been reserved for CP 1 by a previous instruction. (g = 7)

No SAS backup condition exists during CP Ø.

Instruction Execution

Following instruction issue in CP Ø

Ai will be reserved during CP 1. (g = 5)

Bi will be reserved during CP 1. (g = 6, i > Ø)

Xi will be reserved during CP 1. (g = 7)

Xi will be reserved from CP 1 onwards for 7+m clock periods where m .GE. Ø is determined by the SCM destination control unit. (g = 5, h = 1, 2, 3..,5)

The B-register destination path will be reserved for CP 1. (g = 6)

The X-register destination path will be reserved for CP 1. (g = 7)

The SCM destination control unit will reserve the X-register destination path in CP 2+m for CP 7+m. (g = 5, h = 1, 2, ...,5)

SHIFT UNIT

gh	COMPASS Code	Clock Period	XXX ijk	AAA ijk	BBB ijk
20	LXi jk	0	.		
		1	S		
21	AXi jk	0	.		
		1	S		
22	LXi Bj,Xk	0	.	.	.
		1	S		
23	AXi Bj,Xk	0	.	.	.
		1	S		
43	MXi jk	0	.		
		1	S		

Instruction Look-ahead

The look-ahead condition holds in any clock period of instruction interpretation, say CP 0, provided

The instruction occupies bits 14 to 0 of the instruction word.

The X-register destination path has not been reserved for CP 1 by a previous instruction.

The B-register destination path has not been reserved for CP 1 by a previous instruction. (gh = 22, 23)

No SAS backup condition exists during CP 0.

Instruction Issue

The instructions issue during CP 0 provided

Xi is free during CP 0.

Bj and Xk are free during CP 0. (gh = 22, 23)

The X-register destination path has not been reserved for CP 1 by a previous instruction.

The B-register destination path has not been reserved for CP 1 by a previous instruction. (gh = 22, 23)

No SAS backup condition exists during CP 0.

Instruction Execution

Following instruction issue in CP 0

Xi will be reserved during CP 1.

The X-register destination path will be reserved for CP 1. for CP 1. (gh = 22, 23)

NORMALIZE UNIT

gh	COMPASS Code		Clock Period	XXX ijk	AAA ijk	BBB ijk	
24	NXi	Bj,Xk	0	. .		.	(j > 0)
			1	N		N	
			2	N		N	
24	NXi	Xk	0	. .			
			1	N			
			2	N			
25	ZXi	Bj,Xk	0	. .		.	(j > 0)
			1	N		N	
			2	N		N	
25	ZXi	Xk	0	. .			
			1	N			
			2	N			

Instruction Look-ahead

The look-ahead condition holds in any clock period of instruction interpretation, say CP 0, provided

The instruction occupies bits 14 to 0 of the instruction word.

The X-register destination path has not been reserved for CP 2 by a previous instruction.

The B-register destination path has not been reserved for CP 2 by a previous instruction.

No SAS backup condition exists during CP 0.

Instruction Issue

The instructions issue in CP 0 provided

Xi, Bj, and Xk are free during CP 0.

The X-register destination path has not been reserved for CP 2 by a previous instruction.

The B-register destination path has not been reserved for CP 2 by a previous instruction.

No SAS backup condition exists during CP 0.

Instruction Execution

Following instruction issue in CP 0

Xi will be reserved during CP 1 and CP 2.

Bj (j > 0) will be reserved during CP 1 and CP 2.

The X-register destination path will be reserved during CP 2.

The B-register destination path will be reserved during CP 2.

FLOATING ADD UNIT

gh	COMPASS Code		Clock Period	XXX AAA BBB ijk ijk ijk
30	FXi	Xj+Xk	0	...
			1	A
			2	A
			3	A
31	FXi	Xj-Xk	0	...
			1	A
			2	A
			3	A
32	DXi	Xj+Xk	0	...
			1	A
			2	A
			3	A
33	DXi	Xj-Xk	0	...
			1	A
			2	A
			3	A
34	RXi	Xj+Xk	0	...
			1	A
			2	A
			3	A
35	RXi	Xj-Xk	0	...
			1	A
			2	A
			3	A

Instruction Look-ahead

The look-ahead condition holds in any clock period of instruction interpretation, say CP 0, provided

The instruction occupies bits 14 to 0 of the instruction word.

The X-register destination path has not been reserved for CP 3 by a previous instruction.

No SAS backup condition exists during CP 0.

Instruction Issue

The instructions issue in CP 0 provided

Xi, Xj, and Xk are free during CP 0.

The X-register destination path has not been reserved for CP 3 by a previous instruction.

No SAS backup condition exists during CP 0.

Instruction Execution

Following instruction issue in CP 0

Xi will be reserved during CP 1, CP 2, and CP 3.

The X-register destination path will be reserved during CP 3.

FLOATING MULTIPLY UNIT

gh	COMPASS Code		Clock Period	XXX AAA BBB ijk ijk ijk
40	FXi	Xj*Xk	0	...
			1	M
			2	M
			3	M
			4	M
41	RXi	Xj*Xk	0	...
			1	M
			2	M
			3	M
			4	M
42	DXi	Xj*Xk	0	...
			1	M
			2	M
			3	M
			4	M

Instruction Look-ahead

The look-ahead condition holds in any clock period of instruction interpretation, say CP 0, provided

The instruction occupies bits 14 to 0 of the instruction word.

The multiply unit is free during CP 0.

The X-register destination path has not been reserved for CP 4 by a previous instruction.

No SAS backup condition exists during CP 0.

Instruction Issue

The instructions issue in CP 0 provided

Xi, Xj, and Xk are free during CP 0.

The multiply unit is free during CP 0.

The X-register destination path has not been reserved for CP 4 by a previous instruction.

No SAS backup condition exists during CP 0.

Instruction Execution

Following instruction issue in CP 0

Xi will be reserved during CP 1 to CP 4.

The multiply unit will be reserved during CP 1.

The X-register destination path will be reserved during CP 4.

DIVIDE UNIT

gh	COMPASS Code		Clock Period	XXX AAA BBB ijk ijk ijk
44	FXi	Xj/Xk	0	...
			1	D
			2	D
			3	D
			4	D
			5	D
			6	D
			7	D
			8	D
			9	D
			10	D
			11	D
			12	D
			13	D
			14	D
			15	D
			16	D
			17	D
			18	D
			19	D
45	RXi	Xj/Xk	0	...
			1	D
			2	D
			3	D
			4	D
			5	D
			6	D
			7	D
			8	D
			9	D
			10	D
			11	D
			12	D
			13	D
			14	D
			15	D
			16	D
			17	D
			18	D
			19	D

Instruction Look-ahead

The look-ahead condition holds in any clock period of instruction interpretation, say CP 0, provided

The instruction occupies bits 14 to 0 of the instruction word.

The divide unit is free during CP 0.

No SAS backup condition exists during CP 0.

Instruction Issue

The instructions issue in CP 0 provided

Xi, Xj, and Xk are free during CP 0.

The divide unit is free during CP 0.

No SAS backup condition exists during CP 0.

Instruction Execution

Following instruction issue in CP 0

Xi will be reserved during CP 1 to CP 19.

The divide unit will be reserved during CP 1 to CP 17.

The SCM destination control unit will be blocked during CP 14.

The X-register destination path will be reserved for CP 19.

EXAMPLES

The following examples display various features of 7600 timing, moving from the simple to the more intricate situations. A uniform construction has been adopted for all examples. The coding sequence to be timed, say EXAMPLE ANY, is always imbedded in the COMPASS routine *TEST* as follows.

```
*            IDENT     TEST
*            ENTRY     TEST,BEGIN,END
*   *
*    EXAMPLE MACRO     P
*            IB7       B0                  START INSTRUCTION TIMING
*            ENDM
*   *
*    RETURN  MACRO
*            IB6       B0                  STOP INSTRUCTION TIMING
*            JP        END
*            ENDM
*   *
*    TEST    DATA      0                   ENTRY/EXIT LINE
*            SX7       A0                  SAVE A0
*            SA7       SAVEA0
```

A few examples depend on the contents of the registers B1, X1, and X2. TEST therefore loads the integer 1 into B1 and the floating point number 1 into X1 and X2 before it enters the example.

```
*            SA1       CONST
*            SB1       1                   B1  =  1
*            BX2       X1                  X2  =  1.
*            JP        BEGIN
*   *
*    END     BSS       0
*   *
*            SA1       SAVEA0
*            SA0       X1
*            JP        TEST
*   *
*    SAVEA0  BSS       1
*    CONST   DATA      1.
*   *
*    BEGIN   BSS       0
*            EXAMPLE   ANY
```

The coding sequence to be timed follows here; it terminates with the macro call

```
*            RETURN
*            END
```

Thus, the coding sequence which will be timed begins with the instruction

```
*            IB7       B0
```

The instruction reads the current contents of the 7600 clock to B7. The sequence finishes with the instruction

```
*            IB6       B0
```

When the sequence is executed by the 7600, the difference B6 - B7 gives the number of clock periods which elapse between issue of the two clock-reading instructions. The total execution time of the sequence can therefore be determined directly by executing it in non-interruptible mode.

Each example has also been analyzed by the program TIME76, using the following main program

```
*       PROGRAM MAIN(TRACK, OUTPUT)
*           INTEGER   START, STOP
*           EXTERNAL  BEGIN, END
*   C
*           START     =  LOCF(BEGIN)
*           STOP      =  LOCF(END)
*           CALL TRACE76(START, STOP, 5LTRACK)
*           CALL TEST
*           STOP
*   C
*       END
```

TRACE76 is a program which interprets instruction execution, beginning at location BEGIN (i.e. at address START) and ending at location END (i.e. at address STOP). Trace data required for program TIME76 is written to file TRACK. (See program L 310 of the CERN Program Library for details on the calling sequences of TRACE76 and TIME76.) TIME76 arbitrarily assumes that the first instruction word to be timed is reached via a JP instruction. Routine TEST has been arranged accordingly.

I. BASIC SETUP

Summary: 1. The first instruction always reads the CPU clock to B7. B7 is not modified subsequently.

2. The last instruction always reads the CPU clock to B6.

3. With no other instruction in between, the two instructions issue in successive clock periods. Thus, the execution time of the example, B6-B7, is one clock period.

```
*           EXAMPLE   I                  1 CLOCK PERIOD EXECUTION TIME
*           RETURN
```

Notes

See Figure I.

IAS(12)	ADDRESS	COMPASS		MNEMONIC		XXXX 0123	XXXX 4567	AAAA 0123	AAAA 4567	BBB 123	BBBB 4567	CLOCK	IFA		SAS A	SAS B	SAS C	01234567	SMALL CORE MEMORY 01234567	01234567	01234567
111	110	I07	00			....	....	....	....	...	....	0	12						01		
		I36	00			....	....	....	....	...	...S	1			12				01		
						....	....	....	....	...	..SS	2							012		
						....	....	....	....	...	..S.	3							12		
		JP			104	....	....	....	....	...	....	4							2		
						....	....	....	....	...	....	5	04						2		
						....	....	....	....	...	....	6	05	04					2		
112						....	....	....	....	...	....	7		05				4	2		
						....	....	....	....	...	....	8						45	2		
						....	....	....	....	...	....	9						45	2		
						....	....	....	....	...	....	10						45	2		
						....	....	....	....	...	....	11						45			
104						....	....	....	....	...	....	12						45			

Figure I

II. PASS INSTRUCTIONS

Summary: 1. Pass instructions issue during the first clock period of their interpretation.

```
*           EXAMPLE   II                5 CLOCK PERIODS EXECUTION TIME

*           NO
*           NO
*           NO
*           NO
*           RETURN
```

Notes

See Figure II.

IAS(12)	ADDRESS	COMPASS	MNEMONIC	XXXX 0123	XXXX 4567	AAAA 0123	AAAA 4567	BBB 123	BBBB 4567	CLOCK	IFA	SAS A	SAS B	SAS C	01234567	SMALL CORE MEMORY 01234567	01234567	01234567
111	110	IB7 B0		....	....	....	....	...	....	0	12					01		
		NO		....	....	....	....	...	...S	1		12				01		
		NO		....	....	....	....	...	...S	2						012		
		NO		....	....	....	....	...	....	3						12		
	111	NO		....	....	....	....	...	....	4	13					2		
		IB6 B0		....	....	....	....	...	....	5		13				2		
				....	....	....	....	...	..S.	6						23		
112				....	....	....	....	...	..S.	7						23		
		JP	104	....	....	....	....	...	....	8						23		
				....	....	....	....	...	....	9	04					23		
				....	....	....	....	...	....	10	05	04				23		
113				....	....	....	....	...	....	11		05				3		
				....	....	....	....	...	....	12				4	45	3		
				....	....	....	....	...	....	13				45		3		
				....	....	....	....	...	....	14				45		3		
				....	....	....	....	...	....	15				45				
104				....	....	....	....	...	....	16				45				

Figure II

III. X-REGISTER RESERVATION

Summary: 1. Instruction issue is delayed due to reservation of Xi, Xj, or Xk.

```
*           EXAMPLE   III                16 CLOCK PERIODS EXECUTION TIME

*           IX3       X1+X2
*           NX4       B3,X3
*           FX6       X1+X4
*           FX5       X6*X1
*           BX5       X1+X2
*           RETURN
```

Notes

See Figure III.

CP 2 Issue of the NX4 instruction is delayed for one clock period because X3 has been reserved by the preceding instruction.

CP 4 Issue of the FX6 instruction is delayed for two clock periods because X4 has been reserved by the preceding instruction.

CP 7 Issue of the FX5 instruction is delayed for three clock periods because X6 has been reserved by the preceding instruction.

CP 11 Issue of the BX5 instruction is delayed for four clock periods because X5 has been reserved by the preceding instruction.

IAS(12)	ADDRESS	COMPASS	MNEMONIC		XXXX 0123	XXXX 4567	AAAA 0123	AAAA 4567	BBB 123	BBBB 4567	CLOCK	IFA	SAS A	SAS B	SAS C	01234567	SMALL CORE MEMORY 01234567	01234567	01234567
111	110	IB7	B0		....	....	....	....	...	....	0	12					01		
		IX3	X1+X2		....	....	....	....	...	...S	1		12				01		
					...L	....	....	....	...	...S	2						012		
		NX4	B3,X3		....	....	....	....	...	....	3						12		
					....	N...	....	....	..N	....	4						2		
					....	N...	....	....	..N	....	5	13					2		
		FX6	X1*X4		....	....	....	....	...	....	6		13				2		
112					....	..A.	....	....	...	....	7						23		
					....	..A.	....	....	...	....	8						23		
					....	..A.	....	....	...	....	9						23		
	111	FX5	X6*X1		....	....	....	....	...	....	10						23		
					....	.M..	....	....	...	....	11						3		
113					....	.M..	....	....	...	....	12						3		
					....	.M..	....	....	...	....	13						3		
					....	.M..	....	....	...	....	14						3		
		BX5	X1+X2		....	....	....	....	...	....	15						3		
		IB6	B0		....	.B..	....	....	...	....	16								
		NO			....	....	....	....	...	..S.	17								
					....	....	....	....	...	..S.	18	14							
					....	....	....	....	...	....	19		14						
	112	JP		104	....	....	....	....	...	....	20						4		
					....	....	....	....	...	....	21	04					4		
					....	....	....	....	...	....	22	05	04				4		
					....	....	....	....	...	....	23		05			4	4		
					....	....	....	....	...	....	24					45	4		
114					....	....	....	....	...	....	25					45	4		
					....	....	....	....	...	....	26					45	4		
					....	....	....	....	...	....	27					45	4		
104					....	....	....	....	...	....	28					45	4		

Figure III

IV. B-REGISTER RESERVATION

Summary: 1. Instruction issue is delayed due to reservation of Bi, Bj, or Bk.

2. B0 is never reserved.

```
*           EXAMPLE   IV                17 CLOCK PERIODS EXECUTION TIME

*           NO
*           SB2       B1
*           SB4       B1+B2
*           SB5       B4
*           UX5       B5,X1
*           NX0       B5,X0
*           SB5       B0
*           UX6       B0,X1
*           SB0       0
*           NX7       B0,X0
*           RETURN
```

Notes

See Figure IV.

CP 3 Issue of the SB4 instruction is delayed for one clock period because B2 has been reserved by the preceding instruction.

CP 5 Issue of the SB5 instruction is delayed for one clock period because B4 has been reserved by the preceding instruction.

CP 7 Issue of the UX5 instruction is delayed for one clock period because B5 has been reserved by the preceding instruction.

CP 9 Issue of the NX0 instruction is delayed for one clock period because B5 has been reserved by the preceding instruction.

CP 11 Issue of the SB5 instruction is delayed for two clock periods because B5 has been reserved by the preceding instruction.

CP 15 The SB0 instruction issues because B0 is not reserved by the preceding instruction.

CP 16 The NX7 instruction issues because B0 is not reserved by the preceding instruction.

CP 17 The IB6 instruction issues because B0 is not reserved by the preceding instruction.

IAS(12)	ADDRESS	COMPASS MNEMONIC		XXXX 0123	XXXX 4567	AAAA 0123	AAAA 4567	BBB 123	BBBB 4567	CLOCK	IFA	SAS A	SAS B	SAS C	SMALL CORE MEMORY 01234567	01234567	01234567	01234567
111	110	IB7	B0	....	....	....	....	...	....	0	12					01		
		NO		....	....	....	....	...	...S	1		12				01		
		SB2	B1	....	....	....	....	...	...S	2						012		
				....	....	....	....	.I.	....	3						12		
		SB4	B1+B2	....	....	....	....	...	....	4	13					2		
				....	....	....	....	...	I...	5		13				2		
	111	SB5	B4	....	....	....	....	...	....	6						23		
112				....	....	....	....	...	.I..	7						23		
		UX5	B5,X1	....	....	....	....	...	....	8						23		
				....	.B..	....	....	...	.B..	9						23		
		NX0	B5,X0	....	....	....	....	...	....	10						23		
113				N...	....	....	....	...	.N..	11						3		
				N...	....	....	....	...	.N..	12						3		
		SB5	B0	....	....	....	....	...	....	13	14					3		
	112	UX6	B0,X1	....	....	....	....	...	.I..	14		14				3		
		SB0		0	..B.	....	....	...	....	15						4		
		NX7	B0,X0	....	....	....	....	...	....	16						4		
	113	IB6	B0	....	...N	....	....	...	....	17	15					4		
				....	...N	....	....	...	..S.	18		15				4		
				....	....	....	....	...	..S.	19						45		
114		JP	104	....	....	....	....	...	....	20						45		
				....	....	....	....	...	....	21	04					45		
				....	....	....	....	...	....	22	05	04				45		
				....	....	....	....	...	....	23		05			4	45		
115				....	....	....	....	...	....	24					45	5		
				....	....	....	....	...	....	25					45	5		
				....	....	....	....	...	....	26					45	5		
				....	....	....	....	...	....	27					45	5		
104				....	....	....	....	...	....	28					45			

Figure IV

V. A-REGISTER RESERVATION

Summary: 1. Instruction issue is delayed due to reservation of Ai or Aj.

```
*            EXAMPLE   V                    19 CLOCK PERIODS EXECUTION TIME

*            SA0       B5
*            SA0       B6
*            NO
*            SA6       STORE
*            SA6       STORE+1
*            SA7       A6+1
*            FX3       X1*X1
*            SB3       X3
*            SA0       1
*            SA1       A0
*            RETURN
*     GAP    BSS       2
*     STORE  BSS       3
```

Notes

See Figure V.

CP 2 Issue of the instruction SA0 B6 is delayed for one clock period because A0 has been reserved by the preceding instruction.

CP 6 Issue of the SA6 instruction is delayed for one clock period because A6 has been reserved by the preceding instruction.

CP 8 Issue of the SA7 instruction is delayed for one clock period because A6 has been reserved by the preceding instruction.

CP 17 Issue of the SA1 instruction is delayed for one clock period because A0 has been reserved by the preceding instruction.

IAS(12)	ADDRESS	COMPASS	MNEMONIC		XXXX 0123	XXXX 4567	AAAA 0123	AAAA 4567	BBB 123	BBBB 4567	CLOCK	IFA	A	SAS B	C	01234567	SMALL CORE MEMORY 01234567	01234567	01234567
111	110	ID7	B0		....	....	....	....	...	....	0	12					01		
		SA0	B5		....	....	....	....	...	...S	1		12				01		
					....	....	I...	....	...	...S	2						012		
		SA0	B6		....	....	....	....	...	....	3						12		
		NO			....	....	I...	....	...	....	4						2		
	111	SA6		117	....	....	....	....	...	....	5	13					2		
					....	....	....	..I.	...	....	6		13				2		
112		SA6		120	....	....	....	....	...	....	7		17				23		
					....	....	....	..I.	...	....	8	14					23	7	
	112	SA7	A6+	1	....	....	....	....	...	....	9	14	20				23	7	
		FX3	X1*X1		....	....	....	...I	...	....	10		14				23	7	0
					...M	....	....	....	...	....	11		21				34	7	0
113					...M	....	....	....	...	....	12						34	7	01
					...M	....	....	....	...	....	13	15					34	7	01
					...M	....	....	....	...	....	14		15				34	7	01
		SB3	X3		....	....	....	....	...	....	15						345	7	01
114	113	SA0		1	....	....	....	....	..I	....	16						45	7	01
					....	....	I...	....	...	....	17						45		01
		SA1	A0		....	....	....	....	...	....	18						45		01
		IB6	B0		.F..	....	.I..	....	...	....	19						45		1
115					.F..	....	....	....	...	..S.	20	16	01				5		1
					.F..	....	....	....	...	..S.	21		16		1		5		
	114	JP		104	.F..	....	....	....	...	....	22				1		56		
					.F..	....	....	....	...	....	23	04			1		56		
					.F..	....	....	....	...	....	24	05	04		1		6		
					.F..	....	....	....	...	....	25		05		1	4	6		
					....	....	....	....	...	....	26				1	45	6		
116					....	....	....	....	...	....	27				1	45	6		
					....	....	....	....	...	....	28				1	45	6		
					....	....	....	....	...	....	29				1	45	6		
104					....	....	....	....	...	....	30					45	6		

Figure V

VI. Xd CONFLICTS

Summary: 1. Xd conflict between multiply and floating add instructions.

2. Xd conflict between floating add and normalize instructions.

3. Xd conflict between normalize and Boolean instructions.

4. Xd conflict between divide instruction and access to SCM.

5. Xd conflict between operand fetch and multiply instructions.

```
*           EXAMPLE   VI                  24 CLOCK PERIODS EXECUTION TIME

*           FX2       X1/X1
*           NO
*           NX0       B0,X1
*           FX4       X0*X0
*           FX5       X0+X0
*           NX6       B0,X0
*           BX7       X0
*           SA3       *+4
*           SB3       A3
*           FX6       X1*X1
*           BX3       -X6
*           RETURN
```

Notes

See Figure VI.

CP 7 Issue of the FX5 instruction is delayed for one clock period because an Xd conflict arises with the preceding instruction.

CP 9 Issue of the NX6 instruction is delayed for one clock period because an Xd conflict arises with the preceding instruction.

CP 11 Issue of the BX7 instruction is delayed for one clock period because an Xd conflict arises with the preceding instruction.

CP 15 The memory reference to address 116 which is made by the SA3 instruction is delayed in the SAS for one clock period because an Xd conflict arises with the divide instruction.

CP 16 Issue of the FX6 instruction is delayed for one clock period because an Xd conflict arises with the divide instruction.

CP 17 Issue of the FX6 instruction is delayed for a further clock period because an Xd conflict arises with the SA3 instruction.

IAS(12)	ADDRESS	COMPASS MNEMONIC		XXXX 0123	XXXX 4567	AAAA 0123	AAAA 4567	B9B 123	B9B 4567	CLOCK	IFA	SAS A	SAS B	SAS C	01234567	SMALL CORE MEMORY 01234567	01234567	01234567
111	110	IB7 B0		....	....	....	....	...	....	0	12					01		
		FX2 X1/X1		....	....	....	....	...	...S	1		12				01		
		NO		..D.	....	....	....	...	...S	2						012		
		NX0 B0,X1		..D.	....	....	....	...	....	3						12		
				N.D.	....	....	....	...	....	4	13					2		
				N.D.	....	....	....	...	....	5		13				2		
	111	FX4 X0*X0		..D.	....	....	....	...	....	6						23		
112				..D.	M...	....	....	...	....	7						23		
		FX5 X0+X0		..D.	M...	....	....	...	....	8						23		
				..D.	MA..	....	....	...	....	9						23		
		NX6 B0,X0		..D.	MA..	....	....	...	....	10						23		
113				..D.	.AN.	....	....	...	....	11						3		
		BX7 X0		..D.	..N.	....	....	...	....	12						3		
	112	SA3	116	..D.	...B	....	....	...	....	13	14					3		
				..DF	....	...I	....	...	....	14		14				3		
		SB3 A3		..DF	....	....	....	...	....	15		16				4		
				..DF	....	....	....	..I	....	16		16				4		
				..DF	....	....	....	...	....	17						4	6	
		FX6 X1*X1		..DF	....	....	....	...	....	18						4	6	
				..DF	..M.	....	....	...	....	19	15					4	6	
114				..DF	..M.	....	....	...	....	20		15				4	6	
				...F	..M.	....	....	...	....	21						456		
				....	..M.	....	....	...	....	22						456		
	113	BX3 -X6		....	....	....	....	...	....	23						456		
		IB6 B0		...B	....	....	....	...	....	24						56		
				....	....	....	....	...	..S.	25						56		
115				....	....	....	....	...	..S.	26						5		
		JP	104	....	....	....	....	...	....	27						5		
				....	....	....	....	...	....	28	04					5		
				....	....	....	....	...	....	29	05	04				5		
				....	....	....	....	...	....	30		05		4				
				....	....	....	....	...	....	31				45				
				....	....	....	....	...	....	32				45				
				....	....	....	....	...	....	33				45				
				....	....	....	....	...	....	34				45				
104				....	....	....	....	...	....	35				45				

Figure VI

VII. Bd CONFLICTS

Summary: 1. Bd conflict between normalize and increment B-register instructions.

2. No exception is made if the destination register is B0.

```
*           EXAMPLE   VII                10 CLOCK PERIODS EXECUTION TIME

*           NX6       B0,X1
*           SB3       B1
*           ZX7       B4,X1
*           SB2       B1
*           NX5       B5,X2
*           SB0       0
*           RETURN
```

Notes

See Figure VII.

CP 2 Issue of the SB3 instruction is delayed for one clock period because a Bd conflict arises with the preceding instruction.

CP 5 Issue of the SB2 instruction is delayed for one clock period because a Bd conflict arises with the preceding instruction.

CP 8 Issue of the SB0 instruction is delayed for one clock period because a Bd conflict arises with the preceding instruction.

IAS(12)	ADDRESS	COMPASS MNEMONIC		XXXX 0123	XXXX 4567	AAAA 0123	AAAA 4567	BBB 123	BBB 4567	CLOCK	IFA	SAS A	SAS B	SAS C	SMALL CORE MEMORY 01234567	SMALL CORE MEMORY 01234567	SMALL CORE MEMORY 01234567	SMALL CORE MEMORY 01234567
111	113	ID7	B0	....	....	....	....	...	....	0	12							01
		NX6	B0,X1	....	....	....	....	...	...S	1		12						01
				....	..N.	....	....	...	...S	2								012
		SB3	B1	....	..N.	....	....	...	....	3								12
		ZX7	B4,X1	....	....	....	....	..I	....	4								2
				....	...N	....	....	...	N...	5	13							2
	111	SB2	B1	....	...N	....	....	...	N...	6		13						2
112		NX5	B5,X2	....	....	....	....	.I.	....	7								23
				....	.N..	....	....	...	.N..	8								23
		SB0		....	.N..	....	....	...	.N..	9								23
	112	ID6	B0	....	....	....	....	...	....	10	14							23
				....	....	....	....	...	..S.	11		14						3
113				....	....	....	....	...	..S.	12								34
		JP	104	....	....	....	....	...	....	13								34
				....	....	....	....	...	....	14	04							34
				....	....	....	....	...	....	15	05	04						34
				....	....	....	....	...	....	16		05			4			4
114				....	....	....	....	...	....	17					45			4
				....	....	....	....	...	....	18					45			4
				....	....	....	....	...	....	19					45			4
				....	....	....	....	...	....	20					45			4
104				....	....	....	....	...	....	21					45			

Figure VII

VIII. SCM READ OPERATIONS

Summary: 1. Simple case of instruction fetch.

2. Simple case of operand fetch.

```
*            EXAMPLE   VIII              13 CLOCK PERIODS EXECUTION TIME

*            JP        ONE
*   GAP1     BSS       4
*   ONE      SA1       TWO
*            RETURN
*   GAP2     BSS       3
*   TWO      DATA      0
```

Notes

See Figure VIII.

CP 3 The jump address 115 enters the IFA.

CP 4 The jump address is available in the IFA and enters the SAS rank A from there.

CP 5 The jump address is available in the SAS rank A and enters the SCM bank address register from there. SCM bank 15 is busy reading the next instruction word during the following nine clock periods.

CP 10 The instruction word enters the instruction stack.

CP 11 The instruction word is available in rank 12 of the instruction stack and enters the CIW from there. The SA1 instruction issues one clock period later.

CP 13 The operand address 122 enters the SAS rank A.

CP 14 The operand address is available in the SAS rank A and enters the SCM bank address register from there. SCM bank 22 is busy reading the operand during the following nine clock periods.

CP 19 The operand enters X1 from the SCM read operand register and is available there one clock period later.

IAS(12)	ADDRESS	COMPASS MNEMONIC		XXXX 0123	XXXX 4567	AAAA 0123	AAAA 4567	BBB 123	BBBB 4567	CLOCK	IFA	SAS A	SAS B	SAS C	01234567	SMALL CORE MEMORY 01234567	01234567	01234567
111	110	IB7 B0		....	....	....	....	...	....	0	12					01		
				....	....	....	....	...	...S	1		12				01		
				....	....	....	....	...	...S	2						012		
		JP	115	....	....	....	....	...	....	3						12		
				....	....	....	....	...	....	4	15					2		
				....	....	....	....	...	....	5	16	15				2		
				....	....	....	....	...	....	6		16				2	5	
112				....	....	....	....	...	....	7						2	56	
				....	....	....	....	...	....	8						2	56	
				....	....	....	....	...	....	9						2	56	
				....	....	....	....	...	....	10						2	56	
115				....	....	....	....	...	....	11							56	
116	115	SA1	122	....	....	....	....	...	....	12	17						56	
		IB6 B0		.F..	....	.I..	....	...	....	13		17					56	
		NO		.F..	....	....	....	...	..S.	14		22					567	
				.F..	....	....	....	...	..S.	15	20						67	2
				.F..	....	....	....	...	....	16		20					7	2
	116	JP	104	.F..	....	....	....	...	....	17							7	0 2
				.F..	....	....	....	...	....	18	04						7	0 2
117				.F..	....	....	....	...	....	19	05	04					7	0 2
				....	....	....	....	...	....	20		05		4			7	0 2
				....	....	....	....	...	....	21				45			7	0 2
120				....	....	....	....	...	....	22				45			7	0 2
				....	....	....	....	...	....	23				45				0 2
				....	....	....	....	...	....	24				45				0
104				....	....	....	....	...	....	25				45				0

Figure VIII

IX. SCM READ AND WRITE OPERATIONS

Summary: 1. Storage of return jump exit word.

2. Instruction fetch.

3. Operand fetch.

4. Operand store.

```
*            EXAMPLE   IX                22 CLOCK PERIODS EXECUTION TIME

*            RJ        ONE
*  GAP1      BSS       3
*  ONE       DATA      Ø
*            SA1       TWO
*            NO
*            NO
*            SA6       TWO+1
*            RETURN
*  GAP2      BSS       3
*  TWO       BSS       2
```

Notes

See Figure IX.

CP 1 Execution of the RJ instruction begins.

CP 3 Execution of the RJ instruction is delayed for four clock periods to let instruction word 112 arrive in the instruction stack before the stack is cleared.

CP 8 Address 114 of the return jump exit word enters the SAS rank A.

CP 9 The address of the return jump exit word is available in the SAS and enters the SCM bank address register from there. SCM bank 14 is busy storing the return jump exit word during the following nine clock periods.

The jump address 115 enters the IFA.

CP 1Ø The jump address is available in the IFA and enters the SAS rank A from there.

CP 11 The jump address is available in the SAS rank A and enters the SCM bank address register from there. SCM bank 15 is busy reading the next instruction word during the following nine clock periods.

CP 16 The instruction word enters the instruction stack.

CP 17 The instruction word is available in rank 12 of the instruction stack and enters the CIW from there. The SA1 instruction issues one clock period later.

CP 22 The result address 124 enters the SAS rank A.

CP 23 The result address is available in the SAS rank A and enters the SCM bank address register from there. SCM bank 24 is busy storing X6 during the following nine clock periods.

IAS(12)	ADDRESS	COMPASS MNEMONIC		XXXX 0123	XXXX 4567	AAAA 0123	AAAA 4567	BBB 123	BBBB 4567	CLOCK	IFA	SAS A	SAS B	SAS C	SMALL CORE MEMORY 01234567	01234567	01234567	01234567
				....	....	....	....	...	....	0	12				01			
				....	....	....	....	...	...S	1		12			01			
				....	....	....	....	...	...S	2					012			
				....	....	....	....	...	....	3					12			
				....	....	....	....	...	....	4					2			
				....	....	....	....	...	....	5					2			
				....	....	....	....	...	....	6					2			
0				....	....	....	....	...	....	7					2			
				....	....	....	....	...	....	8					2			
		RJ	114	....	....	....	....	...	....	9		14			2			
				....	....	....	....	...	....	10	15				2	4		
				....	....	....	....	...	....	11	16	15				4		
				....	....	....	....	...	....	12		16				45		
				....	....	....	....	...	....	13						456		
				....	....	....	....	...	....	14						456		
				....	....	....	....	...	....	15						456		
				....	....	....	....	...	....	16						456		
115				....	....	....	....	...	....	17						456		
116	115	SA1	123	....	....	....	....	...	....	18	17					456		
		NO		.F..	....	.I..	....	...	....	19		17				56		
		NO		.F..	....	....	....	...	....	20		23				567		
	116	SA6	124	.F..	....	....	....	...	....	21	20					67	3	
		IB6 B0		.F..	....	....	..I.	...	....	22		20				7	3	
		NO		.F..	....	....	....	...	..S.	23		24				7	0 3	
				.F..	....	....	....	...	..S.	24						7	0 34	
117				.F..	....	....	....	...	....	25						7	0 34	
				....	....	....	....	...	....	26	21					7	0 34	
				....	....	....	....	...	....	27		21				7	0 34	
120	117	JP	104	....	....	....	....	...	....	28						7	01 34	
				....	....	....	....	...	....	29	04						01 34	
				....	....	....	....	...	....	30	05	04					01 4	
				....	....	....	....	...	....	31		05		4			01 4	
				....	....	....	....	...	....	32				45			1 4	
121				....	....	....	....	...	....	33				45			1	
				....	....	....	....	...	....	34				45			1	
				....	....	....	....	...	....	35				45			1	
104				....	....	....	....	...	....	36				45			1	

Figure IX

X. DIVIDE UNIT DELAYS OPERAND STORE INSTRUCTION

Summary: 1. Operand store is delayed in SAS by divide instruction.

2. A further operand store creates an SAS backup condition.

3. The request for an instruction word is thus delayed in the IFA.

4. The address of a jump out of stack cannot enter IFA and is lost.

```
*            EXAMPLE   X                    35 CLOCK PERIODS EXECUTION TIME

*            FX0       X1/X2
*            SB4       TWO
*            SB5       THREE
*            FX3       X1+X2
*            FX6       X2*X3
*            SA6       ONE
*            SA7       B4
*            SA6       B5
*            NO
*            ZR        FOUR
*     GAP    BSS       3
*     ONE    BSS       1
*     TWO    BSS       1
*     THREE  BSS       1
*     FOUR   RETURN
```

Notes

See Figure X.

CP 15 The memory reference to address 117 which is made by the first SA6 instruction is delayed in the SAS for one clock period because an Xd conflict arises with the divide instruction. The memory reference to address 120 (B4) therefore enters the SAS rank B.

CP 16 The reference to address 120 is delayed in the SAS rank B because the request in the SAS rank A must be delivered first.

The memory reference to address 121 (B5) therefore enters the SAS rank C.

The NO instruction issues despite the SAS backup condition.

The instruction word address 115 is delayed in the IFA for three clock periods due to the SAS backup condition.

CP 19 The jump address 122 is lost because the IFA still holds the request for word 115.

CP 25 The instruction word 115 enters the instruction stack.

CP 26 No more instruction words are on the way to the instruction stack. A request to fetch the missing word 122 enters the IFA.

CP 27 The request for word 122 is available in the IFA and enters the SAS rank A from there.

A request for word 123 enters the IFA.

CP 28 The request for word 122 is available in the SAS rank A and enters the SCM bank address register from there.

The request for word 123 is available in the IFA and enters the SAS rank A from there.

CP 33 The instruction word at address 122 enters the instruction stack.

CP 34 The instruction word from address 122 is available in rank 12 of the instruction stack and enters the CIW from there. The IB6 instruction issues one clock period later.

The instruction word at address 123 enters the instruction stack.

IAS(12)	ADDRESS	COMPASS	MNEMONIC		XXXX 0123	XXXX 4567	AAAA 0123	AAAA 4567	BBB 123	BBBB 4567	CLOCK	IFA	SAS A	SAS B	SAS C	01234567	SMALL CORE MEMORY 01234567	01234567	01234567
111	110	IB7	B0		....	....	....	....	...	....	0	12					01		
		FX0	X1/X2		....	....	....	....	...	...S	1		12				01		
		SB4		120	D...	....	....	....	...	...S	2						012		
	111	SB5		121	D...	....	....	....	...	I...	3	13					12		
		FX3	X1*X2		D...	....	....	....	...	.I..	4		13				2		
					D..A	....	....	....	...	....	5						23		
					D..A	....	....	....	...	....	6						23		
112					D..A	....	....	....	...	....	7						23		
		FX6	X2*X3		D...	....	....	....	...	....	8	14					23		
					D...	..M.	....	....	...	....	9		14				23		
113					D...	..M.	....	....	...	....	10						234		
					D...	..M.	....	....	...	....	11						34		
					D...	..M.	....	....	...	....	12						34		
	112	SA6		117	D...	....	....	....	...	....	13						34		
		SA7	B4		D...	....	....	..I.	...	....	14						4		
114		SA6	B5		D...	....	....	...I	...	....	15		17				4		
	113	NO			D...	....	....	..I.	...	....	16	15	17	20			4		
					D...	....	....	....	...	....	17	15		20	21		4	7	
					D...	....	....	....	...	....	18	15			21		4	7	0
		ZR	B0,	122	D...	....	....	....	...	....	19	15						7	01
					D...	....	....	....	...	....	20		15					7	01
					....	....	....	....	...	....	21							5 7	01
					....	....	....	....	...	....	22							5 7	01
					....	....	....	....	...	....	23							5 7	01
					....	....	....	....	...	....	24							5 7	01
					....	....	....	....	...	....	25							5 7	01
115					....	....	....	....	...	....	26							5	01
					....	....	....	....	...	....	27	22						5	1
					....	....	....	....	...	....	28	23	22					5	
					....	....	....	....	...	....	29		23					5	2
					....	....	....	....	...	....	30								23
					....	....	....	....	...	....	31								23
					....	....	....	....	...	....	32								23
					....	....	....	....	...	....	33								23
122					....	....	....	....	...	....	34								23
123	122	IB6	B0		....	....	....	....	...	....	35	24							23
					....	....	....	....	...	..S.	36		24						23
					....	....	....	....	...	..S.	37								234
		JP		104	....	....	....	....	...	....	38								34
					....	....	....	....	...	....	39	04							4
					....	....	....	....	...	....	40	05	04						4
					....	....	....	....	...	....	41		05			4			4
124					....	....	....	....	...	....	42					45			4
					....	....	....	....	...	....	43					45			4
					....	....	....	....	...	....	44					45			4
					....	....	....	....	...	....	45					45			4
104					....	....	....	....	...	....	46					45			

Figure X

XI. SAS BACKUP DELAYS INSTRUCTION ISSUE

Summary: 1. Instruction fetch and operand fetch cause an SCM bank conflict.

2. The SCM bank conflict leads to an SAS backup condition.

3. The SAS backup condition prevents instruction issue.

```
*           EXAMPLE   XI                14 CLOCK PERIODS EXECUTION TIME

*           SA3       *+102B
*           NO
*           SB3       3
*           SB4       4
*           RETURN
```

Notes

See Figure XI.

CP 0 The IFA initiates a fetch which occupies SCM bank 12 from CP 2 onwards.

CP 3 The operand fetch initiated by the SA3 instruction is delayed in SAS rank A since it requires access to SCM bank 12. The instruction fetch initiated by the look-ahead condition in CP 2 is therefore delivered to SAS rank B and thus causes an SAS backup condition from CP 4 onwards.

CP 4 Issue of the SB4 instruction is delayed until the SAS backup condition has been resolved in CP 13.

IAS(12)	ADDRESS	COMPASS MNEMONIC		XXXX 0123	XXXX 4567	AAAA 0123	AAAA 4567	BBB 123	BBBB 4567	CLOCK	IFA	SAS A	SAS B	SAS C	01234567	SMALL CORE MEMORY 01234567	01234567	01234567
111	110	IB7 B0		....	....	....	....	...	....	0	12					01		
		SA3	212	....	....	....	....	...	...S	1		12				01		
		NO		...F	....	...I	....	...	...S	2						012		
	111	SB3	3	...F	....	....	....	...	....	3	13	12				12		
				...F	....	....	....	..I	....	4		12	13			2		
				...F	....	....	....	...	....	5		12	13			2		
				...F	....	....	....	...	....	6		12	13			2		
112				...F	....	....	....	...	....	7		12	13			2		
				...F	....	....	....	...	....	8		12	13			2		
				...F	....	....	....	...	....	9		12	13			2		
				...F	....	....	....	...	....	10		12	13			2		
				...F	....	....	....	...	....	11		12	13					
				...F	....	....	....	...	....	12			13			2		
		SB4	4	...F	....	....	....	...	....	13						23		
	112	IB6 B0		...F	....	....	....	...	I...	14	14					23		
				...F	....	....	....	...	..S.	15		14				23		
				...F	....	....	....	...	..S.	16						234		
		JP	104	....	....	....	....	...	....	17						234		
113				....	....	....	....	...	....	18	04					234		
				....	....	....	....	...	....	19	05	04				234		
				....	....	....	....	...	....	20		05			4	234		
114				....	....	....	....	...	....	21					45	34		
				....	....	....	....	...	....	22					45	4		
				....	....	....	....	...	....	23					45	4		
				....	....	....	....	...	....	24					45	4		
184				....	....	....	....	...	....	25					45			

Figure XI

XII. JUMP OUT OF STACK

Summary: 1. Simple case of jump out of stack.

```
*            EXAMPLE   XII                12 CLOCK PERIODS EXECUTION TIME

*            JP        NEXT
*   GAP      BSS       2
*   NEXT     RETURN
```

Notes

See Figure XII.

CP 3 A jump out of stack is performed since location 113 is not currently available from the instruction stack. A request to fetch location 113 is generated.

CP 4 The request for location 113 is held in IFA and enters the SAS rank A from there.

CP 5 The request for location 113 is held in the SAS rank A and enters the SCM bank 13 address register from there.

CP 10 Location 113 is held in the read operand register of SCM bank 13 and enters rank 12 of the instruction stack from there.

CP 11 The new instruction word is available in the instruction stack and enters the CIW from there. The IB6 instruction issues in the following clock period.

IAS(12)	ADDRESS	COMPASS	MNEMONIC	XXXX 0123	XXXX 4567	AAAA 0123	AAAA 4567	BBB 123	BBBB 4567	CLOCK	IFA	SAS A	SAS B	SAS C	01234567	SMALL CORE MEMORY 01234567	01234567	01234567
111	110	IB7 B0		....	....	....	....	...	....	0	12					01		
				....	....	....	....	...	...S	1		12				01		
				....	....	....	....	...	...S	2						012		
		JP	113	....	....	....	....	...	....	3						12		
				....	....	....	....	...	....	4	13					2		
				....	....	....	....	...	....	5	14	13				2		
				....	....	....	....	...	....	6		14				23		
112				....	....	....	....	...	....	7						234		
				....	....	....	....	...	....	8						234		
				....	....	....	....	...	....	9						234		
				....	....	....	....	...	....	10						234		
113				....	....	....	....	...	....	11						34		
114	113	IB6 B0		....	....	....	....	...	....	12	15					34		
				....	....	....	....	...	..S.	13		15				34		
				....	....	....	....	...	..S.	14						345		
		JP	104	....	....	....	....	...	....	15						45		
				....	....	....	....	...	....	16	04					5		
				....	....	....	....	...	....	17	05	04				5		
				....	....	....	....	...	....	18		05			4	5		
115				....	....	....	....	...	....	19					45	5		
				....	....	....	....	...	....	20					45	5		
				....	....	....	....	...	....	21					45	5		
				....	....	....	....	...	....	22					45	5		
104				....	....	....	....	...	....	23					45			

Figure XII

XIII. HOLE IN STACK

Summary: 1. Instruction look-ahead fetches the two words of the calling routine that follow the subroutine call.

2. As the subroutine is too short to fill the instruction stack, the two words of the calling routine are still in the stack upon return to the calling routine.

3. The instruction look-ahead is not active in the calling routine since there is no coincidence with rank 11 or 12.

4. The situation is detected and recovered when the CIW is empty and no more instruction words are on their way to the stack.

```
*            EXAMPLE   XIII              32 CLOCK PERIODS EXECUTION TIME

*            SB1       BACK
*            JP        SUB
*   BACK     SB0       0
*            SB0       0
*            SB0       0
*            SB0       0
*            NO
*            RETURN
*   SUB      JP        B1
```

Notes

See Figure XIII.

CP 6 Word 112 enters rank 12 of the instruction stack.

CP 10 Word 113 enters rank 12 of the instruction stack, thus displacing word 112 to rank 11.

CP 13 Word 115, the first word of the subroutine, enters rank 12 of the instruction stack, thus displacing words 112 and 113 to ranks 10 and 11 respectively.

CP 14 Word 116 enters rank 12 of the instruction stack, thus displacing words 112 and 113 to ranks 9 and 10 respectively.

CP 17 The subroutine exits via a jump in stack. No request is made for a further instruction word since coincidence is with rank 9 of the instruction stack.

CP 19 Word 113 enters the CIW from rank 10 of the instruction stack. No request is made for a further instruction word since coincidence is with rank 10 of the instruction stack.

CP 21 The CIW becomes empty since word 114 is not in the instruction stack. No request is made for word 114 yet, because a word requested earlier (i.e. word 117) is not yet available in the stack.

CP 22 Now that no further instruction words are on their way to the stack, and word 114 is missing, a request for this word is made. The further sequence of events is the same as if a jump out of stack would have issued during clock period 22.

IAS(12)	ADDRESS	COMPASS MNEMONIC		XXXX 0123	XXXX 4567	AAAA 0123	AAAA 4567	BBB 123	BBBB 4567	CLOCK	IFA	SAS A	SAS B	SAS C	SMALL CORE MEMORY 01234567	01234567	01234567	01234567
111	110	IB7	B0	....	....	....	....	...	....	0	12				01			
				....	....	....	....	...	...S	1		12			01			
		SB1	112	....	....	....	....	...	...S	2					012			
		NO		....	....	....	....	I..	....	3					12			
				....	....	....	....	...	....	4	13				2			
				....	....	....	....	...	....	5		13			2			
	111	JP	115	....	....	....	....	...	....	6					23			
112				....	....	....	....	...	....	7	15				23			
				....	....	....	....	...	....	8	16	15			23			
				....	....	....	....	...	....	9		16			23	5		
				....	....	....	....	...	....	10					23	56		
113				....	....	....	....	...	....	11					3	56		
				....	....	....	....	...	....	12					3	56		
				....	....	....	....	...	....	13					3	56		
115				....	....	....	....	...	....	14					3	56		
116				....	....	....	....	...	....	15	17					56		
				....	....	....	....	...	....	16		17				56		
	115	JP	B1+ 0	....	....	....	....	...	....	17						567		
	112	SB0	0	....	....	....	....	...	....	18						67		
		SB6	0	....	....	....	....	...	....	19						7		
	113	SB0	0	....	....	....	....	...	....	20						7		
		SB0	0	....	....	....	....	...	....	21						7		
117				....	....	....	....	...	....	22						7		
				....	....	....	....	...	....	23	14					7		
				....	....	....	....	...	....	24	15	14				7		
				....	....	....	....	...	....	25		15				4	7	
				....	....	....	....	...	....	26						45		
				....	....	....	....	...	....	27						45		
				....	....	....	....	...	....	28						45		
				....	....	....	....	...	....	29						45		
114				....	....	....	....	...	....	30						45		
115	114	NO		....	....	....	....	...	....	31	16					45		
		IB6	B0	....	....	....	....	...	....	32		16				45		
				....	....	....	....	...	..S.	33						456		
				....	....	....	....	...	..S.	34						56		
		JP	104	....	....	....	....	...	....	35						6		
				....	....	....	....	...	....	36	04					6		
				....	....	....	....	...	....	37	05	04				6		
116				....	....	....	....	...	....	38		05		4		6		
				....	....	....	....	...	....	39				45		6		
				....	....	....	....	...	....	40				45		6		
				....	....	....	....	...	....	41				45		6		
				....	....	....	....	...	....	42				45				
104				....	....	....	....	...	....	43				45				

Figure XIII

XIV. BANK CONFLICT DELAYS JUMP ADDRESS IN SAS

Summary: 1. A jump out of stack transfers control to a location in an SCM bank which is busy.

2. The instruction fetch is therefore delayed in the SAS.

```
*            EXAMPLE   XIV                20 CLOCK PERIODS EXECUTION TIME

*            SA1       OUT+40B
*            NO
*            JP        OUT
*   GAP      BSS       2
*   OUT      RETURN
```

Notes

See Figure XIV.

CP 7 The request for instruction word 114 is delayed in the SAS rank A until SCM bank 14 has completed the read/write cycle for the operand in word 154.

IAS(12)	ADDRESS	COMPASS	MNEMONIC	XXXX XXXX AAAA AAAA BBB BBBB 0123 4567 0123 4567 123 4567	CLOCK	IFA	SAS A	SAS B	SAS C	SMALL CORE MEMORY 01234567	01234567	01234567	01234567
111	110	IB7 B0		....	0	12							01
		SA1	154	....S	1		12						01
		NO		.F.. .... .I.. ....S	2								012
				.F..	3	13	54						12
				.F..	4		13						2 4
	111	JP	114	.F..	5								234
				.F..	6	14							234
112				.F..	7	15	14						234
				.F..	8		14	15					234
				....	9		14	15					234
113				....	10		14	15					234
				....	11		14	15					34
				....	12		14	15					34
				....	13		14	15					3
				....	14			15					4
				....	15								45
				....	16								45
				....	17								45
				....	18								45
114				....	19								45
115	114	IB6 B0		....	20	16							45
				....S.	21		16						45
				....S.	22								456
		JP	104	....	23								56
				....	24	04							6
				....	25	05	04						6
				....	26		05			4			6
116				....	27					45			6
				....	28					45			6
				....	29					45			6
				....	30					45			6
104				....	31					45			

Figure XIV

XV. SAS BACKUP DELAYS JUMP ADDRESS IN IFA

Summary: 1. A bank conflict leads to an SAS backup condition.

2. The address of a jump out of stack thus remains in the IFA until the SAS backup condition is resolved.

```
*            EXAMPLE  XV                    21 CLOCK PERIODS EXECUTION TIME

*            NO
*            SA1      *+42B
*            JP       OUT
*   GAP      BSS      2
*   OUT      RETURN
```

Notes

See Figure XV.

CP 4 The operand address 152 is delayed in the SAS rank A until SCM bank 12 has completed the read/write cycle for the instruction word 112. The request for instruction word 113 is therefore directed into the SAS rank B, thus causing an SAS backup condition.

CP 6 The jump address 114 is delayed in the IFA until the current SAS backup condition is resolved.

IAS(12)	ADDRESS	COMPASS MNEMONIC		XXXX 0123	XXXX 4567	AAAA 0123	AAAA 4567	BBB 123	BBBB 4567	CLOCK	IFA	SAS A	SAS B	SAS C	SMALL CORE MEMORY 01234567	01234567	01234567	01234567
111	110	IB7 B0		....	....	....	....	...	....	0	12							01
		NO		....	....	....	....	...	...S	1		12						01
		SA1	152	....	....	....	....	...	...S	2								012
				.F..	....	.I..	....	...	....	3	13							12
				.F..	....	....	....	...	....	4	13	52						2
	111	JP	114	.F..	....	....	....	...	....	5		52	13					2
				.F..	....	....	....	...	....	6	14	52	13					2
112				.F..	....	....	....	...	....	7	14	52	13					2
				.F..	....	....	....	...	....	8	14	52	13					2
				.F..	....	....	....	...	....	9	14	52	13					2
				.F..	....	....	....	...	....	10	14	52	13					2
				.F..	....	....	....	...	....	11	14	52	13					
				.F..	....	....	....	...	....	12	14		13					2
				.F..	....	....	....	...	....	13	14							23
				.F..	....	....	....	...	....	14	15	14						23
				.F..	....	....	....	...	....	15		15						234
				.F..	....	....	....	...	....	16								2345
				....	....	....	....	...	....	17								2345
113				....	....	....	....	...	....	18								2345
				....	....	....	....	...	....	19								2345
114				....	....	....	....	...	....	20								2345
115	114	IB6 B0		....	....	....	....	...	....	21	16							345
				....	....	....	....	...	..S.	22		16						45
				....	....	....	....	...	..S.	23								456
		JP	104	....	....	....	....	...	....	24								56
				....	....	....	....	...	....	25	04							6
				....	....	....	....	...	....	26	05	04						6
				....	....	....	....	...	....	27		05			4			6
116				....	....	....	....	...	....	28					45			6
				....	....	....	....	...	....	29					45			6
				....	....	....	....	...	....	30					45			6
				....	....	....	....	...	....	31					45			6
104				....	....	....	....	...	....	32					45			

Figure XV

XVI. SAS BACKUP DELAYS JUMP ADDRESS + 1 IN IFA

Summary: 1. A bank conflict arises between operand fetch and instruction fetch.

2. The address of a jump out of stack is therefore entered into the SAS rank B.

3. The SAS backup condition delays the request for the second instruction word in the IFA.

```
*            EXAMPLE   XVI                22 CLOCK PERIODS EXECUTION TIME

*            NO
*            SA1       *+43B
*            JP        OUT
*   GAP      BSS       2
*   OUT      RETURN
```

Notes

See Figure XVI.

CP 5 The instruction word address 113 is delayed in the SAS rank A until SCM bank 13 has completed the read/write cycle for the operand 153.

CP 6 The request for instruction word 114 enters the SAS rank B and thus creates an SAS backup condition from the following clock period onwards.

CP 7 The request for instruction word 115, the second word requested as a result of the JP instruction, is delayed in the IFA until the current SAS backup condition is resolved.

IAS(12)	ADDRESS	COMPASS MNEMONIC	XXXX 0123	XXXX 4567	AAAA 0123	AAAA 4567	BBB 123	BBBB 4567	CLOCK	IFA	SAS A	SAS B	SAS C	01234567	SMALL CORE MEMORY 01234567	01234567	01234567
111	110	IB7 BC	....	....	....	....	...	....	0	12					01		
		NO	....	....	....	....	...	...S	1		12				01		
		SA1 153	....	....	....	....	...	...S	2						012		
			.F..	....	.I..	....	...	....	3	13					12		
			.F..	....	....	....	...	....	4	13	53				2		
	111	JP 114	.F..	....	....	....	...	....	5		13				23		
			.F..	....	....	....	...	....	6	14	13				23		
112			.F..	....	....	....	...	....	7	15	13	14			23		
			.F..	....	....	....	...	....	8	15	13	14			23		
			.F..	....	....	....	...	....	9	15	13	14			23		
			....	....	....	....	...	....	10	15	13	14			23		
			....	....	....	....	...	....	11	15	13	14			3		
			....	....	....	....	...	....	12	15	13	14			3		
			....	....	....	....	...	....	13	15	13	14			3		
			....	....	....	....	...	....	14	15	13	14					
			....	....	....	....	...	....	15	15		14			3		
			....	....	....	....	...	....	16	15					34		
			....	....	....	....	...	....	17		15				34		
			....	....	....	....	...	....	18						345		
			....	....	....	....	...	....	19						345		
113			....	....	....	....	...	....	20						345		
114			....	....	....	....	...	....	21						345		
	114	IB6 BD	....	....	....	....	...	....	22	16					345		
115			....	....	....	....	...	..S.	23		16				345		
			....	....	....	....	...	..S.	24						456		
		JP 104	....	....	....	....	...	....	25						56		
			....	....	....	....	...	....	26	04					56		
			....	....	....	....	...	....	27	05	04				6		
			....	....	....	....	...	....	28		05			4	6		
116			....	....	....	....	...	....	29					45	6		
			....	....	....	....	...	....	30					45	6		
			....	....	....	....	...	....	31					45	6		
			....	....	....	....	...	....	32					45	6		
104			....	....	....	....	...	....	33					45			

Figure XVI

XVII. THE THREE CASES OF LOOK-AHEAD

Summary: 1. Coincidence with rank 12, one more word coming.

2. Coincidence with rank 11, no more word coming.

3. Coincidence with rank 12, no more word coming.

```
*            EXAMPLE   XVII                18 CLOCK PERIODS EXECUTION TIME

*            FX6       X1*X1
*            FX6       X6*X6
*            NX6       X6
*            SA6       GAP
*            JP        ONE
*   GAP      BSS       1
*   ONE      RETURN
```

Notes

See Figure XVII.

CP -1 Instruction word 11Ø enters the CIW from rank 12 of the instruction stack. A request for one further instruction word (112) is generated, since one instruction word (111) is currently on its way to the stack.

CP 7 The look-ahead condition holds for the NX6 instruction. The next instruction word (111) is held in rank 11 of the instruction stack. A request for one further instruction word (113) is generated, since no instruction words are currently on their way to the stack.

CP 17 As the JP instruction issues, the next instruction word (113) is held in rank 12 of the instruction stack. A request for two further instruction words (114 and 115) is generated, since no instruction words are currently on their way to the stack.

IAS(12)	ADDRESS	COMPASS MNEMONIC		XXXX XXXX AAAA AAAA BBB BBBB 0123 4567 0123 4567 123 4567	CLOCK	IFA	SAS A B C	SMALL CORE MEMORY 01234567 01234567 01234567 01234567
111	110	ID7 B0		....	0	12		01
		FX6 X1*X1		....S	1		12	01
				.... ..M. ....S	2			012
				.... ..M.	3			12
				.... ..M.	4			2
				.... ..M.	5			2
		FX6 X6*X6		....	6			2
112				.... ..M.	7			2
				.... ..M.	8	13		2
				.... ..M.	9		13	2
				.... ..M.	10			23
		NX6 B0,X6		....	11			3
				.... ..N.	12			3
				.... ..N.	13			3
	111	SA6	112	....	14			3
113				....I.	15			3
				....	16		12	3
		JP	113	....	17			23
	113	IB6 B0		....	18	14		23
				....S.	19	15	14	2
				....S.	20		15	2 4
		JP	104	....	21			2 45
				....	22	04		2 45
				....	23	05	04	2 45
				....	24		05	4 2 45
114				....	25			45 2 45
115				....	26			45 45
				....	27			45 45
				....	28			45 45
104				....	29			45 5

Figure XVII

XVIII. RECOVERY FROM LOSS OF A JUMP ADDRESS

Summary: 1. A bank conflict leads to an SAS backup condition.

2. The request for the next instruction word is thus delayed in the IFA.

3. When a jump out of stack issues, its address cannot enter IFA and is lost.

4. The situation is detected and put right when the flow of instruction words to the instruction stack ceases.

```
*            EXAMPLE   XVIII              31 CLOCK PERIODS EXECUTION TIME

*            NX1       X1
*            SA1       *+2
*            SA2       A1+1
*            SB0       0
*            JP        DONE
*     GAP    BSS       2
*     DONE   RETURN
```

Notes

See Figure XVIII.

CP 6 The request for operand 112 is delayed in the SAS rank A until SCM bank 12 has completed the read/write cycle for this instruction word.

CP 7 The request for operand 113 enters the SAS rank B and thus creates an SAS backup condition from the following clock period onwards.

CP 8 The request for instruction word 114 is delayed in the IFA until the current SAS backup condition is resolved.

CP 10 The jump address 115 is lost because the IFA still holds the request for word 114.

CP 21 The instruction word 114 enters the instruction stack.

CP 22 No more instruction words are on the way to the instruction stack. A request to fetch the missing word 115 enters the IFA. The further sequence of events is the same as if the jump out of stack would have issued during clock period 22.

IAS(12)	ADDRESS	COMPASS MNEMONIC		XXXX 0123	XXXX 4567	AAAA 0123	AAAA 4567	BBB 123	BBB 4567	CLOCK	IFA	SAS A	SAS B	SAS C	SMALL CORE MEMORY 01234567	01234567	01234567	01234567
111	110	IB7	B0	....	....	....	....	...	....	0	12						01	
		NX1	B0,X1	....	....	....	....	...	...S	1		12					01	
				.N..	....	....	....	...	...S	2							012	
				.N..	....	....	....	...	....	3	13						12	
		SA1	112	....	....	....	....	...	....	4		13					2	
				.F..	....	.I..	....	...	....	5							23	
	111	SA2	A1+ 1	.F..	....	....	....	...	....	6		12					23	
112		SB0	0	.FF.	....	..I.	....	...	....	7		12					23	
				.FF.	....	....	....	...	....	8	14	12	13				23	
				.FF.	....	....	....	...	....	9	14	12	13				23	
113	112	JP	115	.FF.	....	....	....	...	....	10	14	12	13				23	
				.FF.	....	....	....	...	....	11	14	12	13				3	
				.FF.	....	....	....	...	....	12	14		13				23	
				.FF.	....	....	....	...	....	13	14		13				23	
				.FF.	....	....	....	...	....	14	14		13				2	
				.FF.	....	....	....	...	....	15	14						23	
				.FF.	....	....	....	...	....	16		14					23	
				..F.	....	....	....	...	....	17							234	
				..F.	....	....	....	...	....	18							234	
				..F.	....	....	....	...	....	19							234	
				....	....	....	....	...	....	20							234	
				....	....	....	....	...	....	21							34	
114				....	....	....	....	...	....	22							34	
				....	....	....	....	...	....	23	15						34	
				....	....	....	....	...	....	24	16	15					4	
				....	....	....	....	...	....	25		16					45	
				....	....	....	....	...	....	26							56	
				....	....	....	....	...	....	27							56	
				....	....	....	....	...	....	28							56	
				....	....	....	....	...	....	29							56	
115				....	....	....	....	...	....	30							56	
116	115	IB6	B0	....	....	....	....	...	....	31	17						56	
				....	....	....	....	...	..S.	32		17					56	
				....	....	....	....	...	..S.	33							567	
		JP	104	....	....	....	....	...	....	34							67	
				....	....	....	....	...	....	35	04						7	
				....	....	....	....	...	....	36	05	04					7	
				....	....	....	....	...	....	37		05		4			7	
117				....	....	....	....	...	....	38				45			7	
				....	....	....	....	...	....	39				45			7	
				....	....	....	....	...	....	40				45			7	
				....	....	....	....	...	....	41				45			7	
104				....	....	....	....	...	....	42				45				

Figure XVIII

XIX. SAS BACKUP DELAYS REFILL OF CIW AFTER JUMP

Summary: 1. A jump out of stack occurs to a location which is currently on its way to the instruction stack. This is not detected. The jump initiates a further fetch of the instruction word and thus causes an SCM bank conflict.

2. The second instruction word fetched by the jump out of stack leads to an SAS backup condition.

3. The first copy of the new instruction word enters the CIW only after the SAS backup condition has been resolved.

```
*            EXAMPLE   XIX                15 CLOCK PERIODS EXECUTION TIME

*            NO
*            JP        NEXT
*     GAP    BSS       1
*     NEXT   NO
*            RETURN
```

Notes

See Figure XIX.

CP -1 The look-ahead condition initiates a fetch of location 112.

CP 4 A jump out of stack is performed since location 112 is not yet available from the instruction stack.

CP 7 Although location 112 is now available from the instruction stack, its transfer to the CIW is prevented by the SAS backup condition.

CP 13 The instruction word from location 112 is copied into the CIW and its pass instruction issues during the following clock period.

IAS(12)	ADDRESS	COMPASS MNEMONIC	XXXX 0123	XXXX 4567	AAAA 0123	AAAA 4567	BBB 123	BBBB 4567	CLOCK	IFA	SAS A	SAS B	SAS C	01234567	SMALL CORE MEMORY 01234567	01234567	01234567
111	110	IB7 B0	....	....	....	....	...	....	0	12					01		
		NO	....	....	....	....	...	...S	1		12				01		
			....	....	....	....	...	...S	2						012		
			....	....	....	....	...	....	3						12		
		JP 112	....	....	....	....	...	....	4						2		
			....	....	....	....	...	....	5	12					2		
			....	....	....	....	...	....	6	13	12				2		
112			....	....	....	....	...	....	7		12	13			2		
			....	....	....	....	...	....	8		12	13			2		
			....	....	....	....	...	....	9		12	13			2		
			....	....	....	....	...	....	10		12	13			2		
			....	....	....	....	...	....	11		12	13					
			....	....	....	....	...	....	12			13			2		
			....	....	....	....	...	....	13						23		
	112	ND	....	....	....	....	...	....	14						23		
		IB6 B0	....	....	....	....	...	....	15						23		
			....	....	....	....	...	..S.	16						23		
112			....	....	....	....	...	..S.	17						23		
113		JP 104	....	....	....	....	...	....	18						23		
			....	....	....	....	...	....	19	04					23		
			....	....	....	....	...	....	20	05	04				23		
			....	....	....	....	...	....	21		05			4	3		
			....	....	....	....	...	....	22					45			
			....	....	....	....	...	....	23					45			
			....	....	....	....	...	....	24					45			
			....	....	....	....	...	....	25					45			
104			....	....	....	....	...	....	26					45			

Figure XIX

XX. FORWARD JUMP ACCELERATED

Summary: 1. A jump out of stack occurs to a location which is currently on its way to the instruction stack.

2. The CIW is refilled as soon as the first copy of the required location is available from the instruction stack.

```
*            EXAMPLE   XX                  13 CLOCK PERIODS EXECUTION TIME
*
*            SB3       B1
*            JP        NEXT
*     GAP    BSS       1
*     NEXT   NO
*            RETURN
```

Notes

See Figure XX.

CP -1 The look-ahead condition initiates a fetch of location 112.

CP 5 A jump out of stack is performed since location 112 is not yet available from the instruction stack.

CP 6 Instruction word 112 enters the instruction stack and is available from rank 12 during the following clock period.

CP 7 Instruction word 112 enters the CIW. The NO instruction issues during the following clock period despite the SAS backup condition.

IAS(12)	ADDRESS	COMPASS	MNEMONIC	XXXX 0123	XXXX 4567	AAAA 0123	AAAA 4567	BBB 123	BBBB 4567	CLOCK	IFA	SAS A	SAS B	SAS C	01234567	SMALL CORE MEMORY 01234567	01234567	01234567
111	110	IB7 B0		....	....	....	....	...	....	0	12					01		
				....	....	....	....	...	...S	1		12				01		
		SB3 B1		....	....	....	....	...	...S	2						012		
				....	....	....	....	..I	....	3						12		
				....	....	....	....	...	....	4						2		
		JP	112	....	....	....	....	...	....	5						2		
				....	....	....	....	...	....	6	12					2		
112				....	....	....	....	...	....	7	13	12				2		
	112	NO		....	....	....	....	...	....	8		12	13			2		
				....	....	....	....	...	....	9		12	13			2		
				....	....	....	....	...	....	10		12	13			2		
				....	....	....	....	...	....	11		12	13					
				....	....	....	....	...	....	12			13			2		
		I86 B0		....	....	....	....	...	....	13						23		
				....	....	....	....	...	..S.	14						23		
				....	....	....	....	...	..S.	15						23		
		JP	104	....	....	....	....	...	....	16						23		
112				....	....	....	....	...	....	17	04					23		
113				....	....	....	....	...	....	18	05	04				23		
				....	....	....	....	...	....	19		05			4	23		
				....	....	....	....	...	....	20					45	23		
				....	....	....	....	...	....	21					45	3		
				....	....	....	....	...	....	22					45			
				....	....	....	....	...	....	23					45			
104				....	....	....	....	...	....	24					45			

Figure XX

XXI. FORWARD JUMP DELAYED

Summary: 1. A jump out of stack occurs to a location which is currently on its way to the instruction stack.

2. The refill of the CIW is blocked during the clock period which follows a jump out of stack.

```
*            EXAMPLE   XXI              13 CLOCK PERIODS EXECUTION TIME
*
*            SB3       B7
*            JP        NEXT
*     GAP    BSS       1
*     NEXT   NO
*            RETURN
```

Notes

See Figure XXI.

CP -1 The look-ahead condition initiates a fetch of location 112.

CP 6 A jump out of stack is performed since location 112 is not yet available from the instruction stack.

CP 7 Although location 112 is now available from the instruction stack the CIW never refills in the clock period following a jump out of stack.

CP 8 Instruction word 112 enters the CIW. The NO instruction issues during the following clock period despite the SAS backup condition.

IAS(12)	ADDRESS	COMPASS	MNEMONIC	XXXX 0123	XXXX 4567	AAAA 0123	AAAA 4567	BBB 123	BBBB 4567	CLOCK	IFA	SAS A	SAS B	SAS C	01234567	SMALL CORE MEMORY 01234567	01234567	01234567
111	110	IB7 B0		....	....	....	....	...	....	0	12					01		
				....	....	....	....	...	...S	1		12				01		
				....	....	....	....	...	...S	2						012		
		SB3 B7		....	....	....	....	...	....	3						12		
				....	....	....	....	..I	....	4						2		
				....	....	....	....	...	....	5						2		
		JP	112	....	....	....	....	...	....	6						2		
112				....	....	....	....	...	....	7	12					2		
				....	....	....	....	...	....	8	13	12				2		
	112	NO		....	....	....	....	...	....	9		12	13			2		
				....	....	....	....	...	....	10		12	13			2		
				....	....	....	....	...	....	11		12	13					
				....	....	....	....	...	....	12			13			2		
		IB6 B0		....	....	....	....	...	....	13						23		
				....	....	....	....	...	..S.	14						23		
				....	....	....	....	...	..S.	15						23		
		JP	104	....	....	....	....	...	....	16						23		
112				....	....	....	....	...	....	17	04					23		
113				....	....	....	....	...	....	18	05	04				23		
				....	....	....	....	...	....	19		05			4	23		
				....	....	....	....	...	....	20					45	23		
				....	....	....	....	...	....	21					45	3		
				....	....	....	....	...	....	22					45			
				....	....	....	....	...	....	23					45			
104				....	....	....	....	...	....	24					45			

Figure XXI

XXII. LOOK-AHEAD DELAYS REFILL OF CIW AFTER JUMP

Summary: 1. The CIW is not refilled after a jump out of stack because an old instruction fetch is still in progress.

2. Refill of the CIW is delayed further by an SAS backup condition.

```
*            EXAMPLE   XXII              17 CLOCK PERIODS EXECUTION TIME

*            SA1       *+8
*            SA2       A1+B1
*            JP        NEXT
*   NEXT     NO
*            RETURN
```

Notes

See Figure XXII.

CP -1 The look-ahead condition initiates a fetch 3f location 112.

CP 2 The look-ahead condition initiates a fetch of location 113.

CP 6 A jump out of stack is performed since location 112 is not yet available from the instruction stack.

CP 7 Although location 112 is now available from the instruction stack, its transfer to the CIW is delayed since the instruction fetch initiated in CP 2 has not yet been completed. (Note the difference to EXAMPLE XIX where refill of the CIW is prevented by an SAS backup condition.)

CP 10 Although the instruction fetch which was initiated in CP 2 is now completed, location 112 still cannot enter the CIW because an SAS backup condition holds since CP 9.

CP 15 The instruction word from location 112 enters the CIW from rank 11 of the instruction stack. Its NO instruction issues during the following clock period. The look-ahead condition does not initiate a further instruction fetch since two instruction words (112 and 113) are currently on their way to the instruction stack.

IAS(12)	ADDRESS	COMPASS MNEMONIC		XXXX 0123	XXXX 4567	AAAA 0123	AAAA 4567	BBB 123	BBBB 4567	CLOCK	IFA	A	SAS B	C	01234567	SMALL CORE MEMORY 01234567	01234567	01234567
111	110	IB7	B0	....	....	....	....	...	....	0	12					01		
		SA1	1	....	....	....	....	...	...S	1		12				01		
				.F..	....	.I..	....	...	...S	2						012		
		SA2	A1+B1	.F..	....	....	....	...	....	3	13	01				12		
				.FF.	....	..I.	....	...	....	4		13			1	2		
				.FF.	....	....	....	...	....	5		02			1	23		
	111	JP	112	.FF.	....	....	....	...	....	6					12	23		
112				.FF.	....	....	....	...	....	7	12				12	23		
				.FF.	....	....	....	...	....	8	13	12			12	23		
				..F.	....	....	....	...	....	9		12	13		12	23		
113				..F.	....	....	....	...	....	10		12	13		12	23		
				....	....	....	....	...	....	11		12	13		12	3		
				....	....	....	....	...	....	12			13		12	23		
				....	....	....	....	...	....	13			13		2	23		
				....	....	....	....	...	....	14			13		2	2		
				....	....	....	....	...	....	15						23		
	112	NO		....	....	....	....	...	....	16						23		
112		IB6	B0	....	....	....	....	...	....	17						23		
				....	....	....	....	...	..S.	18						23		
				....	....	....	....	...	..S.	19						23		
113		JP	104	....	....	....	....	...	....	20						23		
				....	....	....	....	...	....	21	04					3		
				....	....	....	....	...	....	22	05	04				3		
				....	....	....	....	...	....	23		05			4	3		
				....	....	....	....	...	....	24					45			
				....	....	....	....	...	....	25					45			
				....	....	....	....	...	....	26					45			
				....	....	....	....	...	....	27					45			
104				....	....	....	....	...	....	28					45			

Figure XXII

XXIII. LINEAR SEQUENCE OF CODE FASTER THAN INSTRUCTION FETCH

Summary: 1. The instructions of a linear sequence of code issue faster than new instruction words arrive at the instruction stack.

2. As soon as the next instruction word is available in the instruction stack it is copied to the CIW.

```
*            EXAMPLE   XXIII              8 CLOCK PERIODS EXECUTION TIME

*            NO
*            NO
*            NO
*            SB3       1
*            NO
*            NO
*   NEXT     RETURN
```

Notes

See Figure XXIII.

CP -1 The look-ahead condition initiates a fetch of location 112.

CP 6 The CIW becomes empty because instruction word 111 issues before location 112 is available from the instruction stack.

CP 7 The CIW is empty. It is being refilled from rank 12 of the instruction stack, which now holds the instruction word from location 112. The IB6 instruction of this word issues during the following clock period.

IAS(12)	ADDRESS	COMPASS MNEMONIC		XXXX XXXX AAAA AAAA B88 8888 0123 4567 0123 4567 123 4567	CLOCK	IFA	SAS A	SAS B	SAS C	SMALL CORE MEMORY 01234567	SMALL CORE MEMORY 01234567	SMALL CORE MEMORY 01234567
111	110	IB7 B0		....	0	12						01
		NO		....S	1		12					01
		NO		....S	2							012
		NO		....	3							12
	111	SB3	1	....	4	13						2
		NO		....I	5		13					2
		NO		....	6							23
112				....	7							23
	112	IB6 B0		....	8	14						23
				....S.	9		14					23
				....S.	10							234
113		JP	104	....	11							34
				....	12	04						34
				....	13	05	04					34
				....	14		05		4			34
114				....	15				45			4
				....	16				45			4
				....	17				45			4
				....	18				45			4
104				....	19				45			

Figure XXIII

XXIV. BANK CONFLICT DELAYS REFILL OF CIW

Summary: 1. Two operand fetches cause an SCM bank conflict.

2. The SCM bank conflict leads to an SAS backup condition.

3. The SAS backup condition prevents the next instruction word from entering the CIW.

```
*   *
*           EXAMPLE   XXIV              17 CLOCK PERIODS EXECUTION TIME
*   *
*           NO
*           SA1       B1
*           NO
*           NO
*           NO
*           NO
*           NO
*           NO
*           NO
*           SA2       B1
*           SA3       A1+B1
*           SBØ       Ø
*           NO
*           NO
*           RETURN
```

Notes

See Figure XXIV.

CP 12 The operand address Ø1 of the SA2 instruction is delayed in SAS rank A for two clock periods because SCM bank 1 is still occupied with the SA1 instruction. The operand address Ø2 of the SA3 instruction therefore enters SAS rank B and thus creates an SAS backup condition during clock periods 13 to 15.

CP 15 The instruction word 114 is available in rank 12 of the instruction stack but its transfer to the CIW during this clock period is prevented by the SAS backup condition.

CP 16 The CIW is filled from rank 12 of the instruction stack and its IB6 instruction issues in the following clock period.

Acknowledgements

The timing program TIME76 and this report owe their existence to a draft by S. Cray, entitled "Control Data 76ØØ Computer System". Some details of 76ØØ instruction timing have been clarified by J.-P. Mueller of Control Data Corporation from engineering support manuals. His help and that of CDC are gratefully acknowledged.

IAS(12)	ADDRESS	COMPASS	MNEMONIC		XXXX 0123	XXXX 4567	AAAA 0123	AAAA 4567	BBB 123	BBBB 4567	CLOCK	IFA		SAS A	B	C	SMALL CORE MEMORY 01234567	01234567	01234567
111	110	IB7	B0		....	....	....	....	...	....	0	12					01		
		NO			....	....	....	....	...	...S	1		12				01		
		SA1	B1		....	....	....	....	...	...S	2						012		
		NO			.F..	....	.I..	....	...	....	3						12		
	111	NO			.F..	....	....	....	...	....	4	13	01				2		
		NO			.F..	....	....	....	...	....	5		13			1	2		
		NO			.F..	....	....	....	...	....	6					1	23		
112		NO			.F..	....	....	....	...	....	7					1	23		
	112	NO			.F..	....	....	....	...	....	8	14				1	23		
		NO			.F..	....	....	....	...	....	9		14			1	23		
		SA2	B1		....	....	....	....	...	....	10					1	234		
113		SA3	A1+B1		..F.	....	..I.	....	...	....	11					1	34		
	113	SB0		0	..FF	....	...I	....	...	....	12	15	01			1	34		
		NO			..FF	....	....	....	...	....	13	15	01	02		1	34		
		NO			..FF	....	....	....	...	....	14	15	01	02			34		
114					..FF	....	....	....	...	....	15	15		02		1	4		
					..FF	....	....	....	...	....	16	15				12	4		
	114	IB6	B1		..FF	....	....	....	...	....	17	16	15			12	4		
					..FF	....	....	....	...	..S.	18		16			12	45		
					..FF	....	....	....	...	..S.	19					12		56	
		JP		104	...F	....	....	....	...	....	20					12		56	
					....	....	....	....	...	....	21	04				12		56	
					....	....	....	....	...	....	22	05	04			12		56	
115					....	....	....	....	...	....	23		05			12	4	56	
116					....	....	....	....	...	....	24					2	45	56	
					....	....	....	....	...	....	25						45	56	
					....	....	....	....	...	....	26						45	56	
					....	....	....	....	...	....	27						45	6	
104					....	....	....	....	...	....	28						45		

Figure XXIV