

PRIMARY BLOCK DIAGRAM

CPU

The CPU described in this manual includes all central processor logic circuits not contained within the I/O Multiplexer, SCM, LCM, functional units, or scanner.

The CPU, together with the functional units, executes the CPU program stored in SCM. The CPU consists primarily of 24 operating registers and a 12-word instruction stack. Data moves to and from the CPU thru the operating registers. Data may be directly addressed in either SCM or LCM.

INSTRUCTION STACK AND ISSUE

The principal components of the instruction stack are the IWS and the IAS. A coincidence test is made between (P) and the contents of each IAS rank during each clock period. Whenever a coincidence occurs, the corresponding 60-bit instruction word in the IWS is sent to the CIW Register.

INSTRUCTION WORD STACK (IWS)

The IWS is a group of twelve registers, individually identified by rank, which hold program instruction words for execution. Data moves through the IWS from Rank 12 to Rank 1. Rank 12 is filled with SCM data and all other ranks are filled with data from the next-highest-order rank. The data shifted from Rank 1 is discarded. Data movement through the IWS only takes place when Shift Stack is present.

Shift Stack is generated by the read address entering into SAS and no SCM bank conflict. It is delayed to arrive coincident with the read data arriving at IWS.

INSTRUCTION ADDRESS STACK (IAS)

The IAS is a group of twelve registers which hold relative SCM program addresses on a one-for-one basis with the program words in the IWS. Rank 1 in the IAS contains the oldest address in the stack, and this address corresponds to the SCM location from which the word in Rank 1 of the IWS was read. All ranks of the IAS receive information from the next-highest-order rank during a Shift Stack condition in a manner analogous to that for the IWS. An address is read into Rank 12 of the IAS from the NSA Register. This address corresponds to the SCM address for the word arriving at Rank 12 of the IWS. The address in Rank 1 of the IAS is discarded.

The outputs of all IAS ranks are compared with (P) each clock period. Rank X Coincidence gates the IWS Rank X contents to the CIW Register.

60348000 A

NEXT STACK ADDRESS (NSA)

The NSA Register, adder, and Advance Stack Flag (ASF) generate the NSA. The NSA is a 16-bit SCM relative address for the next sequential word required by the IWS. The inputs to the NSA Register consist of a branch address from the P Register or the contents of the NSA Register plus one or zero from the NSA adder. One will be added to the NSA when ASF is set.

P REGISTER

The P Register contains the current program execution address.

There are four possible sources of data for the P Register. Read Data Bits 36-53 enter the P Register from the SRO Register in SCM as a part of the exchange package. K enters the P Register during conditional branch and return jump sequences. K+Bi enters during the unconditional jump instruction (02). P+1 enters during straight-line coding.

Under branch conditions, the P bits enter the IFA Register in SCM where they are added to RAS to form the fetch address of an instruction word. The P Register contents are compared with the breakpoint address. Coincidence between the two generates an error exit request. The P bits enter the Register Selection Network to become bits 36-53 in the first word of the exchange package. The contents of the P Register become the return jump exit (RJX) address during the return jump sequence.

CURRENT INSTRUCTION WORD (CIW) REGISTER

The CIW Register is divided into four 15-bit parcels. All four parcels are loaded in one clock period when an instruction word is read from the IWS. The highest-order parcel in the instruction word is issued first. An instruction issues from the CIW Register when the conditions in the functional units and operating registers are such that the functions required in the instruction execution may be performed to completion without conflicting with a previously issued instruction. The other parcels are then shifted left in the CIW Register by either 15 bits or 30 bits, depending on the instruction format for the instruction issued.

The Exchange Sequence Register And Counter provides three bits during the exchange sequence which become the i, j, and k designators that gate Xi, Aj, and Bk into the exchange package in SCM. The counter outputs also sequence the P, PSD, and support registers into the exchange package.

EXCHANGE JUMP ADDRESS (XJA)

The data in the XJA and XJA Operand Registers is cleared and new data is entered at the beginning of each exchange sequence. The XJA Register holds an address which, when combined with the quantity in the XJA Operand Register, becomes the exchange sequence initial address. The quantity in the XJA Operand Register is zero for all XJA inputs except K+Bj which must have RAS added to it.

There are several ways of initiating an exchange sequence and several sources for the exchange package initial address. The normal termination for an exchange package execution interval is caused by an Exchange Exit Instruction (013) in the associated program. In this case the Exit Mode Flag in the PSD Register determines the source of the exchange sequence initial address. If the flag is set, K+Bj is selected; if clear, NEA is the initial address. Abnormal terminations for an exchange package execution interval are caused by I/O Multiplexer interrupt requests or by error condition flags in the PSD Register. The four combinations of situations are mutually exclusive.

PSD AND SUPPORT REGISTERS

PROGRAM STATUS DESIGNATION (PSD) REGISTER

The PSD Register consists of 18 individual program status designation flags. Six of these flags are mode designators and 12 are condition designators as follows:

Bit	Flag	
17	Exit	Mode
16	Monitor	
15	Step	
14	Indefinite	
13	Overflow	
12	Underflow	Condition
11	LCM Parity	
10	SCM Parity	
9	LCM Block Range	
8	SCM Block Range	
7	LCM Direct Range	
6	SCM Direct Range	
5	Program Range (Error Stop)	
4	Breakpoint	
3	Step	
2	Indefinite	
1	Overflow	
0	Underflow	

The PSD Register is loaded along with the other CPU registers during an exchange sequence. All 18 bits are entered into the register at this time. The six mode designators remain unaltered throughout the execution interval for the exchange package. The 12 condition designators may be set by transient conditions which occur during the execution interval. All flags are stored in the SCM exchange package at the end of the execution interval.

The execution interval for an exchange package may be terminated by an error condition which occurred during this interval. The monitor program must determine the nature of the error condition by analyzing the flags in the PSD portion of the exchange package in SCM. This monitor program may then clear the flag in the exchange package and resume execution of the object program or it may take some alternate action.

SUPPORT REGISTERS

The support registers assist the operating registers during the execution of programs. The registers are entered with SCM Read Data during the execution of an exchange sequence. The information is not altered during the execution interval for an exchange package. When the execution interval has been completed, the data in these registers is sent back to SCM.

Reference Address - SCM (RAS) Register

(RAS) are added to the relative address specified by the instruction to form the absolute SCM address.

Field Length - SCM (FLS) Register

Relative SCM addresses are compared with (FLS). If a relative address equals or exceeds (FLS), the SCM Block Range Condition Flag or SCM Direct Range Condition Flag in the PSD Register will set.

Reference Address - LCM (RAL) Register

(RAL) are added to the relative address specified by the instruction to form the absolute LCM address.

Field Length - LCM (FLL) Register

Relative LCM addresses are compared with (FLL). If a relative address equals or exceeds (FLL), the LCM Block Range Condition Flag or LCM Direct Range Condition Flag in the PSD Register will set.

Normal Exit Address (NEA) Register

NEA is used as the absolute SCM address for the new exchange package during an exchange exit (013) instruction when the Exit Mode Flag in the PSD Register is clear.

Error Exit Address (EEA) Register

EEA is used as the absolute SCM address for the exchange package that will terminate the program whenever an error exit occurs during the execution interval for an exchange package.

B REGISTERS

The B Registers are intended primarily for indexing functions in program execution. The B0 Register does not physically exist in the hardware. In the execution of instructions this register appears to contain all zero bits.

INPUTS

Only one data source may transmit data in a given clock period. It is gated to the B Register by access control. The increment input is present during 60-67 instruction execution and is gated by Go 2 Cycle To Bi. The normalize inputs are gated by Go Normalize during 24 and 25 instruction execution. The SCM inputs are present during an exchange sequence and are gated by signals from Exchange Destination Control. The Boolean Unit data is gated by Go 2 Cycle To Bj during 26 instruction execution. The multiplexer inputs are present during 016 and 017 instruction execution when the j designator is not equal to zero. They are gated by Go Read Channel.

OUTPUTS

Bi and Bj are used by the Jump Test circuit for 04 thru 07 instructions. Bi is one input to the K+Bi adder. It is used to form the word count during 02 Jump instruction execution. Bj is used by the K+Bj adder for block copy instructions 011 and 012 and the exchange exit instruction with the exit flag set.

BPA REGISTER

A program may be executed in small sections by inserting a breakpoint address in the BPA portion of the program's exchange package in SCM. During an exchange sequence, the BPA from the exchange package in SCM is entered into the BPA Register.

When (BPA) are coincident with (P), the Breakpoint Coincidence signal is sent to Exchange Jump Control and sets the Breakpoint Condition Flag in the PSD Register to initiate an exchange sequence.

X REGISTERS

The eight X Registers are the principal operating registers for the CPU. These registers are each 60 bits in length and serve as the source and destination for operands in execution of the arithmetic instructions. The X Registers receive data from all nine of the functional units and from SCM and LCM.

INPUTS

Translator

The translator merges the special case and sign data from the functional units into complement control signals. The signals are used by Complement Control to generate the bit patterns for the special case floating point formats or to complement the entire word when the result from the functional unit should be negative.

It also uses the special case data from the functional units to generate and transmit overflow, underflow, or indefinite signals to the PSD Register.

Input To X Network

Every functional unit has at least one result data path and several of the floating point units have multiple 60-bit result data paths to the X Registers. Instruction issue control insures that only one of the nine functional units sends a result to the X Registers during any given clock period. The data flow from the functional units is treated in two groups. One group, consisting of the Multiply, Divide, Shift, and Normalize Units, is treated in a static merge network. Data from these units flows thru the Complement Control network before entering the X Registers. The second group, consisting of the remaining functional units and the storage units, merges with the data from the Complement Control network in a second static merge network. The 60 bits from this last network are delivered to all eight X Registers. The data is entered into a specific register selected by the outputs of the X Register Selection Network.

X Register Access Control and Selection

The X Register access control and selection functions determine the timing and the specific register selection for each word entered into an X Register. The X Register Selection Network selects the destination X Register for all two cycle (two clock period) instructions. The i and j designators are enabled by one of several gating signals to select an Xi or an Xj Register depending on the instruction.

A four-register delay chain is provided in X Register Access Control for instructions with execution times exceeding two clock periods. For example, Register One supplies an extra clock period of delay for the three-clock period normalize instruction's i designator value.

OUTPUTS

The contents of all eight of the X Registers enter a static distribution network along with the i, j, and k designators from the CIW Register. The distribution network decodes the i, j, and k designators and gates the contents of the selected Xi, Xj and Xk registers to the nine functional units, SCM, and LCM.

The Xi outputs to SCM stores X0 thru X7 in the exchange package. The Xj output to Jump Test is used during the 03 branch instructions. The Xk and Xj outputs to LCM provide an address and a storage word respectively for the LCM direct read and write instructions.

A REGISTERS

The eight A Registers are the vehicle for addressing SCM for operands and storing results.

Registers A1 thru A5 are used to address SCM in reading data from SCM to an X Register. A read SCM reference is initiated whenever one of these A Registers is the destination in execution of an increment instruction. The data from the relative SCM address is delivered to the corresponding X Register. For example, an increment instruction with a destination of the A2 Register causes an SCM reference with the SCM data sent to the X2 Register.

Registers A6 and A7 are used to address SCM in writing data into SCM from an X Register. A write SCM reference is initiated whenever one of these A Registers is the destination during execution of an increment instruction. The data from the corresponding X Register is delivered to the SWS to be written into the relative SCM address specified in the A Register.

INPUTS

Only one data source may transmit data in a given clock period. The input from the Increment Unit is gated into the selected A Register by Go 2 Cycle to Ai during 50-57 instruction execution. The SCM input fills A0 thru A7 during the exchange sequence.

OUTPUTS

The data in the A Register specified by the j designator is delivered to the Increment Unit and SCM each clock period. The data in the A0 Register, the starting relative SCM address for block copy instructions, is delivered to SCM and LCM each clock period. The SCM data path, from the distribution network, is used for storing the A Register data in the exchange package in SCM during an exchange sequence. The Increment Unit data path provides data from Aj for 50, 54, 55, 60, 64, 65, 70, 74, and 75 instruction execution.

SECONDARY BLOCK DIAGRAM PART OF INSTRUCTION STACK, ISSUE, AND BRANCH

INSTRUCTION STACK

The instruction stack consists of the instruction word and address paths on CPU 2.0 and the control circuits on sheet CPU 2.1. The principal components of the instruction stack are the IWS and the IAS. A coincidence test is made between (P) and the contents of each IAS rank during each clock period. Whenever a coincidence occurs, the corresponding 60-bit instruction word in the IWS is sent to the CIW Register. If more than one coincidence occurs, all are sent simultaneously to the CIW Register and the results are merged as a logical sum (A+B).

INSTRUCTION WORD STACK (IWS)

The IWS is a group of twelve 60-bit registers which hold program instruction words for execution. Data moves from IWS Rank 12 to IWS Rank 1 at a rate of one rank per clock period only when Shift Stack is present. Rank 12 is filled with SCM data and all other ranks are filled with data from the next-highest-order rank. The data shifted from Rank 1 is discarded.

Shift Stack is generated by the read address entering into SAS and no SCM bank conflict. It is delayed to arrive coincident with the read data arriving at IWS.

INSTRUCTION ADDRESS STACK (IAS)

The IAS is a group of twelve 18-bit registers which hold relative SCM program addresses on a one-for-one basis with the program words in the IWS. Rank 1 in the IAS contains the oldest address in the stack, and this address corresponds to the SCM location from which the word in Rank 1 of the IWS was read. All ranks of the IAS are clear/entered with information from the next-highest-order rank during a Shift Stack clock period in a manner analogous to that for the IWS. An address is read into Rank 12 of the IAS from the NSA Register. This address corresponds to the SCM address for the word arriving at Rank 12 of the IWS. The address in Rank 1 of the IAS is discarded.

The outputs of all IAS ranks are compared with (P) each clock period. Bit group coincidence is detected on the HA modules. Since bits 16 and 17 of IAS Rank 12 receive no input and a constant one, respectively, bits 16 and 17 of the P Register

must be zero and one, respectively, to achieve coincidence. The group results enter the HB modules where they are checked for word coincidence. Word X Coincidence gates the IWS Rank X contents to the CIW Register.

During exchange and return jump instructions, the IAS is cleared, voiding the IWS. No further instructions can enter the CIW Register until the IWS is loaded with a new sequence of instructions.

Operation

The Next Stack Address (NSA) for the IAS is a 16-bit SCM relative address for the next sequential word required by the IWS. It is generated in the NSA Register and NSA Adder modules.

The two possible sources of data for the NSA Register are the P Register and the NSA Loop. The contents of the P Register enter at the end of a clock period in which a P to NSA condition exists. P to NSA is generated by Enable P to NSA-1 or Enable P to NSA-2 and Shift Stack. The contents of the NSA Register plus one enter via the NSA Loop at the end of a clock period in which ASF is set. ASF is set by Shift Stack and $\bar{P}$ to NSA. When both conditions exist in the same clock period, the P to NSA condition takes priority. When neither condition exists, the same value constantly reenters the NSA Register via the NSA Loop.

P to NSA signals the arrival of the last fetch word requested prior to a program branch. Thus, all IWS words have arrived from SCM, their corresponding addresses have entered into the IAS, and the branch address can now enter the NSA Register.

Address

When Shift Stack is present and ASF is not set, the address entered into IAS Rank 12 is a copy of the contents of the NSA Register. When Shift Stack is present and ASF is set, the address entered into IAS Rank 12 is (NSA) plus one. Generally, the address that enters IAS also loops back to enter the NSA Register during the same clock period. The only exception to this is when P to NSA and ASF occur in the same clock period. This causes (P) to enter the NSA Register as (NSA) plus one enters IAS Rank 12.

INSTRUCTION ISSUE

The instruction issue function is performed by the CIW Register and associated circuits on the right side of sheet CPU 2.0 and the additional control circuits on CPU 2.1.

Program instruction words are read one at a time from the Instruction Word Stack into the Current Instruction Word (CIW) Register for execution. Each instruction word is divided into four 15-bit parcels. There may be as many as four instructions in the CIW Register at one time. These instructions must be executed in sequence and the proper allowance made for the mixture of one-parcel and two-parcel instruction formats.

An instruction issues from the CIW Register when the conditions in the functional units and operating registers are such that the functions required in the instruction execution may be performed to completion without conflicting with a previously issued instruction. This is indicated in the hardware by the presence of Enable Issue and Registers Free. The issue process requires one clock period. During this clock period the g, h, i, j, and k designators in the instruction are sent to all appropriate parts of the CPU. The proper control flags are set at the end of this clock period to execute the functions required in the instruction. Also, at the end of the clock period, the data in the CIW Register is altered to position the next instruction for execution. Since there is no further record of the instruction once it issues from the CIW Register, it must be executed to completion in a fixed-time framework. No delays are allowable from issue to delivery of data to the destination operating registers.

CURRENT INSTRUCTION WORD (CIW) REGISTER

Operation

The CIW Register is divided into four 15-bit parcels. All four parcels are loaded in one clock period from the IWS. The highest-order parcel in the instruction word is issued first. The other parcels are then shifted left in the CIW Register by either 15 bits or 30 bits, depending on the instruction format for the instruction issued.

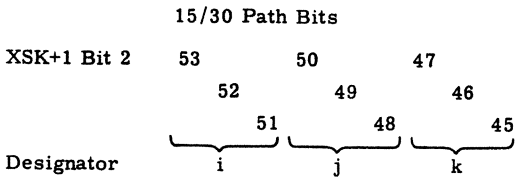
The CIW Register clears and new data enters at the end of a clock period in which Go Issue exists. Go Issue is generated by Registers Free and Enable Issue. The presence of either Sample IWS or Gate 15/30 Path determines whether the input data is from the IWS or from the lower parcels of the CIW Register. The absence of both gates during Go Issue enters a 60-bit word of zeros into the register.

IWS Path

Data from the IWS path is selected whenever Sample IWS is present. If coincidence occurs between (P) and IAS, an instruction word enters the CIW Register; otherwise, all zeros enter the register.

15/30 Path

Data enters from the 15/30 path whenever Gate 15/30 Path is present. Circuits on the LG modules determine which data is entered. It will be either the CIW shifted 15 or 30 bits to the left or XSK+1. A translator in the LG modules uses the g, h, and i designators to discover the number of parcels in the issuing instruction and determine the subsequent shift. The g, h, and i designators are zeros during part of the exchange sequence allowing the XSK+1 bits to enter the CIW Register as follows:



Gate 15/30 Path gates the shifted CIW to the input of the CIW Register after determining that there is at least one instruction left in the word and that the current instruction is not a branch.

Zeros fill the CIW Register during an exchange exit instruction (013). The zero value of the g and h designators is used to generate various exchange sequence control signals such as XSF and XIF.

PARCEL COUNT

The parcel count reflects the number of parcels in the CIW that have been issued. The parcel counting function is performed by the Parcel Counter Register (PKR) on the 1CC7 module and a translator on the LK module. The (PKR) change each time an instruction issues by an amount equal to the number of parcels in the issuing instruction.

OUTPUTS

The following concerns those outputs not discussed above.

Read Stack indicates that the instruction currently positioned in the upper parcel of the CIW Register is the last instruction in the word. When this instruction issues, the CIW Register will be loaded with a new 60-bit word from the IWS.

Set Error Flag sets the Program Range Condition Flag in the PSD Register when an error exit code (00) issues from the CIW Register. It is inhibited during the dead start, exchange sequence, and branch out of stack conditions when the CIW Register normally contains zeros.

Set XSF indicates that the execution of the current program instruction word is completed. It sets the Exchange Sequence Flag to continue the exchange sequence. It is generated as RIF clears when the following conditions are satisfied:

1. LCM is not busy.
2. All previously issued instructions have delivered their results to the operating registers.
3. No SCM bank conflicts occurred that prevented an instruction fetch address from leaving the IFA Register.

Go LCM Block sets the Go LCM Block Flag in LCM Control. It initiates an LCM block copy and is generated during the 011 and 012 instructions when the conditions listed under Set XSF above are satisfied.

BRANCH

Branch instructions are executed in the portion of the CPU that contains the program address register (P) and those control flags required to sequence the entry of a new address into the P Register as well as the register (RJX) and control flags required to execute the return jump instruction.

The P and RJX Register modules are located on this sheet but the associated control flags are shown on several other sheets. A discussion of the branch functions and the P and RJX Registers follows. For information on the control flags, refer to the text for the sheet on which the flag is shown.

NORMAL BRANCH EXECUTION

A normal branch instruction, as distinguished from a return jump instruction, has three cases to consider in execution timing. One case occurs when the jump criterion is not met in a conditional branch instruction. This is referred to as the fall thru case.

A second case, called branch in stack, occurs when the jump is taken to an address currently within the IAS. A third case, called branch out of stack, occurs when the jump is taken out of the range of the instruction stack. Each of these cases has a separate timing sequence.

All normal branch instructions begin execution with the setting of the GJF. The conditions for setting GJF are similar to the conditions for issue of a computation instruction. GJF causes the instruction to issue only in the fall thru case. The other two cases involve further sequence control flags.

When the jump criterion is met, a new program sequence address is entered into the P Register at the end of the clock period in which GJF is set. JCF sets at the same time to indicate that the jump has been completed. The branch instruction then issues in the following clock period. If the new program address is in the IAS, the next instruction word may be read directly into the CIW Register in this clock period. If the new program address is not in the IAS, an SCM reference must be initiated to read the new program instruction word. This process is a function of the instruction stack Fetch Control (CPU 2.1).

RETURN JUMP EXECUTION

Execution of a return jump instruction begins with the setting of GJF. GFJ enters the new program sequence address into the P Register. This is the K portion of the return jump instruction and becomes the address of the exit word for the called subroutine. At the same time, the previous address in the P Register is copied into the RJX Register. This is the address of the instruction word currently in the CIW Register plus one. RJF is set at the end of this clock period to continue the sequence.

RJF is a delay mechanism to allow the completion of any SCM instruction word references. RJF statically clears all ranks of the IAS for as long as it is set. When all instruction words previously requested have arrived at the IWS, RJF clears and SXF sets to continue the sequence. The contents of the P Register are transmitted to the IFA Register during the last clock period in which RJF is set.

SXF causes the SCM reference to store the special word required for the called subroutine exit. The data for this word is in the RJX Register. The address for the SCM reference is in the P Register and in the IFA Register. SXF remains set until the SCM reference has been initiated by the acceptance of (IFA) in the SAS. SXF then clears and JCF sets to complete the sequence. The contents of the P Register are advanced one count during the last clock period in which SXF is set.

JCF allows the return jump instruction to issue from the CIW Register. This flag is set for only one clock period. During this clock period the P Register contains the entrance address (K+1) for the called subroutine. The IFA Register contains this same address. No coincidence is possible between (P) and the IAS because of the previous clearing of the IAS. The instruction stack fetch control will therefore initiate the SCM references to read up the beginning of the called subroutine.

P REGISTER

The P Register is an 18-bit register which contains the current program execution address. The current contents of the P Register are cleared and new data is entered at the end of a clock period in which Enter P exists.

The P Register enters into a large number of functions so it is physically reproduced six times (one-third of a P Register on each HA module) to allow the proper fanout of information. All hardware copies of the P Register are entered with the same data in the same clock period and under the same conditions.

P Input Merge

There are four possible sources of data for the P Register. Read Data Bits 36-53 enter the P Register from the SRO Register in SCM as a part of the exchange package. K Bits 0-17 enter the P Register during conditional branch and return jump sequences. K+Bi enters during the unconditional jump instruction (02). P+1 enters during straight-line coding.

Outputs

See the IAS paragraphs above for discussions on two of the P Register outputs.

P Bits 0-17 plus two Bit 17s from the adjacent P Register are tested for zero on CPU 2.4. (P) equal to zero generates an error exit request. Under branch conditions, P Bits 0-15 enter the IFA Register in SCM where they are added to RAS to form the fetch address of an instruction word. The contents of the P Register become the return jump exit (RJX) address during the return jump sequence. (See RJX Register below.) The contents of P are subtracted from the SCM field length (FLS) in the LH module. (P) $\geq$ FLS generates an error exit request. This terminates the execution interval for the exchange package at the end of the current instruction word. The P Register contents are compared with the breakpoint address in the LR module. Coincidence generates an error exit request. P Bits 0-17 enter the LT module to become bits 36-53 in the first word of the exchange package.

RETURN JUMP EXIT (RJX) REGISTER

The RJX Register holds the return jump exit address during execution of the return jump instruction. The current contents of the P Register (CIW address plus one) enter the RJX Register at the end of a clock period in which Enter RJX exists and remain there until the execution of the next return jump instruction.

The information in the RJX Register is transmitted to the SWS at the proper time to construct and store in SCM a called subroutine exit word. The called subroutine exit word is 60 bits long and is formed in the SWB; bits 0-29 are zeros, bits 30-47 are the RJX bits, and bits 48-59 are a constant instruction code of 0400₈. This word is stored in an SCM location determined by K of the return jump instruction and RAS.



SECONDARY BLOCK DIAGRAM FETCH AND ISSUE CONTROL

FETCH CONTROL

The instruction stack consists of the instruction word and address paths on sheet CPU 2.0 and the fetch control circuits on the LO and HK modules on CPU 2.1.

Words are requested for the IWS two words ahead of the word currently being executed. Since there are still at least two words left to be executed in the IWS when a word from IWS Ranks 1 thru 10 is being executed in the CIW Register, no words are requested. However, when the Rank 11 word is gated to the CIW Register, one word must be requested from SCM in order to remain two words ahead. Similarly, when the Rank 12 word is gated to the CIW Register, or under branch conditions, two words must be requested.

The fetch control circuits monitor the CIW Register and instruction stack and fetch one word or two words for the IWS depending upon detected conditions. One word is requested from SCM when coincidence is detected between IAS Word 11 and (P). Two words are fetched when coincidence is detected between IAS Word 12 or when there is no coincidence between (P) and any of the IAS ranks.

COINCIDENCE TEST

The test for coincidence is made in the LO module. When all three bit groups for the same word are coincident with the like bits of the P Register, the three inputs will be logical ones and Word X Coincidence will be generated. Word 11 Coincidence exists when the contents of IAS Rank 11 are the same as (P) and implies that the CIW Register is about to read the next to last word in the IWS. When this word is read to the CIW Register F1F is set to request an instruction word from SCM. Word 12 Coincidence exists when the contents of IAS Rank 12 are the same as (P) and implies that the CIW Register is about to read the last word in the IWS. When this word is read to the CIW Register, both fetch flags (F1F and F2F) are set to request a total of two instruction words from SCM. No Coincidence is generated when no IAS Ranks are coincident with (P) and denotes that a two instruction word fetch is required.

ACCESS CONTROL FLAG AND FETCH TRANSLATORS

The Access Control Flag and Fetch Translators generate the signals outlined below:

1. Fetch One Word Flag (F1F) - F1F sets whenever the instruction stack requires a word from SCM and the request has not yet been honored by the SAS.
2. Fetch Two Words Flag (F2F) - F2F sets whenever the instruction stack requires two words from SCM and neither request has yet been honored by the SAS. F1F is always set when F2F is set.
3. Marker One Word Flag (M1F) - M1F sets whenever the instruction stack requires a word from SCM which has not yet arrived at the IWS.
4. Marker Two Words Flag (M2F) - M2F sets whenever the instruction stack requires two words from SCM which have not yet arrived at the IWS. M1F is always set when M2F is set.
5. Store Exit Flag (SXF) - SXF sets at the same time that RJJF is cleared and continues the sequence for the return jump instruction. It remains set until SCM access control has accepted the address for writing the RJX Register data into SCM. SXF then clears and JCF (CPU 2.4) sets.
6. Jump Out of Stack Flag (JOF) - JOF sets when a branch instruction is executed which requires jumping to a program address not currently held in the IAS.
7. Enable P to NSA-1, -2 - These signals are used in the instruction stack to generate P To NSA. P To NSA signals the arrival of the last fetch word requested prior to a program branch. The NSA Register is then set to the new program sequence address (contents of P Register) and the marker flags are set in anticipation of the instruction words for the new program sequence. It also clears JOF.
8. Return Jump Flag (RJF) - RJF sets at the end of a clock period in which the CIW Register contains a return jump instruction and GJJF is set. This flag remains set until all words requested from SCM for the IWS have arrived at the IWS. SXF then sets and RJF clears.

9. Out of Stack Flag (OSF) - OSF sets whenever the CIW Register is ready for the next word from the IWS and no word is available.
10. Enter IFA-1, -2 - These conditions are ORed together in the IFA Register module to, effectively, become a single condition. This condition occurs on a program branch resulting in a jump out of stack. It clears the IFA registers and fills them with the contents of the P Register and RAS.

ISSUE CONTROL

The instruction issue function is performed by the control circuits on CPU 2.1 and by the CIW Register and associated circuits on sheet CPU 2.0. See CPU 2.0 description for the definition of instruction issue.

A, B, X REGISTER RESERVATIONS

There are eight register busy flags associated with each set of registers - A, B, and X. A flag is set to reserve a specific register when an instruction issues from the CIW Register that will deliver a result to that register. The flag remains set until it is cleared at the same time that the result is actually transmitted to the reserved register. This flag prevents subsequent instructions from reading the contents of this register until the new data has been delivered.

The register busy flags are located on the HC modules. The flag(s) that is set when an instruction issues is determined by a translation of the instruction's i and, in the case of the X and B Registers, j designators and five outputs of the Go Unit Select Translator. See CPU 3.2 for a description of the Go Unit Select Translator outputs.

Before the next instruction in the CIW Register can issue, its operand and result register needs are checked against the register busy flags. Coincidence between an i, j, or k designator and a set register busy flag generates a register busy signal that is used in the Registers Free Translator. The instruction will not issue until the results of the previous instruction are delivered to the reserved/requested register; at this time the appropriate selection bits and clear reservation signal clear the selected register busy flag.

The Registers Quiet signal indicates that no register busy flags are set; no registers are reserved.

REGISTERS FREE

This condition indicates that the operating registers involved in the instruction waiting to issue are now free. That is, there are no register busy flags set for an A, B, or X Register that is either an operand or a destination register for the current instruction in the CIW Register.

GO UNIT CIRCUIT

The go unit circuit consists of the Go Unit Select Translator, Go Unit Flags, and Go Unit Fanout.

Go Unit Select Translator

The Go Unit Select Translator translates the g, h, i, and j designators of the instruction in the upper parcel of the CIW Register into the outputs listed. See CPU 3.2 for a more-detailed description of the Go Unit Select Translator.

The outputs enable setting Go Unit Flags or operating register busy flags (see A, B, X, Register Reservations above) at the end of a clock period in which the instruction issues.

Go Unit Flags

With the exception of LCM Busy and Divide Busy, all Go Unit Flags set at the end of the clock period in which the applicable instruction issues and clear during the following clock period. The LCM Busy and Divide Busy Flags set at the end of the clock period in which the applicable instruction issues and clear when the block copy has been completed, when the LCM word has entered an X Register or LCM Bank Operand Register, or when the Divide Unit is available to perform another divide operation.

The primary function of the Go Unit Flags is to control the execution of all instructions involving functional units. With the exception of the multiply and divide units, data flows continuously thru the functional units. The outputs are discarded, however, unless the applicable Go Unit Flag is set. The Go Unit Flag, or a copy, gates the output of the functional unit to the destination operating register(s). Also, all instructions that execute in two clock periods send a Go 2 Cycle signal to the operating register's access control which selects the correct result register in the following clock period.

The Multiply and Divide Busy Flags control data movement through the respective functional units and block further inputs for one and 17 clock periods respectively following instruction issue.

Go Unit Fanout

The Go Increment To Storage Flag becomes Go Test SCM Direct Range in the Go Unit Fanout. The Divide Busy Flag becomes the lower five fanout outputs. Enter Xk, Enter Exponent, and Divide Busy are complements of the Divide Busy Flag; Release Rank 1, 2 Borrows and Begin Sequence are not inverted.

ENABLE ISSUE TRANSLATOR

Go Exit

This condition denotes the presence of an exchange exit instruction in the current program sequence.

Set GJF

This condition is ANDed with Registers Free and a clock pulse on the 1CC7 module to set the Go Jump Flag.

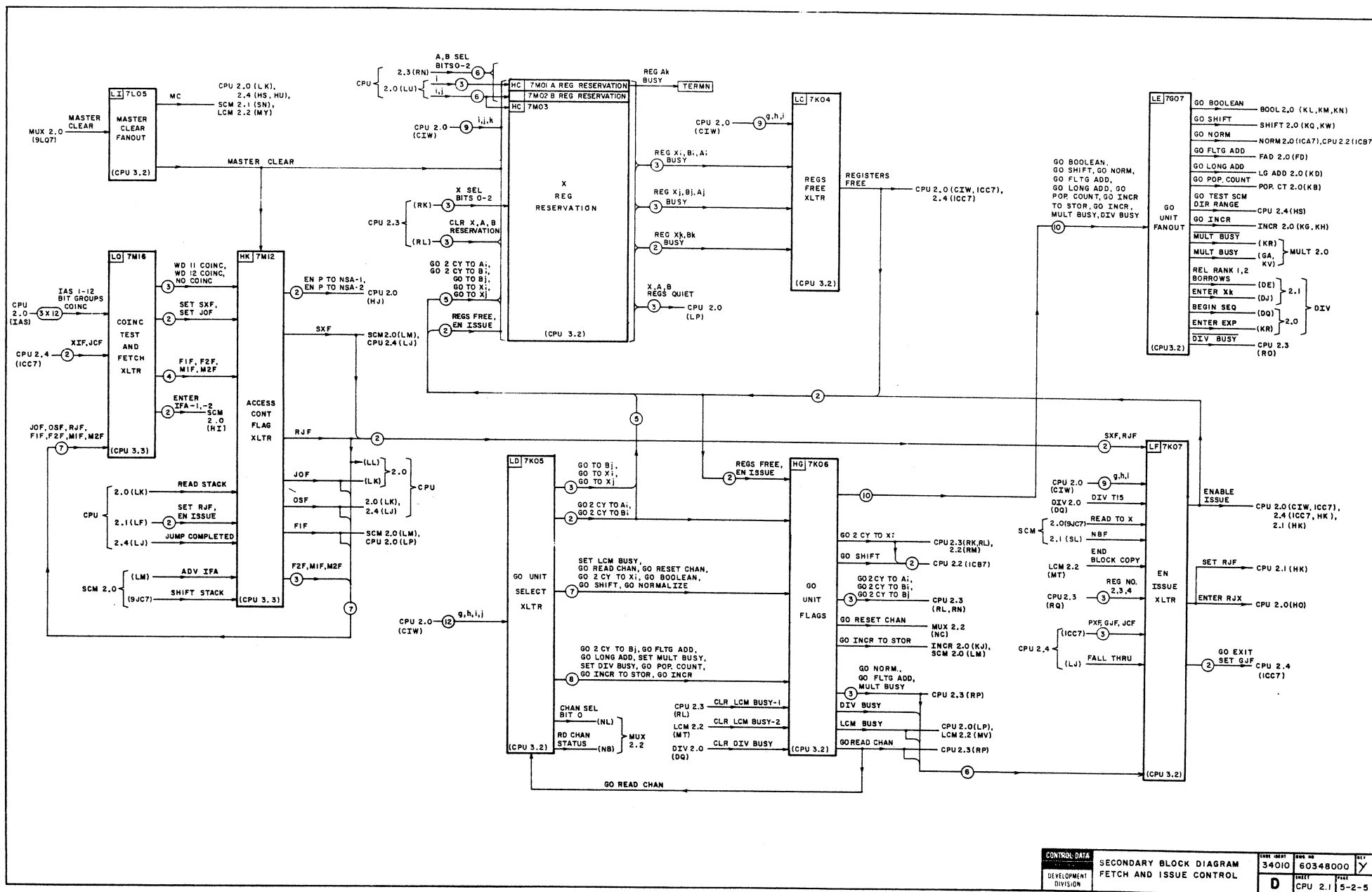
Enter RJX, Set RJF

These conditions are generated when a return jump instruction is ready for execution. Enter RJX gates the P Register contents into the RJX Register. Set RJF is ANDed with a clock pulse on the HK module to set the Return Jump Flag.

Enable Issue

Enable Issue indicates that the functional units, register destination paths, and storage access path are such as to allow issue of the current instruction in the CIW Register. This condition does not take into account the possible register reservation problems.

The Enable Issue translation primarily prevents the conflict that would occur if two or more instructions attempted to deliver results to the X Registers during the same clock period. Thus, an instruction is not allowed to issue if the data to the destination X Register will conflict with data from another functional unit already processing data.



SECONDARY BLOCK DIAGRAM X REGISTERS DATA PATHS

The eight X Registers are the principal operating registers for the CPU. These registers are each 60 bits in length and serve as the source and destination for operands in execution of the arithmetic instructions. Data will remain in an X Register until a control condition generated in the X Register Access Control (CPU 2.3) specifically clears the data and enters new data. Only one X Register can be cleared and entered with new data at the end of any given clock period.

The X Registers receive data from all nine of the functional units and from SCM and LCM. The X Register input control (ICB7 Module) and access control circuits regulate the entry of data into the X Registers. The contents of the X Registers are sent to all nine of the functional units, branch control, SCM, and LCM. The merge and distribution functions are performed in 60-bit static networks preceding and following the X Registers.

X REGISTER INPUT CONTROL TRANSLATOR

The X Register Input Control Translator uses the special case information from the functional units to generate and transmit Overflow, Underflow, or Indefinite signals to the PSD Register for setting the corresponding condition flag. This transmission occurs in the same clock period that data would arrive at the X Register under no special case condition. The complement of the Overflow signal sets bit 59 of the X Register input path to a zero when an Overflow condition exists. This is the first step in generating the special bit pattern of the overflow condition exponent.

The X Register Input Control Translator merges the special case and sign treatment information from the functional units into complement control signals. These signals generate the bit patterns for the special case floating point formats or to complement the entire word when the result from the functional unit should be negative. The signals sent to Complement Control by the input control are: Complement 0 - 47, Complement 48 - 57, and Complement 58, 59.

INPUT TO X NETWORK

Every functional unit has at least one result data path and several of the floating point units have multiple 60-bit result data paths to the X Registers. Merging of the input to X data is performed in 60-bit static networks preceding the X Registers (RC and RD modules).

A functional unit outputs logical ones to the X Registers during all clock periods except those in which a meaningful result is ready for entry into the destination X Register. Issue Control insures that only one of the nine functional units sends a result to the X Registers during any given clock period.

The data flow from the functional units is treated in two groups. One group, consisting of the Multiply, Divide, Shift, and Normalize Units is treated in a static merge network. Data from these units flows thru another network in the RC and RD modules (complement control) before entering the X Registers. The second group, consisting of the remaining functional units and the storage units, merges with the data from the complement control network, in another static merge network. The 60 bits from this last network are delivered to all eight X Registers. The data is entered into the selected register when the Enter X condition is present.

COMPLEMENT CONTROL

The complement control network complements all, part, or none of the 60 bits of input data depending upon the complement control signals received from the X Register Input Control Translator. The complementing network performs the following functions:

1. Complements the Multiply or Divide Unit output if the result coefficient is negative.

2. Complements the Normalize or Shift Unit output if the result is positive. The result received from each of these units is the complement of the true result.
3. Packs a complete overflow, complete underflow, or indefinite quantity into the destination X Register if the corresponding error condition was detected by any of the floating point units during instruction execution in these units.
4. Complements the logical ones input from the Multiply, Divide, Shift, and Normalize Units to zeros during clock periods in which either SCM or LCM delivers data to the X Registers. Succeeding merges of these zero bits with logical ones from the remaining functional units result in zeros from the nine functional units. This allows the entry of correct data from either SCM or LCM.

GO LCM TO X GENERATOR

Go LCM to X is generated during clock periods in which the LCM Ready-2 condition for either banks 0 - 3 or banks 4 - 7 exists and neither Go 2 Cycle To Xi nor Register Number 1 exists. Its presence gates the X Register input quantity from LCM into the X Register input path. The Enter X condition enables entry to the destination X Register. The absence of Go LCM to X blocks the input from LCM, forcing the LCM input to the X Register to zeros.

SELECTION AND DISTRIBUTION

The contents of all eight of the X Registers enter a static distribution network along with the i, j, and k designators from the CIW Register. The distribution network decodes the i, j, and k designators and gates the contents of the selected Xi, Xj and Xk registers to the nine functional units, branch control, SCM, and LCM.

Certain functional units require multiple sign bit inputs. Consequently, bit 59 of both the Xj and Xk quantities are entered into fanout networks that distribute multiple copies of these sign bits.

The Xi and Xk quantities also enter a static selection network along with bit 1 of the h designator. The network gates Xk to the Shift Unit when h1 is a one (22 and 23 instructions). It gates Xi when h1 is a zero (20 and 21 instructions).

The lower 19 bits of the X0 Register are unconditionally sent to LCM.

SECONDARY BLOCK DIAGRAM
A, B, BPA REGISTERS DATA PATHS;
X, A, B REGISTERS ACCESS CONTROL AND SELECTION

A, B, BPA REGISTERS DATA PATHS

A REGISTERS

The eight A Registers are the vehicle for addressing SCM for operands and storing results. The A0 Register is not used in SCM addressing for operands. It holds the starting relative SCM address for block copy instructions and may be used for some indexing functions.

Registers A1 thru A5 are used to address SCM in reading data from SCM to an X Register. A read SCM reference is initiated whenever one of these A Registers is the destination in execution of an increment instruction. The data from the relative SCM address is delivered to the corresponding X Register.

Registers A6 and A7 are used to address SCM in writing data into SCM from an X Register. A write SCM reference is initiated whenever one of these A Registers is the destination in execution of an increment instruction. The data from the corresponding X Register is delivered to the SWS to be written into the relative SCM address specified in the A Register. Such an increment instruction reserves the A Register but not the X Register.

Data remains in an A Register until a control condition generated in access control specifically clears the register and enters new data. Only one A Register may be cleared and entered with data at the end of any given clock period. The control condition which causes this entry is the Enter A condition. The selection of the proper A Register is specified by A Selection Bits 0-2.

Inputs

Data enters the A Registers from the Increment Unit and SCM. Since only one of these sources may transmit data in a given clock period, these data paths are merged without selection in a static network in the RA modules prior to the A Registers. Data enters the selected A Register only when Enter A is present.

Selection And Distribution

The RI modules receive 18 bits of data from each of the A Registers plus three bits from the j designator portion of the CIW Register. The data in the A Register specified by the j designator is delivered to the Increment Unit and SCM each clock period.

60348000 A

The data in the A0 Register is delivered to SCM and LCM each clock period. The A0 Register contains the starting SCM address for block copy instructions. The SCM data path is used for storing the A Register data in the exchange package in SCM during an exchange sequence. The address data for SCM is transmitted directly from the Increment Unit.

B REGISTERS

The B Registers are intended primarily for indexing functions in program execution. The B0 Register does not physically exist in the hardware. In the execution of instructions this register appears to contain all zero bits. Information to be stored in the B0 Register is, in effect, discarded.

Data remains in a B Register until a control condition generated in access control specifically clears the register and enters new data. Only one B Register can be cleared and entered with new data at the end of any given clock period.

Input to B Network

There are five sources of data for entry into the B Registers. All five data paths are merged without selection at the B Register end of the data paths since only one of these sources may transmit data in a given clock period. The data is entered in a selected B Register only when Enter B is present. The selection of the proper B Register is specified by the B Selection Bits 0 - 2.

During an exchange sequence, when the selection bits specify B0, the breakpoint address from SCM is entered into the BPA Register.

Selection and Distribution

The RI modules receive 18 bits of data from each of the B Registers plus the i, j, and k designators from the CIW Register. Each of the three groups of data paths, Bi, Bj, and Bk carries information from a single B Register in any given clock period. The B Register for each group is specified by the i, j, or k designator appropriate for that group.

BPA REGISTER

A program may be executed in small sections during a debugging phase by inserting a breakpoint address in the BPA portion of the program's exchange package in SCM. During an exchange sequence, the BPA from the exchange package in SCM is entered into the BPA Register. The (BPA) are continuously compared with (P). When coincidence occurs Breakpoint Coincidence initiates an exchange sequence.

BPA MERGE

During an exchange sequence, but prior to entering a new value into the BPA Register, the (BPA) are sent to SCM for storage in the exchange package. This occurs when the XSK=0 and XSF signals are received from the exchange sequence control circuits.

X, A, B REGISTERS ACCESS CONTROL AND SELECTION

The X, A, and B Registers Access Control and Selection functions determine the timing and the register selection for each word entered into an X, A, or B Register. In addition, they release the register reservations at the proper time to correspond with information arrival at the X, A, or B Registers.

X REGISTER ACCESS CONTROL

X Register Access Control contains four 4-bit registers (RQ module). Information flows from register number 4 through the other three registers at a rate of one register each clock period. A delay of zero to four clock periods may be obtained depending on which register is entered with the data.

Preceding each register is a data switch (RP module) which chooses between the data from the preceding register in the chain or new data. Data flows from register to register unless a controlling functional unit flag is set.

No two functional units may deliver data to an X Register in the same clock period. The responsibility for avoiding conflicts of this type rests with Issue Control rather than access control. An instruction is not allowed to issue if the data to the destination X Register will conflict with data from another functional unit which is already in process.

When the divide instruction issues from the CIW Register, the i designator is captured in a three-bit register (R0 module). It is retained in the three-bit register for 18 clock periods by Divide Busy. At divide time 15 (Div T15) in the divide sequence, the data in the three-bit register (Divide Tag Bits 0-2) is gated into the X Register access chain. At the same time, Divide T15 becomes the flag that indicates the presence of a valid three-bit code. The four bits then move from register to register and control the data entry into the X Register four clock periods later.

An SCM reference which results in reading a word into a destination X Register is treated the same as a functional unit. Conflicts between an SCM reference and a previously issued divide instruction are prevented in the SAS. An SCM reference is delayed if such a conflict would occur. SCM to X Tag Bits 0-2 are clear/entered into a three-bit register (R0 module) every clock period.

Read To X is generated during increment instructions that cause an SCM storage reference (50-57). It is transmitted from SCM when the X Register destination code is in the three-bit register and gates the three-bit code along with a Flag into register number 4. The four bits control the data entry into the proper X Register four clock periods later.

The i designator changes to a new value when an instruction issues. Since the Go Unit Flag (e. g. Go Normalize) also sets at this time, the i value must be saved or delayed for one clock period in order to get the correct value into an access control register. This function is performed by a delay register on the R0 module. The i Delayed output is also used in the X, A, and B Register selection circuits.

A multiply instruction requires five clock periods for execution. The Multiply Busy Flag sets at the end of the first of these clock periods, and the data is transmitted from the Multiply Unit to the destination X Register in the last clock period. At the end of the second clock period i Delayed is entered into register number 3 along with a Flag. Three clock periods later this code will control the entry of the multiply result into the appropriate X Register.

The operation of the floating add and normalize instructions is similar to multiply except that their execution times are four and three clock periods respectively.

DESTINATION X REGISTER SELECTION

The Destination X Register Selection function consists of circuits on the RK modules and a translator on the RL module. Its purpose is to choose a source of X selection bits and to gate that choice to the X Registers and to Issue Control simultaneously. The X selection bits, gated by Enter X, enable the appropriate X Register input data path and, gated by Clear X Reservation, clear the related register busy flag.

The RK module circuits choose selection bits from among four sources. The contents of register number one are selected whenever the Flag is set in that register. XJC Bits 0-2 are selected when XJ To X is present. The i Delayed bits are selected whenever the Go Two Cycle To Xi flag is set. These three selections will not conflict due to the mutual exclusivity of their origins. If none of the three selections is present, LCM To X Tag Bits 0-2 are selected.

LCM To X Tag Bits 0-2 are the delayed j designator gated by LCM Busy. The LCM Busy flag sets when the 014 instruction issues and clears when the requested word arrives at the X Register. LCM Ready-1 clears the LCM Busy flag and X Register reservations, and enters the data into the X Register.

A REGISTER ACCESS CONTROL

This function is performed by gates on the RN and RL modules. Its purpose is to choose a source of A selection bits and to gate that choice to the A Registers and to Issue Control simultaneously. The A selection bits, gated by Enter A, enable the appropriate A Register input data path and, gated by Clear A Reservation, clear the related register busy flag.

The gates on the RN module choose selection bits from between two sources. The i Delayed bits are selected whenever the Go Two Cycle To Ai flag is set. XJC Delayed Bits 0-2 are selected when XJ To A, B is present.

B REGISTER ACCESS CONTROL

The B Register access control and selection functions are performed by circuits on five modules. B Register Access Control on modules RO, RP, and RQ determines the timing for three-clock period instructions. The circuits on the RN and RL modules provide the selection bits and associated gating signals.

The B Register Access Control consists of two small registers and associated gating. One register provides j Delayed for use in register selection.

The second register is the Bj Access Control Register. This register holds a three-bit register designation code plus a flag to indicate a valid reference. The j Delayed bits enter this register during normalize (24, 25) or read channel (016, 017) instructions.

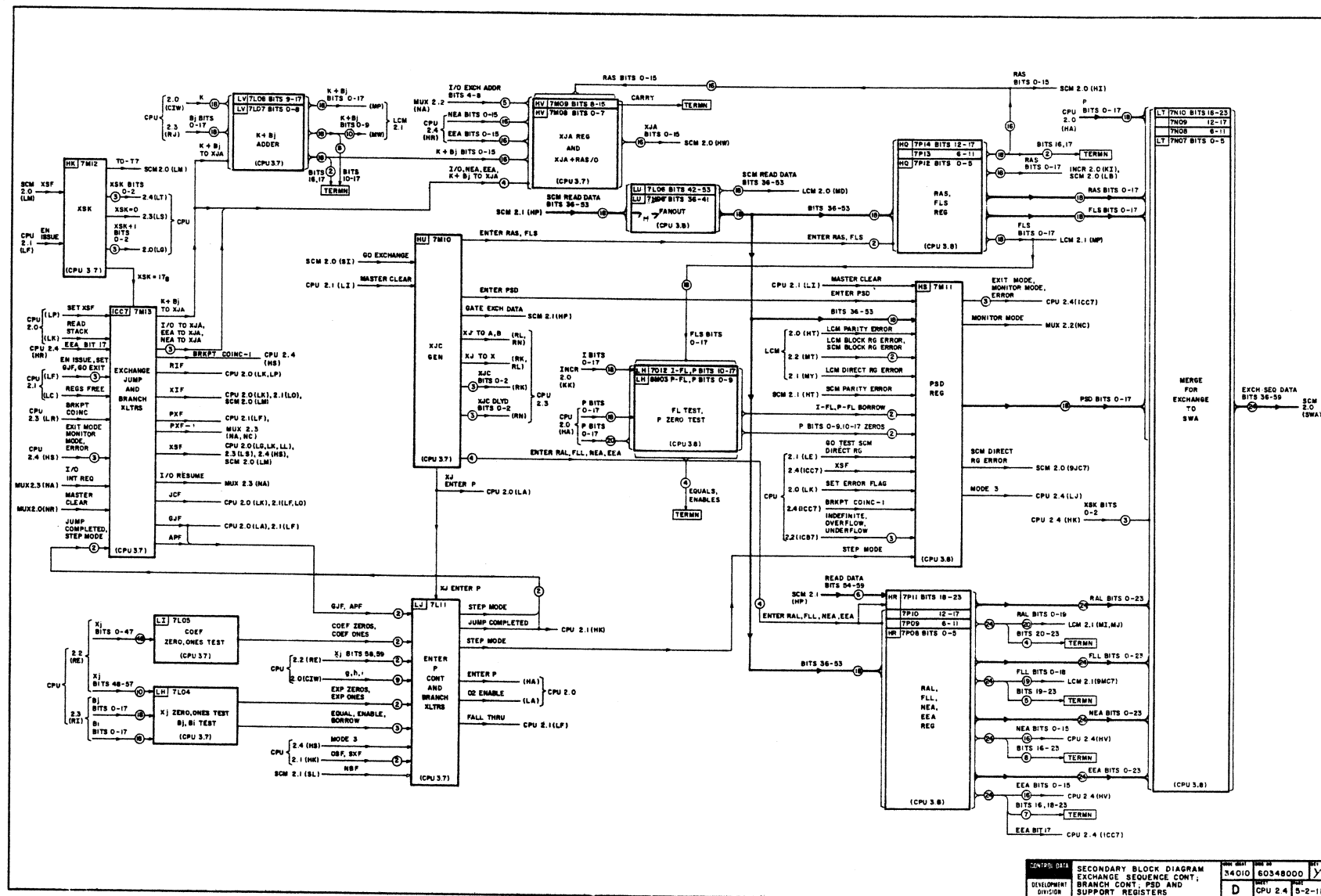
The Normalize Unit is the only functional unit with a three clock period execution time that delivers results to a B Register. If $j \neq 0$, Go Normalize gates j Delayed into the Bj Access Control Register and becomes the Flag that indicates the presence of a valid three-bit code. If $j = 0$, the gating function is inhibited. In this case, the destination B Register is B0 and the data is discarded.

The 016 and 017 instructions also require three clock periods for execution. The Go Read Channel flag sets when the instruction issues. The circuit then operates the same as it does during normalize instructions.

The selection function consists of the gates associated with the B Registers on the RN and RL modules. Its purpose is to choose a source of B selection bits and to gate that choice to the B Registers and to Instruction Issue Control simultaneously. The B selection bits, gated by Enter B, enable the appropriate B Register input data path and, gated by Clear B Reservation, clear the related register busy flag.

The gates on the RN module choose selection bits from among four sources. The i Delayed bits are selected whenever the Go Two Cycle To Bi flag is set. XJC Delayed Bits 0-2 are selected when XJ To A, B is present. 3 Cycle To Bj Bits 0-2 are selected whenever the 3 Cycle To Bj Flag is set. The j Delayed bits are selected whenever the Go Two Cycle To Bj flag is set. These selections will not conflict due to the mutual exclusivity of their origins.

The B0 Register does not exist in the hardware. Selecting B0 actually enables the BPA Register input data path. For this reason, B Selection Bits 0-2 can equal zero only when the exchange package path is selected. Also, Enter B and Clear B Reservation are generated when the i or j designators equal zero only during the exchange sequence.



DETAILED-MODULES DIAGRAM PART OF INSTRUCTION STACK; PART OF BRANCH

INSTRUCTION STACK

The instruction stack consists of the instruction word and address paths on CPU 3.0 and the control circuits on sheet CPU 3.3. The principal components of the instruction stack on CPU 3.0 are the IWS and the IAS. A coincidence test is made between (P) and the contents of each IAS rank during each clock period. Whenever a coincidence occurs, the corresponding 60-bit instruction word in the IWS is sent to the CIW Register. If more than one coincidence occurs, all are sent simultaneously to the CIW Register and the results are merged as a logical sum (A+B).

INSTRUCTION WORD STACK (IWS)

The IWS is a group of twelve 60-bit registers which hold program instruction words for execution. The twelve registers are individually identified by rank. Data moves through the IWS from Rank 12 to Rank 1. Rank 12 is filled with SCM data and all other ranks are filled with data from the next-highest-order rank. The data shifted from Rank 1 is discarded. Data movement through the IWS only takes place during a clock period in which Shift Stack is present. Odd-clock FFs, not shown on the diagram, prevent data flow of more than one rank per clock period.

Shift Stack is generated by the read address entering into SAS and no SCM bank conflict. It is delayed to arrive coincident with the read data arriving at IWS.

INSTRUCTION ADDRESS STACK (IAS)

The IAS is a group of twelve 18-bit registers which hold relative SCM program addresses on a one-for-one basis with the program words in the IWS. Rank 1 in the IAS contains the oldest address in the stack, and this address corresponds to the SCM location from which the word in Rank 1 of the IWS was read. All ranks of the IAS are clear/entered with information from the next-highest-order rank during a Shift Stack clock period in a manner analogous to that for the IWS. An address is read into Rank 12 of the IAS from the NSA Register. This address corresponds to the SCM address for the word arriving at Rank 12 of the IWS. The address in Rank 1 of the IAS is discarded.

The outputs of all IAS ranks are compared with (P) each clock period. Six-bit groups are compared on each HA module. Note that bits 16 and 17 of IAS Rank 12 receive no input and a constant one, respectively. Since this keeps bits 16 and 17 of all ranks at a zero and one, respectively, bits 16 and 17 of the P Register must be zero and one, respectively, to achieve coincidence. The group results enter the HB modules where they are checked for word coincidence. Word X Coincidence gates the IWS Rank X contents to the CIW Register.

During exchange and return jump instructions, the IAS is cleared. Since coincidence cannot occur until the IWS is loaded with a new sequence of instructions, the IWS is, effectively, void and no further instructions can enter the CIW Register during this interval.

Operation

The NSA (Next Stack Address) Register, adder, Advance Stack Flag (ASF) and associated input circuits generate the NSA. The NSA Register is a clear/enter type register with gated clock pulse control. ASF is a clear/enter bit register which is set at the end of a clock period in which Shift Stack is present. The sole purpose of this flag is to advance the count in the NSA Register in the clock period following the Shift Stack condition. The NSA is a 16-bit SCM relative address for the next sequential word required by the IWS.

The two possible sources of data for the NSA Register are the P Register and the NSA Loop. The contents of the P Register enter at the end of a clock period in which a P To NSA condition exists. The contents of the NSA Register plus one enter via the NSA Loop at the end of a clock period in which ASF is set. When both conditions exist in the same clock period, the P To NSA condition takes priority. When neither condition exists, the same value constantly reenters the NSA Register via the NSA Loop.

P To NSA signals the arrival of the last fetch word requested prior to a program branch. Thus, all IWS words have arrived from SCM, their corresponding addresses have entered into the IAS, and the branch address can now enter the NSA Register.

Address

When Shift Stack is present and ASF is not set, the address entered into IAS Rank 12 is a copy of the contents of the NSA Register. When Shift Stack is present and ASF is set, the address entered into IAS Rank 12 is (NSA) plus one. Generally, the address that enters IAS also loops back to enter the NSA Register during the same clock period. The only exception to this is when P To NSA and ASF occur in the same clock period. This causes (P) to enter the NSA Register as (NSA) plus one enters IAS Rank 12.

BRANCH

Branch instructions are executed in a portion of the CPU which is not well defined as a functional unit. This portion of the CPU contains the program address register (P) and those control flags required to sequence the entry of a new address into the P Register as well as the register (RJX) and control flags required to execute the return jump instruction.

The P and RJX Registers are located on this sheet but the associated control flags are shown on several other sheets. A general discussion of the branch functions, both normal and return jump, and detailed discussions of the P and RJX Registers follow. For detailed information on the control flags, refer to the text for the sheet on which the flag is shown.

NORMAL BRANCH EXECUTION

A normal branch instruction, as distinguished from a return jump instruction, has three cases to consider in execution timing. One case occurs when the jump criterion is not met in a conditional branch instruction. This is referred to as the fall thru case.

A second case, called branch in stack, occurs when the jump is taken and the destination address is currently within the IAS. A third case, called branch out of stack, occurs when the jump is taken out of the range of the instruction stack. Each of these cases has a separate timing sequence.

All normal branch instructions begin execution with the setting of the GJF (CPU 3.2, 3.7). This occurs when the conditions specified by the instruction have been resolved.

These conditions involve register data which may be in process in functional units as a result of previously issued instructions. The conditions for setting GJF are therefore similar to the conditions for issue of a computation instruction. GJF allows the branch instruction to issue from the CIW Register only in the fall thru case. The other two cases involve further sequence control flags.

A branch instruction in which the jump criterion is met requires the entry of a new program sequence address into the P Register. The P Register is cleared and entered with a new address at the end of the clock period in which GJF is set. JCF (CPU 3.7) is set at the same time to indicate that the jump has been completed. The branch instruction then issues from the CIW Register in the following clock period. If the new program address is in the IAS, the next instruction word may be read directly into the CIW Register in this clock period. If the new program address is not in the IAS, an SCM reference must be initiated to read the new program instruction word. This process is a function of the Instruction Stack - Fetch Control (CPU 3.3).

RETURN JUMP EXECUTION

Execution of a return jump instruction begins with the setting of GJF in a manner similar to that required for a normal branch instruction. GJF causes two register actions in the clock period in which it is set. The contents of the P Register are cleared and the new program sequence address is entered. This is the K portion of the return jump instruction and becomes the address of the exit word for the called subroutine. Also during this clock period, the previous address in the P Register is copied into the RJX Register. This is the address of the instruction word currently in the CIW Register plus one. RJF (CPU 3.2, 3.3) is set at the end of this clock period to continue the sequence.

RJF is a delay mechanism to allow the completion of any SCM instruction word references. RJF statically clears all ranks of the IAS for as long as it is set. When all instruction words previously requested have arrived at the IWS, RJF clears and SXF (CPU 3.3) is set to continue the sequence. The contents of the P Register are transmitted to the IFA Register during the last clock period in which RJF is set.

SXF causes the SCM reference to store the special word required for the called sub-routine exit. The data for this word is in the RJX Register. The address for the SCM reference is in the P Register and in the IFA Register. SXF remains set until the SCM reference has been initiated by the acceptance of (IFA) in the SAS. SXF then clears and JCF is set to complete the sequence. The contents of the P Register are advanced one count during the last clock period in which SXF is set.

JCF allows the return jump instruction to issue from the CIW Register. This flag is set for only one clock period. During this clock period the P Register contains the entrance address (K+1) for the called subroutine. The IFA Register contains this same address. No coincidence is possible between (P) and the IAS because of the previous clearing of the IAS. The instruction stack fetch control will therefore initiate the SCM references to read up the beginning of the called subroutine.

P REGISTER

The P Register is an 18-bit register which contains the current program execution address. This register is a clear/enter-type register with gated clock pulse control. The current contents of the P Register are cleared and new data is entered at the end of a clock period in which Enter P exists.

The P Register enters into a large number of functions so it is physically reproduced six times (one-third of a P Register on each HA module) to allow the proper fanout of information. All hardware copies of the P Register are entered with the same data in the same clock period and under the same conditions.

Inputs

There are four possible sources of data for the P Register (LA modules). The selection of one of these is made on the basis of three control conditions. Each condition specifies one of the data paths while the absence of all three selects the fourth data path. Read Data Bits 36-53 enter the P Register from the SRO Register in SCM as a part of the exchange package. K Bits 0-17 enter the P Register during conditional branch and return jump sequences. K+Bi enters during the unconditional jump instruction (02). P+1 enters during straight-line coding and increases (P) by one count in the clock period following the entry of a new word into the CIW Register.

Outputs

See the IAS paragraphs above for discussions on two of the P Register outputs.

P Bits 0-17 plus two Bit 17s from the adjacent P Register Fanout are tested for zero on CPU 3.8. When (P) equal zero, the Program Range Condition Flag sets generating an error exit request. Under branch conditions, P Bits 0-15 enter the IFA Register in SCM where they are added to RAS to form the fetch address of an instruction word. The contents of the P Register become the return jump exit (RJX) address during the return jump sequence. (See RJX Register below.) The contents of P are subtracted from the SCM field length (FLS) in the LH module. When $(P) \geq FLS$, the SCM Direct Range Condition Flag sets generating an error exit request. This terminates the execution interval for the exchange package at the end of the current instruction word. The P Register contents are compared with the breakpoint address in the LR module. If coincidence occurs and the exchange sequence flag (XSF) is clear an error exit request is generated. P bits 0-17 enter the LT module to become bits 36-53 in the first word of the exchange package.

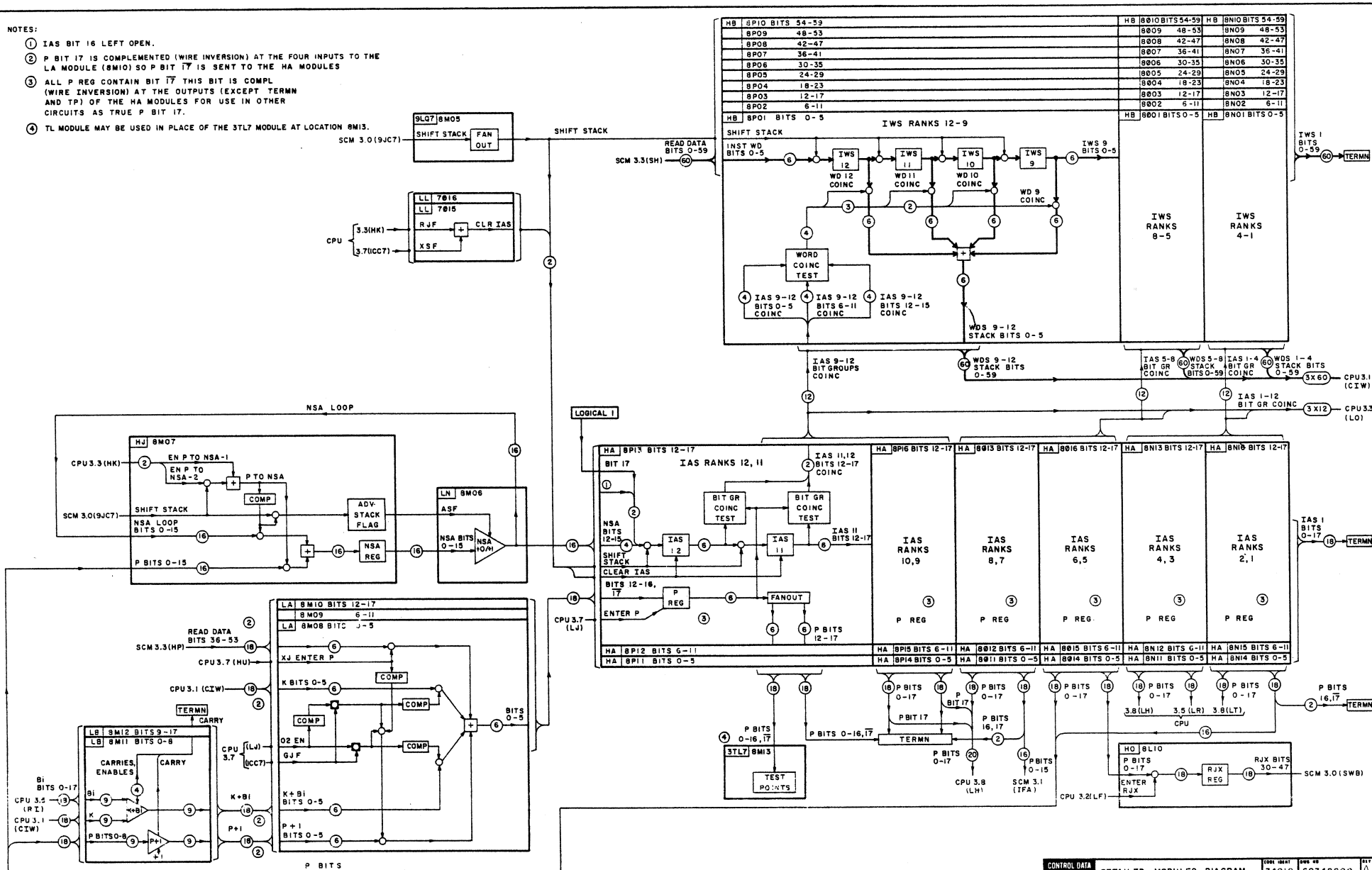
RETURN JUMP EXIT (RJE) REGISTER

The RJX Register is an 18-bit clear/enter type register with gated clock pulse control. It holds the return jump exit address during execution of the return jump instruction. The current contents of the P Register (CIW address plus one) enter the RJX Register at the end of a clock period in which Enter RJX exists and remain there until the execution of the next return jump instruction.

The information in the RJX Register is transmitted to the SWS at the proper time to construct and store in SCM a called subroutine exit word. The called subroutine exit word is 60 bits long and is formed in the SWB; bits 0-29 are zeros, bits 30-47 are the RJX bits, and bits 48-59 are a constant instruction code of 0400_g. This word is stored in an SCM location determined by K of the return jump instruction and RAS.

NOTES:

- ① IAS BIT 16 LEFT OPEN.
- ② P BIT 17 IS COMPLEMENTED (WIRE INVERSION) AT THE FOUR INPUTS TO THE LA MODULE (8M10) SO P BIT 17 IS SENT TO THE HA MODULES
- ③ ALL P REG CONTAIN BIT 17 THIS BIT IS COMPL (WIRE INVERSION) AT THE OUTPUTS (EXCEPT TERMN AND TP) OF THE HA MODULES FOR USE IN OTHER CIRCUITS AS TRUE P BIT 17.
- ④ TL MODULE MAY BE USED IN PLACE OF THE 3TL7 MODULE AT LOCATION 8M13.



DETAILED-MODULES DIAGRAM PART OF INSTRUCTION ISSUE

The instruction issue function is performed by the CIW Register and associated circuits on sheet CPU 3.1 and the additional control circuits on CPU 3.2.

Program instruction words are read one at a time from the Instruction Word Stack into the Current Instruction Word (CIW) Register for execution. Each instruction word is divided into four 15-bit parcels as described in the System Reference Manual. There may be as many as four instructions in the CIW Register at one time. These instructions must be executed in sequence and the proper allowance made for the mixture of one-parcel and two-parcel instruction formats. Instruction format is described in the Reference Manual.

An instruction issues from the CIW Register when the conditions in the functional units and operating registers are such that the functions required in the instruction execution may be performed to completion without conflicting with a previously issued instruction. This is indicated in the hardware by the presence of Go Issue which is comprised of Enable Issue and Registers Free. The issue process requires one clock period. During this clock period the g, h, i, j, and k designators in the instruction are sent to all appropriate parts of the CPU. The proper control flags are set at the end of this clock period to execute the functions required in the instruction. Also, at the end of the clock period, the data in the CIW Register is altered to position the next instruction for execution. Since there is no further record of the instruction once it issues from the CIW Register, it must be executed to completion in a fixed-time framework. No delays are allowable from issue to delivery of data to the destination operating registers.

CURRENT INSTRUCTION WORD (CIW) REGISTER

OPERATION

The CIW Register is physically located on the HD, HE, and HF modules. Due to fanout considerations, the upper 15 bits are distributed over a total of 11 modules, nine HD and two HE modules, while the lower 45 bits are located on just five HF modules.

The CIW Register is divided into four 15-bit parcels. All four parcels are loaded in one clock period when an instruction word is read from the IWS. The highest-order parcel in the instruction word is issued first. The other parcels are then shifted left in the CIW Register by either 15 bits or 30 bits, depending on the instruction format for the instruction issued. The lower-order parcels are replaced with zeros as the data is shifted left in the register.

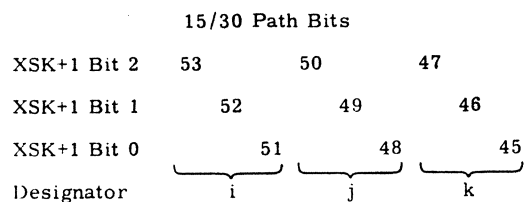
The CIW Register is cleared and new data enters at the end of a clock period in which Registers Free and Enable Issue exist. The presence of either Sample IWS or Gate 15/30 Path determines whether the input data is from the IWS or from the lower parcels of the CIW Register. The absence of both gates during Go Issue enters a 60-bit word of zeros into the register.

IWS PATH

Data from the IWS path is selected whenever Sample IWS is present. If coincidence occurs between (P) and IAS, an instruction word enters the CIW Register; otherwise, a blank word - all zeros - enters the register. Sample IWS exists whenever Read Stack is present (see Translators paragraph below) except during the part of the exchange sequence when XIF is present. XIF inhibits the IWS path when the 15/30 path is entering the i, j, and k designators into the CIW Register that read the A, B, and X Registers into the exchange package.

15/30 PATH

Data enters from the 15/30 path whenever Gate 15/30 Path is present. Circuits on the LG modules determine which data is entered. It will be either the CIW shifted 15 or 30 bits to the left, with vacated lower parcels filled with zeros, or XSK+1. The translator in the LG modules uses the g, h, and i designators to discover the number of parcels in the issuing instruction and determine the subsequent shift. The g, h, and i designators are zeros during the exchange sequence when XSF is set allowing the XSK+1 bits to enter the CIW Register as follows:



Gate 15/30 Path gates the shifted CIW to the input of the CIW Register after determining that there is at least one instruction left in the word and that the current instruction is not a branch. JCF prevents the shift following return jump (010) and branch (02-07) instructions.

Neither 15 Bit Instruction nor 30 Bit Instruction is enabled during an exchange exit instruction (013). Since the IWS path is also inhibited at this time, zeros fill the CIW Register. The zero value of the g and h designators is used to generate various exchange sequence control signals such as XSF and XIF.

Gate 15/30 Path is generated by XSF so it can gate the XSK+1 bits to the input of the i, j, and k portions of the CIW Register.

PARCEL COUNT

The parcel count reflects the number of parcels in the CIW that have been issued. The parcel counting function is performed by the Parcel Counter Register (PKR) on the ICC7 module and a translator on the LK module. The (PKR) change each time an instruction issues by an amount equal to the number of parcels in the issuing instruction. When the last parcel in the CIW issues, the PKR is cleared by Set PK 0 and Set PK 1. It is also cleared prior to new instruction sequences by Dead Start, XSF, or JCF.

TRANSLATORS

The following concerns those translator outputs not discussed above.

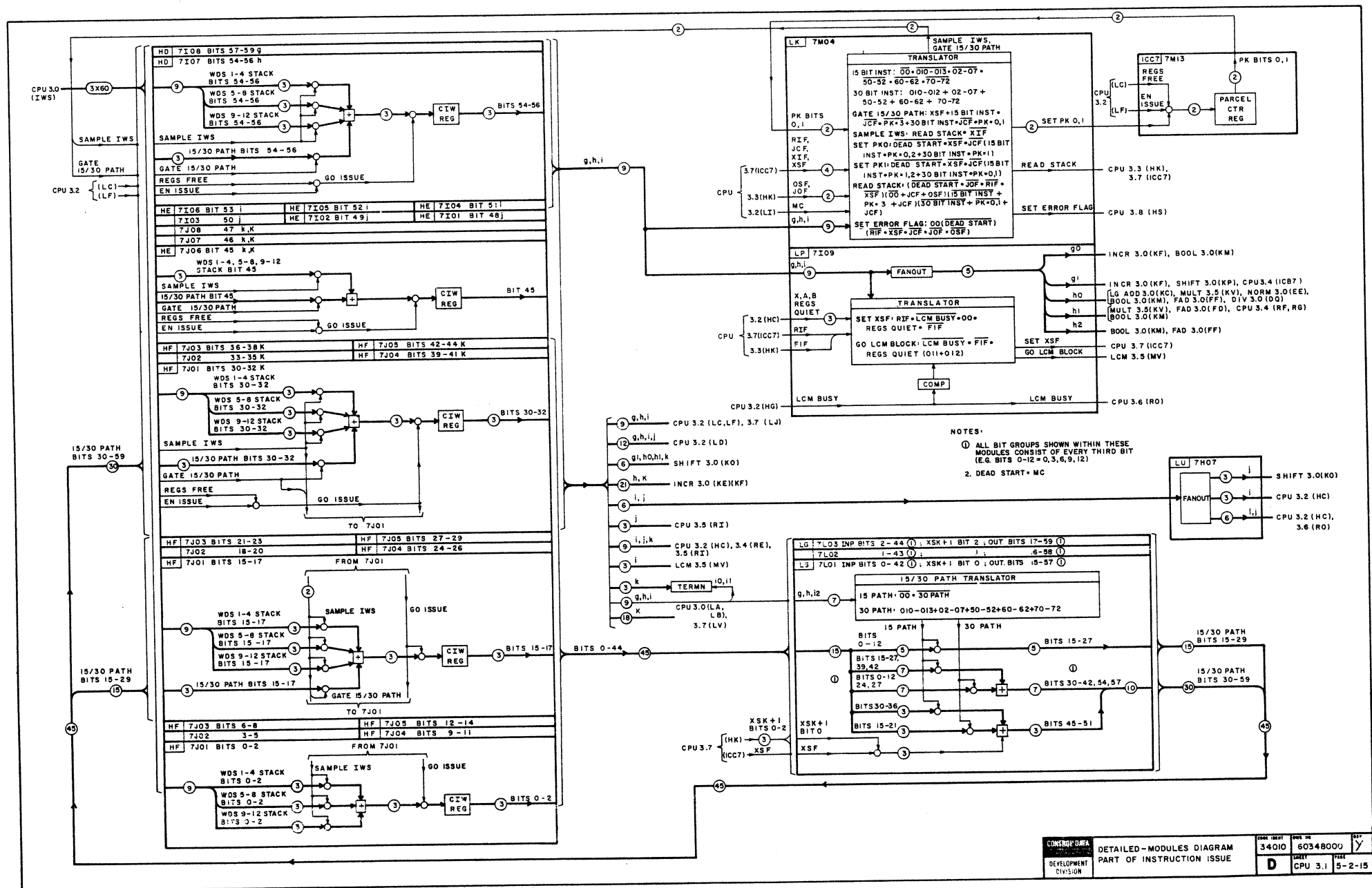
Read Stack indicates that the instruction currently positioned in the upper parcel of the CIW Register is the last instruction in the word. When this instruction issues, the CIW Register will be loaded with a new 60-bit word from the IWS.

Set Error Flag sets the Program Range Condition Flag in the PSD Register when an error exit code (00) issues from the CIW Register. It is inhibited during the dead start, exchange sequence, and branch out of stack conditions when the CIW Register normally contains zeros.

Set XSF indicates that the execution of the current program instruction word is completed. It sets the Exchange Sequence Flag to continue the exchange sequence. It is generated as RIF clears when the following conditions are satisfied:

1. LCM is not busy.
2. All previously issued instructions have delivered their results to the operating registers.
3. No SCM bank conflicts occurred that prevented an instruction fetch address from leaving the IFA Register.

Go LCM Block sets the Go LCM Block Flag in LCM Control. It initiates an LCM block copy and is generated during the 011 and 012 instructions when the conditions listed under Set XSF above are satisfied.



DETAILED-MODULES DIAGRAM INSTRUCTION ISSUE CONTROL

The instruction issue function is performed by the control circuits on CPU 3.2 and by the CIW Register and associated circuits on sheet CPU 3.1. See CPU 3.1 description for the definition of instruction issue.

A, B, X REGISTER RESERVATIONS

There are eight register busy flags associated with each set of registers - A, B, and X. These flags are bit registers with separate set and clear inputs which are forced clear on a dead start. A flag is set to reserve a specific register when an instruction issues from the CIW Register that will deliver a result to that register. The flag remains set until it is cleared at the same time that the result is actually transmitted to the reserved register. This flag prevents subsequent instructions from reading the contents of this register until the new data has been delivered.

The register busy flags are located on the HC modules. The flag(s) that is set when an instruction issues is determined by a translation of the instruction's i and, in the case of the X and B Registers, j designators and five outputs of the Go Unit Select Translator.

Referring to the lower portion of the Go Unit Select Translator, it can be seen that the 5X instructions, with an i designator value of one through five, generate Go 2 Cycle to Ai and Go To Xi so an A and an X Register must be reserved for these instructions. This is necessary because a read from SCM to register Xi is also provided by these instructions. Go To Bj and Go To Xi are generated by 24-26 instructions so B and X Registers must be reserved. This is done because the B Register receives the shift count and the X Register receives the normalized result. All other instructions that reserve a result register will reserve only one X, A, or B Register.

Before the next instruction in the CIW Register can issue, its operand and result register needs are checked against the register busy flags. Coincidence between the outputs of the i, j, and k sense translators and a set register busy flag generates a register busy signal that is used in the Registers Free Translator. The instruction will not issue until the results of the previous instruction are delivered to the reserved/requested register; at this time the appropriate selection bits and clear reservation signal clear the selected register busy flag.

The Registers Quiet signal indicates that no register busy flags are set; no registers are reserved.

REGISTERS FREE

This condition indicates that the operating registers involved in the instruction waiting to issue are now free. That is, there are no register busy flags set for an A, B, or X Register that is either an operand or a destination register for the current instruction in the CIW Register.

The Registers Free Translator, as shown on the diagram, consists of the Registers Free translation and eight intermediate translations. The intermediate translations, provided to make the Registers Free translation less cumbersome, are enabled by the g, h, and i designators of the instruction in the upper parcel of the CIW Register. These are compared in the Registers Free translation with the outputs of the operating registers' reservation circuits. At least one signal within each parentheses must be as shown to enable Registers Free. For example, a 22 instruction generates XI and BJ. Therefore, registers Xi and Bj must be free (Xi Busy, Bj Busy) to generate Registers Free. All other conditions will be met, even if other registers are busy, since no other intermediate translations will be enabled.

GO UNIT CIRCUIT

The go unit circuit consists of the Go Unit Select Translator, Go Unit Flags, Go Unit Fanout, Multiply Busy Fanout, and Divide Busy Fanout.

GO UNIT SELECT TRANSLATOR

The Go Unit Select Translator translates the g, h, i, and j designators of the instruction in the upper parcel of the CIW Register into the outputs listed. The outputs are generated only when the input designators satisfy the conditions in the first three columns of the translator table. For instance, Go To Xi is not enabled by 5X instructions with i designator values equal to zero, six, or seven but is enabled by 5X instructions with i designator values of one through five.

The outputs enable setting Go Unit Flags or operating register busy flags (see A, B, X, Register Reservations above) at the end of a clock period in which the instruction issues.

GO UNIT FLAGS

With the exception of LCM Busy and Divide Busy, all Go Unit Flags are clear/enter bit registers with gated clock pulse control. A clear/enter flag sets at the end of the clock period in which the applicable instruction issues and clears during the following clock period. The LCM Busy and Divide Busy Flags are bit registers with separate set and clear inputs. These flags set at the end of the clock period in which the applicable instruction issues and clear when the block copy has been completed, when the LCM word has entered an X Register or LCM Bank Operand Register, or when the divide unit is available to perform another divide operation.

The primary function of the Go Unit Flags is to control the execution of all instructions involving functional units. With the exception of the multiply and divide units, data flows continuously thru the functional units. The outputs are discarded, however, unless the applicable Go Unit Flag is set. The Go Unit Flag, or a copy, gates the output of the functional unit to the destination operating register(s). Also, all instructions that execute in two clock periods send a Go 2 Cycle signal to the operating register's access control which selects the correct result register in the following clock period.

The Multiply and Divide Busy Flags control data movement through the respective functional units and block further inputs for one and 17 clock periods respectively following instruction issue.

FANOUTS

The Go Increment To Storage Flag becomes Go Test SCM Direct Range in the Go Unit Fanout. The Divide Busy Flag becomes the Divide Busy Fanout outputs shown. Enter Xk, Enter Exponent, and Divide Busy are complements of the Divide Busy Flag; Release Rank 1, 2 Borrows and Begin Sequence are not inverted.

ENABLE ISSUE TRANSLATOR

GO EXIT

This condition denotes the presence of an exchange exit instruction in the current program sequence.

SET GJF

This condition is ANDed with Registers Free and a clock pulse on the ICC7 module to set the Go Jump Flag. See GJF discussion in the CPU 3.7 text.

ENTER RJX, SET RJF

These conditions are generated when a return jump instruction is ready for execution. Enter RJX gates the P Register contents into the RJX Register. Set RJF is ANDed with a clock pulse on the HK module to set the Return Jump Flag. See the CPU 3.0 and 3.3 text for further discussion.

ENABLE ISSUE

Enable Issue indicates that the functional units, register destination paths, and storage access path are such as to allow issue of the current instruction in the CIW Register. This condition does not take into account the possible register reservation problems (See Registers Free above.)

The Enable Issue translation primarily prevents the conflict that would occur if two or more instructions attempted to deliver results to the X Registers during the same clock period. Thus, an instruction is not allowed to issue if the data to the destination X Register will conflict with data from another functional unit already processing data. For example, the normalize instructions (24+25) are three clock period instructions so they are prevented from issuing when an instruction has previously issued which has three clock periods remaining before delivering its results to an X Register or when a floating add instruction (four clock periods) has issued in the preceding clock period.

The 014 instruction (Read LCM to Xj) generates Enable Issue with no restrictions (other than NBF) since the data from LCM is repeated each clock period until the X Register input path is free.

Block 5 Cycle prevents the five clock period multiply instructions (40-42) from issuing when the divide unit results or an SCM word are within five clock periods of entering an X Register.

The 5X instructions generate Enable Issue with no restrictions (other than NBF) since a conflict between an SCM reference and a previously issued divide instruction is resolved in SCM Storage Address Stack control.

7600 MULTIPLICATION

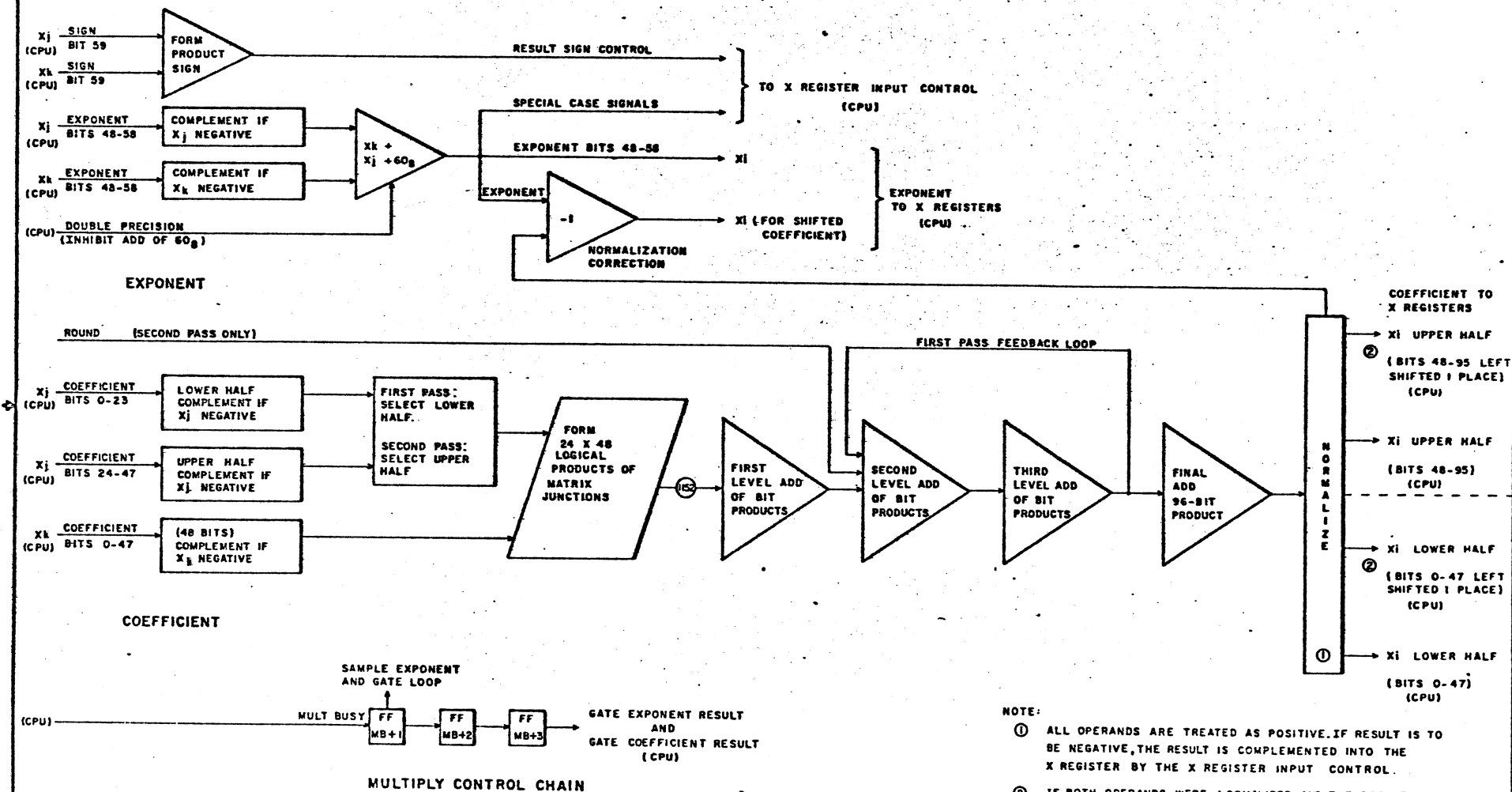
The multiply unit is constructed as a pipeline which in this case has 5 stages, each stage requiring one minor cycle for completion. As can be seen from the timing diagram, each of the three adders is in use for 2 stages, therefore inputs to the pipeline can only occur every two minor cycles, i.e. once every 55 n.sec. Each multiplication takes $5 \times 27.5 = 137.5$ n.sec. but multiplications can be completed at a peak rate of one every 55 n.sec.

The attached notes give a summary of the mantissa arithmetic. The exponent arithmetic is quite straightforward and is done in parallel without having any effect on the overall timing.

N.B. The stages of the carry save adders take less than 27.5 n.sec.

Indication of Sequence of Events

1. Set everything up
2. $X_{jL} * X_k$ and first level add
of 1152 product bits (Adder 1)
3. $X_{jU} * X_k$ and first level add
of 1152 product bits (Adder 1) Second and third level adds of
bits from $X_{jL} * X_k$ (Adders 2 & 3)
4. Second and third level adds of
 $X_{jU} * X_k$ and bits from second
and third level adds for $X_{jL} * X_k$
(Adders 2 & 3)
5. Final Add



DEVELOPMENT DIVISION	PRIMARY BLOCK DIAGRAM MULTIPLY UNIT		EQUIPMENT	
	SIZE	DRAWING NO.	REV	
	C	60348000	A	
	SHEET	MULT 1.0	PAGE	5-7-1

PRIMARY BLOCK DIAGRAM MULTIPLY UNIT

The floating multiply unit executes the following three instructions:

- 40 Floating product of (X_j) times (X_k) to X_i
- 41 Rounded floating product of (X_j) times (X_k) to X_i
- 42 Floating double precision product of (X_j) times (X_k) to X_i

The X_i coefficient and X_i exponent are formed separately with the proper consideration taken for single or double precision. If single precision is specified, the upper half of the coefficient result and an exponent (corrected for single precision) are sent to the X register. Otherwise, for double precision, the lower half of the coefficient result and the exponent result are sent to the X register.

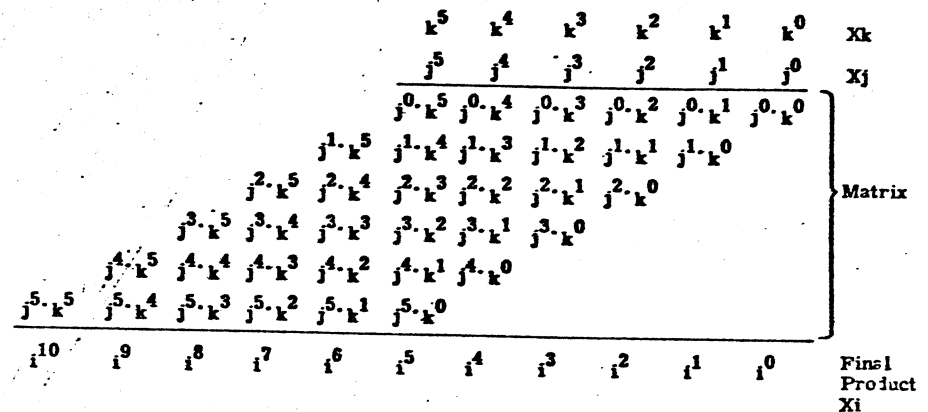
If both operands are normalized, the result will be normalized. If rounding is specified, a round bit is generated and added to bit 47 of the coefficient result.

The multiply instructions require five clock periods for execution. Multiply instructions may enter the multiply unit every other clock period. Thus, during a five-clock-period segment, more than one multiply instruction may be active in the unit.

COEFFICIENT MULTIPLY

Coefficient multiply is basically the same as a multiply done with paper and pencil. Every bit of the multiplier is multiplied times every bit of the multiplicand. In binary arithmetic this is the AND (logical product) function.

Just as in pencil and paper multiplication the bit-by-bit products are arranged in a matrix as shown below with each row offset by one bit. The columns of the matrix are then added to obtain the final product, X_i .



The multiply is performed in two steps. First the lower 24 bits of X_j are multiplied times all 48 bits of X_k and the columns of the resulting matrix are partially added. Then the upper 24 bits of X_j are multiplied times all 48 bits of X_k and the resulting matrix is added to the partial sums and partial carries from the first pass to form the 96-bit double precision product. Each step in the process involves forming 1152 binary products. These 1152 bits of data must then be added in the proper groupings to form a combined sum.

First Pass

Second Pass	k^5	k^4	k^3	k^2	k^1	k^0					
				j^5	j^4	j^3					
	$j^3 \cdot k^5$	$j^3 \cdot k^4$	$j^3 \cdot k^3$	$j^3 \cdot k^2$	$j^3 \cdot k^1$	$j^3 \cdot k^0$					
	$j^4 \cdot k^5$	$j^4 \cdot k^4$	$j^4 \cdot k^3$	$j^4 \cdot k^2$	$j^4 \cdot k^1$	$j^4 \cdot k^0$					
$j^5 \cdot k^5$	$j^5 \cdot k^4$	$j^5 \cdot k^3$	$j^5 \cdot k^2$	$j^5 \cdot k^1$	$j^5 \cdot k^0$						
ps10	ps9	ps8	ps7	ps6	ps5	ps4	ps3	Results of First Level Add ps2 ps1 ps0 } Add Loop Bits from First Pass			
pc10	pc9	pc8	pc7	pc6	pc5						
			ps7	ps6	ps5	ps4	ps3				
			pc7	pc6	pc5	pc4	pc3	pc2			
ps10	ps9	ps8	ps7	ps6	ps5	ps4	ps3	ps2	ps1	ps0	Results of Second and Third Level Adds }
pc10	pc9	pc8	pc7	pc6	pc5	pc4	pc3				
i^{10}	i^9	i^8	i^7	i^6	i^5	i^4	i^3	i^2	i^1	i^0	Final Add

60343000 Λ

The Carry Save Adder used in the Floating Multiply Unit permits the addition of columns of numbers without using the time consuming carry and satisfy checks typical of full adders. Instead, each level of add has two outputs-pseudo sums and pseudo carries, defined as follows:

(Pseudo carries are displaced one bit to the left because they affect the next significant bit position).

Thus, the inputs to the First Level Add are the logical products produced by the matrix. The inputs to the Second Level Add are the pseudo sums and pseudo carries from the First Level Add and the inputs to the Third Level Add are the pseudo sums and pseudo carries from the Second Level Add. The output from the Third Level Add is a pseudo sum and pseudo carry bit for each bit position of the product.

After the first pass, the output from the Third Level Add is shifted right 24 places, fed back into the Second Level Add, and merged with the results of the second pass through the matrix. The Second and Third Level Adds are then repeated for the entire product until a pseudo sum and a pseudo carry bit again represent each bit position of the product. This time the output of the Third Level Add is fed to the Final Add network and is reduced to one bit for each bit position of the final product (96 bits).