SYSTEMS CONCEPTS

520 THIRD STREET SAN FRANCISCO, CALIFORNIA 94107

SYSTEMS CONCEPTS

SA-10 USER'S MANUAL

Serial Numbers 107 and Above

TELEPHONE: 415-442-1500 TWX: 910-372-6062

TABLE OF CONTENTS

(revised March 1, 1975)

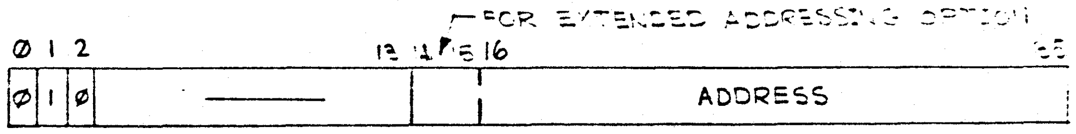
SA-10 USER'S MANUAL

Introduction

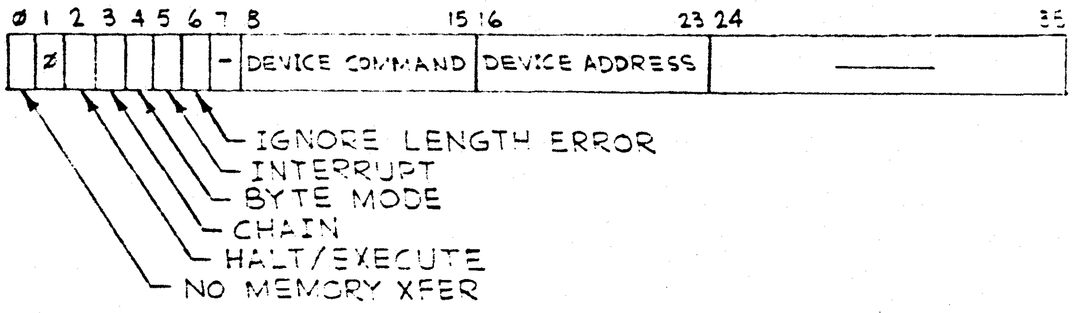
The Systems Concepts SA-10 IBM-to-DEC Subsystem Adapter connects to a PDP-10 input-output bus and memory bus and to any peripheral devices which can be connected to a standard IBM System-360 selector channel, and it enables the PDP-10 to run these devices as if they were connected to a standard IBM System/360 selector channel. One SA-10 simulates either two or four selector channels. To run the SA-10, the PDP-10 program compiles a channel program which contains commands to be sent to device controllers and pointers to data areas to be read or written. Then the PDP-10 starts up the SA-10 with commands sent over the input-output bus, whereupon the SA-10 proceeds to read the channel program and transfer data over the memory bus without further CPU intervention until the end of the channel program is reached, or an error occurs, or one of the instructions in the channel program requests that the CPU be interrupted.

Channel Programs

There are three types of words in channel programs: device command words, data chain words, and transfers in channel. The format of a transfer in channel is:



When the channel executes a transfer in channel, it transfers to the location specified in the address field and continues executing channel commands from there. A device command word has the following format:



First bit 2 (the "halt/execute" bit) is examined to determine whether this command should be executed at all; if the bit is 0, this is the end of the current channel program and the controller should halt, store status, and interrupt the CPU. If this bit is a one, the channel proceeds.

Next the channel examines bit 4, the "byte mode" bit, to determine whether the data to be transferred (if any) will be entire thirty-six bit words, nine 8-bit bytes per two words, or thirty-two bit words with four eight-bit bytes per word. In this second case, the "word count" in the data chain words that follow (if any) is actually a byte count.

The channel next examines bit 0 to see whether any data is to be transferred to or from memory in the course of executing this command. "Data" here includes arguments to seek commands, search commands, and so forth, in addition to "data" in the usual sense of information to actually be stored on or retrieved from the device. If there is to be data transferred, bit 0 will be a 0 and one or more data chain words will follow this command to specify where the data is to come from or go to in memory.

The data transfer, if any, now happens. The direction of the transfer is determined from the low bit of the command being sent to the device (0=read, 1=write). Aside from this, the contents of the command do not directly affect channel operation. If a length error occurs during the transfer, the channel examines bit 6 of the device command word, the "ignore wrong length" bit, to see whether it should ignore the error (the bit is 1) or complain about it (the bit is 0). If it chooses to complain, it does so by storing status and interrupting. If some other error occurs, the channel complains unconditionally.

At the conclusion of the data transfer, the channel examines bit 3 of the device command word to see whether the command just executed was to be chained to another device command or not. The channel sends this information along to the device, which may expect certain commands to have been preceeded by certain other commands in the same chain, or <u>not</u> to have been preceded by certain other commands in the same chain. Also, the channel uses this bit to decide whether to continue executing the channel program without delay (when command chaining is not specified) or to first wait for the device to send a status byte which contains Device End, indicating that the device is completely finished with the operation started by the command just executed, and to further examine this status byte to see if it contains Status Modifier, in which case the channel is to skip over whatever command follows the one just executed. (Typically, the command just executed will be something like "search ID equal" on a 2314 disk, which will return "status modifier" if the ID of the record which is just beginning is the same as the ID sent from the memory of the processor, and the command which will be skipped in case will be a transfer in channel back to the "search ID equal" command. Thus, the channel would stay in a loop executing these two words until the ID of the record about to be read was the same as the ID of the record desired, whereupon the program would leave the loop and probably transfer the record.)

Last, the channel examines bit 5 of the device command word, the "interrupt" bit. If it is on, the channel stores some status and interrupts the CPU. The status which is stored will have the "program interrupt flag" bit set.

The format of data chain words, which are used in conjunction with device command words as described above, is as follows:

The sign bit specifies whether further data chain words follow this one (1 means this is the last one). The word count specifies the 2's complement of the number of words to be transferred (or, in byte mode, the 2's complement of the number of bytes to be transferred) and the address field specifies the address of the first word to be transferred. If the address field is entirely zero and this data chain word pertains to a read operation (the data is being transferred into memory), the data should be ignored and the contents of memory should not be modified. The word count field tells how much data to ignore.

Let us consider now an example of data transfer from a 2314 disk to memory. In the simplest case, three commands must be sent to the disk controller: a seek to position the head assembly to the correct cylinder and select the correct head, a search to find the correct portion of the track (this command must be repeatedly executed until the right sector comes along), and a read data to transfer the data from the disk to memory. The first two of these commands require data to be sent to the disk controller. For the seek command, the controller uses the information in the data sent out to tell the disk where the head assembly should be and to specify which head should be listened to. For the search, the controller compares the information coming from memory with the information coming from the disk to see if the disk is in the correct portion of its rotation. A simple channel program to execute these commands would contain eight words:

Device command word: memory transfer, execute, command chain, byte mode, "seek" command code, desired device address.

Data chain word: no further data chain words, 6 bytes to be transferred (i.e. word count =-6), address of the 6 bytes to be sent.

Device command word: memory transfer, execute, command chain, byte mode, "search ID equal" command code, desired device address.

Data chain word: no further data chain words, 5 bytes to be transferred, address of the 5 bytes to be sent.

Transfer in channel back to the above device command word.

Device command word: memory transfer, execute, no command chain, word mode, "read data" command code, desired device address.

Data chain word: no further data chain words, expected size in words of the record, first address where record should be put.

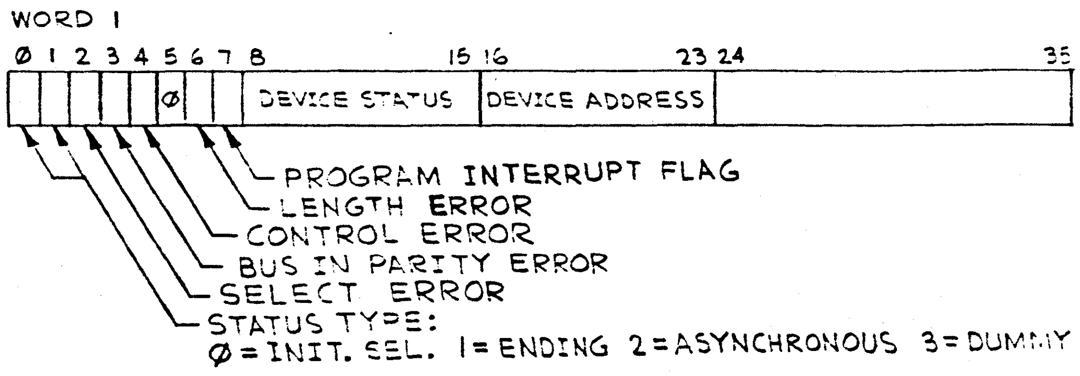
Device command word: halt.

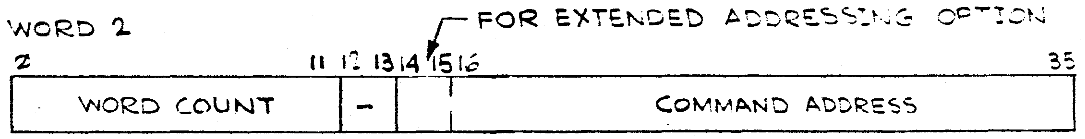
The program must also ahead of time put the data to be supplied for the seek and search commands into the places specified in the channel program.

Status

As used in connection with IBM-compatible peripherals, a status byte is a piece of information about the state of a device transmitted from the device to the channel when the channel initiates communication to the device to send it a command, when the channel finishes communicating with a device after the command has been sent, or when the device decides on its own that something interesting has happened. In the last case, the channel is not obligated to accept the status byte from the device if it doesn't want to. In the SA-10, the channel will accept a status byte which the device transmits on its own initiative only when it is idle, and this sort of status is referred to as asynchronous status.

The first type of status is called initial selection status and the second is called ending status. The bits of the status byte have names which are independent of what type of status the byte is, but the exact meaning of a bit is dependent upon not only what other bits in the status byte are on but also upon when the status byte is generated. The names of the status bits, reading from left to right, are Attention*, Status Modifier, Control Unit End*, Busy*, Channel End, Device End, Unit Check*, and Unit Exception*. The bits with asterisks denote conditions which are of potential interest to the PDP-10 program, and whenever a status byte with any of those bits on is presented to the channel, the channel stores the status byte in memory and stops processing the channel program. Any asynchronous status presented to the channel will be stored in memory. Initial selection status, if all is well, should either be entirely zero or have Channel End on and Busy off. In the second case, the command just sent to the device was an immediate command which requred no data transfer and which has already been completely executed, and this initial selection status is treated as if it had been ending status. Other values of initial selection status indicate conditions which the program should know about and are stored in memory. Ending status will be stored if it has interesting bits on, or if the channel detected some error during the execution of the command, or if the device command word in the channel program had the Interrupt bit on, or if the next device command word in the channel program says to halt. In certain circumstances, upon transmitting a piece of status information to the channel, the device forgets the information. Therefore, means must exist to ensure that status information does not get forgotten once it is stored in memory. Associated with each channel in the SA-10 is a status flag which is set whenever the channel stores status information in memory and which the channel expects the program to clear once it has examined the status and decided what to do about it. When the channel stores status, it first checks to see that the status flag is currently off. If it is on, it waits for it to go off. It then stores two words:





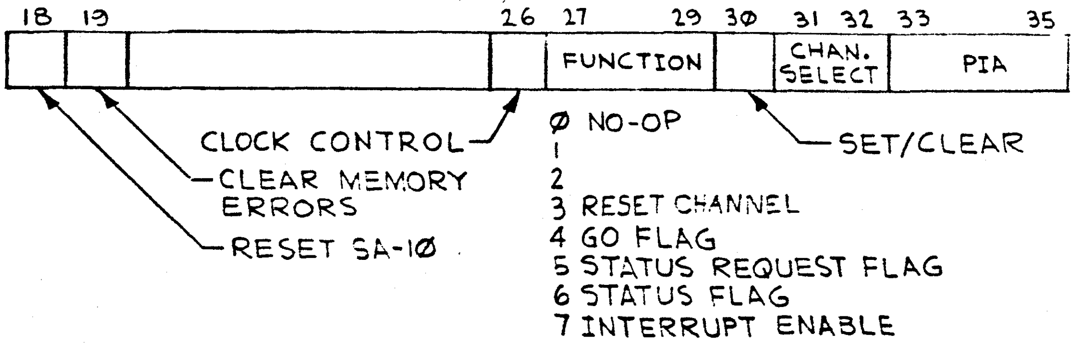
at the base address plus four times the channel number plus one and plus two. After storing the status, the channel will wait for the program to clear the status flag before returning to its normal idle state unless this status was stored as a result of the Interrupt bit being on in a device command word in a channel program, in which case after storing the status the channel continues executing the channel program.

Also associated with each channel is a priority interrupt enable flag, a go flag, and a status request flag. A channel will request an interrupt on the channel which is assigned to the SA-10 as a whole whenever its priority interrupt enable flag and status flag are both on and under no other circumstances. All four of the flags associated with each channel can be directly set or cleared by the PDP-10 program, so it would seem that if the PDP-10 program wants to cause a channel to request an interrupt, it need merely turn on the status flag and priority interrupt enable flag for that channel, and this will in fact cause an interrupt. However, this also creates timing problems because the status flag will be on without the channel having stored any status information in memory (except it's possible that the channel did store status just about the same time that the program turned on the status flag). To avoid this, the program should never turn on the status flag (though it is expected to clear it after examining the status that was stored). Instead, there is a status request flag which the program can turn on. If the channel is idle when the status request flag is turned on, the channel will generate some dummy status, store it in memory, and turn on the status flag and thereby cause an interrupt (if it is enabled and the SA-10 has a channel). The channel itself clears the status request flag whenever it stores status in memory. If the channel is running at the time the status request flag is turned on, sooner or later it is going to store some sort of status in memory of its own accord, thereby causing the interrupt which the program desired when it turned on the status request flag.

As for the other information in the status stored in memory, the Select Error bit means that no device responded to an attempt to select the device address given in the Device Address field; the Device Status field is meaningless. The Bus In Parity Error bit means that some byte if information transmitted to the channel by the device had bad parity. The Control Error bit means that some unexpected sequence of control signals which the channel could not deal with was received. The Length Error bit signifies that the amount of data which the channel program told the channel to transfer was either more or less than the amount the device expected. The Program Interrupt bit indicates that this status was stored as the result of the Interrupt bit having been on in a device command word in the channel program; furthermore, the status will be error-free ending status. The Device Status and Device Address fields give the most recent values of these numbers, which should be considered in the light of the previously described bits. The Word Count field gives the present value of the word count and the Command Address points to the device command word in the channel program which follows the one which caused the error. There may be one or more data chain words between the bad device command word and the one the Command Address field points at. If there is only one, the information in the Word Count field of the stored status and the Word Count and Address fields of the data chain word can be used to determine what word of memory was about to be transferred at the time the status was stored.

I/O Bus Commands to the Channel

The only instructions used in normal operation of the SA-10 are CONO and CONI (CONSO, CONSZ). The formal of the CONO is:



When the go flag is turned on, if the channel is idle, it picks up the contents of the base address plus the channel number times four and executes it. This word will typically be a transfer in channel to the real channel program, which will be elsewhere. Once the channel has started running, it will continue until that channel or the entire SA-10 is reset, or until an error occurs, or until it hits a halt in the channel program. Once having stored status, the channel will do no further pro essing until the program clears the status flag. If the program sets the go flag before clearing the status flag, the channel will begin executing the new channel program in preference to storing status in response to the status request flag, which in turn will be done in preference to accepting and storing asynchronous status from some device. When a device reports that is is unhappy, in many cases the program must issue a sense command to find out why before issuing any other commands to that device, and it is a programming convenience to be able to set up a channel program to read back a device's sense data and to execute that program before the channel does anything else. To do this, do not clear the status flag until after examining the status which has been stored, and if you decide to issue a sense command, set up that channel program and turn on the go flag and then clear the status flag.

The operation of status and status request flags is described more fully under Status. The program should never turn on the status flag and it never has to turn off the status request flag.

The priority interrupt enable flag must be on for a particular channel to be able to request an interrupt. (Of course, the SA-10 must also be assigned to some channel and the interrupt system and that interrupt channel must be turned on.)

A particular channel in the SA-10 may be reset, in which case it will stop doing whatever it is doing (if anything), send out a System

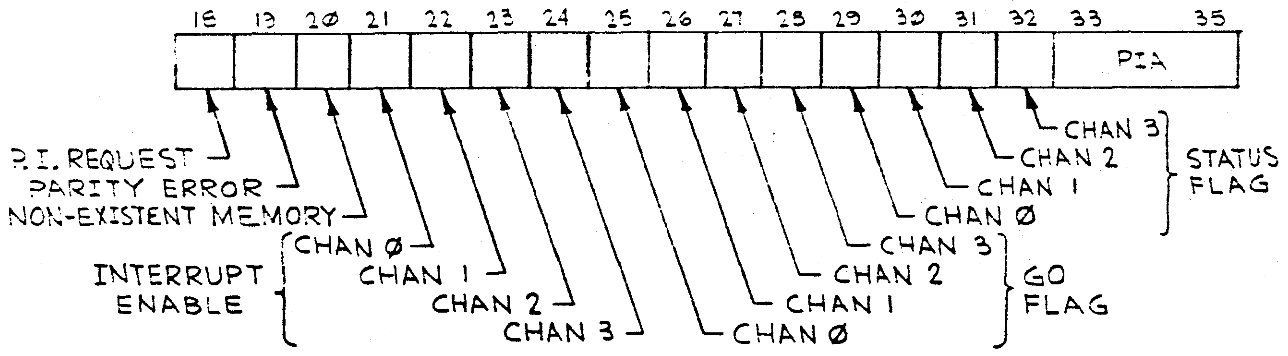
Reset signal to all the devices which are connected to it, and then be in its normal idle, wherein it waits for its go flag or status flag to be turned on or for some device to present asynchronous status.

Also, the entire SA-10 may be reset, which is equivalent to resetting each of its channels, clearing the memory error flags, and starting the SA-10 clock.

If the clock control bit is on, bit 30 is also examined. If it is on, the SA-10 clock is started if it is not running. If bit 30 is off and the clock is running, it is stopped. If bit 30 is off and the clock is stopped, one clock pulse is generated. The clock must be running for normal operation of the SA-10.

The clear memory error bit clears the non-existent memory flag and the parity error flag and has no other effect.

The format of the halfword read by CONI, CONSO, AND CONSZ is:



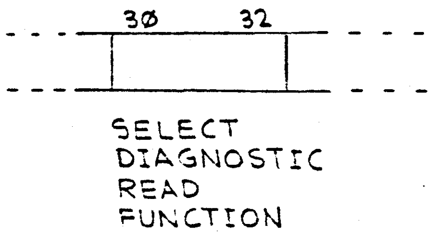
The priority interrupt request bit means that the SA-10 is requesting an interrupt for some reason, (assuming the SA-10 is assigned to an active priority interrupt channel): either the non-existent memory flag, or the parity error flag, or some channel's status flag and priority interrupt enable flag are on.

The non-existent memory flag and parity error flag mean, respectively, that some channel attempted to reference non-existent memory and that a word was read from memory with bad parity. If either of these bits is on, the SA-10 will perform no further memory cycles until they are cleared. Any channel which subsequently tries to access memory (and also the channel which ariginated the erring reference, if it was a read) will hang until the memory error flags are cleared. The channel which originated the erring reference can be determined as described under Diagnostic Features.

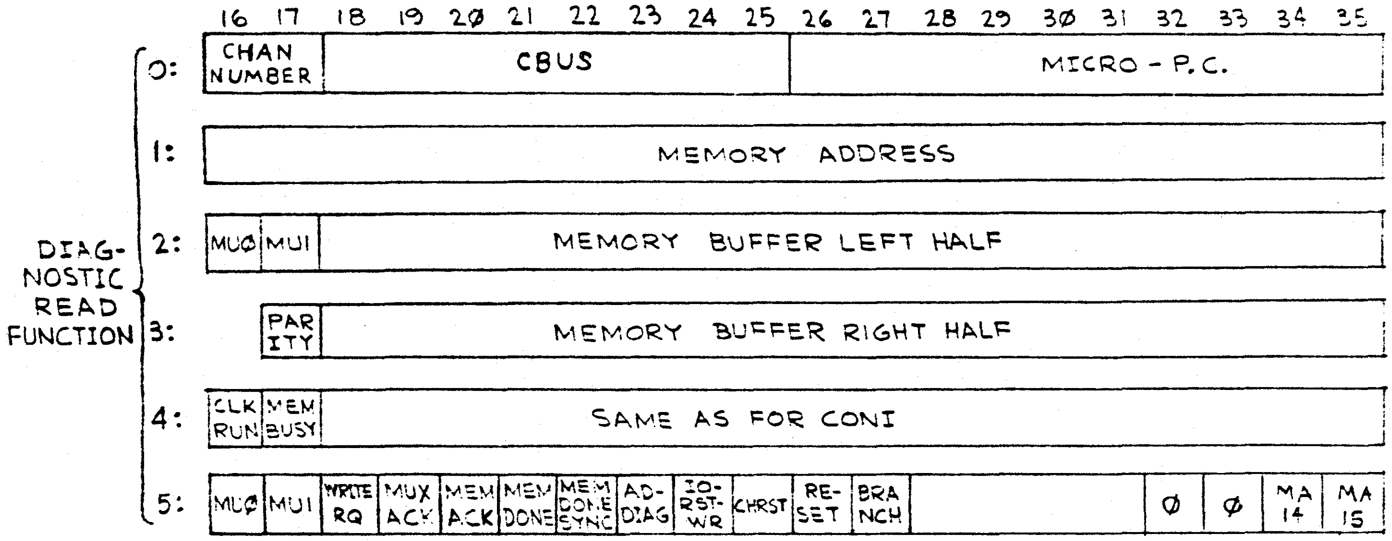
The priority interrupt enable, go and status flags for each channel are all directly available in the conditions. The low three bits give the number of the priority interrupt channel to which the SA-10 is assigned.

Diagnostic Features

Various registers and busses in the SA-10 can be read with DATAI. The things which can be read are divided into six groups; which of these groups will be read is selected by a CONO before the DATAI:



The formats of the words read in by DATAI is:



Words Ø and 5 will read back as garbage while the SA-10 clock is running, but words 1 through 4 can be read at any time. If the SA-10 performs a reference to non-existent memory or detects a word with bad parity in memory, the number of the channel originating the memory reference is in the memory user bits, MUØ and MU1, the address is in the Memory Address, and the word read or about to be stored is in the memory buffer. The parity bit shows the bit read from memory on a read but is always zero on a write.

The significance of the other information read in by DATAI is explained in the SA-10 Maintenance Manual.

DATAO is used to set the contents of the SA-10's micro-instruction register. This instruction must not be executed while the SA-10 clock is running. For an explanation of the format of SA-10 micro-instructions, refer to the SA-10 Maintenance Manual.

SA10-A ADDITIONS

Block Multiplexor Mode

The Block Multiplexor Mode provides the ability to overlap data transfers and device waiting periods, such as seeks and rotational delays, in a nearly optimal manner with little program overhead.

In this mode the CPU maintains a separate command list for each device. When the channel encounters a wait condition when processing a command, such as a seek requiring arm motion, the channel saves the address of the next command and enters a wait loop. When a device completes execution of a function, the corresponding command list is reactivated at the saved address. The CPU is interrupted only when the entire list has been processed or if an error occurs.

The CPU can specify the Block Multiplexor Mode for each channel by setting bit 0 in the corresponding base address word. This word now points to a device list instead of a command list. The device list contains one entry for each device, in one of the formats shown below.

0 7	8 11	12 13	14	16 35	
Device Addr	1101	x		Command Addr	Start Device
Device Addr	1110	x		Command Addr	Waiting On Device
Device Addr	1111	X		Command Addr	Terminated Device
X	0000	X		X	End Of List

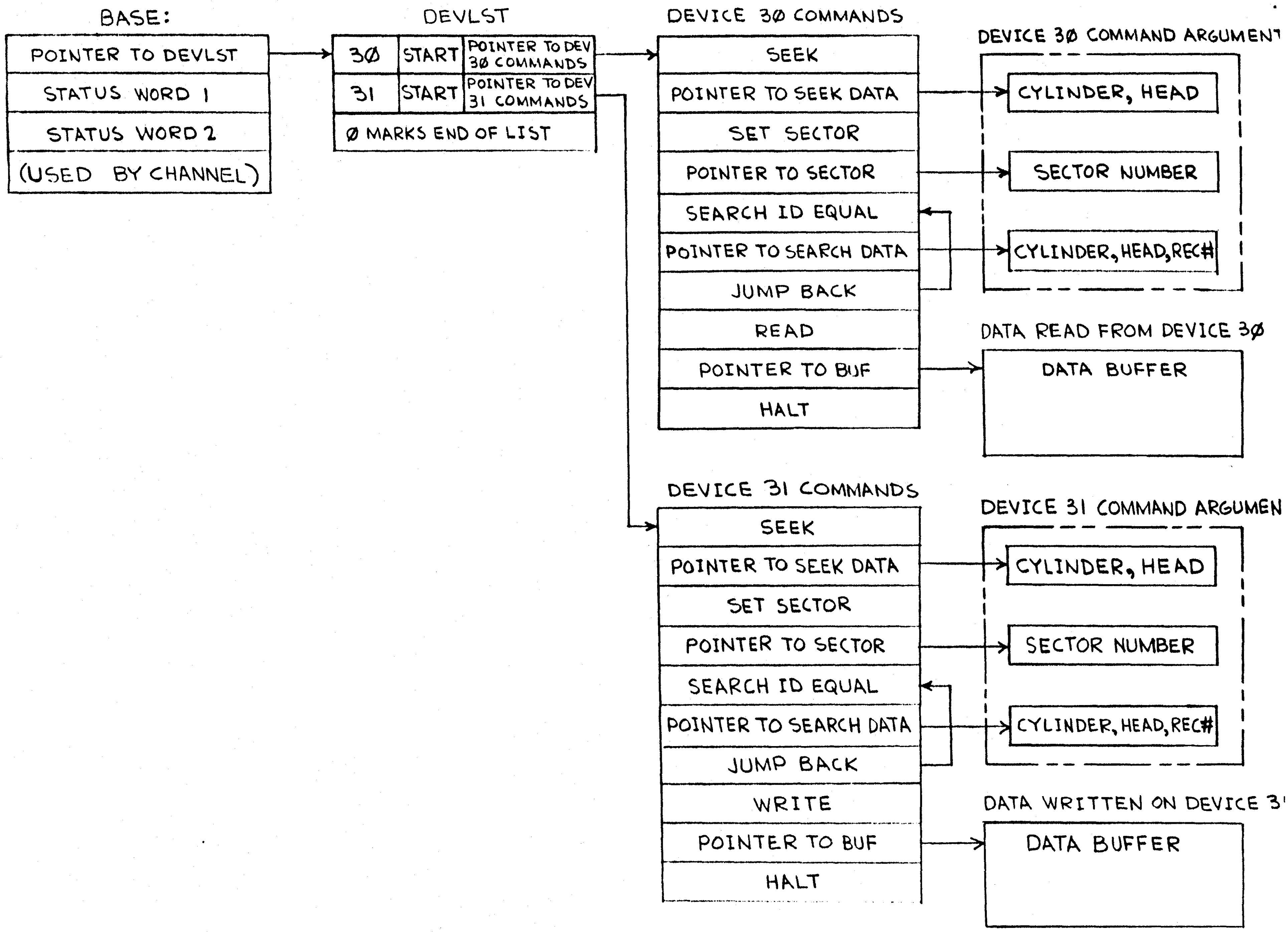
for Extended Addressing Option

The CPU enters a "Start Device" word in the device list and sets the GO flag. The channel responds when idle by scanning the device list for a start entry. When one is found, command execution begins at the address specified. If a chained command is executed for which a channel end is received without a device end, the device list entry is rewritten to the "wait" state with the command address field pointing to the next command to be executed. The search of the device list is resumed (since the GO flag may have been set more than once) until another "start" or the end of list is found. When asynchronous status is received while in the idle loop, if it contains an error or no device end it is stored and reported as usual. Otherwise the device list is searched for a word in the "wait" state with a matching device address field. If found, command processing is resumed, otherwise the status is stored and reported. If an error or halt is encountered while processing commands, the device list entry is set to the "terminated" state. In addition, if the high order (sign) bit is on in the halt command, the channel will not wait for the status flag to be cleared before proceeding to the idle loop. This permits processing commands for another device concurrently with interrupt service.

Example

Assume two 3330-type disk drives, device addresses 30 and 31. It is desired to read a record from device 30, cylinder 1, head 3, record 2 into INBUF, and write OUTBUF onto device 31, cylinder 5, head 6, record 4. The CPU sets up data as shown below and sets GO.

```
BASE:   BYTE (12)6000(24)DEVLST    ;POINTER TO DEVLST IN BLOCK MUX. MODE
STW1:   0                          ;STATUS WORD 1 STORED HERE
STW2:   0                          ;STATUS WORD 2
STW3:   0                          ;USED BY CHANNEL

...

DEVLST: BYTE (8)30(4)15(24)D30LST  ;START, POINTER TO DEV 30 COMMANDS
        BYTE (8)31(4)15(24)D31LST  ;START, POINTER TO DEV 31 COMMANDS
        0                          ;END OF LIST MARKER

...

D30LST: BYTE (8)70,7,30            ;SEEK
        BYTE (12)-6(24)SEEK30      ;POINTER TO SIX BYTE SEEK ARG
        BYTE (8)70,43,30           ;SET SECTOR
        BYTE (12)-1(24)SECT30      ;POINTER TO SECTOR
D30LUP: BYTE (8)70,61,30           ;SEARCH ID EQUAL
        BYTE (12)-5(24)SRCH30      ;POINTER TO FIVE BYTE SEARCH ARG
        BYTE (12)2000(24)D30LUP    ;LOOP IF WRONG RECORD
        BYTE (8)40,6,30            ;READ DATA (WORD MODE)
        BYTE (12)-RECLEN(24)INBUF  ;POINTER TO BUFFER
D30HLT: BYTE (8)200                ;HALT WITHOUT HANGING

SEEK30: BYTE (8)0,0,0,1,0,3
SECT30: BYTE (8)40
SRCH30: BYTE (8)0,1,0,3,2

INBUF:  BLOCK RECLEN

...

D31LST: BYTE (8)70,7,31            ;SEEK
        BYTE (12)-6(24)SEEK31      ;POINTER TO SIX BYTE SEEK ARG
        BYTE (8)70,43,31           ;SET SECTOR
        BYTE (12)-1(24)SECT31      ;POINTER TO SECTOR
D31LUP: BYTE (8)70,61,31           ;SEARCH ID EQUAL
        BYTE (12)-5(24)SRCH31      ;POINTER TO FIVE BYTE SEARCH ARG
        BYTE (12)2000(24)D31LUP    ;LOOP IF WRONG RECORD
        BYTE (8)40,5,31            ;WRITE DATA (WORD MODE)
        BYTE (12)-RECLEN(24)OUTBUF ;POINTER TO BUFFER
D31HLT: BYTE (8)200                ;HALT WITHOUT HANGING

SEEK31: BYTE (8)0,0,0,5,0,6
SECT31: BYTE (8)140
SRCH31: BYTE (8)0,5,0,6,4

OUTBUF: BLOCK RECLEN
```


BASE:
POINTER TO DEVLST
STATUS WORD 1
STATUS WORD 2
(USED BY CHANNEL)
DEVLST
30
START
POINTER TO DEV 3Ø COMMANDS
31
START
POINTER TO DEV 31 COMMANDS
Ø MARKS END OF LIST
DEVICE 3Ø COMMANDS
SEEK
POINTER TO SEEK DATA
SET SECTOR
POINTER TO SECTOR
SEARCH ID EQUAL
POINTER TO SEARCH DATA
JUMP BACK
READ
POINTER TO BUF
HALT
DEVICE 3Ø COMMAND ARGUMENT
CYLINDER, HEAD
SECTOR NUMBER
CYLINDER, HEAD, REC#
DATA READ FROM DEVICE 3Ø
DATA BUFFER
DEVICE 31 COMMANDS
SEEK
POINTER TO SEEK DATA
SET SECTOR
POINTER TO SECTOR
SEARCH ID EQUAL
POINTER TO SEARCH DATA
JUMP BACK
WRITE
POINTER TO BUF
HALT
DEVICE 31 COMMAND ARGUMEN
CYLINDER, HEAD
SECTOR NUMBER
CYLINDER, HEAD, REC#
DATA WRITTEN ON DEVICE 3
DATA BUFFER

When the GO flag is set the channel finds the "Start" for device 30 and performs the seek. When Channel End occurs, DEVLST is altered to BYTE (8)30(4)16(24)D30LST+2 and the scan resumes, finding the "Start" for device 31. That seek is then issued, and on Channel End DEVLST+1 is altered to BYTE (8)31(4)16(24)D31LST+2. The channel finds the zero at DEVLST+2 and idles. When Device End is received from device 31, the matching wait is found at DEVLST+1 and the set sector is issued. On Channel End DEVLST+1 is set to BYTE (8)31(4)16(24)D31LST+4. When the seek completes on device 30, its set sector is similarly given. If device 30 reaches its rotational target first, search, read, and halt are done and status is stored. The channel then waits for device 31 to reach the desired position.

Read Backwards

This command acts like a read, except that bytes are stored in memory in reverse order, from right to left, decrementing the address after each transfer. In BYTE mode, the low two bits of the specified byte count (for each data chain word) are used to position the first byte so that the last byte is stored in bits 0-7. Bits to the right of the first byte are zeroed. In WORD mode, bit 7 of the command is used to indicate a record containing an odd number of words, and causes the right half of the first byte read to be discarded.

Improved Error Recovery

Certain error conditions, such as an uncorrectable read error on a 3330, are retried automatically. The channel keeps a pointer to the first word of the last command, and resumes execution at that address upon controller request. If the error is recovered in this manner, the program will receive no indication anything was wrong except through the error counts maintained by the controller.

Some controller faults, such as an uncorrectable control store error, result in an inability to complete a signaling sequence. The channel responds by issuing a selective reset and reporting a control check to the CPU. If a memory error occurs, the program can cause a selective reset, along with the control check indication, by giving a "reset" CONO with bit 30 set. A HALT I/O instruction may be simulated by giving a "reset" CONO with bit 29 off and bit 30 set. This function stores no status of its own.

Miscellaneous

If bit 7 is set on a BYTE mode control or write command, the first byte is taken from positions 16-23. This permits the same data to be used as an argument to a seek command and a subsequent search ID command.

SA10-B ADDITIONS

Two additional data transfer modes are provided, making a total of four. The mode of a transfer is determined by bits 1 and 4 of the device command word governing the transfer:

Bit 1	Bit 4	Mode
0	0	WORD
0	1	BYTE
1	0	NATURAL
1	1	TAPE COMPATIBILITY

Two memory words are broken into data bytes when writing, or formed from data bytes when reading, as shown below according to the mode. (Byte 0 is the first; in a partial byte, bit 0 is the leftmost.)

Mode	First Memory Word					Second Memory Word					
WORD	0 7 0	8 15 1	16 23 2	24 31 3	32 35 4 0-3	0 7 5	8 15 6	16 23 7	24 31 8	32 35 4 4-7	←Byte ←Bits
BYTE	0 7 0	8 15 1	16 23 2	24 31 3	32 35 xxx	0 7 4	8 15 5	16 23 6	24 31 7	32 35 xxx	←Byte
NATURAL	0 7 0	8 15 1	16 23 2	24 31 3	32 35 4 0-3	0 3 4 4-7	4 11 5	12 19 6	20 27 7	28 35 8	←Byte ←Bits
TAPE C.	0 7 0	8 15 1	16 23 2	24 31 3	32 35 4 4-7	0 7 5	8 15 6	16 23 7	24 31 8	32 35 9 4-7	←Byte ←Bits

SYSTEMS CONCEPTS 520 THIRD STREET SAN FRANCISCO, CALIFORNIA 94107

SA-10
Subsystem Adaptor
Theory of Operation

Serial Numbers 107 and Above

TELEPHONE: 415-442-1500 TWX: 910-372-6062

TABLE OF CONTENTS

(Revised March 1, 1975)

INTRODUCTION

The Systems Concepts SA-10 is designed to interface IBM-compatible I/O control units to a DEC PDP-10. One SA-10 can simulate four IBM selector channels; the SA-10 is also available in a two channel version. Figure 1 shows a typical installation.

At the heart of the SA-10 is a microprocessor with a 16 bit instruction word and 768 words of read-only memory (ROM). Each subchannel has associated with it a ROM instruction address register (IAR), and the microprocessor alternates between them. Thus four independent microprograms can be running (almost) concurrently in the shared microcode.

The subchannel microprograms are not necessarily run in consecutive order. In fact, the standard algorithm runs them in an eight-clock cycle as follows: 0,1,0,2,0,1,0,3, giving the first two subchannels a higher data rate capability.

For a definition of the outboard interface, refer to IBM publication GA22-6974-1, "IBM System/360 and System/370 Interface--Channel to Control Unit, Original Equipment Manufacturers Information". For programming information, refer to "SA-10 USER'S MANUAL".

DATA STRUCTURES

The channel microcode has access to the following data items:

MRB	a 36 bit source of data from PDP-10 memory
MB	a 36 bit sink for data to PDP-10 memory shared between the channels
R0,R1,R2,R3	four 36 bit registers used for manipulating memory data
A0,A1,A2,A3	four 24 bit registers used for manipulating memory addresses and word counts. An A register can be incremented by +1 or -1 in one clock tick; this is the only arithmetic hardware in the SA-10. The first three registers are sometimes referred to as WC (word count), CA (current address), and PC (channel program counter); the fourth register saves the PC for retry purposes.
MA	a 22 bit sink for addresses to PDP-10 memory shared between the channels.
BUF	a FIFO used to buffer data to and from the I/O control unit. The capacity is 16 bytes.
CBUS	an 8 bit bus onto which can be gated almost anything
XBUS	an 8 bit bus, driven from CBUS or CBUS with its halves swapped, which supplies data to BUF and the R registers
TDS	two digits of temporary storage that can form the right half of the byte whose left half is bits 32-35 of an R register. If bit 14 of the microinstruction is off, TDS1 is selected. If bit 14 is on, TDS0 is selected. TDS0 is used to halve the data byte that crosses the word boundary of a 36 bit PDP-10 word.

Figure 2 shows some of the data structure of the channel. The notation in parentheses shows the number of bits. For instance MRB(36x4) means that there are four MRB registers (one per subchannel) and that they are 36 bits wide.

The R registers have ten different write enable signals so that the CBUS can be loaded into ten different byte positions:

B0L:	bits 0-3	B0R:	bits 4-7
B1L:	bits 8-11	B1R:	bits 12-15
B2L:	bits 16-19	B2R:	bits 20-23
B3L:	bits 24-27	B3R:	bits 28-31
B4L:	bits 32-35	B4R:	TDS

The same convention is used for gating bytes of the R registers back onto the CBUS.

CHANNEL FLAGS

Four bits per channel are visible both to the microcode and to the PDP-10. Their operation is described in the SA-10 USER'S MANUAL under "I/O Bus commands to the Channel".

000	GO FLAG (also called BUSY)
001	STATUS REQUEST FLAG
010	STATUS FLAG
011	INTERRUPT ENABLE

SUBCHANNEL TAGS

Of these eight bits, the first six directly control protocol lines to the control units. The last two bits are internal to the subchannel.

000	ADDRESS OUT
001	SELECT OUT (also raises HOLD OUT)
010	SERVICE OUT
011	NOT OPERATIONAL OUT (inverted at cable driver)
100	SUPPRESS OUT
101	COMMAND OUT
110	WRITE (as opposed to READ)
111	BUFFER ENABLE (the FIFO)

OPERATION STATUS BITS

As seen by the ON and OFF instructions, these bits are as follows:

000	STATUS TYPE bit 0
001	STATUS TYPE bit 1
010	SELECT ERROR
011	BYTE MODE (four bytes per word)
100	CONTROL ERROR
101	not used
110	LENGTH ERROR
111	PROGRAM INTERRUPT

where the two TYPE bits are decoded as follows:

00	Dummy
01	Asynchronous
10	Initial selection
11	Ending

None of the above bits do anything in the channel except BYTE MODE.

Warning: when the operation status bits are gated onto the CBUS to be used as a source, they appear somewhat differently. The first two bits are inverted and BYTE MODE has been replaced by another signal.

C0	NOT STATUS TYPE bit 0
C1	NOT STATUS TYPE bit 1
C2	SELECT ERROR
C3	BUS IN PARITY ERROR
C4	CONTROL ERROR
C5	0
C6	LENGTH ERROR
C7	PROGRAM INTERRUPT

the status code, in its complemented form, now becomes:

00	Ending
01	Initial selection
10	Asynchronous
11	Dummy

This CBUS format of the status bits is the one used when the channel stores a status byte in PDP-10 memory.

SUBCHANNEL IN BITS

Another group of eight bits that can be gated on the CBUS is as follows:

C0	ADDRESS IN
C1	SELECT IN
C2	BUFFER EMPTY (FIFO)
C3	OPERATIONAL IN
C4	REQUEST IN
C5	STATUS IN
C6	BUFFER CYCLE REQUEST (FIFO)
C7	BUFFER HALT [illegible]

The five "IN" signals are read directly from the control unit; the other three signals refer to the state of the 16 byte subchannel FIFO.

CHANNEL FLAGS IN

When the channel flags are selected as a CBUS source, the byte looks like this:

C0	GO (BUSY)
C1	STATUS REQUEST
C2	STATUS FLAG
C3	not used
C4	WORD COUNT OK (no overflow)
C5	TIMER (25.6 usec)
C6	not used
C7	not used

THE MICROINSTRUCTION SET

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
GOTO	0	0	0	0	0	0	a									

This instruction causes the next instruction to be taken from location a, which can be any location in ROM. Currently the ROM is 768 words long; up to 1024 words may be installed.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PULSE	0	0	0	0	0	1	f		a							

Two of the four possible functions specified by f are used:

f=00	clear all eight subchannel tags
f=01	clear the (7) operation status bits

The next instruction is taken from an address formed by concatinating bits 6-7 of the current location with bits 8-15 of the pulse instruction.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OFF	0	0	w		0	n			a							

The function of this instruction is to turn off a single bit somewhere. The w field selects the group as follows:

w=01	channel flags
w=10	subchannel tags
w=11	operation status bits

Within each group, the n field selects the individual bit, as specified in the sections above describing each group.

The next instruction address is formed the same way as the PULSE instruction.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ON	0	0	w		1	n			a							

This instruction turns a single bit on, and works just like the OFF instruction.

```
        0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
       ------------------------------------------------
XANC    0  1  r     y     b        d           a
       ------------------------------------------------
```

The function of this instruction is to manipulate the 24-bit A registers. The y field specifies which of the A registers will be involved: WC (00), CA (01), or PC (10).
The r field specifies one of the 36-bit R registers.
The b field simultaneously specifies what is gated onto the CBUS:

00	IARB8-15 (diagnostic purposes)
01	ADDR16-23
10	ADDR24-31
11	ADDR32-35,0,0,ADDR14,ADDR15

and what is gated onto the ADRIN bus:

00	Incremented value of ADDR
01	Word count field (1,REG1-1$,1,1,1,0) or decremented value of ADDR if d=10xx
10	Address field (reg16-35)
11	Base address + 4 times channel number

The d field specifies the destination:

0000	R0-7	"
0001	R8-15	"
0010	R16-23	"
0011	R24-31	from XBUS
0100	R32-35,TDS	"
0101	BUS OUT	"
0110	subchannel FIFO	"
0111	"	" (also causes swap)
1000	start memory read [illegible]	
1001	conditionally start memory read [illegible]	
1010	start memory write	
1011	start dummy cycle (diagnostic purposes) [illegible]	
1100	A register from ADRIN bus	
1101	R register (all 36 bits) from MPB	
1110	R32-35, R12-15	
1111	---	

The four memory functions above hang until the memory interface is free (see "HANGING", below). If a memory cycle is to be started, MB is loaded from the specified R register, MA is loaded from the specified A register, and the A register is loaded from the ADRIN bus. The 1001 conditional read initiates a fetch cycle if the result will not exceed the word count.
The next instruction is taken from an address formed by concatenating bits 6-11 of the current location with the a field (bits 12-15) of the XANC instruction. The XANC, TEST, and XFER instructions cannot jump out of the current 16-word page.

	0	1	2 3	4 5 6 7	8 9 10 11	12 13 14 15
TEST	1	0	r	s	c	a

The TEST instruction can jump to either of two places, depending on whether a specified "branch" condition is met. If the condition is not met, the next instruction address is formed as in the XANC instruction. If a branch does occur, the address is the same except that the least significant bit is complemented. (See "BRANCHING" below.)

If an R register is involved, it is specified by the r field.

The s field specifies a data source. (See the XFER instruction below.)

The c field specifies the condition to be tested:

0nnn	CBUS bit nnn true
1000	CBUS not equal zero
1001	CBUS bits 0+2+3+6+7 true (error condition test)
1010	REG16-23 = MRB0-~~15~~7(same device test)
1011	NOT CBUS bits 4,5 on and 6,7 off (read backwards test)
1100	CBUS4 and not CBUS1 (byte mode test)
1101	CBUS nonzero or REG14-15 nonzero (skip test)

	0	1	2 3	4 5 6 7	8 9 10 11	12 13 14 15
XFER	1	1	r	s	d	a

This is a general-purpose move instruction. The r, d, and a fields are the same as the XANC instruction.

The s field for the XFER and TEST instructions specifies the source of the data:

0000	R0-7
0001	R8-15
0010	R16-23
0011	R24-31
0100	R32-35,TDS
0101	CHANNEL FLAGS IN
0110	---
0111	---
1000	SUBCHANNEL B/C IN BITS
1001	"
1010	SUBCHANNEL TAGS
1011	OPERATION STATUS BITS
1100	0
1101	MRB to R register
1110	subchannel FIFO buffer
1111	" (also causes swap)

IMPLICIT DATA PATHS

Mention has already been made of the TDS registers and how they communicate with the CBUS as the right half of byte 4 of the R registers. On a memory read, when the contents of MRB are moved to an R register, the bottom four bits are moved into TDS (called REG36-39 on the schematics) as will as into R32-35. On a memory write, if the source is R3, TDS replaces the bottom four bits of R3.

XBUS normally follows CBUS, but sometimes is gated from CBUS swapped, under control of the SWAP signal. SWAP may be generated by source = 17 (FIFO with swap). This also modifies the effect of destination codes 0-5 to write the right half of XBUS into one byte position lower than normal (dest = 0 writes TDS, dest = 1 writes bits 4-7, dest = 2 writes bits 12-15, etc.) Dest = 7 also gives SWAP (FIFO with swap). This modifies source codes 0-5 to gate one lower byte position to the right half of CBUS. (Source = 0 gates TDS, source = 1 gates bits 4-7, source =2 gates bits 12-15, etc.)

The word count register, A0, is incremented automatically on XFER instructions which reference the subchannel FIFO (succes fully). This happens unconditionally in byte mode, and also when CR14 and CR15 are both true (jumping to the end of a four word block).

On XANC instructions that start memory cycles, the specified A register, in addition to being used as a memory address, is written from the ADRIN bus. The typical quantity to gate onto the ADRIN bus would be the incremented value of the register.

TIMING CONTROL

A 20 MHz crystal oscillator is counted down to a 5MHz clock with a duty cycle of 25%. Most of the SA-10 runs off a clock gated by CLEN, a flipflop under the control both of the PDP-10 and console switches.

SCC, a three bit counter (sht 13), develops a major cycle of eight clock ticks. A jumperable gating structure on SCC develops two NXT-SCA lines which specify which subchannel will run on the next clock tick. The NXT-SCA lines gate the correct IAR onto the ROM address lines. The clock loads the ROM output into the CR register, the gated IAR into the IARB register (mainly for diagnostic purposes), and the NXT-SCA lines into the SCA register (all on sht 11). Thus each microinstruction requires two phases, a fetch (using NXT-SCA) and an execution (using SCA), and each phase overlaps some other channel doing the opposite phase.

A few bits used to control critical timing paths (e.g. READ-OK, sht 3,10) are selected by channel number on the fetch phase and clocked into flip-flops so that they will be stable very early in execution phase.

The 1600 nsec SCC cycle is further counted down to produce a TIMER signal with a period of 25.6 usec, which in turn is counted to 15 to detect non-existent memory in about 400 usec.

HANGING

The IAR for each channel is usually modified at the end of execution phase and stored back into the RAM chips. The write enables to these RAM chips are gated in several flavors:
a) a GOTO writes all the bits. b) PULSE, OFF, and ON instructions write only bits 8-15, leaving the RAM chips for bits 6-7 unmodified. c) XANC, TEST, and XFER instructions write only the last four bits. d) sometimes no bits at all are written. This last case is called "hanging".

A HANG condition causes the same instruction to be executed over and over until the HANG condition goes away.

1. MEM-GO-HANG (sht 7) is generated when an attempt to start a memory cycle finds the memory interface busy.
2. MRB-HANG (sht 9) is generated when an attempt is made to reference fetch data in MRB and the memory is busy on behalf of that channel.
3. BUF-HANG (sht 8) is generated when the microprocessor is temporarily blocked from the subchannel FIFO buffer.

BRANCHING

Branching has been discussed above under the TEST op. Since the condition tested is sometimes asynchronous to the SA-10 clock, the situation is treated carefully. Firstly, only one bit is changed (IAR15) as a result of the BRANCH condition. Since it is possible for a channel to have a fetch phase immediately following the execution phase, IAR15 is given most of a clock tick to stabilize as follows: it is used as a ROM address bit which controls the output multiplexor internal to the Tri-State PROM chips, which drive the CR register. (The CR can also be loaded from the PDP-10 I/O BUS (for diagnostic purposes), or a small integer constituting a GOTO to one of a few fixed locations (used for channel resets).)

There is one BRANCH condition that is not included in the TEST op. If BUF-OP is true (the FIFO buffer is being used either as a source or as a destination), and CR15 and CR14 are both false (jumping to the beginning of a four word block), a BRANCH will occur if for some reason it is not reasonable to transfer another byte (device done or word count reached).

SUBCHANNEL AND FIFO

The term SUBCHANNEL here refers to that portion of the logic which is exactly duplicated four (or two)=TIMES AND WHICH DEALS with the control unit interface. It contains cable drivers and receivers, a BUSOUT register, a 16 byte FIFO, and control logic. Figure 4 shows the basic data paths.

The FIFO is organized as a 16 byte ring buffer in two 16x4 RAM chips. The buffer pointers are two four bit counters, one associated with the microcode, and one with the device. (THE microcode and the device cannot take buffer cycles on the same clock tick.) The full and empty conditions are handled by remembering which side took the last cycle. If the two pointers are equal, the buffer is either full or empty, and the side that took the last cycle cannot take another one until the other side takes a cycle.

Once a block transfer is initiated, the transmission of bytes is controlled by SERVICE IN requests from the control unit. The subchannel honors these requests by taking device buffer cycles independently of the microcode. A typical handshake sequence is shown in figure 5. The SA-10 is equipped to handle the High-Speed Transfer Feature where DATA IN requests alternate with SERVICE IN.

If a control unit with the IO Error Alert Feature sends DISCONNECT IN because it is bewildered, the PANIC bit is set in the subchannel, causing a jump to microcode location 0004 (for that channel only).

MEMORY

The portion of the SA-10 that deals with PDP-10 memory, and is not replicated once per channel, is termed the memory interface. Included are a 36 bit data register (MB), a 22 bit address register (MA), parity circuitry, and the control logic necessary to follow the memory bus protocol.

The memory interface can only be doing one memory cycle at a time. The time during which the memory interface is off doing its own thing is defined by the clock synchronous flip-flop MC-BUSY, during which time further attempts to start memory cycles are blocked.

An attempt by the microcode to start a memory cycle generates MEM-GO (sht 7) if a cycle really is to be started. MEM-GO sets MC-BUSY and the memory interface proceeds on its own. For a write cycle, the microcode is done at this point. For a read, the memory interface remembers which channel requested the cycle in bits MU0 and MU1 (sht 3) and loads the appropriate MRB register with the data. An attempt to use MRB will cause a hang until MC-BUSY goes away.

Figure 6 shows the timing of some of the memory interface signals.

IO BUS

DATAO is used to load the CR register for diagnostic purposes, and should be issued only when the clock is off. -CLEN gates the IO-BUS bits to the CR inputs (sht 11), and the DATAO CLR pulse clocks CR (sht 10). DATAO also goes to a plug to an external device (sht 6). (The most likely external device would be a portable PROM programmer.)

DATAI reads the diagnostic bus back to the PDP-10. DATAI allows RD-DIAG to set on the next clock (sht 8), and on the following clock DIAG-ON-BUS sets, enabling the cable drivers. DATAI going away dc resets both flip-flops. The inputs to the diagnostic bus must have been specified by a previous CONO, which information is saved in the CON register (sht 8), and placed on the DIAG-SEL lines by RD-DIAG (sht 10).

CONO has several functions. If bit 18 is on, the SA-10 is reset (see RESETS, below). If bit 26 is on, bit 30 is interpreted as a clock control function. The three possible cases of Stop, Single pulse, and Start are shown in figure 7. A flag write sequence (bit 27 on) is also shown. A subchannel reset (bit 27 off, 28 on) is similar. The PI channel number is stored as part of the CON register. If bit 19 is on and there has been a memory error, MC-CONTIN (sht 7) will be set, allowing the memory interface to continue and reset the error flags.

CONI is very straightforward; it merely enables the cable drivers where the information bits are already gated. No attempt is made to synchronize with the clock.

RESETS

The most drastic flavor of reset is the term RSTA (sht 8) which is caused by SA-10 power up, PDP-10 general reset, or a CONO with bit 18 true. RSTA sets the RESET flip-flop (sht 13) which is guaranteed to stay true through one complete 1600ns major cycle. RESET forces CLEN true and jams a GOTO 0 into CR.

A CONO with bit 27 false and bit 28 true is synchronized by SCH-RST-RQ and SCH-RST (sht 3), waits for the right channel number to come around on the NXT-SCA lines, then jams a GOTO into CR. The bottom two bits of the GOTO are bits 29 and 30 of the CONO.

As mentioned above, a DISCONNECT IN from a control unit causes a GOTO 0004.

CONSOLE

There is a group of four channel switches on the console. Whenever the NXT-SCA lines point to a switch that is on, a signal called SCH-SEL-SW is generated (sht 8). If the ADR-IGN switch is on, or if the address in IAR matches the address switches, SCH-SEL-SW goes on to become SW-COND. SW-COND can be used as a 'scope sync, to stop the clock, or to control when the lights register is loaded.

The clock will stop on SW-COND if the ADR-HLT-SW is on, and one may continue with the CONT button. If ADR-IGN-SW is on, the clock will stop on each instruction. If, in addition, all four channel switches are on, a single-clock function is obtained.

There is a 20 bit register, the sole purpose of which is to hold data for the 20 console lights. The register's input is the DIAG bus, and the three LITE-SEL switches select one of six possible sources to the DIAG bus. If the LC switch is in the COND position, the lights register is loaded on the tick following the one where SW-COND is true (i.e., execution phase of the selected instruction). The delay is achieved by the CONDB flip-flop. If the selected instruction starts a memory cycle, CONDB is saved in the MA register as MWATCH. If the LC=SWITCH is in the MEM position, the lights register is loaded at the end of any memory cycle during which MWATCH is true. If the switch is in the center position, both CONDB and MWATCH conditions are displayed.

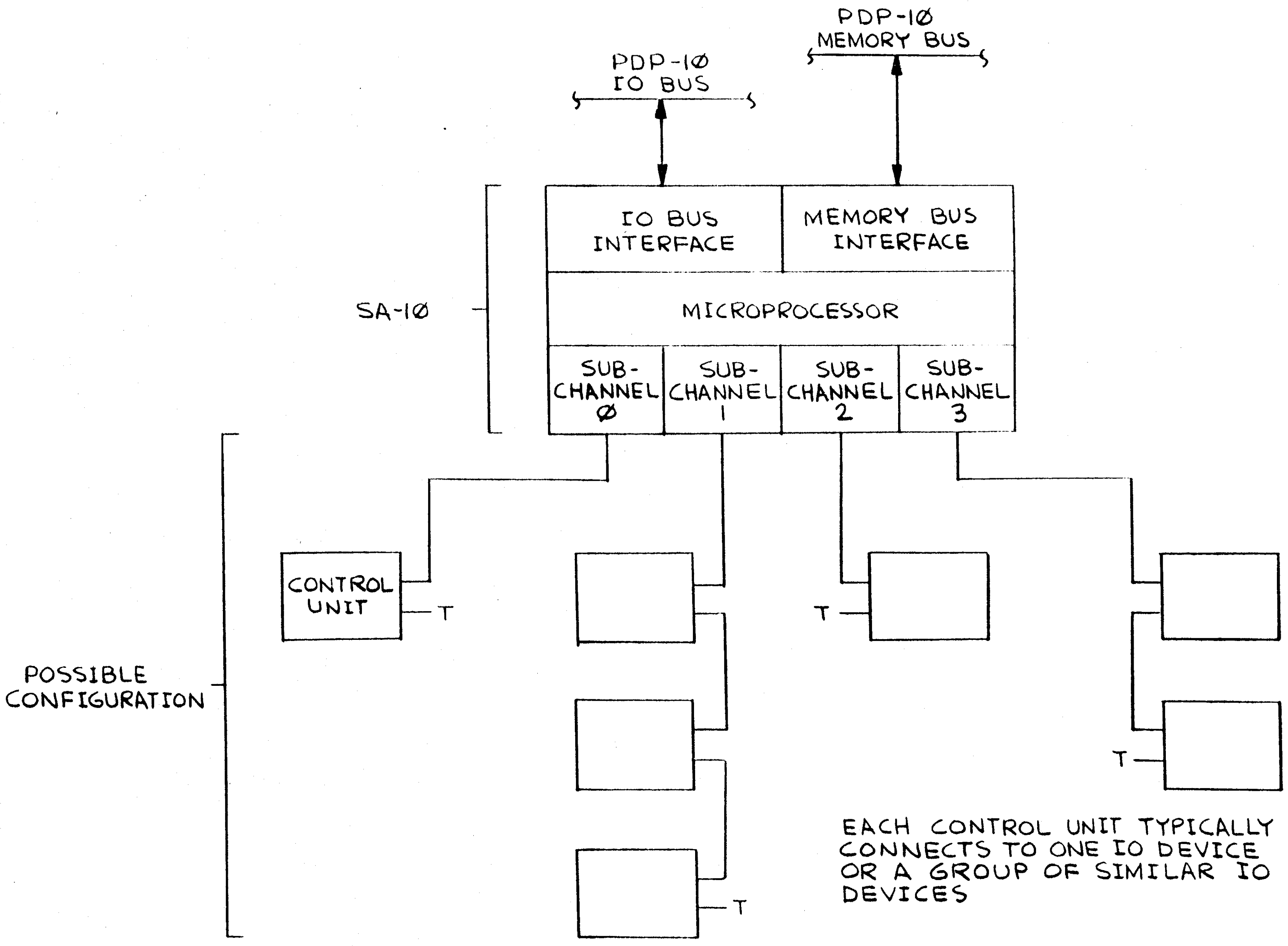


Figure 1. The SA-10 in a system.

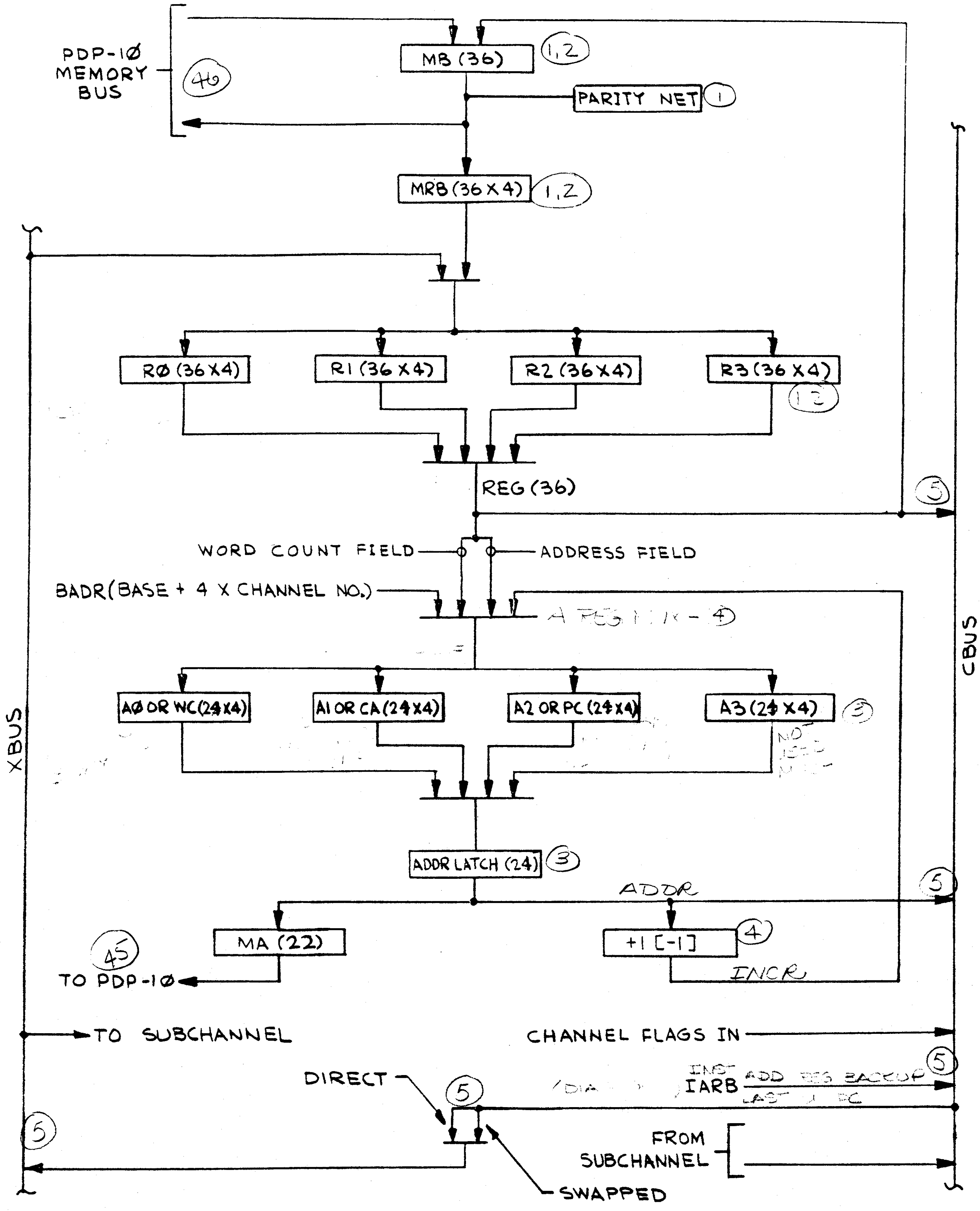


Figure 2. Address and Word data paths.

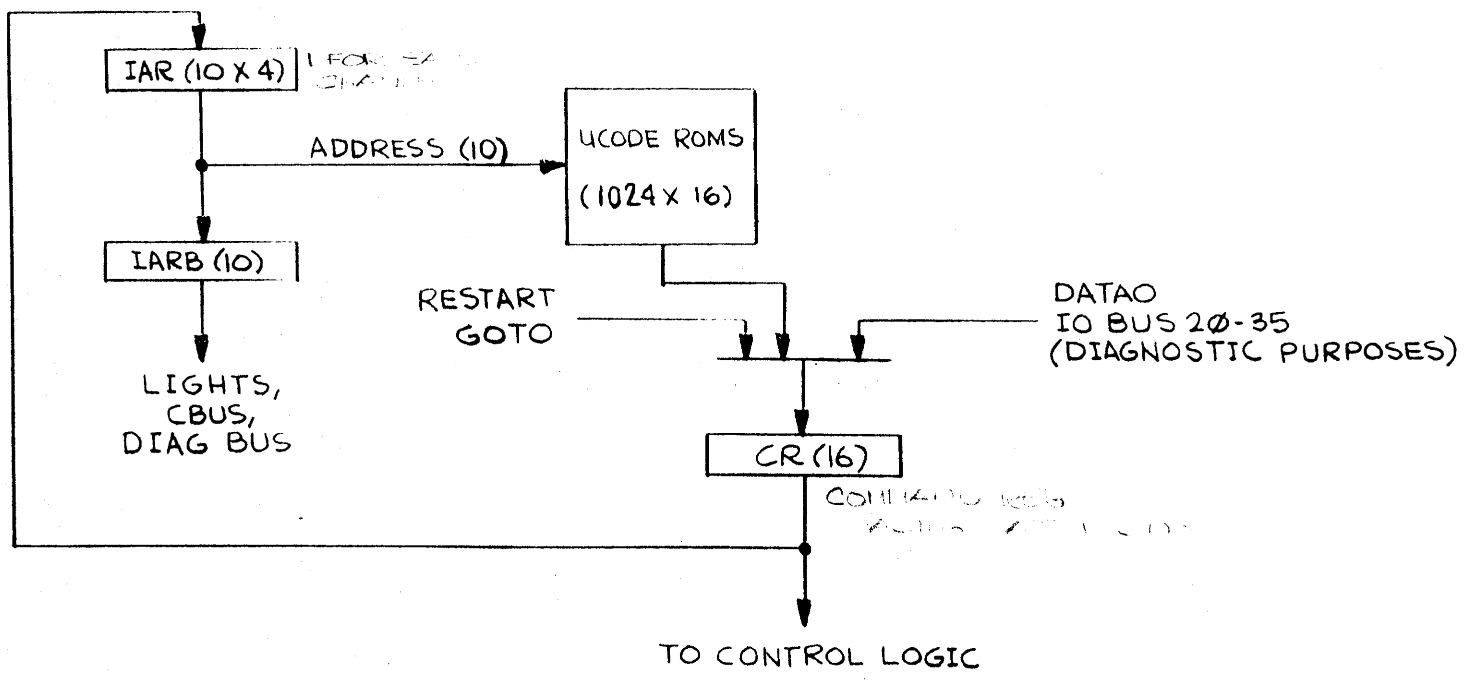


Figure 3. Microprocessor.

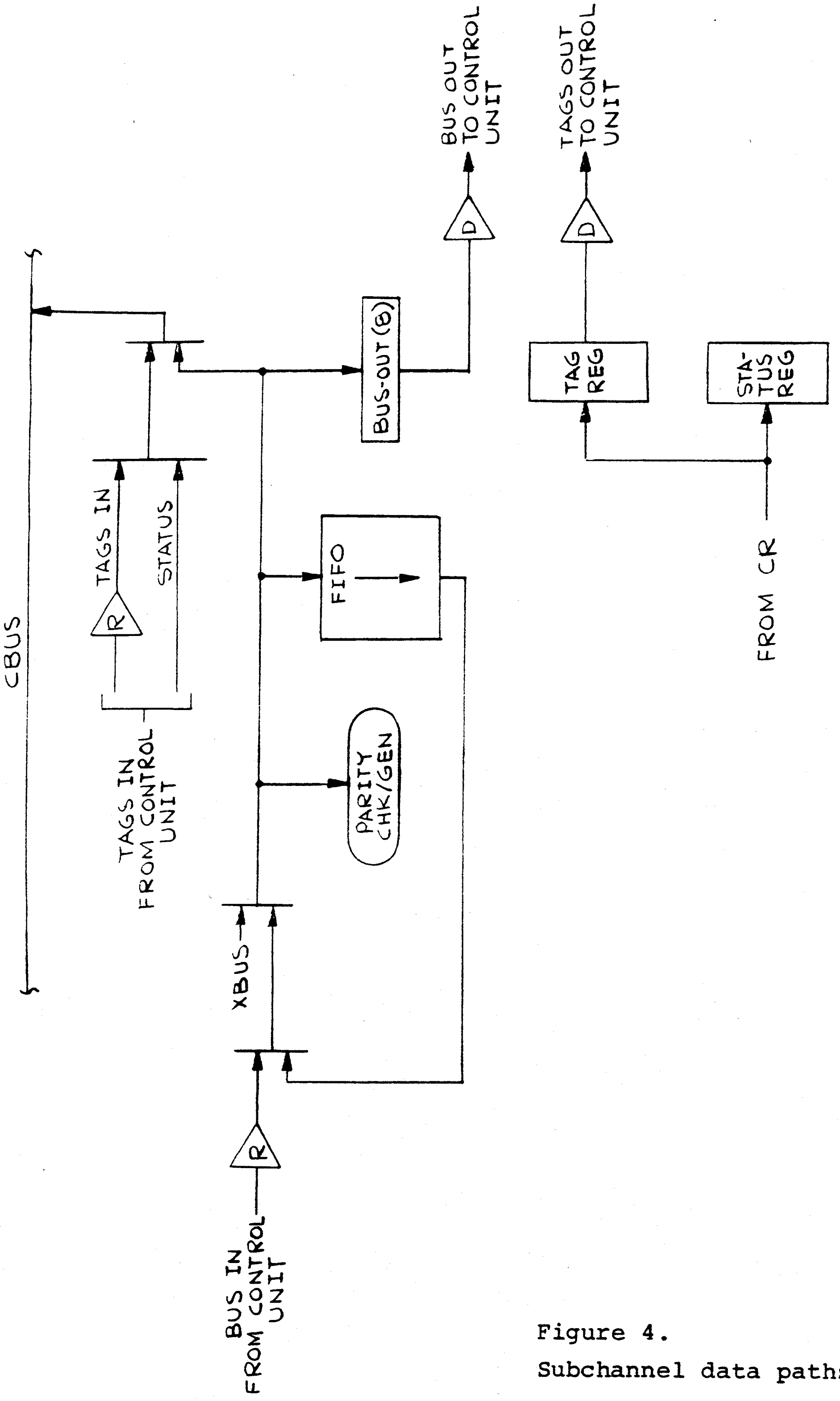


Figure 4.
Subchannel data paths.

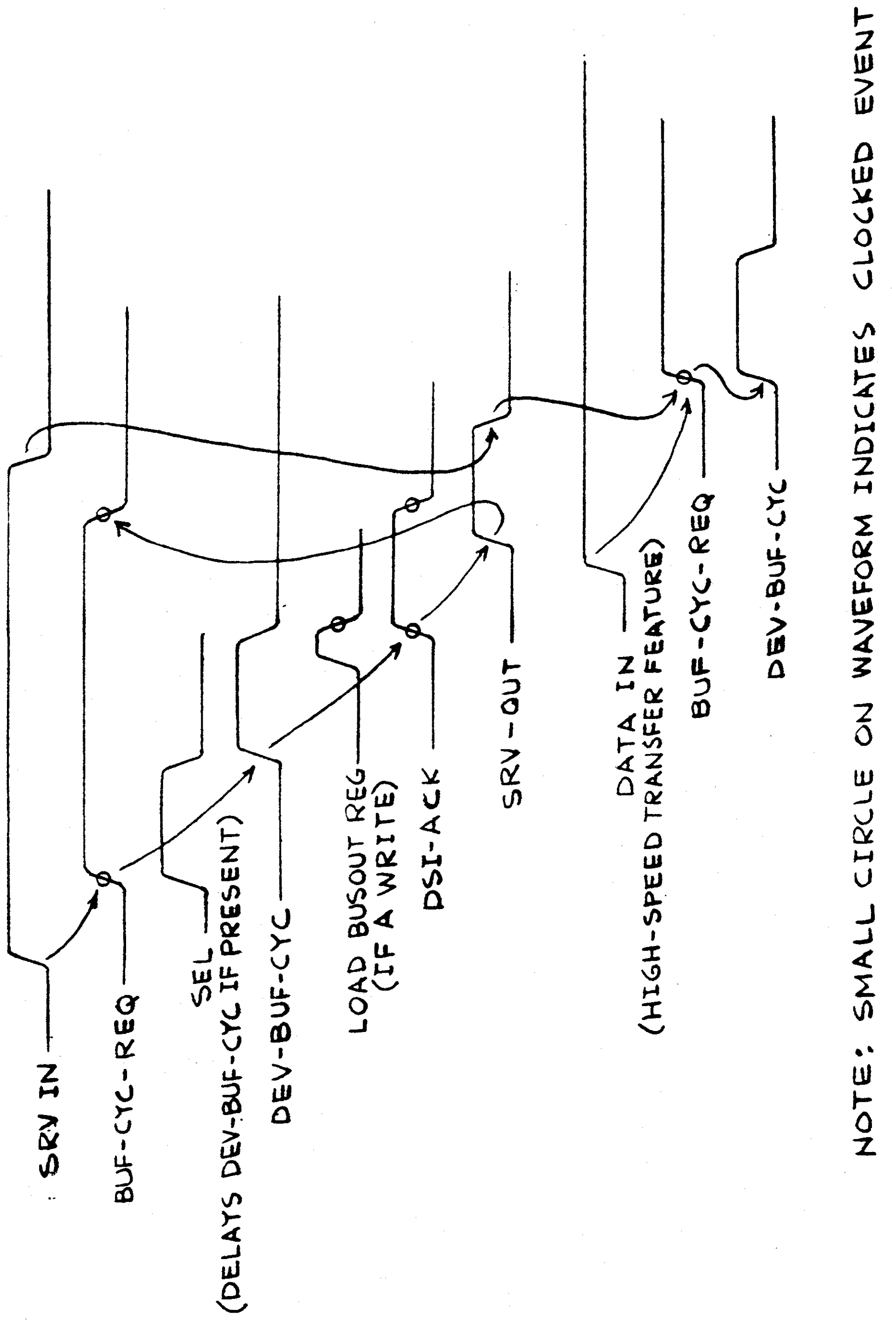

SRV IN
BUF-CYC-REQ
SEL
(DELAYS DEV-BUF-CYC IF PRESENT)
DEV-BUF-CYC
LOAD BUSOUT REG
(IF A WRITE)
DSI-ACK
SRV-OUT
DATA IN
(HIGH-SPEED TRANSFER FEATURE)
BUF-CYC-REQ
DEV-BUF-CYC
NOTE: SMALL CIRCLE ON WAVEFORM INDICATES CLOCKED EVENT

Figure 5. Handshake with control unit.

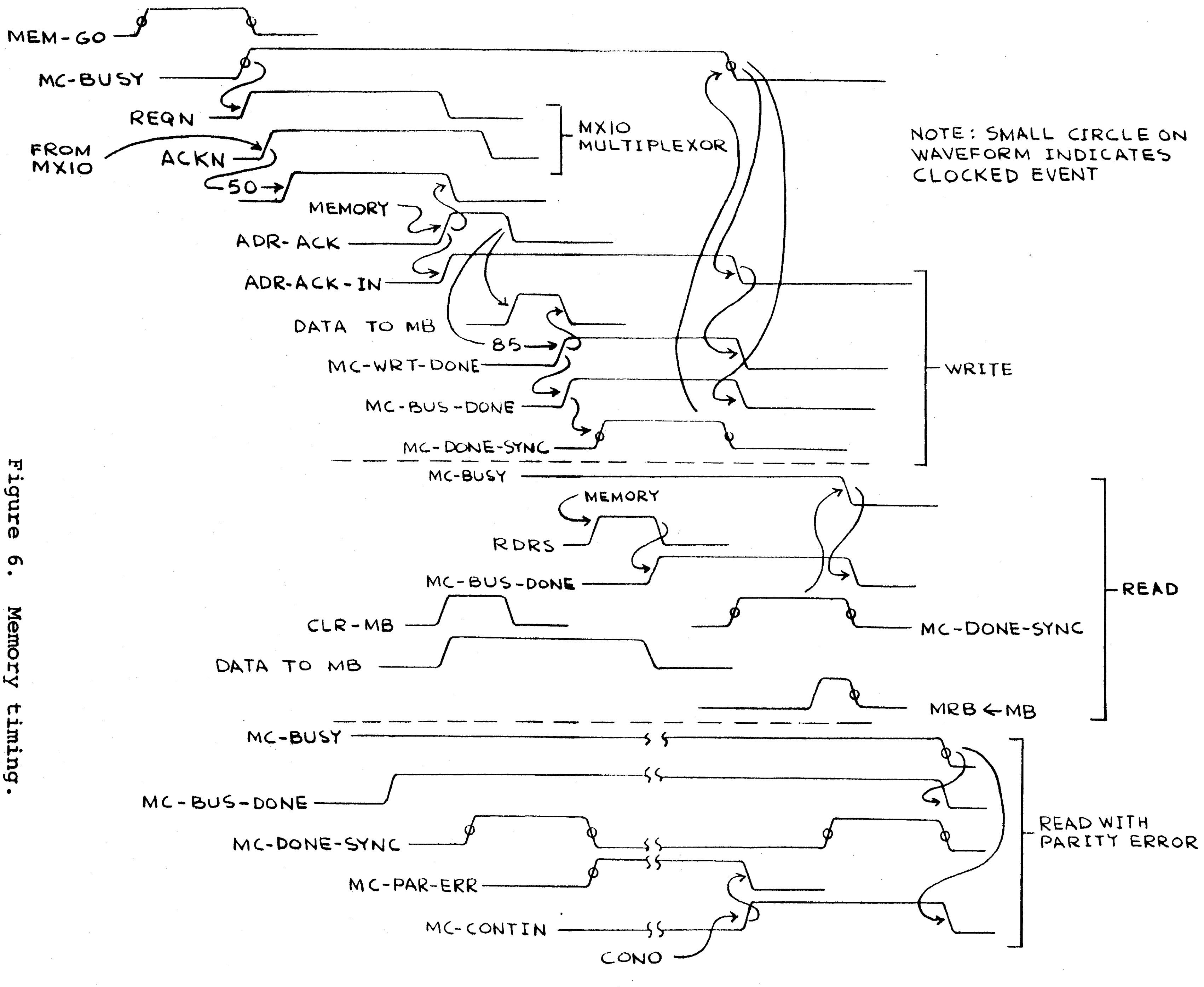
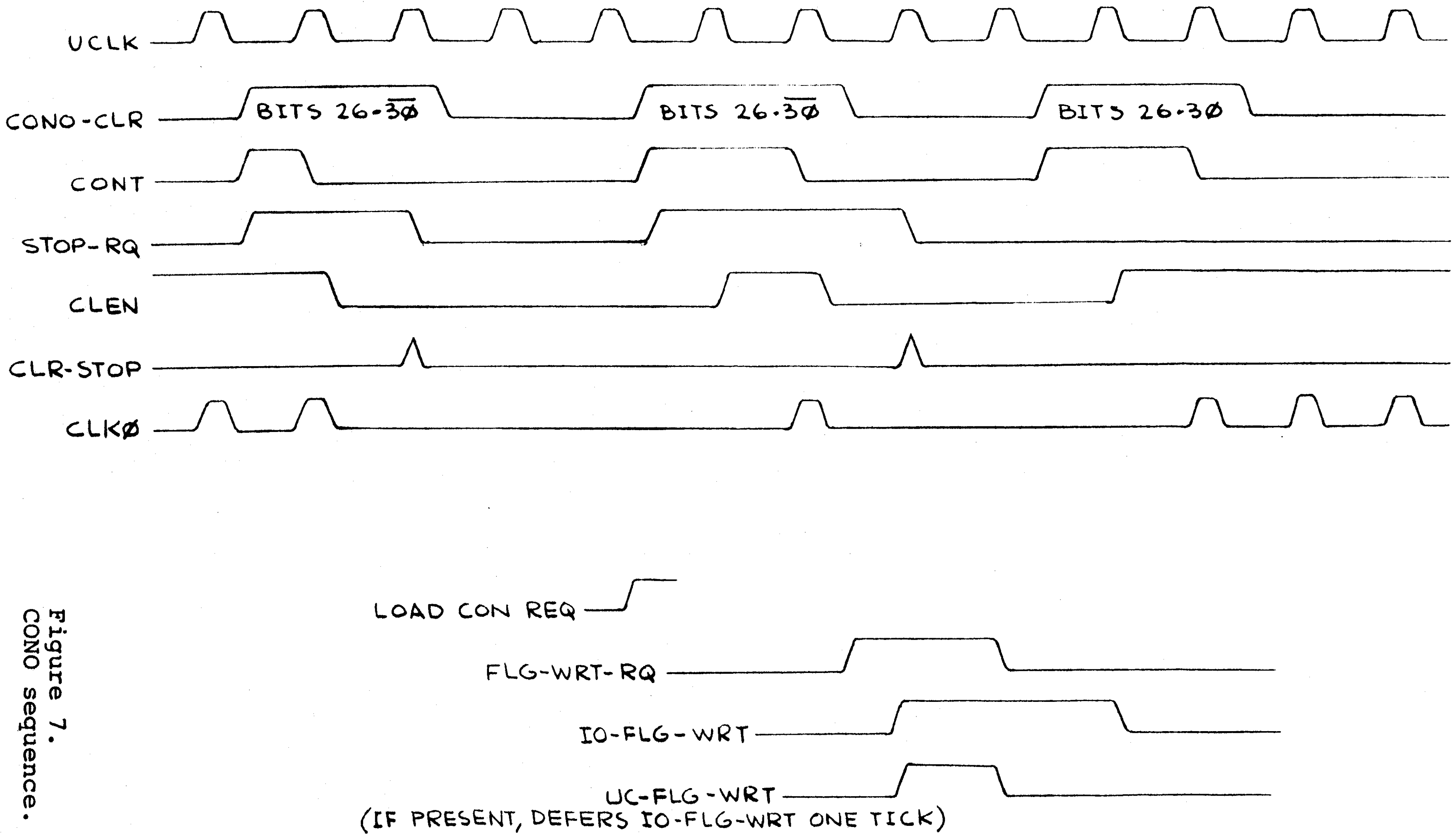


Figure 6. Memory timing.



NOTE: SMALL CIRCLE ON WAVEFORM INDICATES CLOCKED EVENT

Figure 7.
CONO sequence.

GLOSSARY

The numbers refer to the drawing sheets.

ACKN	7	memory bus multiplexor (MX10) acknowledge
ADDR13-35	3	latches hold A reg outputs stable while they are being written
ADR.CR4	9	XANC microinstruction with bit 4 true
ADR.CR5	9	XANC microinstruction with bit 5 true
ADR-CRY-CMPL	4	ADR carry complement (for decrementing)
ADR-CRY-ENB	4	ADR carry enable
ADR-HLT-SW	10	address halt switch
ADR-IGN-SW	10	address ignore switch
ADRIN13-35	4	multiplexed input to the A registers
ADR-SEL	9	-CR0.CR1, XANC instruction
ADRSEL0-1	4	select source for ADRIN bus
AS6-15	10	address select switches
AS=IAR	11	IAR matches console address switches
BADR25-31	4	switches control base address
BRANCH	10	when true, low order bit of IAR is complemented
BUF-BRANCH	8	buffer reference branch because data transfer finished
BUF-DUMMY	8	an attempt to take a FIFO cycle somehow inhibited
BUF-HANG	8	microcode hang until FIFO IS AVAILABLE
BUF-OP	8	FIFO reference, source or destination
BUS-OUT<CBUS	9	destination is BUS OUT register
BUSY	10	
BUSY-SY	10	
BYTE-MODE	8	BYTE MODE, selected by SCA number
CBUS0-7	5 and 14	main 8-bit data path
CBUS<B0-3	9	source field is 00xx
CBUS<B4-5	9	source field is 001x
CBUS<DEV	9	source field is 10xx
CLEN	13	clock enable
CLK(B1)	9	
CLK(B1)-	7	
CLK(B2)	9	
CLK(B2)-	7	
CLK(B3)	9	
CLK(B4)	9	
CLK0	13	Clock gated with CLEN
CLKT1	13	
CLR-MB-LT	7	clear MB (left)
CLR-MB-RT	7	clear MB (right)
CLR-STA-S0	10	clear operation status bits, subchannel 0
CLR-STOP	13	clears STOP-RQ
CLR-TAG-S0	10	clear subchannel tags, subchannel 0
CON27-32	8	IO BUS bits saved from last CONO
CONDB	10	console condition flip-flop
CONT(NO),(NC)	10	continue switch

CONT	13	continue after clock stop
CR0-15	11	register where microinstructions are decoded
CRD0-15	11 and 12	3-state bus that is the input to CR
CROBAR	7	from power supply sequencer
CRY16	4	carry out of incrementor bit 16
CRY20	4	" " " " 20
CRY24	4	" " " " 24
CRY28	4	" " " " 28
CRY32	4	" " " " 32
DECR	4	controls incrementor to decrement
DEV-DONE	10	"DEV-DONE" signal gated from the appropriate subchannel
DEV-DONE-SY	10	
DEV-MATCH	9	same device address in block multiplexor mode
DAIG16-35	5 and 6	20-bit diagnostic bus
DIAG-ON-BUS	8	set one clock after RD-DIAG; gates IO BUS drivers
DIAG-SEL-A,B	10	controls gating to DIAG bus
DIAG-SEL1-2	10	controls gating to DIAG bus
DST-BUF-OP	9	destination is subchannel FIFO
DST=7	9	Destination field = 7, referring to subchannel FIFO with byte halves swapped
EXT-FN	10	-CR0.-CR1.-CR2.-CR3
EXT-SENSE	6	response from external mystery device
FIX-WC	8	signal used to increment WC if final word is not completely filled
FLG-WRT-RQ	8	remembers CONO to set or reset flag
HI0	8	+3 volts to panel 0
HI1	7	+3 volts to panel 1
HI2	7	+3 volts to panel 2
HOLD-ADDR	9	enable to A reg holding latch
IAR6-15	11	Instruction Address Register, 4x10 RAM
IAR7-14	12	IAR inverters
IARB6-15	11	value of IAR for this instruction
IGN-SPLIT-BYTE	8	attempt to move split byte in byte mode is treated as a no-op
INCR13-35	4	five 4-bit partial sums of A+1
INCR-WC	8	automatic increment of word count
INTR	8	PDP-10 interrupt--channel or memory error
IOB<STAT	8	gates IO BUS drivers
IOBUS26-35		from IO BUS receivers
IO-DAT16-35	6	to IO BUS drivers for CONI or DATAI
IO-FLG-WRT	8	flag set or reset due to CONO--clocked ff
JAM-CR	11	CR is to be loaded from special source
LC-SW-COND	10	light control switch
LC-SW-MEM	10	" " "
LD-ADR	9	signal used to load an A register
LD-ADRA-E	4	controlled by LD-ADDR and preceeding carries
LD-ADR-OP	9	destination is an A register
LD-CR	10	
LD-IAR12-15	10	

LD-IAR6-7	10	
LD-IAR8-11	10	
LD-LR	10	load the lights register
LD-MA	7	load MA from A reg--memory cycle
LD-MB-LT	7	load MB (left) from R reg--memory cycle
LD-MB-RT	7	(right)
LD-MRB	7	MB to MRB at end of memory cycle
LD-STA-S0	10	load operation status bits, subchannel 0
LD-TAG-S0	10	load subchannel tags, subchannel 0
LIT16-35	6	20 light drivers
LITE-SEL0-2	10	display selection switches
LOC	7	D.C. voltage from power sequencer to be connected to PWR ON to bring power up
MA14-35	3	memory address register
MADR<MA	7	gate address to memory
MB0-19	1	memory buffer register
MB20-35	2	memory buffer register
MBD32-35	2	data to be stored in bottom half-byte of word
MB<MBUS-LT	7	catch fetch data
MB<MBUS-RT	7	(right)
MBP	9	parity bit for MB
MB-PAR-A	1	parity on MB0-11
MB-PAR-B	1	parity on MB12-23
MB-PAR-C	1	parity on MB24-35
MB-PAR-EV	9	MB (including MBP) has even parity
MBUS<MB-LT	7	strobe pulse to level shifters, memory data left half
MBUS<MB-RT	7	strobe pulse to level shifters, memory data right half, and control signals
MC-ADR-ACK		Address Acknowledge from cable receiver
MC-ADR-ACK-IN	7	latch remembers ADR-ACK
MC-ADRA	7	ADR-ACK to this unit
MC-BUS-DONE	7	done with memory bus
MC-BUSY	7	memory interface busy--subsequent references will hang
MC-CONTIN	7	catches PDP-10 acknowledgement of SA-10A memory error
MC-DONE-SYNC	7	follows MC-BUS-DONE, clocked
MC-ENB	7	MX10 says this is our memory cycle
MC-ERROR	7	memory error: parity or NX-MEM
MC-FINISH	7	signal which resets MC-BUSY
MC-NXM	7	non-existent memory error
MC-PAR-ERR	7	bad parity on a fetch
MC-REQ-CYC	7	memory cycle request
MC-SET-DONE	7	sets MC-BUS-DONE on write or NX-MEM
MC-WRRQ	3	marks memory cycle as a write
MC-WRT-DONE	7	write done--forms trailing edge of pulses to memory
MC-ZAP	7	abort memory cycle
MEM-GO	7	begin memory cycle
MEM-GO-A	7	memory cycle very likely

MEM-GO-HANG	7	hang because memory interface busy
MHZ10	13	=10Mhz
MHZ5	13	=5Mhz
MPX-CLR	7	abort signal to MX10 multiplexor
MRB0-19	1	memory read buffer RAMs
MRB20-35	2	memory read buffer RAMs
MRB-HANG	9	hang due to fetch not yet compleye
MTR-OUT	8	metering out signal to device controllers. True if a CONI (to any device) has occurred in last 24 msec.
MU0-1	3	remember which channel started current memory cycle
MU0(B1)	9	
MU1(B1)	9	
MWATCH	3	marks memory cycle for console display
NXMS	10	memory busy for 200 usec = non-existent memory
NXT-SCA0-1	13	govern which channel will execute next
OPR-GROUP	10	-CR0.-CR1
PANIC	11	subchannel has received DISCONNECT
PI1-7	8	interrupt lines to PDP-10
PIA0-2	8	IO-BUS33-35 from last CONO: interrupt channel
POR	7	power on reset from power supply sequencer
PWRON	7	signal to power sequencer for turn-on
RCBA0-2	9	address lines for those multiplexors which gate REG bits to the right half of CBUS
RCBE	9	REG CBUS enable to multiplexors
RD-DIAG	8	clock-synchronous DATAI
READ-OK	8	word count not yet reached on this channel
READ-OK-SY	10	
READS-DONE	8	word count not yet reached
REG0-19	1	outputs of R reg RAMs
REG20-35	2	outputs of R reg RAMs
REG36-39	2	outputs of TDS RAMs
REG<CBUS	9	everything but a 36-bit wide source
REG<MRB	9	source field is 1101
REM	7	signal to power sequencer for remote turn-on
REQN	7	memory bus multiplexor (MX10) request
RESET	13	
ROMEH	12	enable line for high-addressed half of PROM array
ROMEL	12	enable line for low-addressed half of PROM array
RST	8	POR + IO-RESET
RSTA	8	condition for complete SA-10A reset
RST-SYNC	13	governs reset of RESET
SO-ADR-OUT	14	ADDRESS OUT
SO-BOR<CBUS	15	transfer CBUS to BUS OUT register
SO-BUF-CYC-REQ	15	flip-flop set by SERVICE IN or DATA IN

S0-BUF-EMP	15	subchannel FIFO is empty
S0-BUF-ENB	14	buffer enabled, a subchannel tag
S0-BUF-HLT	15	buffer halt--device done or no more data
S0-BUFR0-7	14	subchannel buffer outputs
S0-BUFW0-7	14	subchannel buffer inputs
S0-BUS-IN	14	eight bits plus parity from control unit
S0-BUS-IN-PERR	15	bad parity from control unit
S0-BUSIN-MUX	14	eight-bit bus in subchannel onto which may be gated tags or status
S0-BUS-OUT	14	eight bits plus parity to control unit
S0-BUSY	8	subchannel 0 BUSY (or GO) flag
S0-BYTEM	15	BYTE MODE--subchannel status bit
S0-CB-DEV+BUF	15	gate device or buffer to CBUS
S0-CB-TAG+STA	15	gate tags or status to CBUS
S0-CBUS<DEV	15	gate device BUS IN TO CBUS
S0-CLK	15	
S0-CLKA	15	
S0-CMD-OUT	14	COMMAND OUT
S0-CTRL-ERR	15	subchannel status bit
S0-DEV-BUF-AVL	15	FIFO available to device
S0-DEV-BUF-CYC	15	device buffer cycle
S0-DEV-DONE	15	branch condition from subchannel: Status in or not Operational in
S0-DEV-LST	15	the last buffer cycle was taken for the device end
S0-DSC-IN	14	DISCONNECT IN from control unit
S0-DSI-ACK	15	true for one clock after S0-DEV-BUF-CYC
S0-DTA-IN	14	DATA IN from control unit (High-Speed Transfer Feature)
S0-DTA-OUT	14	DATA OUT to control unit (response to DATA IN)
S0-DTA-OUT-A	15	
S0-ENB-BUS-IN	15	enable BUS IN through receivers
S0-ENB-STA	15	enable status bits through multiplexor
S0-HLD-OUT	14	HOLD OUT
S0-INT-EN	8	subchannel 0 interrupt enable flag
S0-INT-STA0-1	15	subchannel status bits
S0-OPL-OUT	14	OPERATIONAL OUT
S0-PANIC	15	control unit is bewildered--causes jump to microcode 0004
S0-PAR-EV	14	even parity on data byte
S0-PROG-INT	15	subchannel status bit
S0-PTRS-EQ	14	the FIFO is either full or empty if the pointers are equal
S0-SCB<CBUS	15	gate CBUS to buffer
S0-SEL	15	SCA register pointing to channel 0
S0-SEL-ERR	15	subchannel status bit
S0-SEL-OUT	14	SELECT OUT
S0-SRV-IN	14	SERVICE IN from control unit
S0-SRV-OUT	14	SERVICE OUT to control unit--response to SERVICE IN
S0-SRV-OUT-A	15	

S0-STA-FLG	8	subchannel 0 status flag
S0-STA-IN	4	STATUS IN
S0-STA-RQ	8	subchannel 0 status request flag
S0-SUP-OUT	14	SUPRESS OUT
S0-SW	10	channel select switch
S0-UC-BUF-AVL	15	FIFO available to microcode
S0-UC-BUF-CYC	15	microcode buffer cycle
S0-UC-W-BUF-CYC	15	microcode write buffer cycle
S0-WRT	14	WRITE (data to device), a subchannel tag
***Preceeding signals are specific to subchannel 0. To ***generalize to subchannel n, add 2n to page number.		
SA0-CONO-CLR	8	first pulse of a CONO
SA0-DATAO-CLR	8	first pulse of a DATAO
SA0-SEL	8	PDP-10 selects SA-10 on I/O bus
SCA0,1	11	subchannel active
SCA0(B1)	9	
SCC0-2	13	major cycle counter
SCH-RST	8	subchanne reset, clocked
SCH-RST-CLR	8	jams a GOTO into CR
SCH-RST-RQ	8	remembers subchannel reset CONO
SCOPE-SYNC	10	
SC-SEL-SW	8	NXT lines match a switch-selected channel
SEL-R3	9	CR2.CR3
SRC-BUF-OP	9	source field is 1110 (FIFO)
SRC-BUF-OP1	9	source field is 1111 (FIFO with byte halves swapped)
STA-FLG	10	
STA-FLG-SY	10	
STA-RQ	10	
STA-RQ-SY	10	
STATUS-INT	8	PDP-10 interrupt from channel
STOP-RQ	13	stop request from CONO
SW-COND	10	console switch conditions met
SW-STOP	10	clock stop for console switches
SWAP	9	byte halves of CBUS are to be swapped as they are gated to XBUS
TIMER	10	12.2 usec waveform
UC-BUF-AVL	8	subchannel FIFO available to microcode
UC-BUF-CYC	8	microcode buffer cycle
UC-FLG-WRT	10	-CR0.-CR1.-CR2.CR3
UCLK	13	ungated clock
WC-OC	3	MSB of ADDR used to signify WC negative
WRT-FLGS-A	8	write BUSY and STATUS REQUEST flags
WRT-FLGS-B	8	write STATUS and INTERRUPT ENABLE flags
WRT-REG-B0L-B4L	9	write pulses to R register destination bytes 0-4, left four bits
WRT-REG-B0R-B4R	9	write pulses to R register destination bytes 0-4, right four bits
XBUS0-7	5	eight bit data path equal either to CBUS, or to CBUS with right and left halves swapped

SYSTEMS CONCEPTS 520 THIRD STREET SAN FRANCISCO, CALIFORNIA 94107

SYSTEMS CONCEPTS

ENGINEERING DRAWING

CONVENTIONS

TELEPHONE: 415-442-1500 TWX: 910-372-6062

SYSTEMS CONCEPTS LOGIC DRAWING CONVENTIONS

Logic Blocks

A given logic block on a drawing contains the following information:

a) Abbreviated form of manufacturer's type number. E.G. 74175 is used for SN74175N, and 3000 for MC3000P.

b) Location. In a cabinet with several wire-wrap panels, for example, 2F11 means second panel from the top, section F, socket number 11. The area at the top of a section is divided into three 14-pin sockets, denoted 31, 32, and 33. Socket 31 is roughly over socket 1, 32 is in the middle, and 33 is roughly over socket 5. On a printed circuit board, designators such as U21 are used, keyed to a parts placement drawing for the board.

c) An asterisk if the block represents a 14-pin DIP which is in a 16-pin socket (or a section of such a DIP). In this case the DIP is always placed in the lower pins of the socket, such that each socket pin number is 1 greater than the corresponding DIP pin number.

d) Gates and inverters are indicated by the form of the logic block as drawn; more complicated functions will have a designation for each pin denoting its function. For instance, a J-K flip-flop might have terminals marked J, K, C, R, S, Q; C is the clock, R the Reset (clear), S the Set (preset), and Q the output.

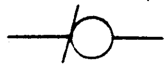
e) For each pin, the pin number. This is always the DIP pin number; if the chip is marked with an asterisk, add 1 to the given number to get the socket pin number. The same pin may be shown in more than one place. One appearance will be considered primary, and the others secondary; the pin number will be in parentheses for a secondary appearance.

Mixed Logic

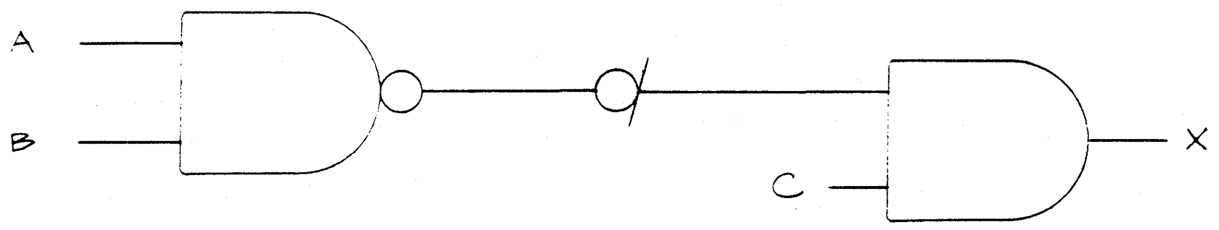
Mixed logic notation is used. An element is drawn showing the function intended by the designer, with inputs and outputs reversed in electrical polarity if needed. (See the example below.)

A signal asserted at a low voltage (ground in TTL) is shown with a nipple at each end of the wire. As an exception, the nipple is omitted on a secondary appearance of a pin. There may be a different signal with the same name, drawn without nipples, such as the other output of a flip-flop.

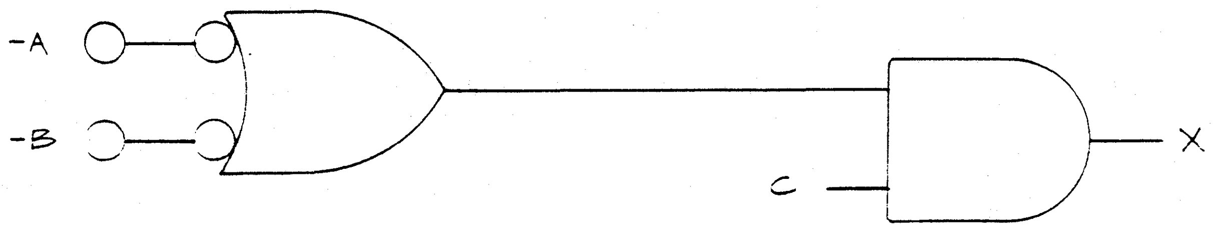
If an input is activated by the denial of a signal, a minus sign is used before the signal name. The signal -X without a nipple is electrically the same as X with a nipple; likewise, -X with a nipple is electrically the same as X without a nipple.

An inverter performs no logical function but only changes the polarity of assertion. Logical inversion (using an output to inhibit an input) is shown by a symbol like  .

$X = \overline{(A \wedge B)} \wedge C$ (all signals true high) would be drawn



$X = (\overline{A} \vee \overline{B}) \wedge C$ (all signals true high) would be drawn



Both drawings represent the same hardware and wiring.

Clocking

Clock inputs are shown with the polarity of the pulse used, assuming a state change after the end of the pulse, i.e. activated by the trailing edge. This is consistent with the common practice of using the same signal both in clocking a device and in determining its mode or input data.

The gate input of a latch has its polarity shown such that assertion causes the latch to pass data from input to output.

Ground, HI, Supply Voltages

Ground (0 volts) is drawn as a triangle pointed down or to the left.

HI (approximately +3 volts) is generated independently for each wire-wrap panel or printed circuit board. Unless otherwise indicated, the HI run connected to a device is the one specific to that panel or board.

Since ground and HI are fixed voltages rather than signals, the nipple of mixed logic is not drawn at the point where the HI label or ground triangle is attached to a wire or to a device.

Unless otherwise shown, ground and supply voltages are connected to each device according to the manufacturer's recommendation.

Discrete Components

In printed circuit boards, discrete components are usually soldered in, and are given reference designators as follows:

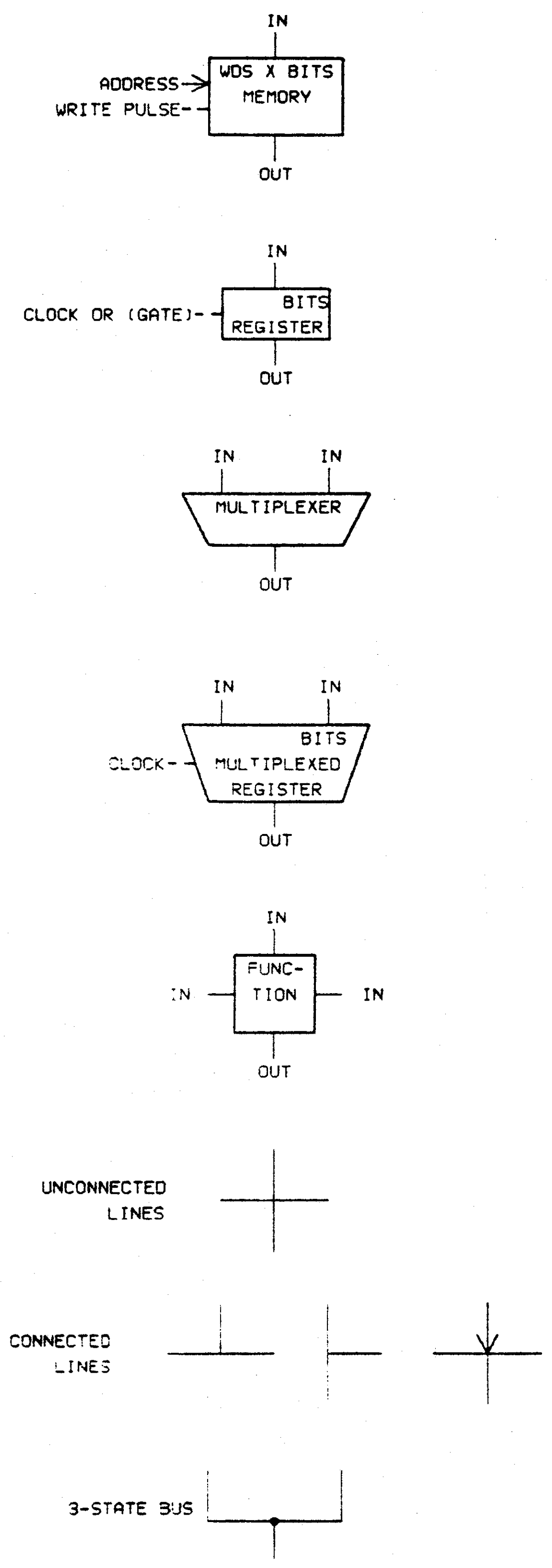
Rn	Resistor or multiple-resistor package
Cn	Capacitor
Qn	Transistor
CRn	Diode
Tn	Transformer
Ln	Inductor
Jn	Jack

In wire-wrap panels, such components are usually mounted on 14- or 16-pin plugs which are then inserted into DIP sockets. The component is then designated by the DIP position in the panel and the socket pin numbers on which its terminals appear.

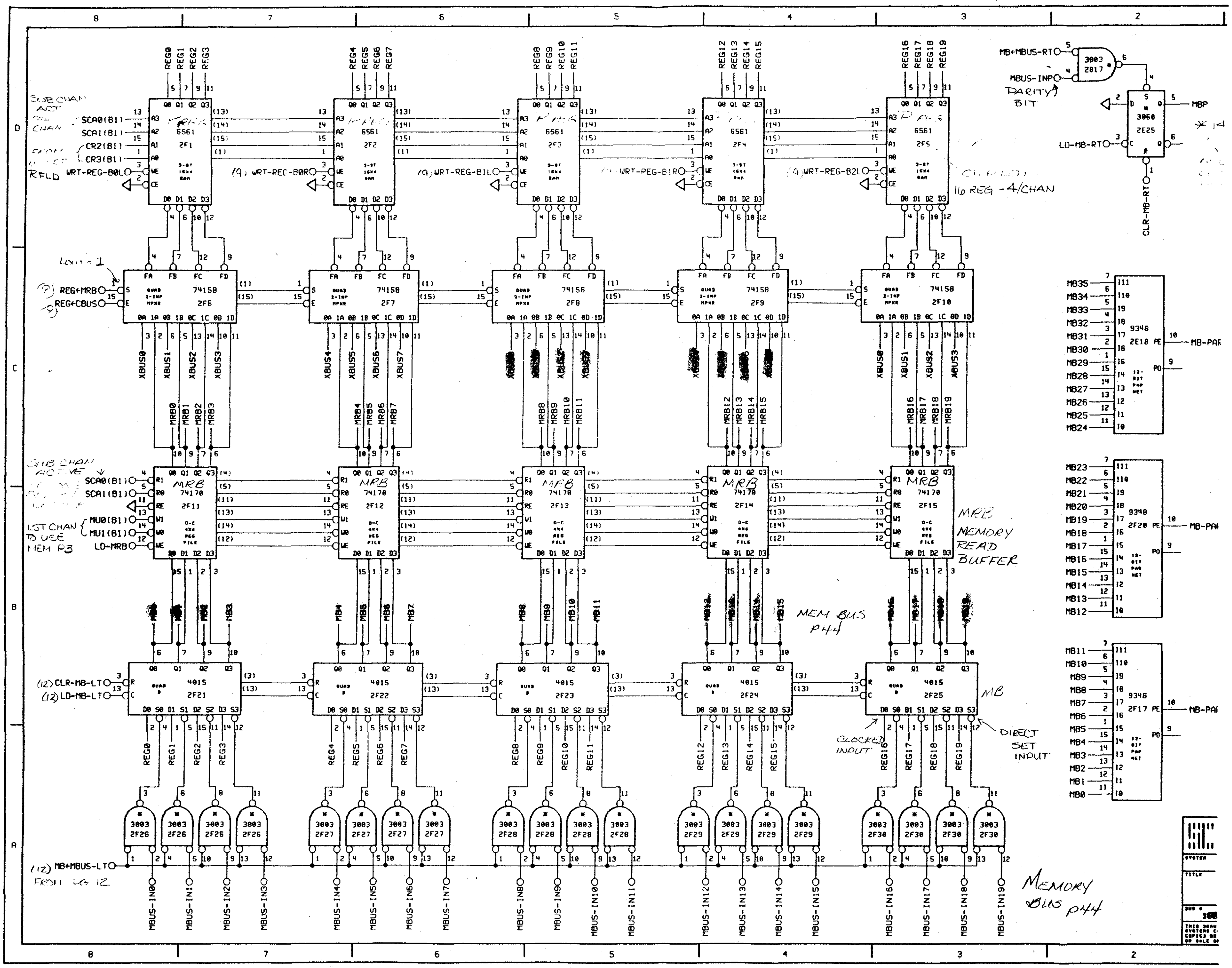
Unless otherwise indicated:

Resistors are 1/4 watt, 5%;
1% resistors are 1/8 watt;
Resistance values are in ohms;
Capacitance values are in microfarads.

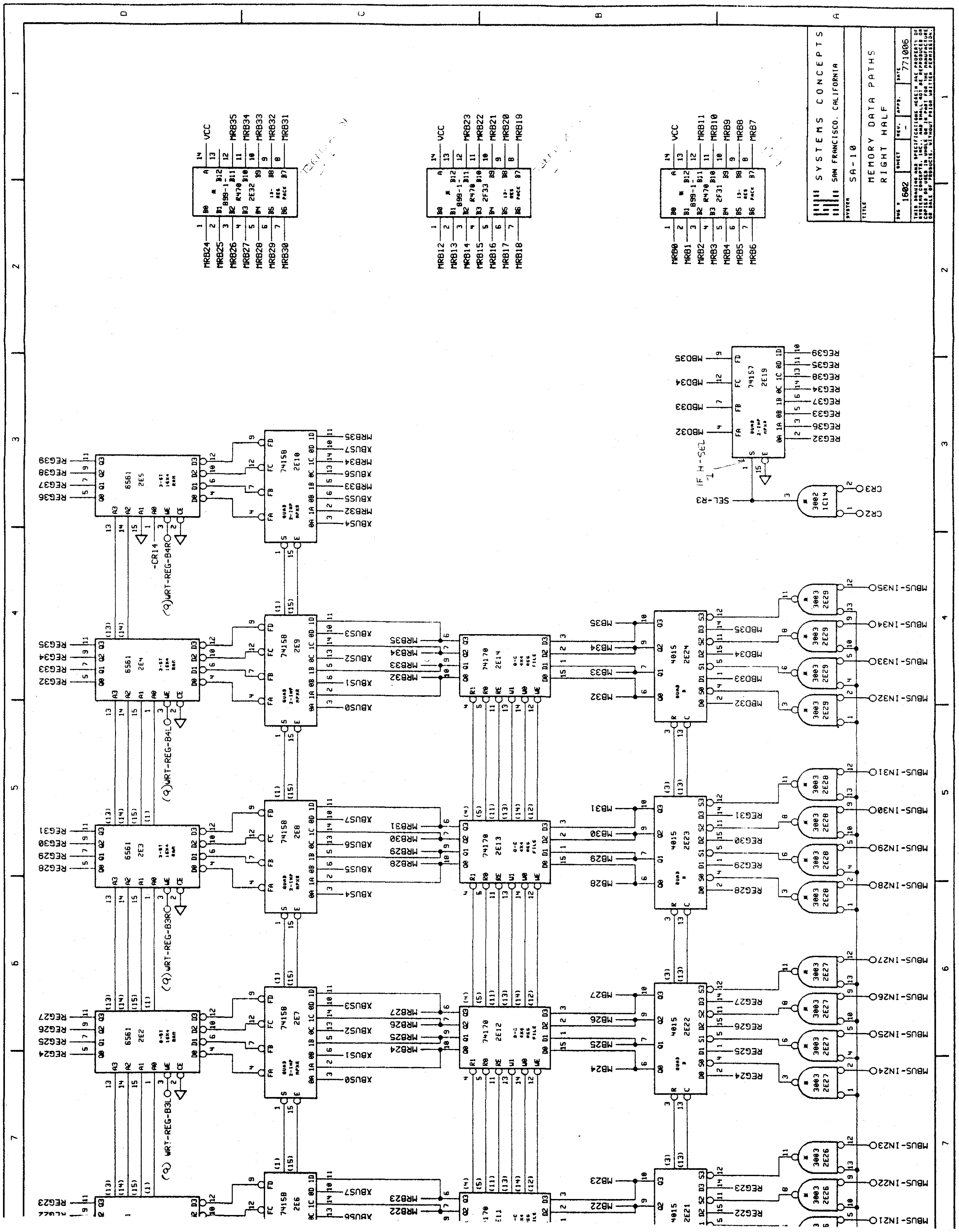
SYSTEMS CONCEPTS BLOCK DIAGRAM CONVENTIONS

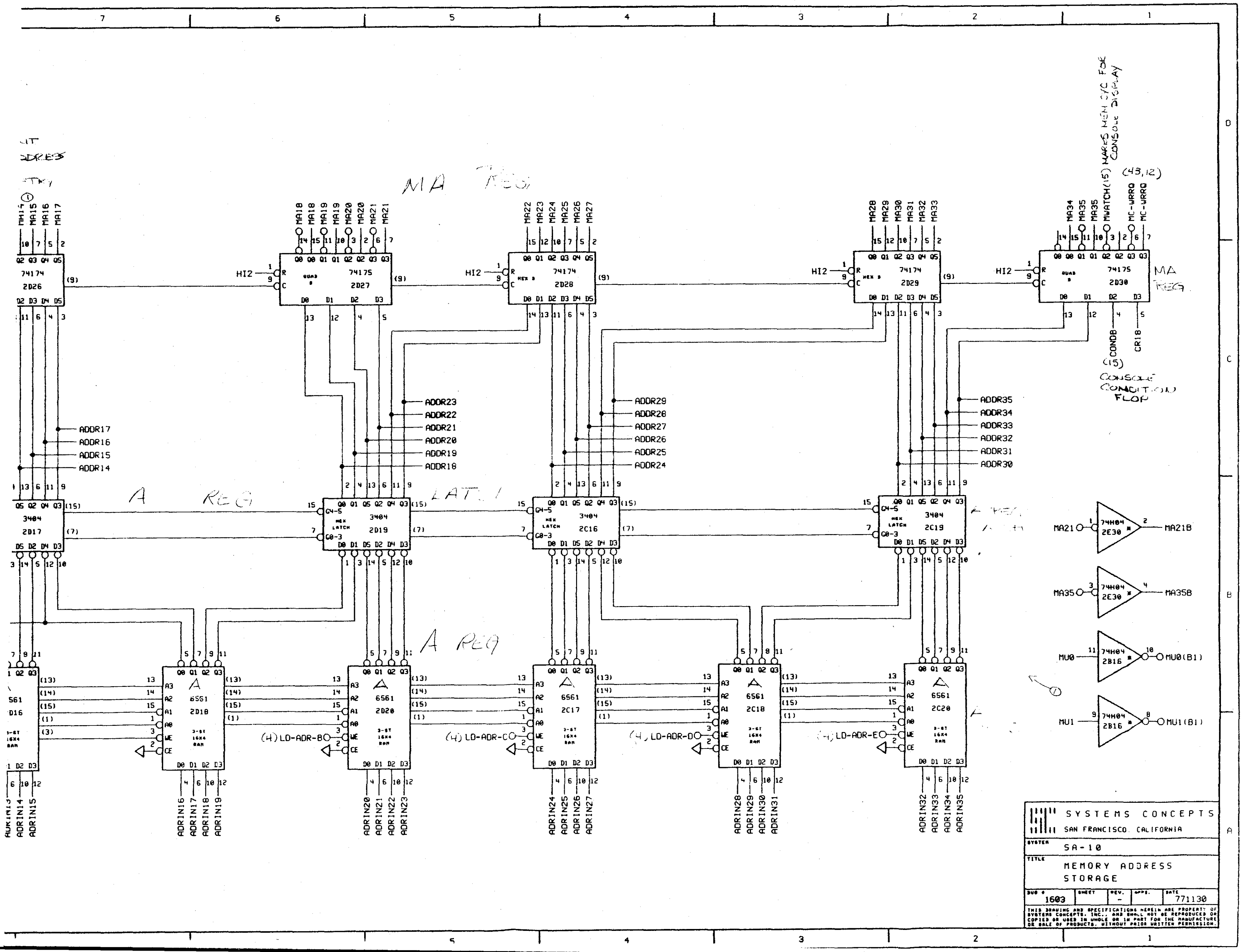


ENGINEERING DRAWINGS
SA-10
SUBSYSTEM ADAPTOR



REG0 REG1 REG2 REG3
REG4 REG5 REG6 REG7
REG8 REG9 REG10 REG11
REG12 REG13 REG14 REG15
REG16 REG17 REG18 REG19
6561 2F1
6561 2F2
6561 2F3
6561 2F4
6561 2F5
SUB CHAN ACT
SCA0(B1)
SCA1(B1)
CR2(B1)
CR3(B1)
WRT-REG-B0L
WRT-REG-B0R
WRT-REG-B1L
WRT-REG-B1R
WRT-REG-B2L
RFLD
MB+MBUS-RT
MBUS-INP
PARITY BIT
3003 2B17
3060 2E25
MBP
LD-MB-RT
CLR-MB-RT
16 REG - 4/CHAN
REG+MRB
REG+CBUS
74158 2F6
74158 2F7
74158 2F8
74158 2F9
74158 2F10
XBUS0 XBUS1 XBUS2 XBUS3
XBUS4 XBUS5 XBUS6 XBUS7
MRB0 MRB1 MRB2 MRB3
MRB4 MRB5 MRB6 MRB7
MRB8 MRB9 MRB10 MRB11
MRB12 MRB13 MRB14 MRB15
MRB16 MRB17 MRB18 MRB19
9348 2E18
9348 2F20
9348 2F17
MB-PAR
SUB CHAN ACTIVE
LST CHAN TO USE MEM P3
MU0(B1)
MU1(B1)
LD-MRB
MRB
74170 2F11
74170 2F12
74170 2F13
74170 2F14
74170 2F15
MRB MEMORY READ BUFFER
MB3 MB4 MB5 MB6 MB7 MB8 MB9 MB10 MB11 MB15
MEM BUS p44
CLR-MB-LT
LD-MB-LT
4015 2F21
4015 2F22
4015 2F23
4015 2F24
4015 2F25
MB
CLOCKED INPUT
DIRECT SET INPUT
3003 2F26
3003 2F27
3003 2F28
3003 2F29
3003 2F30
MB+MBUS-LT
FROM LG 12
MBUS-IN0 MBUS-IN1 MBUS-IN2 MBUS-IN3
MBUS-IN4 MBUS-IN5 MBUS-IN6 MBUS-IN7
MBUS-IN8 MBUS-IN9 MBUS-IN10 MBUS-IN11
MBUS-IN12 MBUS-IN13 MBUS-IN14 MBUS-IN15
MBUS-IN16 MBUS-IN17 MBUS-IN18 MBUS-IN19
MEMORY BUS p44


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA
SA-10
MEMORY DATA PATHS
RIGHT HALF
1602
771006
THIS DRAWING AND SPECIFICATIONS HEREIN ARE PROPERTY OF SYSTEMS CONCEPTS, INC. AND SHALL NOT BE REPRODUCED OR COPIED OR USED IN WHOLE OR IN PART FOR THE MANUFACTURE OR SALE OF PRODUCTS WITHOUT PRIOR WRITTEN PERMISSION.
74158
74157
74170
4015
6561
3003
899-1-R470
WRT-REG-B4R
WRT-REG-B4L
WRT-REG-B3R
WRT-REG-B3L
SEL-R3
IF H-SEL

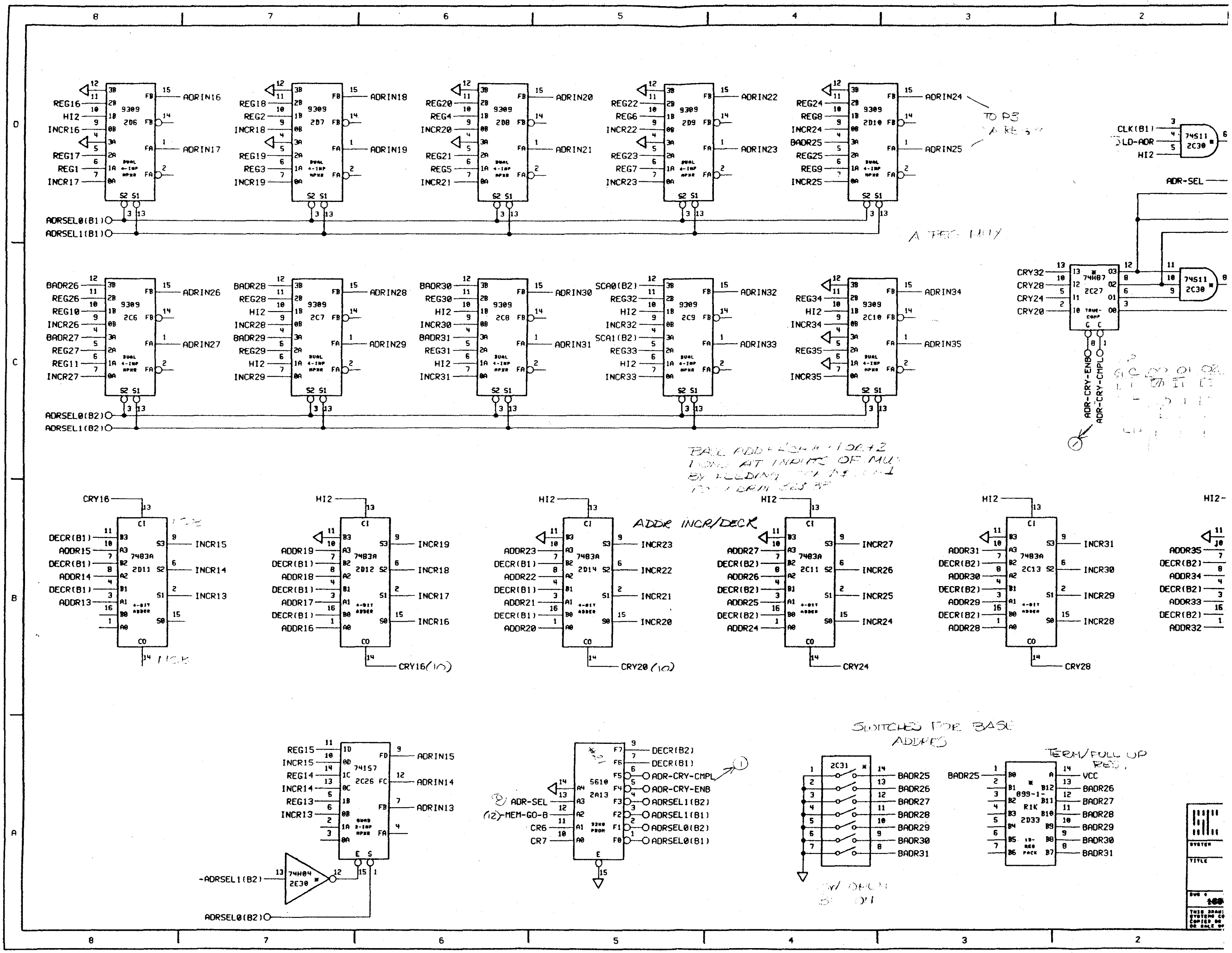


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

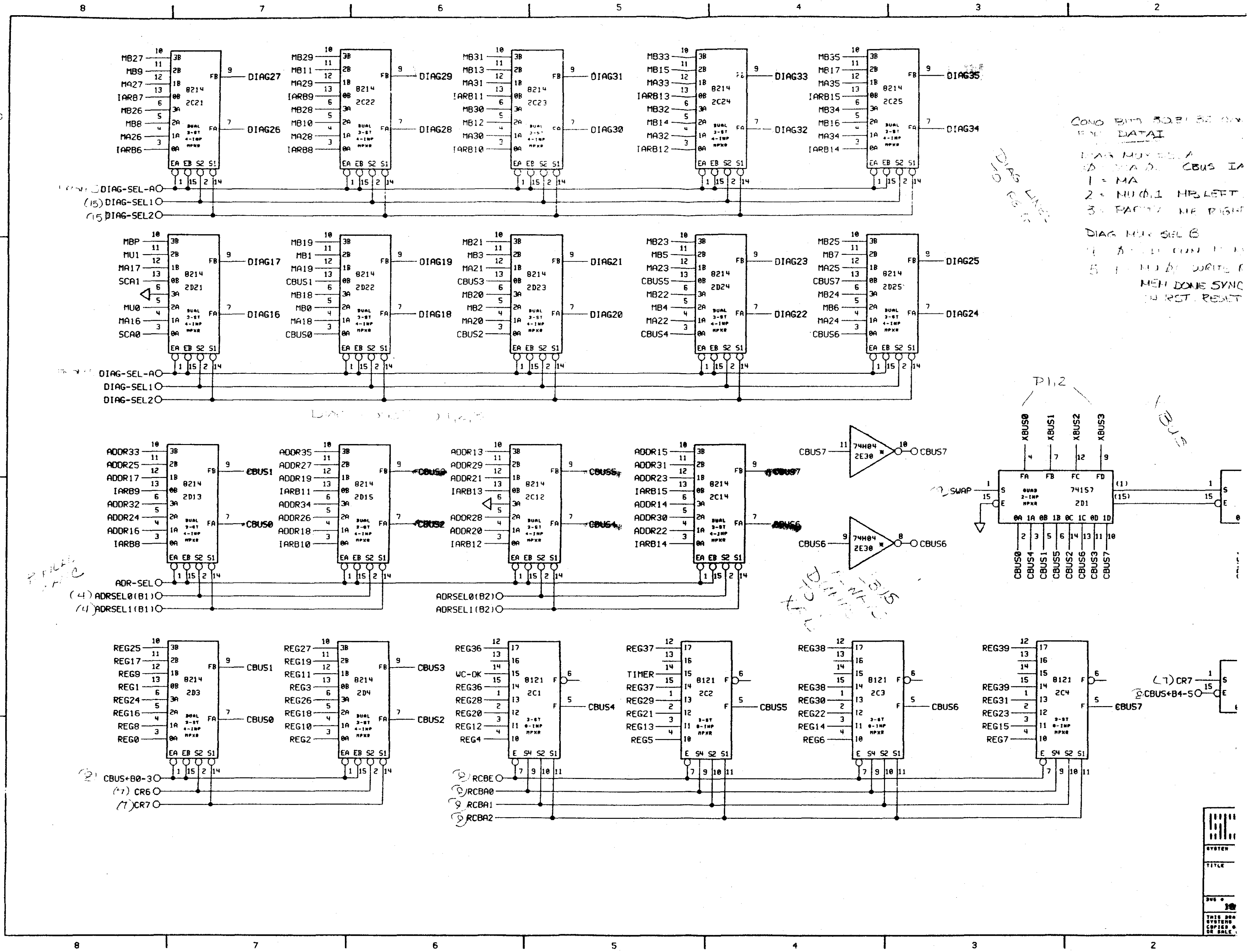
SYSTEM: SA-10

TITLE: MEMORY ADDRESS STORAGE

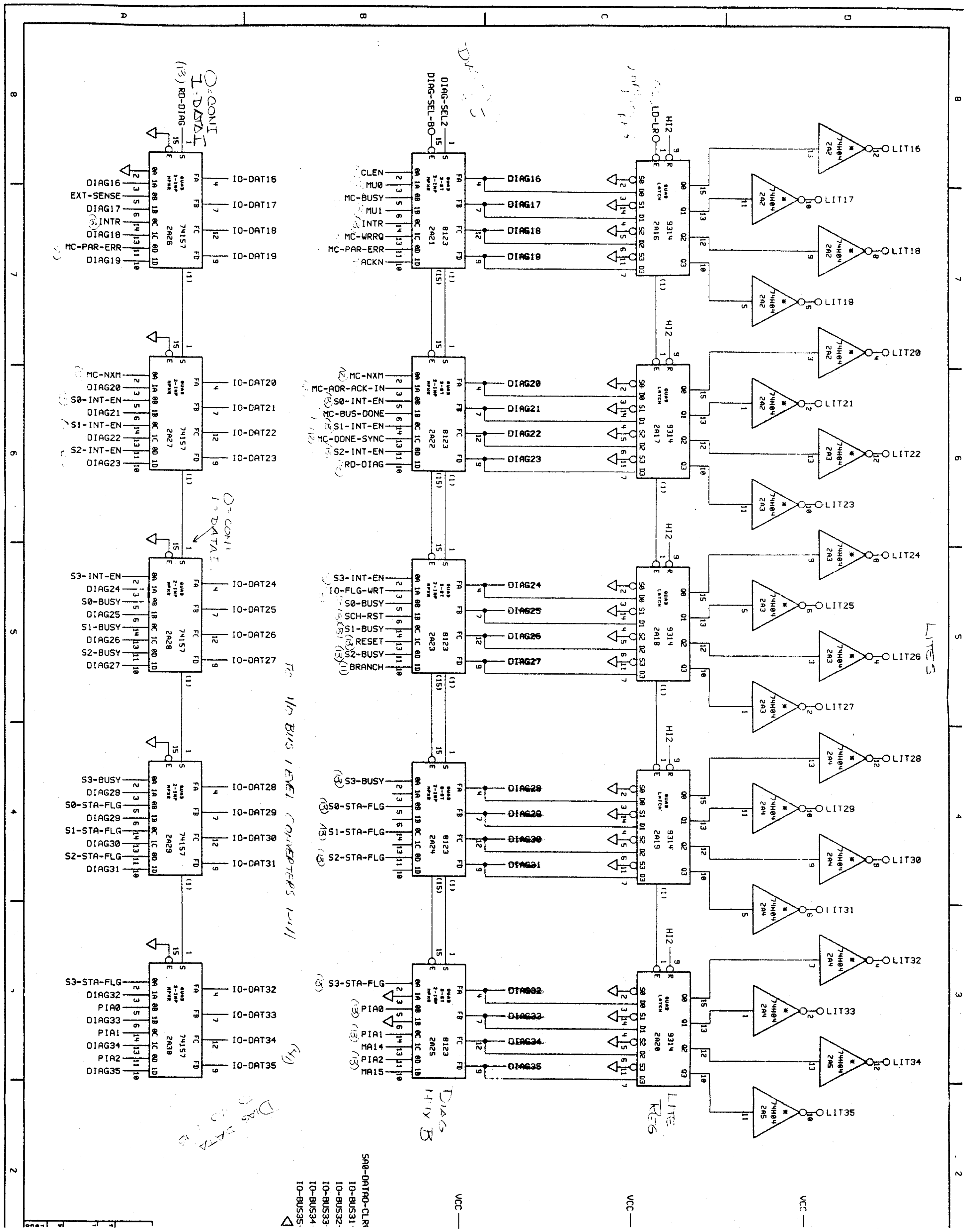
DWG #	SHEET	REV.	APPD.	DATE
1603		-		771130



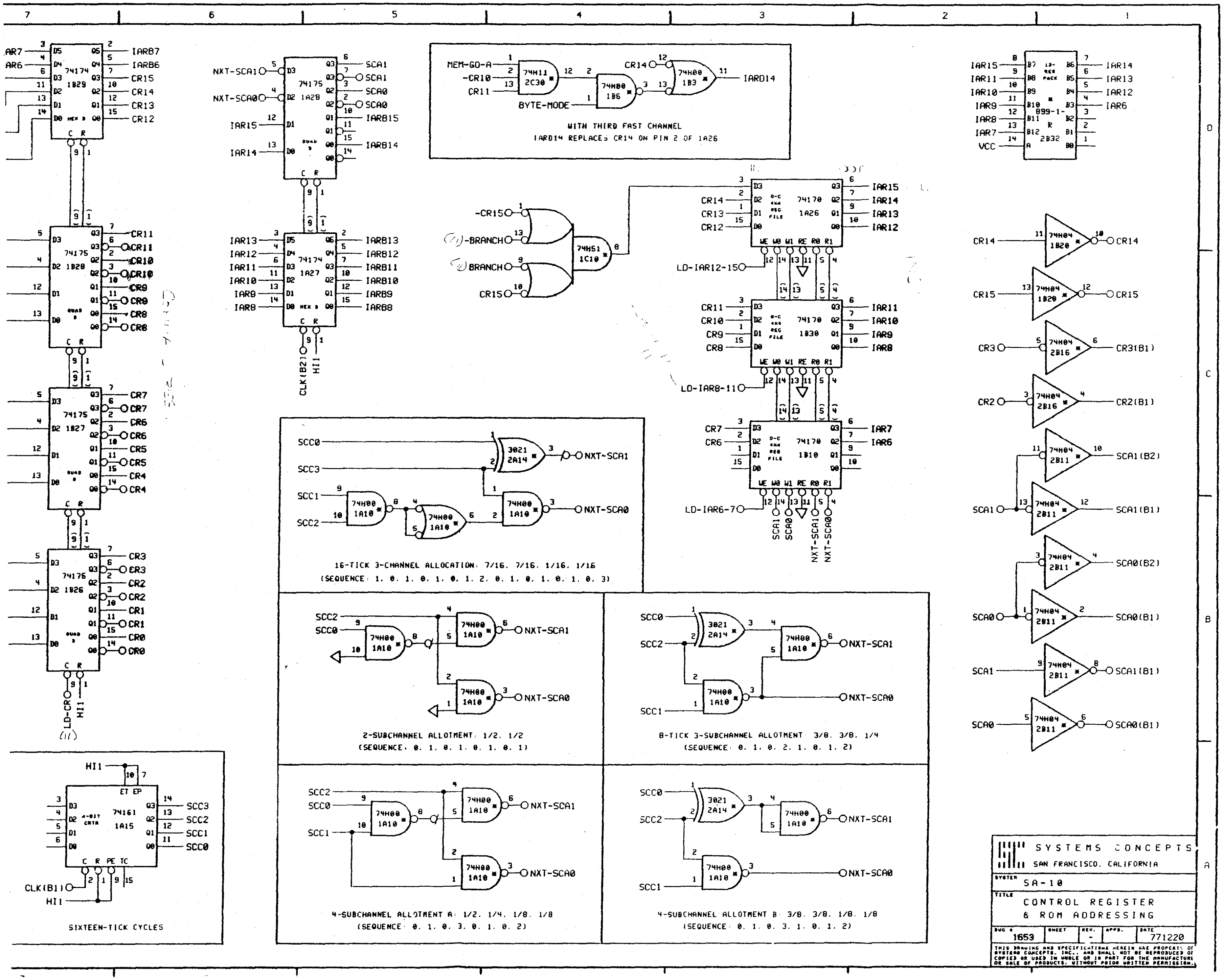
8 7 6 5 4 3 2
D C B A
REG16 HI2 INCR16 REG17 REG1 INCR17 9309 2D6 DUAL 4-INP MPXR ADRIN16 ADRIN17
REG18 REG2 INCR18 REG19 REG3 INCR19 9309 2D7 ADRIN18 ADRIN19
REG20 REG4 INCR20 REG21 REG5 INCR21 9309 2D8 ADRIN20 ADRIN21
REG22 REG6 INCR22 REG23 REG7 INCR23 9309 2D9 ADRIN22 ADRIN23
REG24 REG8 INCR24 BADR25 REG25 REG9 INCR25 9309 2D10 ADRIN24 ADRIN25
TO P3 A REG
ADRSEL0(B1) ADRSEL1(B1)
A REG MUX
CLK(B1) LD-ADR HI2 74S11 2C30
ADR-SEL
BADR26 REG26 REG10 INCR26 BADR27 REG27 REG11 INCR27 9309 2C6 ADRIN26 ADRIN27
BADR28 REG28 HI2 INCR28 BADR29 REG29 HI2 INCR29 9309 2C7 ADRIN28 ADRIN29
BADR30 REG30 HI2 INCR30 BADR31 REG31 HI2 INCR31 9309 2C8 ADRIN30 ADRIN31
SCA0(B2) REG32 HI2 INCR32 SCA1(B2) REG33 HI2 INCR33 9309 2C9 ADRIN32 ADRIN33
REG34 HI2 INCR34 REG35 INCR35 9309 2C10 ADRIN34 ADRIN35
CRY32 CRY28 CRY24 CRY20 74H87 2C27 TRUE-COMP 74S11 2C30
ADR-CRY-ENB ADR-CRY-CMPL
ADRSEL0(B2) ADRSEL1(B2)
BASE ADDRESS / DON'T AT INPUTS OF MUX BY FEEDING
ADDR INCR/DECR
CRY16 DECR(B1) ADDR15 DECR(B1) ADDR14 DECR(B1) ADDR13 7483A 2D11 4-BIT ADDER INCR15 INCR14 INCR13
HI2 ADDR19 DECR(B1) ADDR18 DECR(B1) ADDR17 DECR(B1) ADDR16 7483A 2D12 INCR19 INCR18 INCR17 INCR16 CRY16
HI2 ADDR23 DECR(B1) ADDR22 DECR(B1) ADDR21 DECR(B1) ADDR20 7483A 2D14 INCR23 INCR22 INCR21 INCR20 CRY20
HI2 ADDR27 DECR(B2) ADDR26 DECR(B2) ADDR25 DECR(B2) ADDR24 7483A 2C11 INCR27 INCR26 INCR25 INCR24 CRY24
HI2 ADDR31 DECR(B2) ADDR30 DECR(B2) ADDR29 DECR(B2) ADDR28 7483A 2C13 INCR31 INCR30 INCR29 INCR28 CRY28
HI2 ADDR35 DECR(B2) ADDR34 DECR(B2) ADDR33 DECR(B2) ADDR32
REG15 INCR15 REG14 INCR14 REG13 INCR13 74157 2C26 QUAD 2-INP MPXR ADRIN15 ADRIN14 ADRIN13
-ADRSEL1(B2) 74H04 2E30 ADRSEL0(B2)
ADR-SEL MEM-GO-B CR6 CR7 5610 2A13 32X8 PROM DECR(B2) DECR(B1) ADR-CRY-CMPL ADR-CRY-ENB ADRSEL1(B2) ADRSEL1(B1) ADRSEL0(B2) ADRSEL0(B1)
SWITCHES FOR BASE ADDRES
2C31 BADR25 BADR26 BADR27 BADR28 BADR29 BADR30 BADR31
SW OPEN IS ON
TERM/PULL UP RES.
BADR25 899-1-R1K 2D33 13-RES PACK VCC BADR26 BADR27 BADR28 BADR29 BADR30 BADR31
SYSTEM TITLE



8
7
6
5
4
3
2
MB27
MB9
MA27
IARB7
MB26
MB8
MA26
IARB6
8214
2C21
DIAG27
DIAG26
MB29
MB11
MA29
IARB9
MB28
MB10
MA28
IARB8
8214
2C22
DIAG29
DIAG28
MB31
MB13
MA31
IARB11
MB30
MB12
MA30
IARB10
8214
2C23
DIAG31
DIAG30
MB33
MB15
MA33
IARB13
MB32
MB14
MA32
IARB12
8214
2C24
DIAG33
DIAG32
MB35
MB17
MA35
IARB15
MB34
MB16
MA34
IARB14
8214
2C25
DIAG35
DIAG34
DIAG-SEL-A
(15) DIAG-SEL1
(15) DIAG-SEL2
DUAL 3-ST 4-INP MPXR
MBP
MU1
MA17
SCA1
MU0
MA16
SCA0
8214
2D21
DIAG17
DIAG16
MB19
MB1
MA19
CBUS1
MB18
MB0
MA18
CBUS0
8214
2D22
DIAG19
DIAG18
MB21
MB3
MA21
CBUS3
MB20
MB2
MA20
CBUS2
8214
2D23
DIAG21
DIAG20
MB23
MB5
MA23
CBUS5
MB22
MB4
MA22
CBUS4
8214
2D24
DIAG23
DIAG22
MB25
MB7
MA25
CBUS7
MB24
MB6
MA24
CBUS6
8214
2D25
DIAG25
DIAG24
DIAG-SEL-A
DIAG-SEL1
DIAG-SEL2
ADDR33
ADDR25
ADDR17
IARB9
ADDR32
ADDR24
ADDR16
IARB8
8214
2D13
CBUS1
CBUS0
ADDR35
ADDR27
ADDR19
IARB11
ADDR34
ADDR26
ADDR18
IARB10
8214
2D15
~~CBUS3~~
~~CBUS2~~
ADDR13
ADDR29
ADDR21
IARB13
ADDR28
ADDR20
IARB12
8214
2C12
~~CBUS5~~
~~CBUS4~~
ADDR15
ADDR31
ADDR23
IARB15
ADDR14
ADDR30
ADDR22
IARB14
8214
2C14
~~CBUS7~~
~~CBUS6~~
ADR-SEL
(4) ADRSEL0(B1)
(4) ADRSEL1(B1)
ADRSEL0(B2)
ADRSEL1(B2)
CBUS7
74H04
2E30
CBUS7
CBUS6
74H04
2E30
CBUS6
P1,2
XBUS0
XBUS1
XBUS2
XBUS3
SWAP
QUAD 2-INP MPXR
74157
2D1
CBUS0
CBUS4
CBUS1
CBUS5
CBUS2
CBUS6
CBUS3
CBUS7
REG25
REG17
REG9
REG1
REG24
REG16
REG8
REG0
8214
2D3
CBUS1
CBUS0
REG27
REG19
REG11
REG3
REG26
REG18
REG10
REG2
8214
2D4
CBUS3
CBUS2
REG36
WC-OK
REG36
REG28
REG20
REG12
REG4
8121
2C1
3-ST 8-INP MPXR
CBUS4
REG37
TIMER
REG37
REG29
REG21
REG13
REG5
8121
2C2
CBUS5
REG38
REG38
REG30
REG22
REG14
REG6
8121
2C3
CBUS6
REG39
REG39
REG31
REG23
REG15
REG7
8121
2C4
CBUS7
(7) CR7
CBUS+B4-5
(2) CBUS+B0-3
(7) CR6
(7) CR7
RCBE
RCBA0
RCBA1
RCBA2
COND BITS 30,31,32
DATAI
DIAG MUX SEL A
1 = MA
2 = MU 0,1 MB LEFT
3 = PARITY MB RIGHT
DIAG MUX SEL B
MEM DONE SYNC
RST RESET
P1,2



(13) RD-DIAG
0=CONI 1=DATAI
DIAG16
EXT-SENSE
DIAG17
INTR
DIAG18
MC-PAR-ERR
DIAG19
74157
2A26
IO-DAT16
IO-DAT17
IO-DAT18
IO-DAT19
MC-NXM
DIAG20
S0-INT-EN
DIAG21
S1-INT-EN
DIAG22
S2-INT-EN
DIAG23
2A27
IO-DAT20
IO-DAT21
IO-DAT22
IO-DAT23
S3-INT-EN
DIAG24
S0-BUSY
DIAG25
S1-BUSY
DIAG26
S2-BUSY
DIAG27
2A28
IO-DAT24
IO-DAT25
IO-DAT26
IO-DAT27
S3-BUSY
DIAG28
S0-STA-FLG
DIAG29
S1-STA-FLG
DIAG30
S2-STA-FLG
DIAG31
2A29
IO-DAT28
IO-DAT29
IO-DAT30
IO-DAT31
S3-STA-FLG
DIAG32
PIA0
DIAG33
PIA1
DIAG34
PIA2
DIAG35
2A30
IO-DAT32
IO-DAT33
IO-DAT34
IO-DAT35
To I/O BUS LEVEL CONVERTERS (4)
DIAG DATA
DIAG-SEL2
DIAG-SEL-B
CLEN
MU0
MC-BUSY
MU1
INTR
MC-WRRQ
MC-PAR-ERR
ACKN
8123
2A21
DIAG16
DIAG17
DIAG18
DIAG19
MC-NXM
MC-ADR-ACK-IN
S0-INT-EN
MC-BUS-DONE
S1-INT-EN
MC-DONE-SYNC
S2-INT-EN
RD-DIAG
2A22
DIAG20
DIAG21
DIAG22
DIAG23
S3-INT-EN
IO-FLG-WRT
S0-BUSY
SCH-RST
S1-BUSY
RESET
S2-BUSY
BRANCH
2A23
DIAG24
DIAG25
DIAG26
DIAG27
S3-BUSY
S0-STA-FLG
S1-STA-FLG
S2-STA-FLG
2A24
DIAG28
DIAG29
DIAG30
DIAG31
S3-STA-FLG
PIA0
PIA1
MA14
PIA2
MA15
2A25
DIAG32
DIAG33
DIAG34
DIAG35
DIAG MUX B
HI2
LD-LR
9314
QUAD LATCH
2A16
2A17
2A18
2A19
2A20
LITE REG
LITES
74H04
2A2
2A3
2A4
2A5
LIT16
LIT17
LIT18
LIT19
LIT20
LIT21
LIT22
LIT23
LIT24
LIT25
LIT26
LIT27
LIT28
LIT29
LIT30
LIT31
LIT32
LIT33
LIT34
LIT35
SA0-DATAO-CLR
IO-BUS31
IO-BUS32
IO-BUS33
IO-BUS34
IO-BUS35
VCC
VCC
VCC



16-TICK 3-CHANNEL ALLOCATION: 7/16, 7/16, 1/16, 1/16
(SEQUENCE: 1, 0, 1, 0, 1, 0, 1, 2, 0, 1, 0, 1, 0, 1, 0, 3)

2-SUBCHANNEL ALLOTMENT: 1/2, 1/2
(SEQUENCE: 0, 1, 0, 1, 0, 1, 0, 1)

8-TICK 3-SUBCHANNEL ALLOTMENT: 3/8, 3/8, 1/4
(SEQUENCE: 0, 1, 0, 2, 1, 0, 1, 2)

SIXTEEN-TICK CYCLES

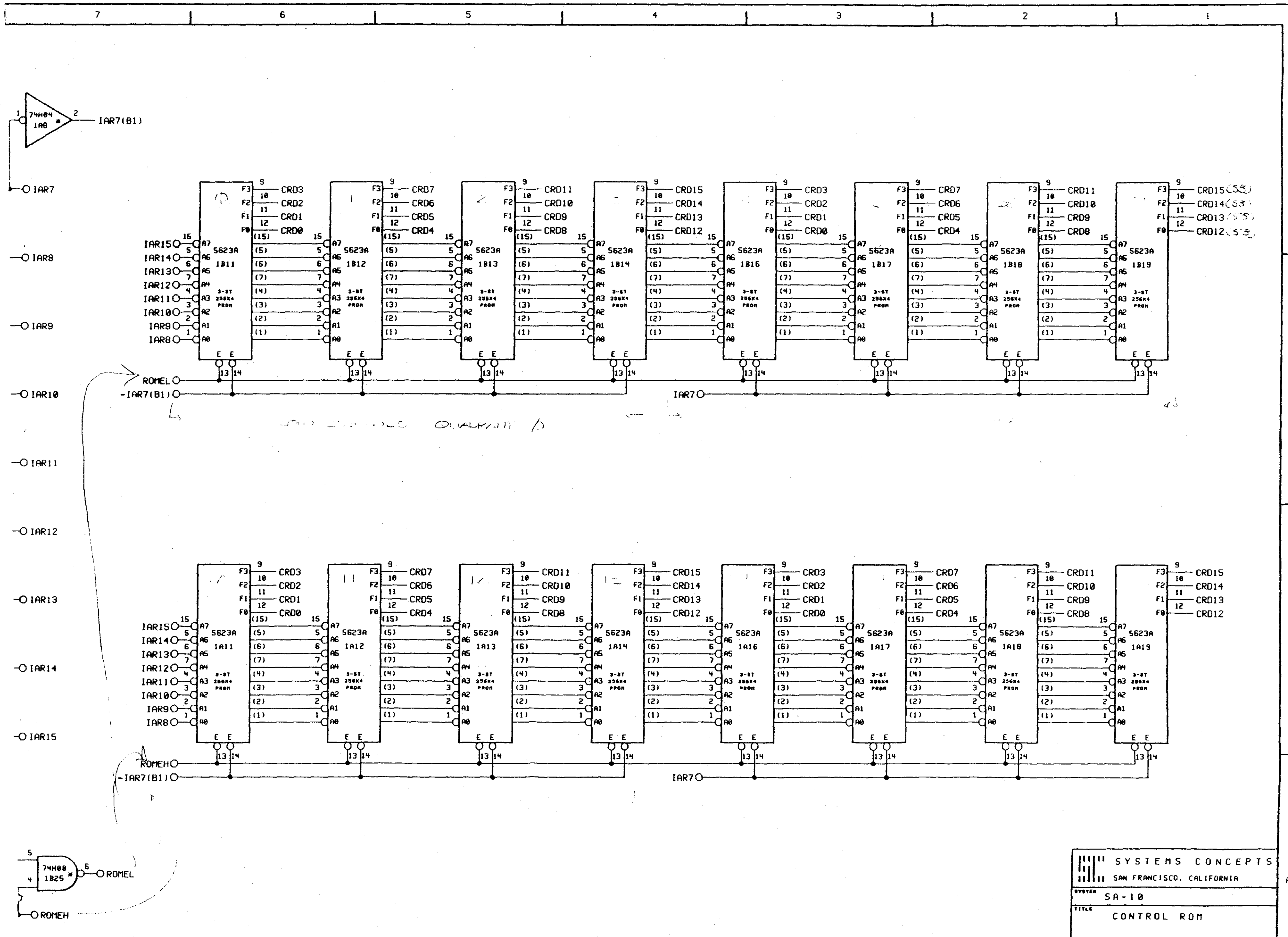
4-SUBCHANNEL ALLOTMENT A: 1/2, 1/4, 1/8, 1/8
(SEQUENCE: 0, 1, 0, 3, 0, 1, 0, 2)

4-SUBCHANNEL ALLOTMENT B: 3/8, 3/8, 1/8, 1/8
(SEQUENCE: 0, 1, 0, 3, 1, 0, 1, 2)

SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM	SA-10
TITLE	CONTROL REGISTER & ROM ADDRESSING

DWG #	SHEET	REV.	APPD.	DATE
1653		-		771220

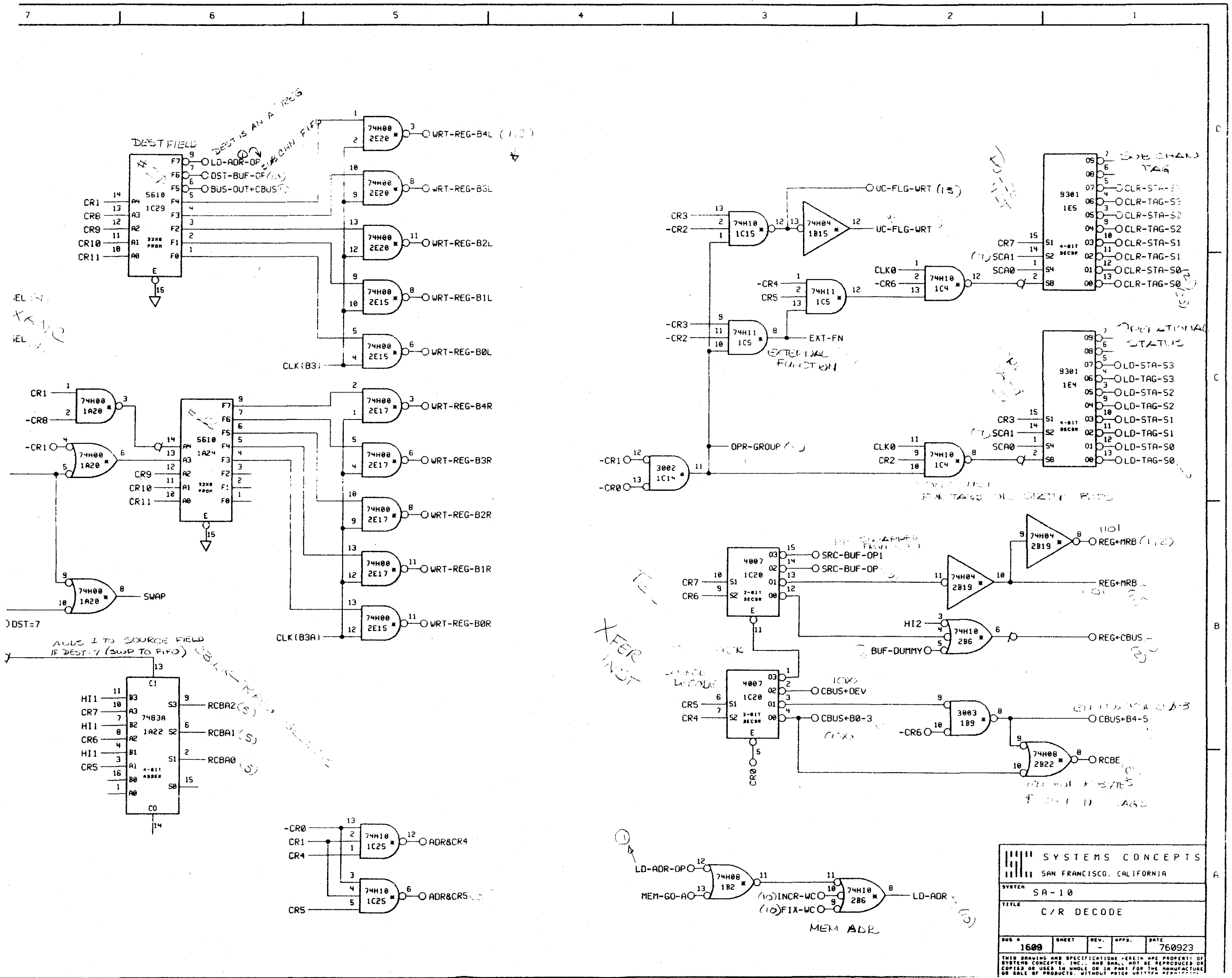


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM
SA-10

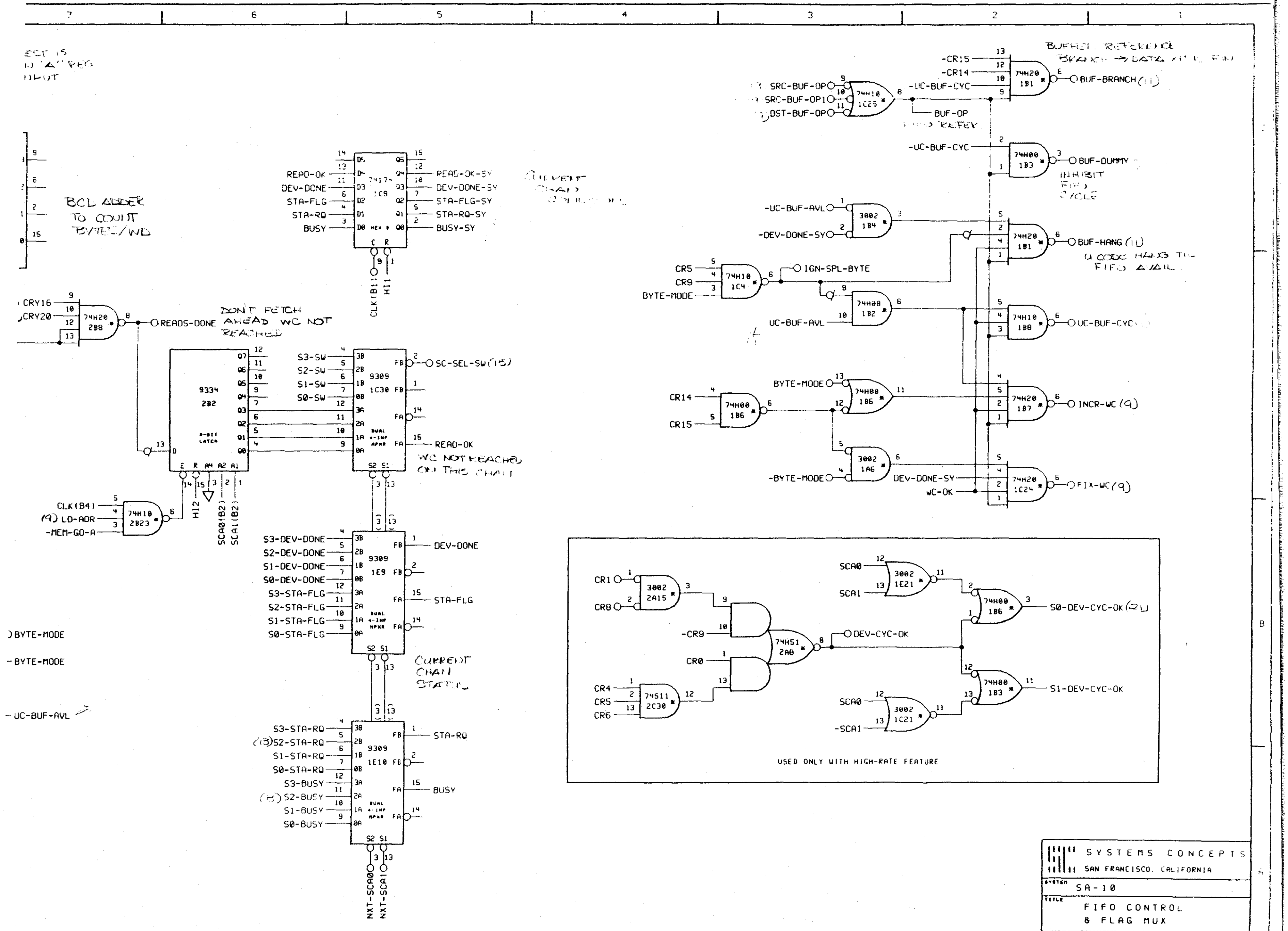
TITLE
CONTROL ROM

DWG #	SHEET	REV.	APPD.	DATE
1600		-		760923



SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

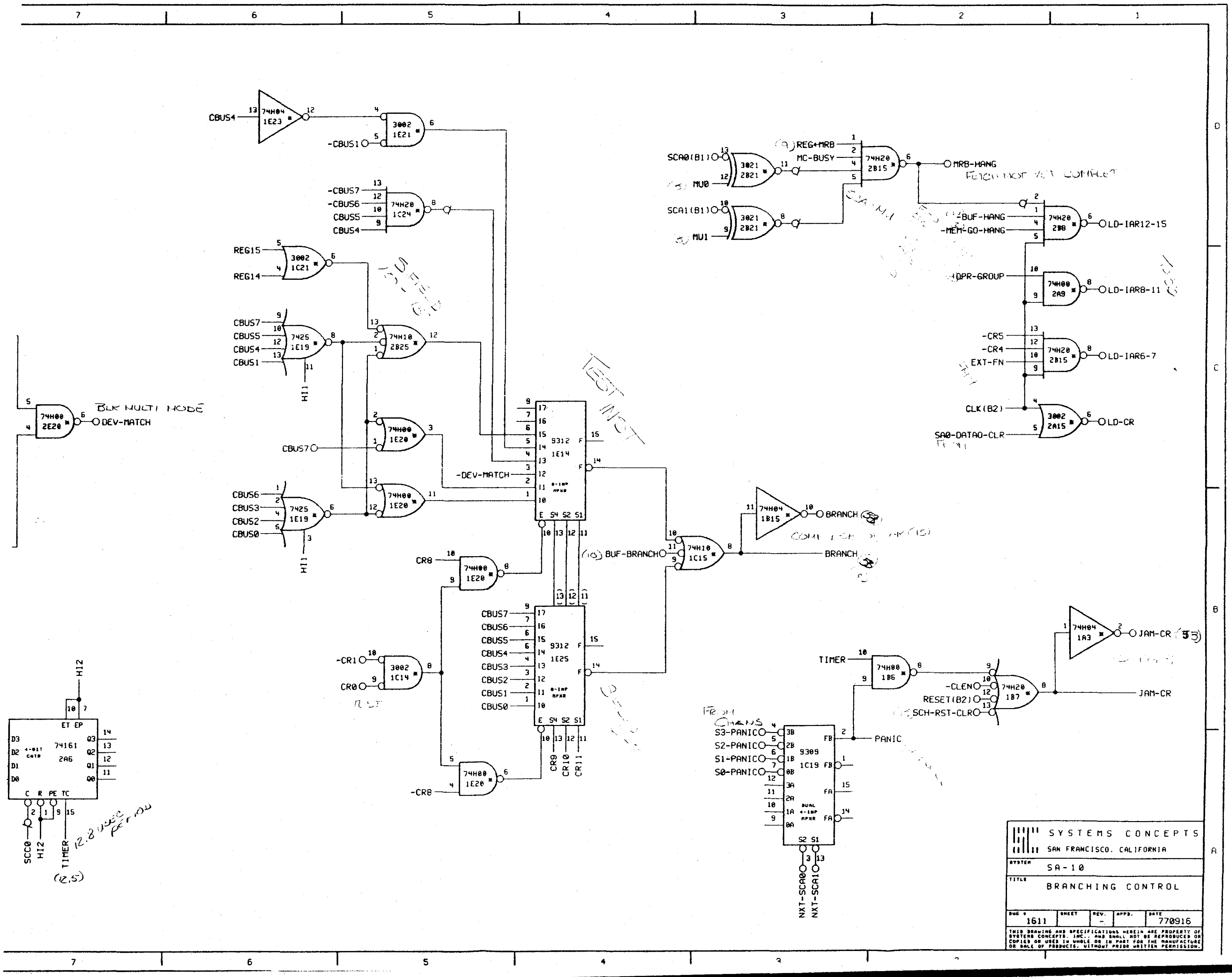
SYSTEM	SA-10
TITLE	C/R DECODE
DWG #	1609
SHEET	
REV.	-
DATE	760923



SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM	SA-10
TITLE	FIFO CONTROL & FLAG MUX

DWG #	SHEET	REV.	APPD.	DATE
1610		-		770321

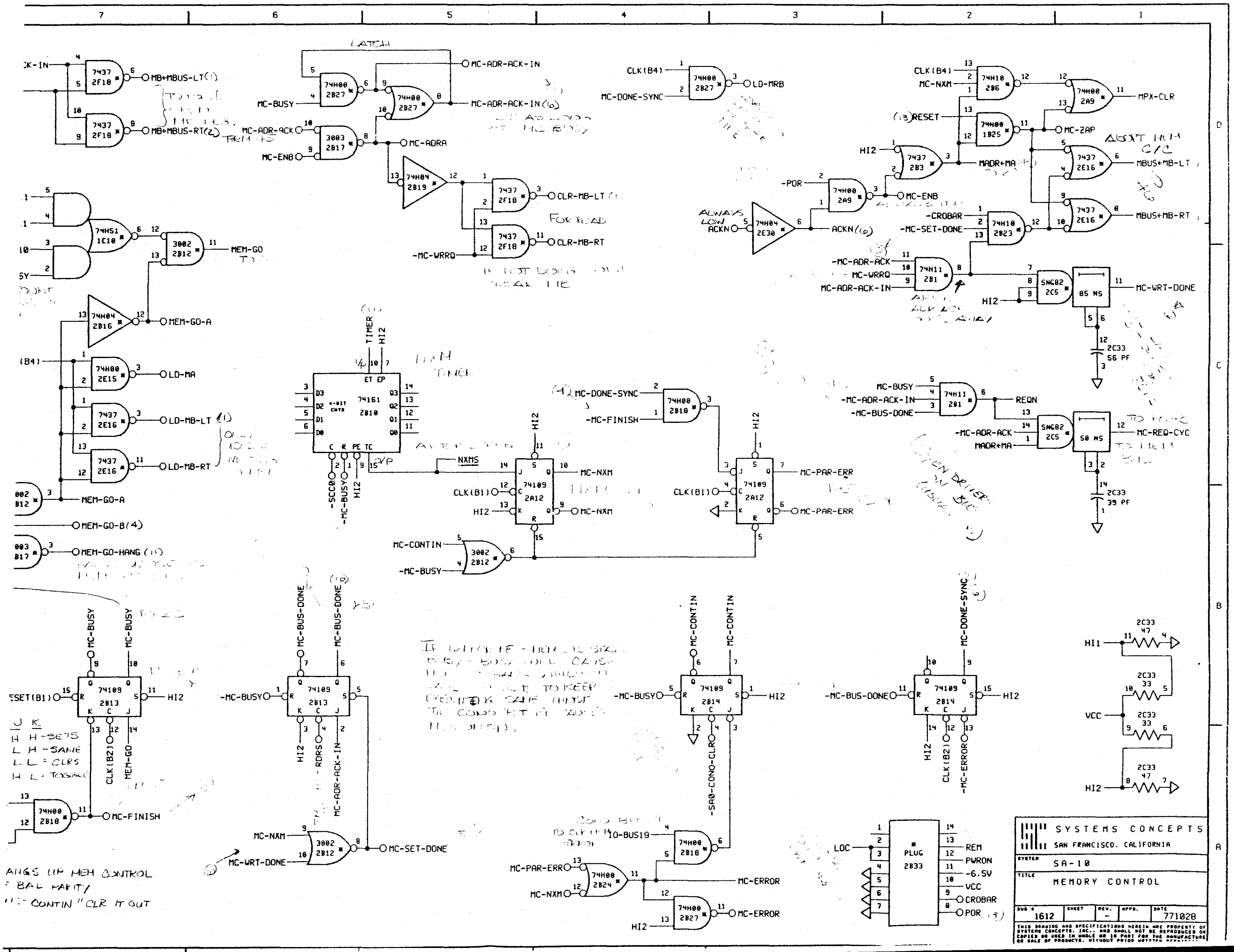


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM
SA-10

TITLE
BRANCHING CONTROL

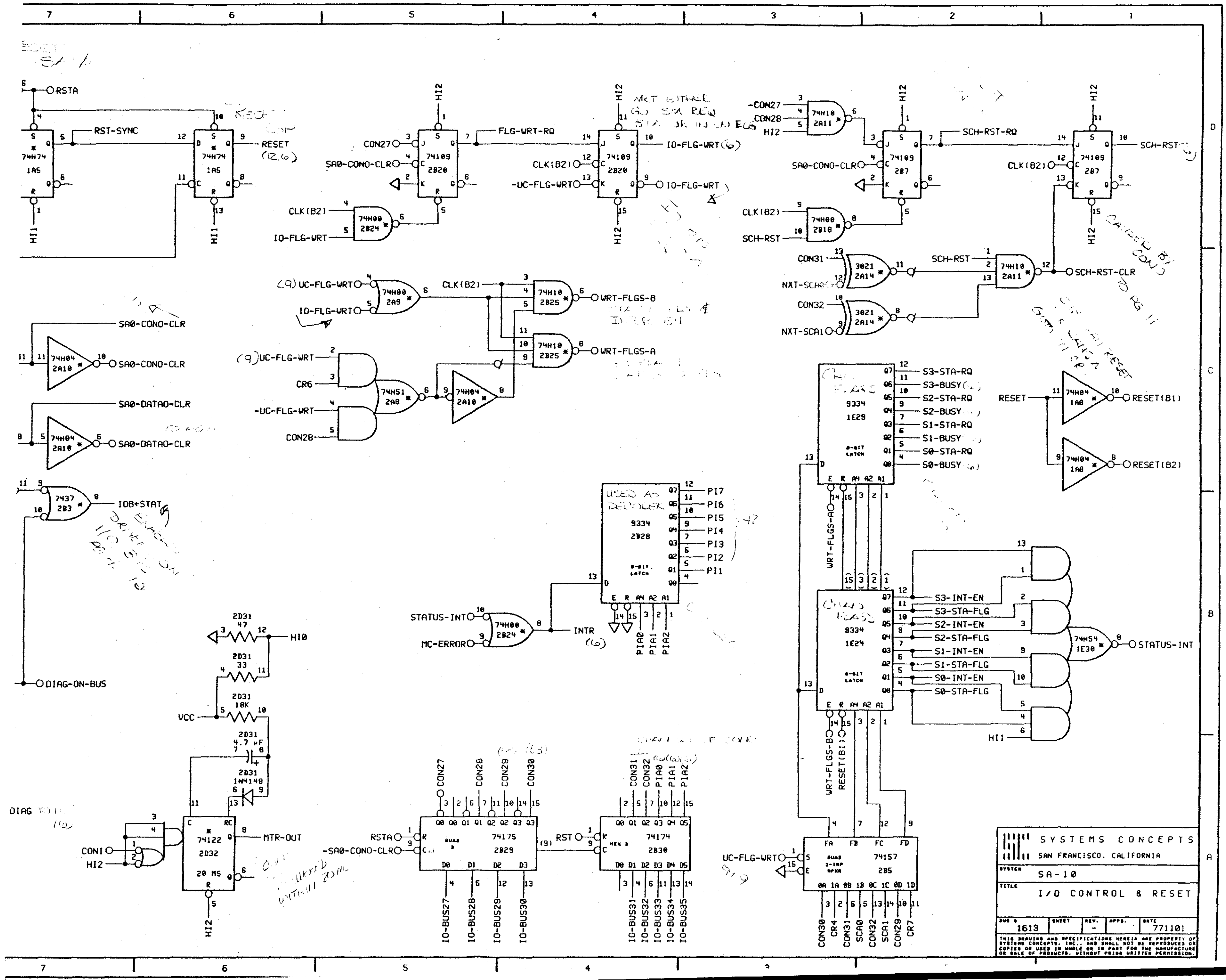
DWG #	SHEET	REV.	APPD.	DATE
1611		-		770916



SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM	SA-10
TITLE	MEMORY CONTROL

DWG #	SHEET	REV.	APPD.	DATE
1612		-		771028



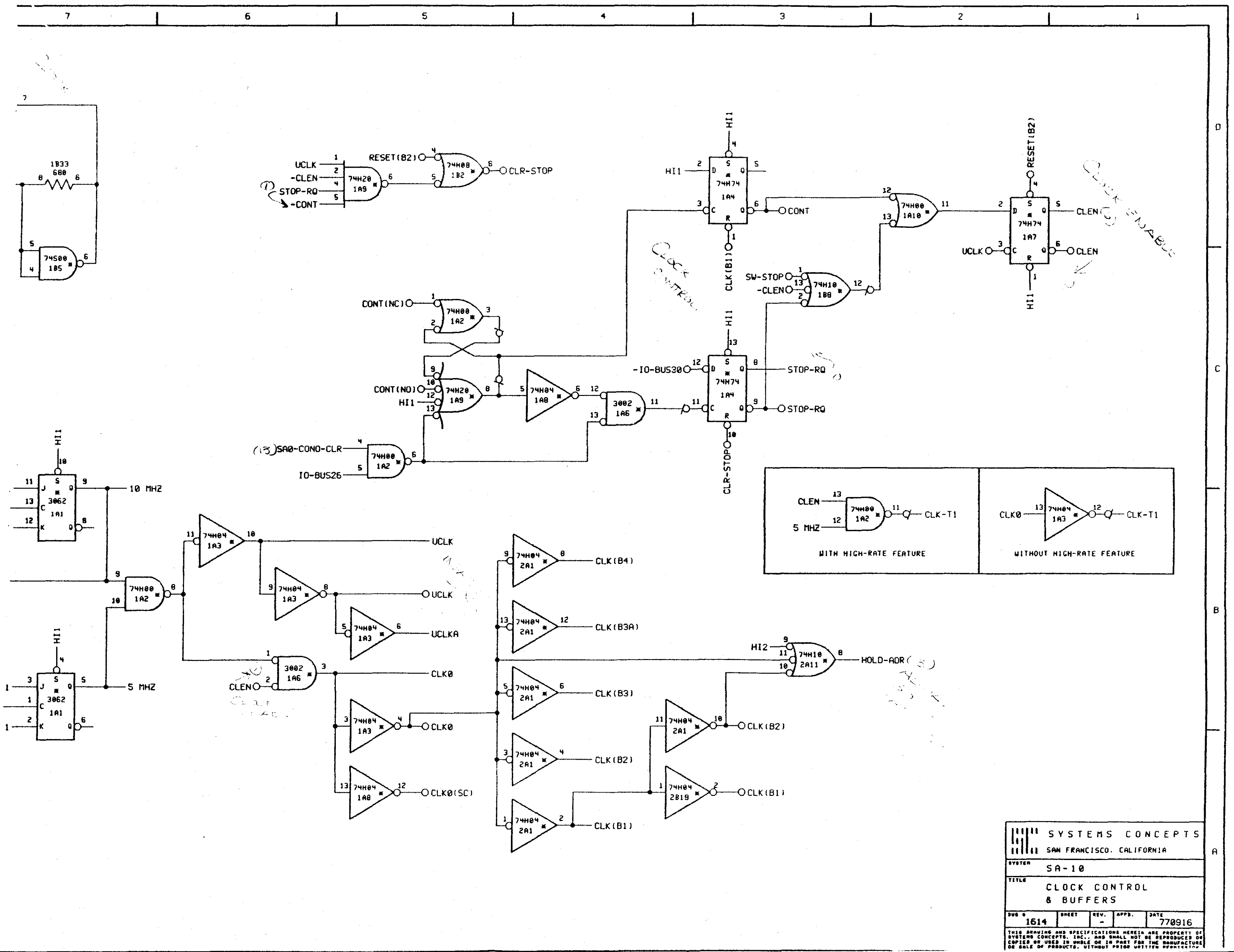
SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM SA-10

TITLE I/O CONTROL & RESET

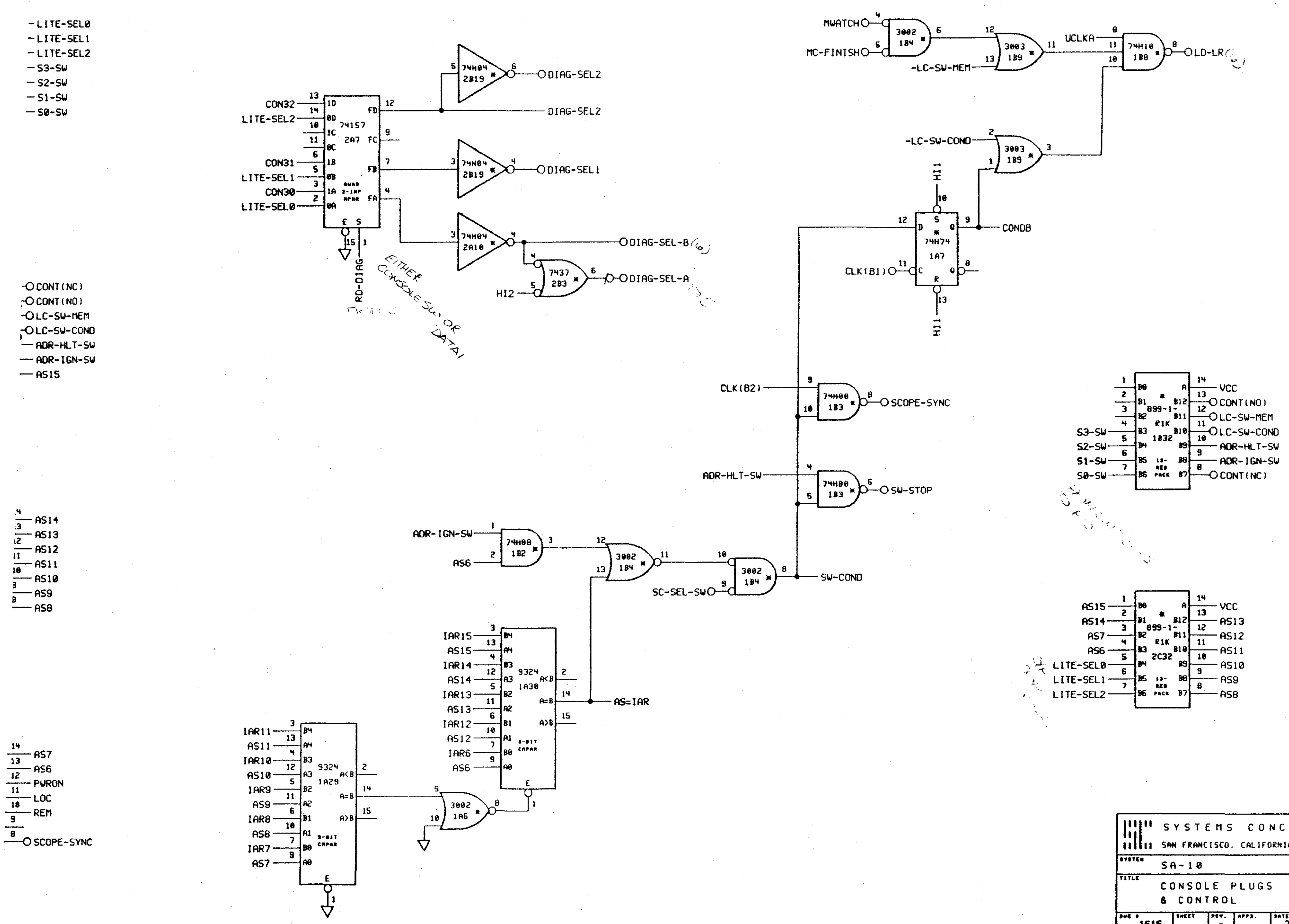
DWG #	SHEET	REV.	APPD.	DATE
1613		-		771101

THIS DRAWING AND SPECIFICATIONS HEREIN ARE PROPERTY OF SYSTEMS CONCEPTS, INC., AND SHALL NOT BE REPRODUCED OR COPIED OR USED IN WHOLE OR IN PART FOR THE MANUFACTURE OR SALE OF PRODUCTS, WITHOUT PRIOR WRITTEN PERMISSION.



SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM	SA-10
TITLE	CLOCK CONTROL & BUFFERS
DWG #	1614
SHEET	
REV.	-
APPD.	
DATE	770916

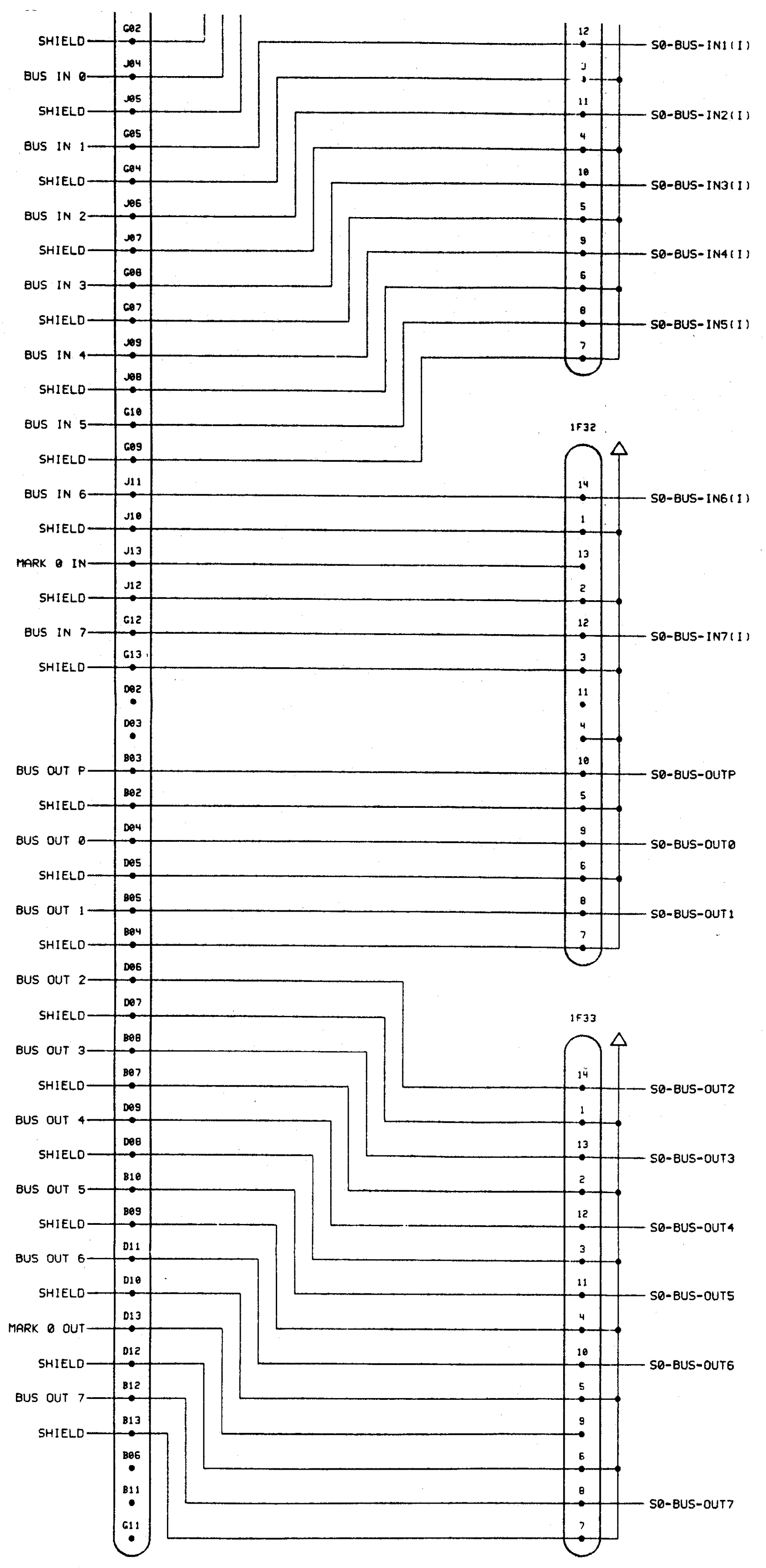


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM: SA-10

TITLE: CONSOLE PLUGS & CONTROL

DWG #	SHEET	REV.	APPD.	DATE
1615		-		771102

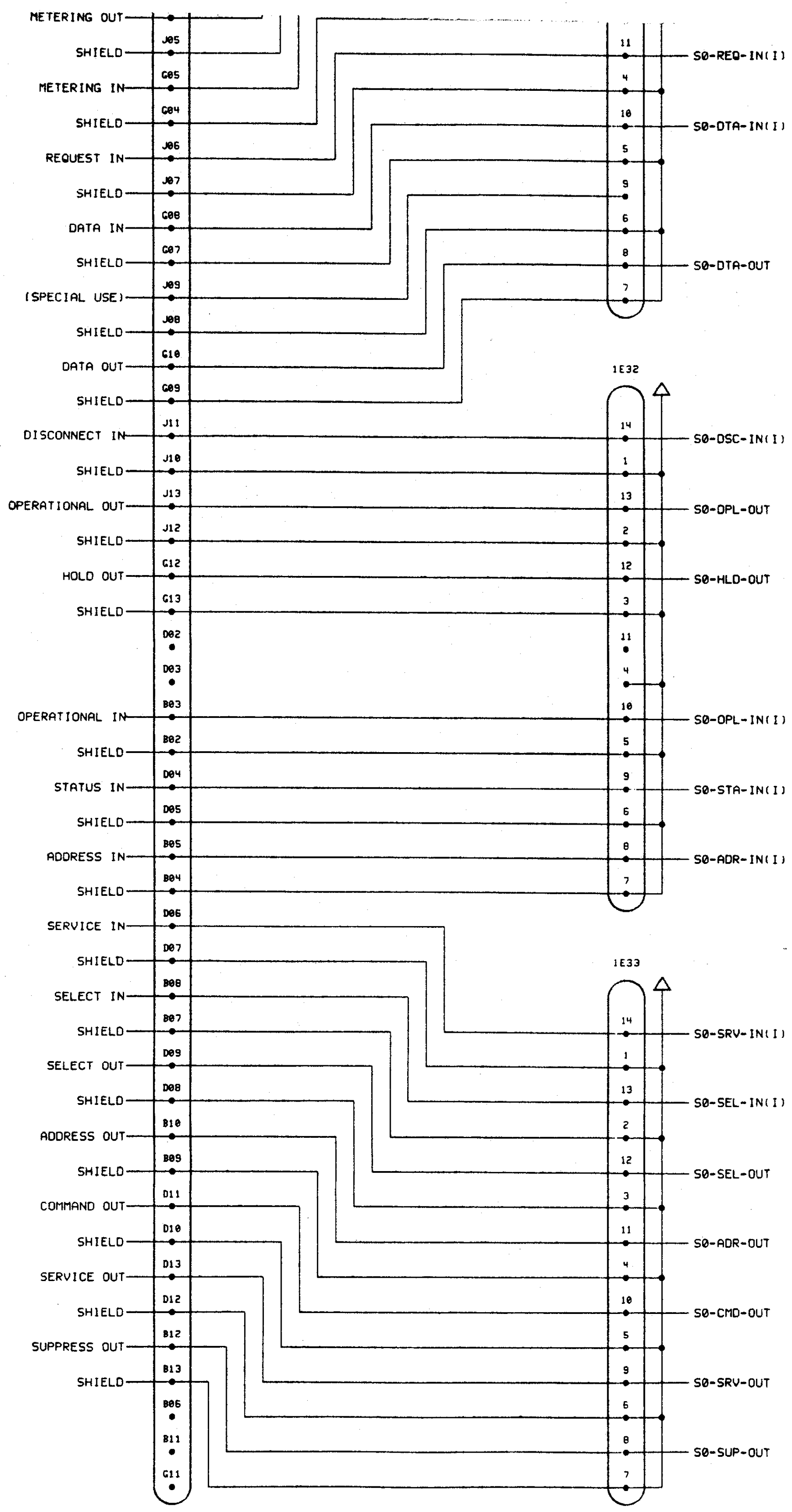


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM SA-10

TITLE SUB CHANNEL 0
IBM BUS INTERFACE

DWG #	SHEET	REV.	APPD.	DATE
1618		-		760914



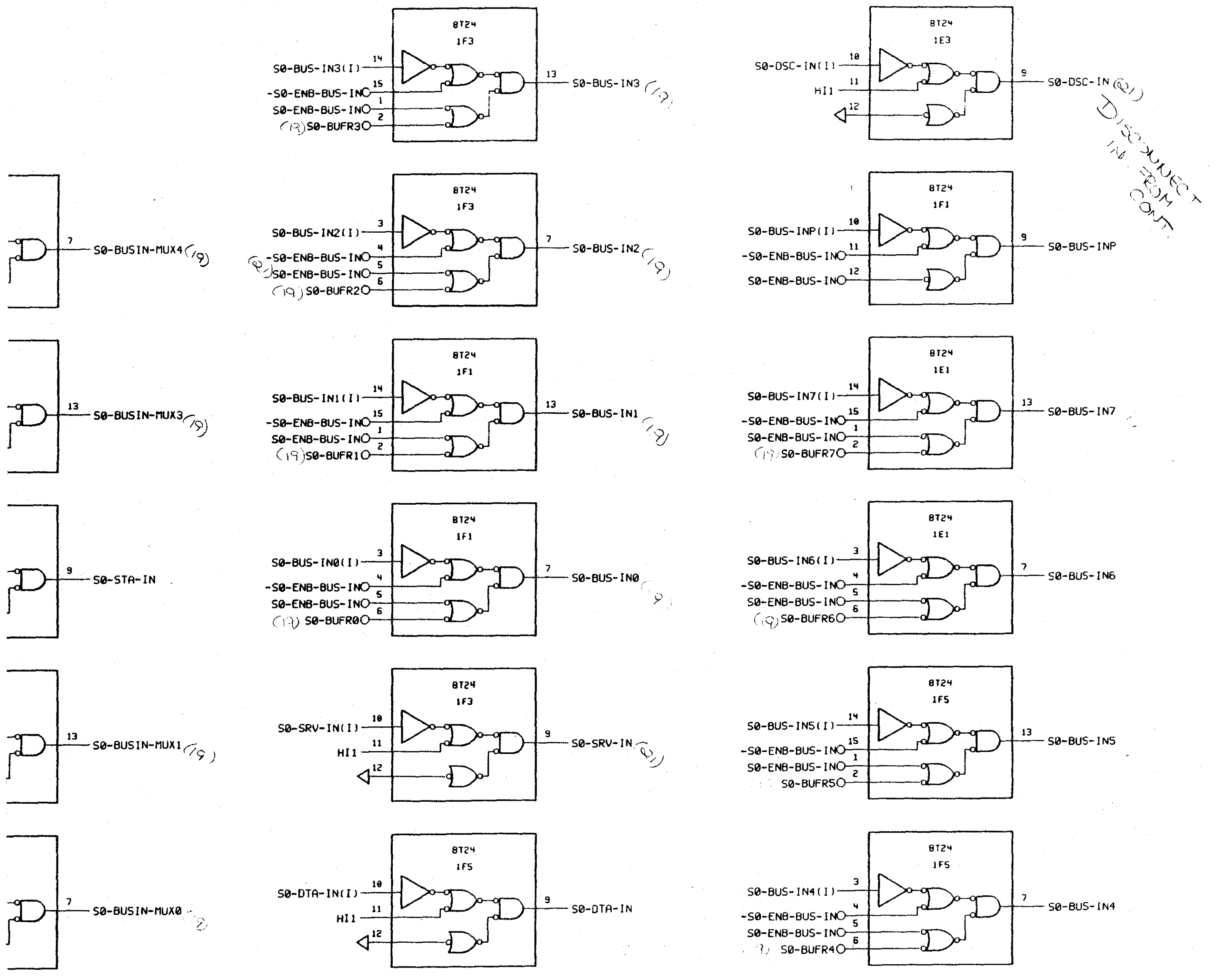
SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM SA-10

TITLE SUB CHANNEL 0
IBM TAG INTERFACE

DWG #	SHEET	REV.	APPR.	DATE
1617		-		760917

THIS DRAWING AND SPECIFICATIONS HEREIN ARE PROPERTY OF SYSTEMS CONCEPTS, INC. AND SHALL NOT BE REPRODUCED OR COPIED OR USED IN WHOLE OR IN PART FOR THE MANUFACTURE OR SALE OF PRODUCTS WITHOUT PRIOR WRITTEN PERMISSION.

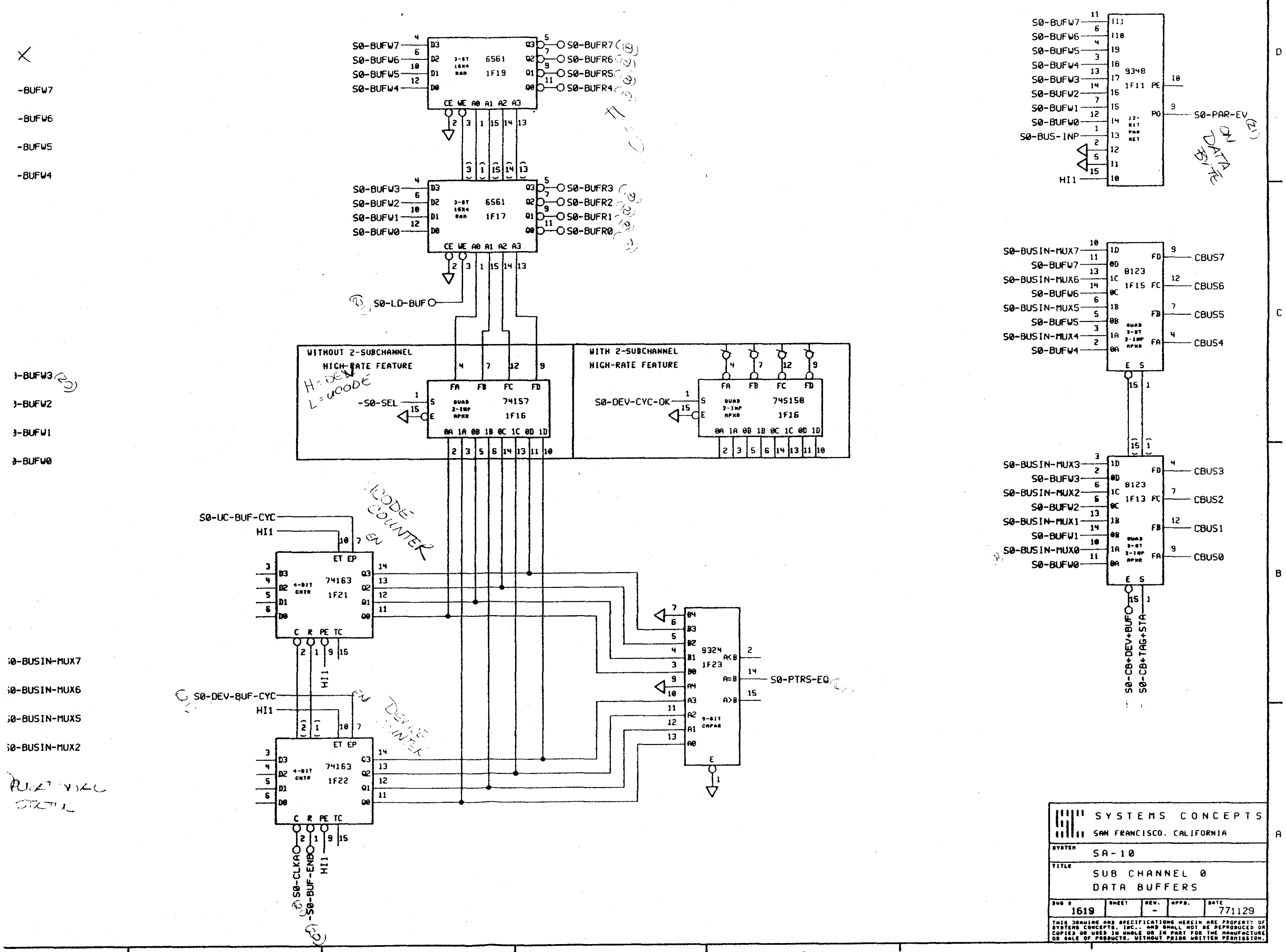


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM: SA-10

TITLE: SUB CHANNEL 0 RECEIVERS

DWG #	SHEET	REV.	APPD.	DATE
1618		-		770321

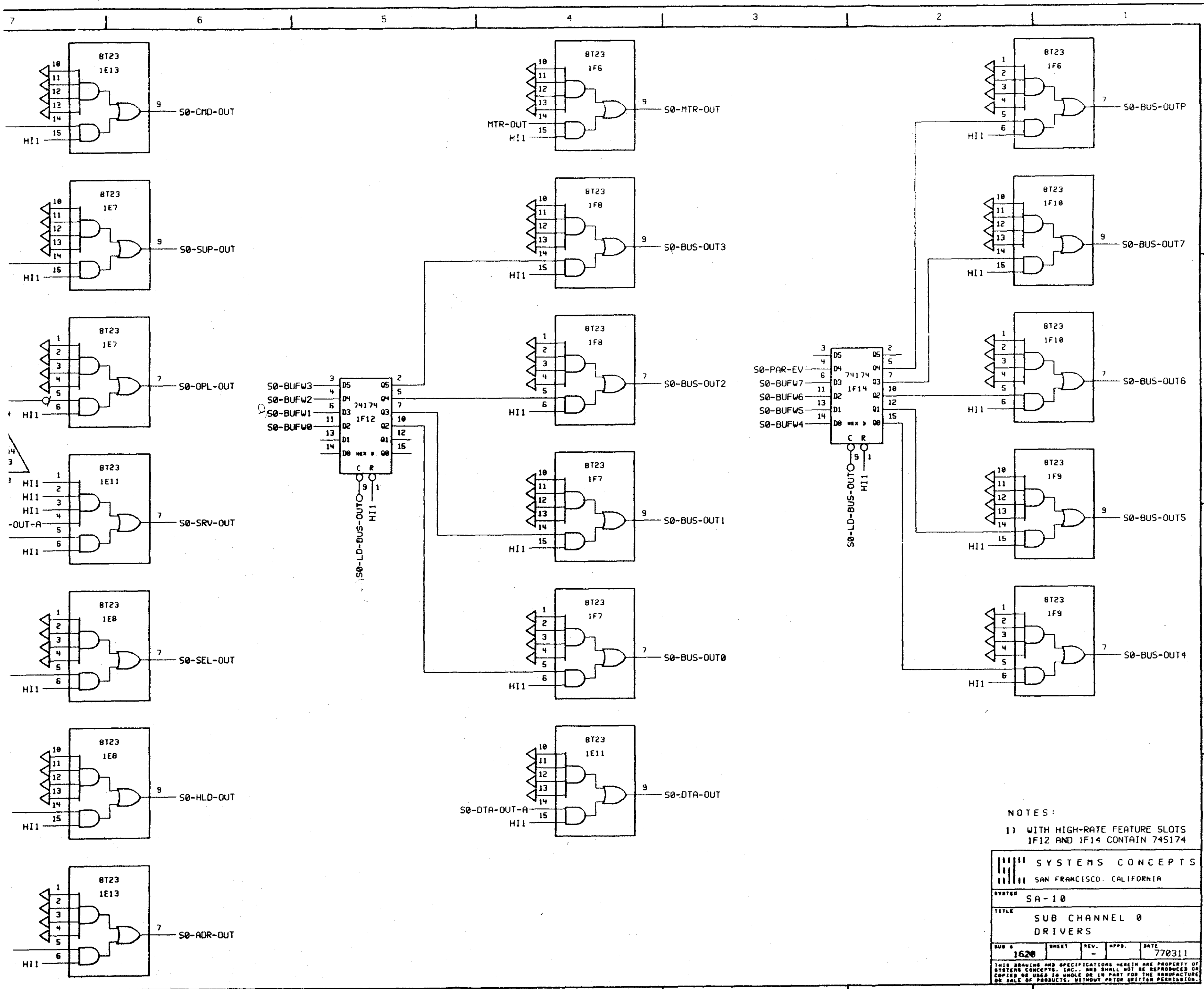


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM
SA-10

TITLE
SUB CHANNEL 0 DATA BUFFERS

DWG #	SHEET	REV.	APPD.	DATE
1619		-		771129



NOTES:

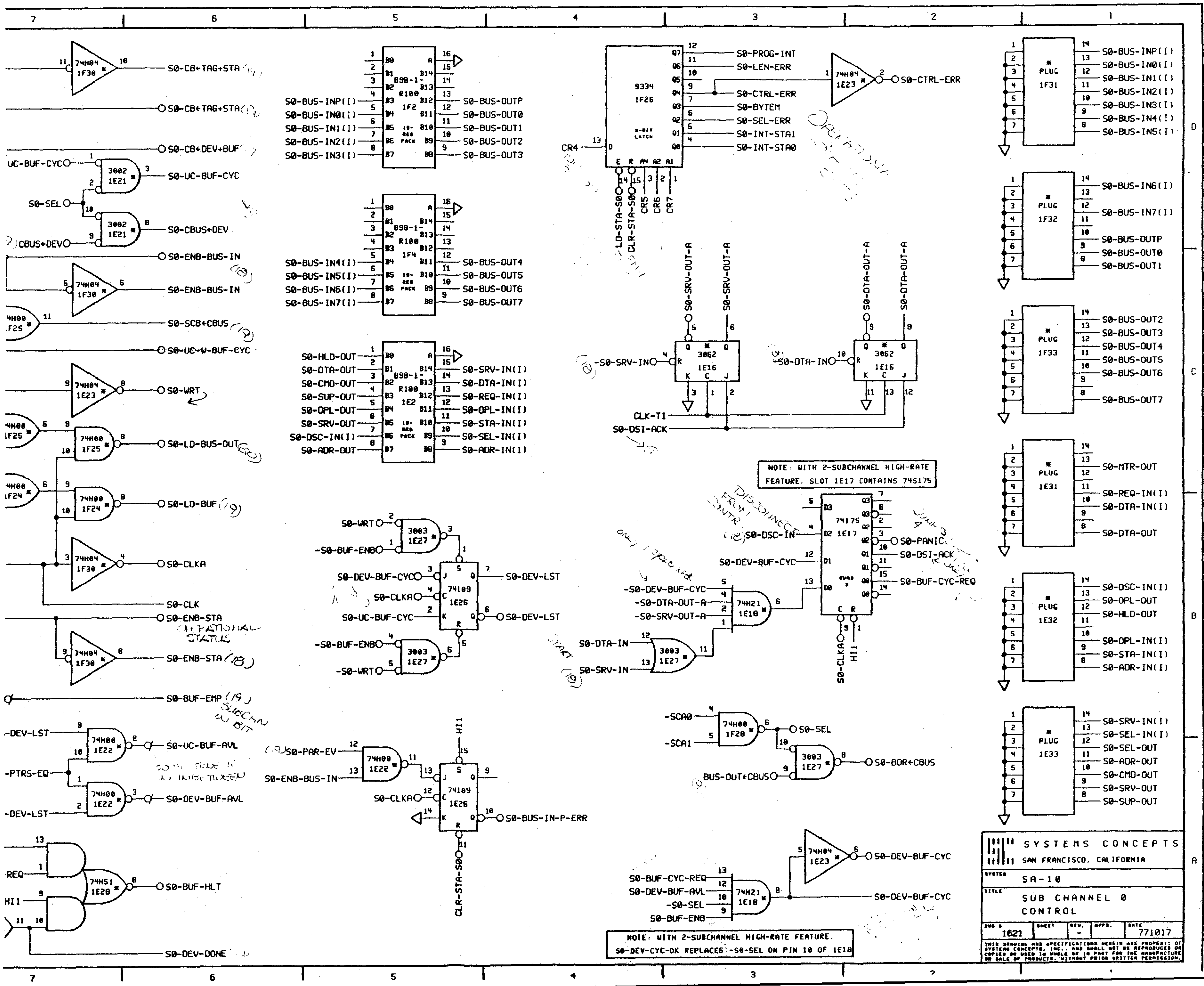
1) WITH HIGH-RATE FEATURE SLOTS 1F12 AND 1F14 CONTAIN 74S174

SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM: SA-10

TITLE: SUB CHANNEL 0 DRIVERS

DWG #	SHEET	REV.	APPD.	DATE
1620		-		770311



NOTE: WITH 2-SUBCHANNEL HIGH-RATE FEATURE. SLOT 1E17 CONTAINS 74S175

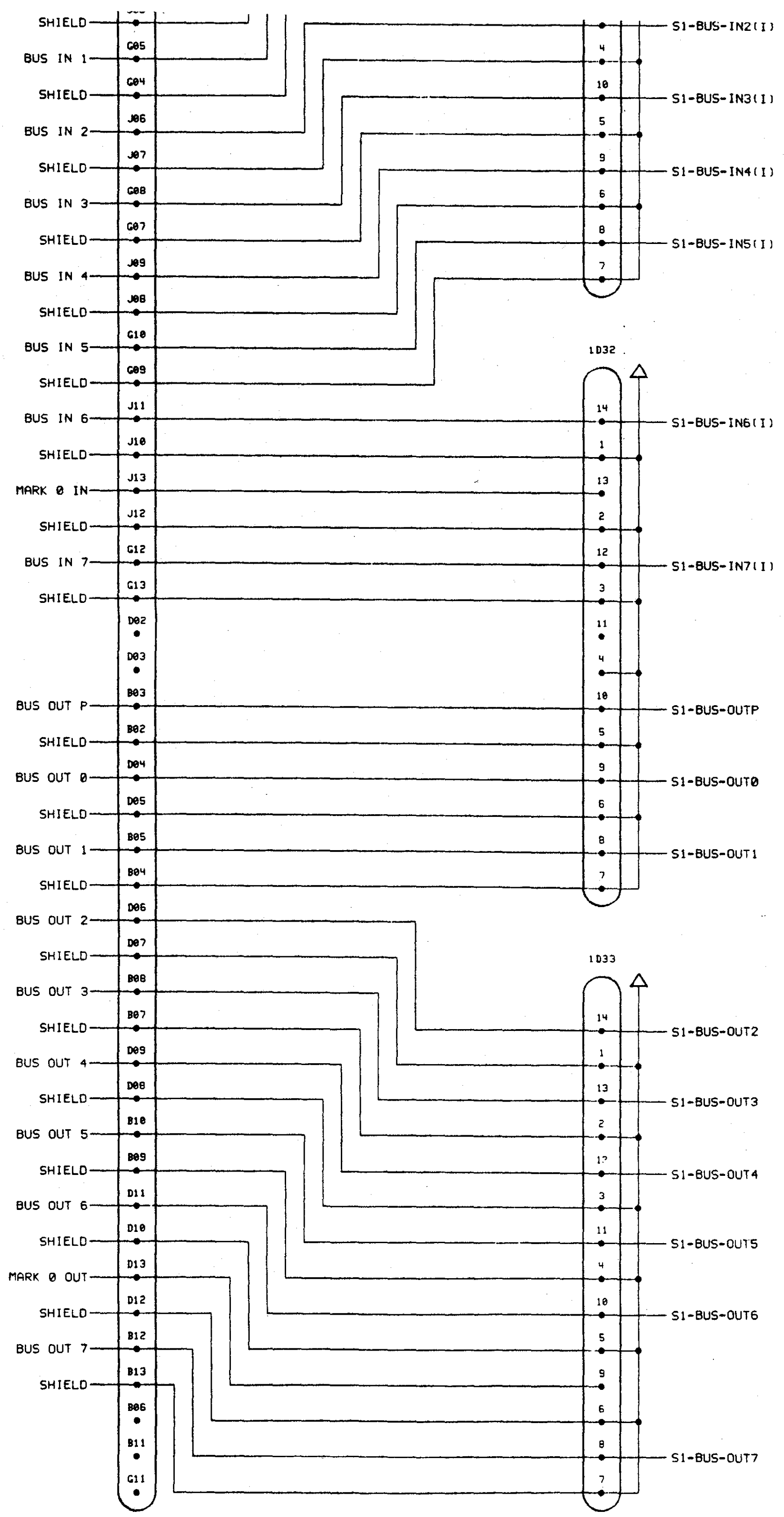
NOTE: WITH 2-SUBCHANNEL HIGH-RATE FEATURE. S0-DEV-CYC-OK REPLACES -S0-SEL ON PIN 10 OF 1E18

SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM	SA-10
TITLE	SUB CHANNEL 0 CONTROL

DWG #	SHEET	REV.	APPD.	DATE
1621		-		771017

THIS DRAWING AND SPECIFICATIONS HEREIN ARE PROPERTY OF SYSTEMS CONCEPTS, INC., AND SHALL NOT BE REPRODUCED OR COPIED OR USED IN WHOLE OR IN PART FOR THE MANUFACTURE OR SALE OF PRODUCTS, WITHOUT PRIOR WRITTEN PERMISSION.



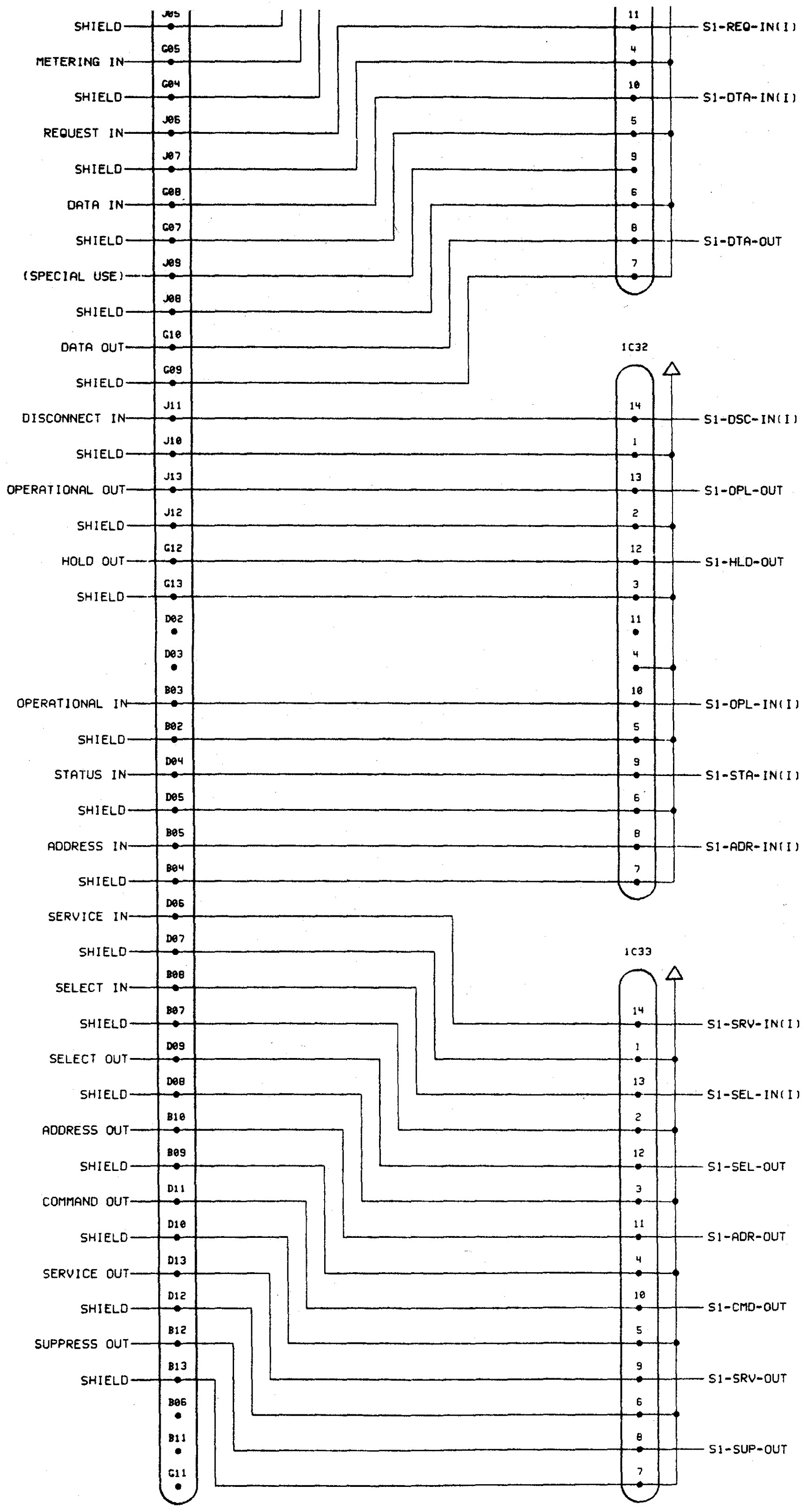
SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM SA-10

TITLE SUB CHANNEL 1
IBM BUS INTERFACE

DWG #	SHEET	REV.	APPD.	DATE
1622		-		760914

THIS DRAWING AND SPECIFICATIONS HEREIN ARE PROPERTY OF SYSTEMS CONCEPTS AND SHALL NOT BE REPRODUCED OR COPIED OR USED IN WHOLE OR IN PART AS THE BASIS FOR THE MANUFACTURE OR SALE OF PRODUCTS WITHOUT PRIOR WRITTEN PERMISSION.

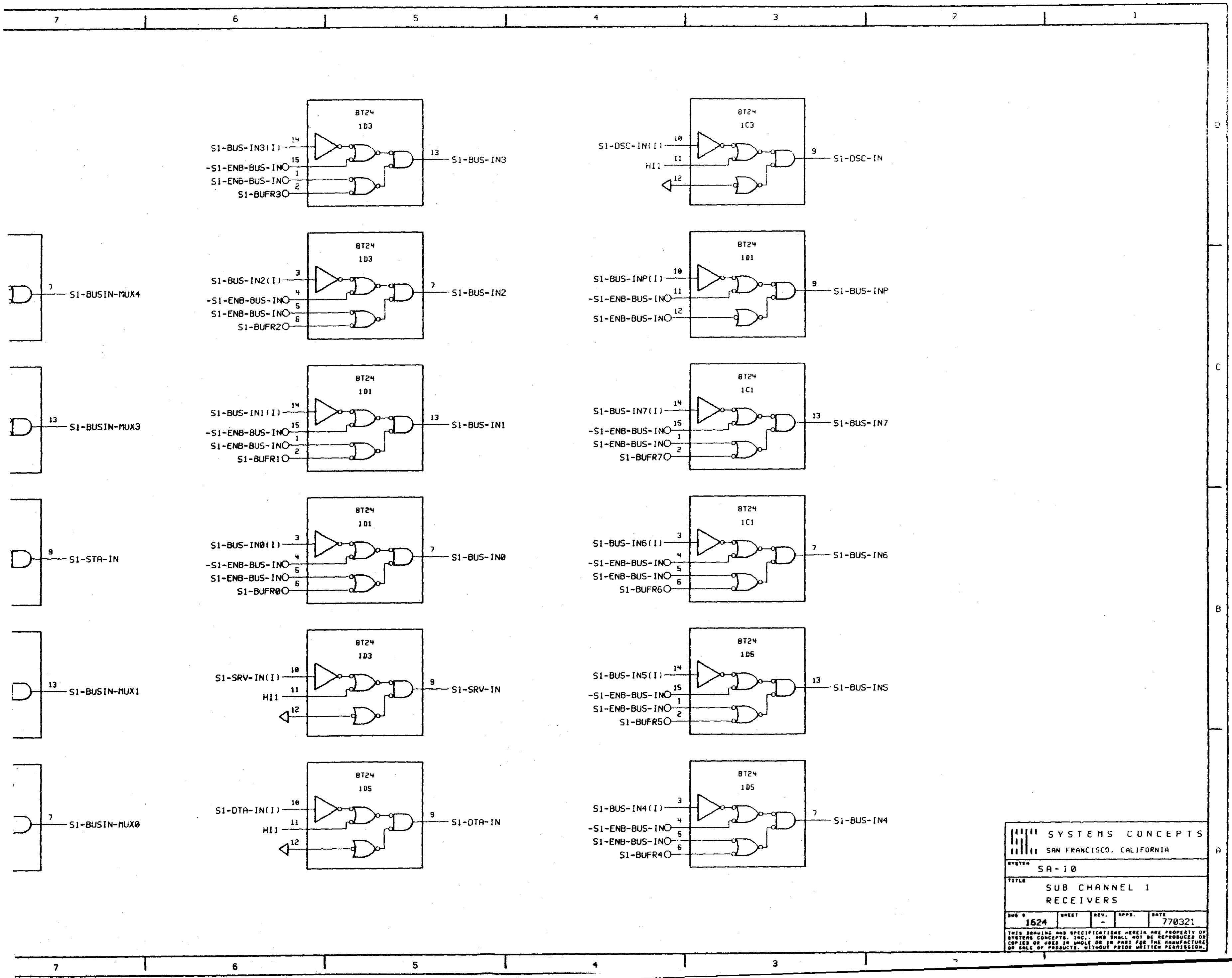


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM SA-10

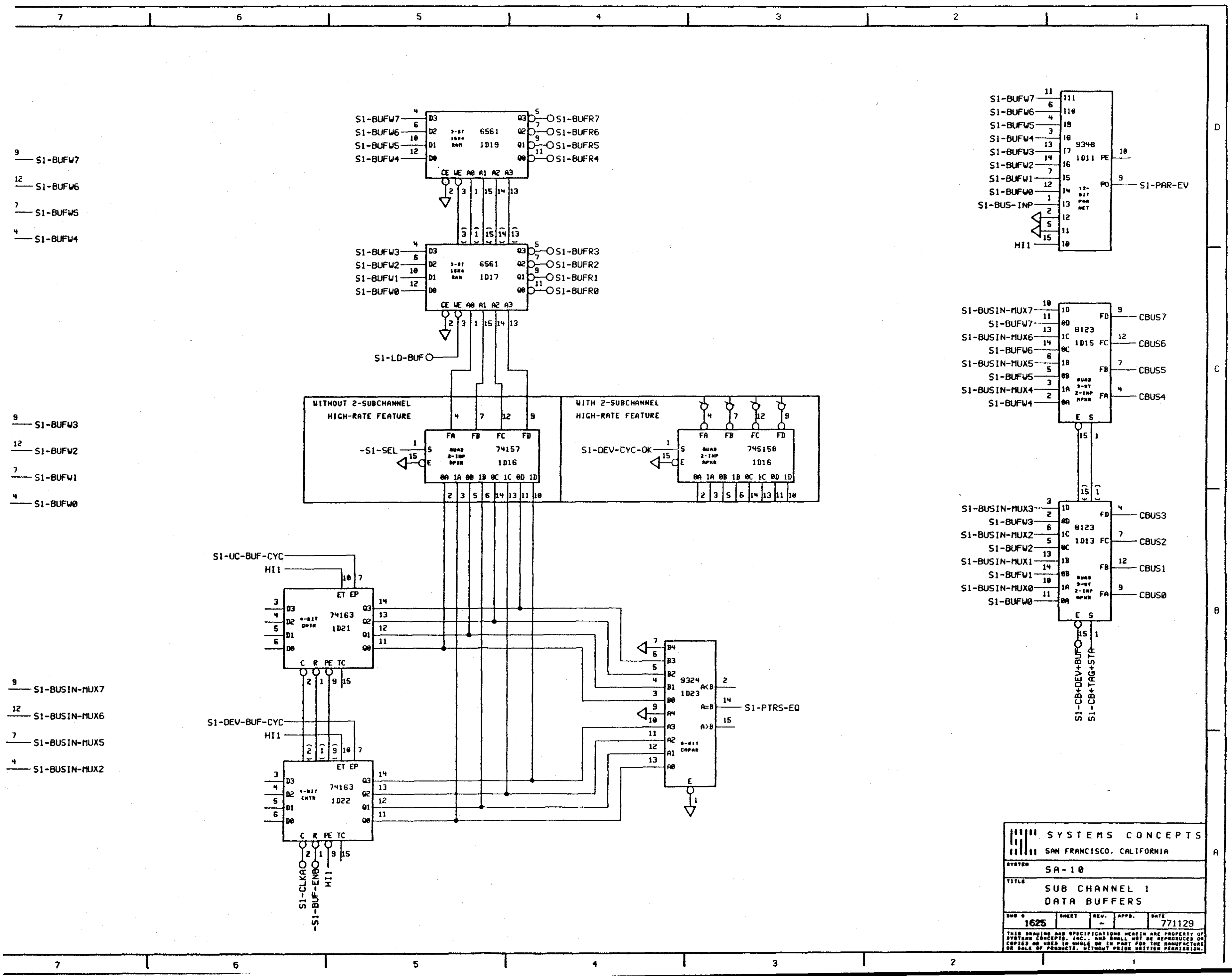
TITLE SUB CHANNEL 1
IBM TAG INTERFACE

DWG #	SHEET	REV.	APPD.	DATE
1620		-		760917



SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM	SA-10
TITLE	SUB CHANNEL 1 RECEIVERS
SUB #	1624
SHEET	
REV.	-
APPD.	
DATE	770321

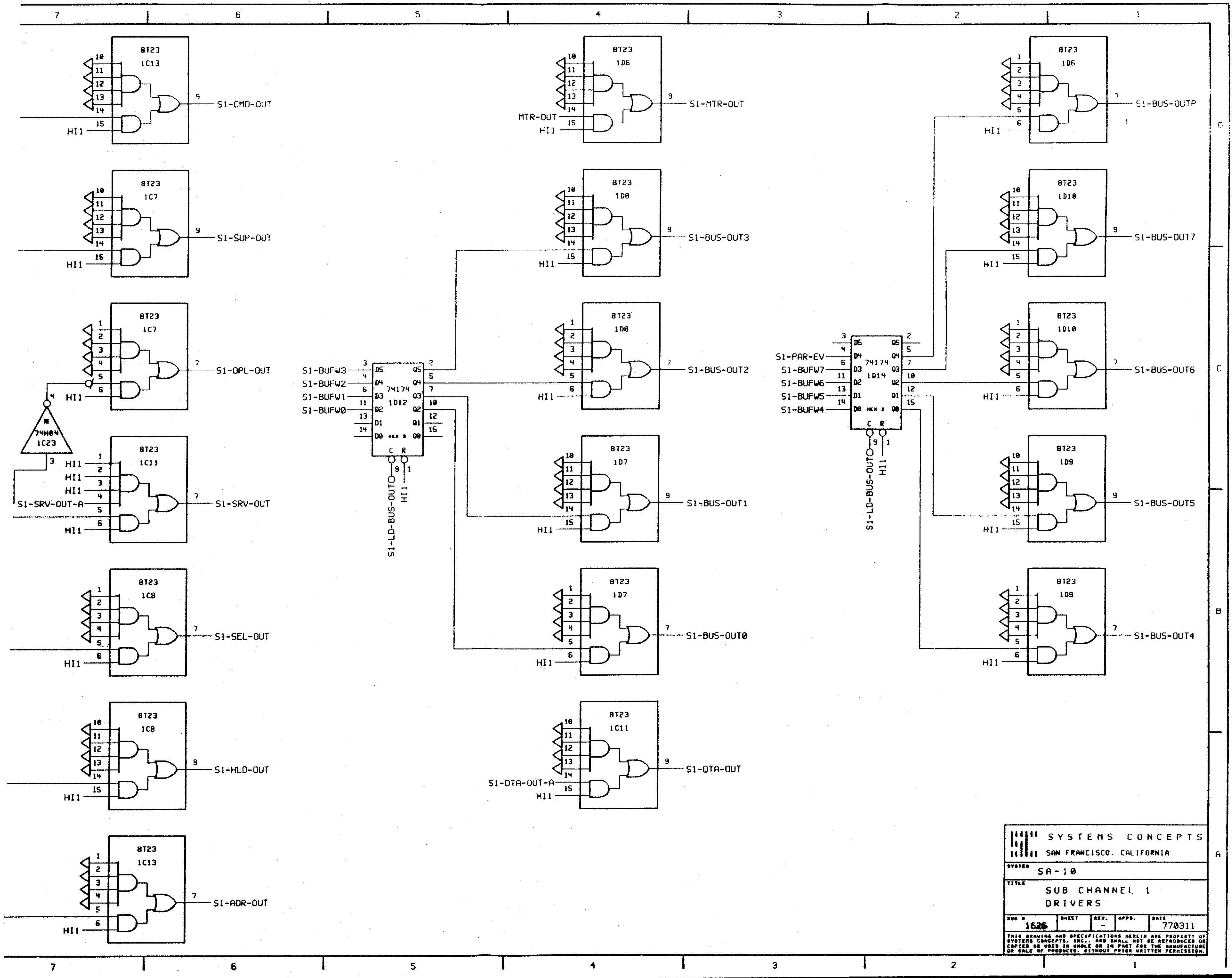


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM: SA-10

TITLE: SUB CHANNEL 1 DATA BUFFERS

DWG #	SHEET	REV.	APPD.	DATE
1625		-		771129



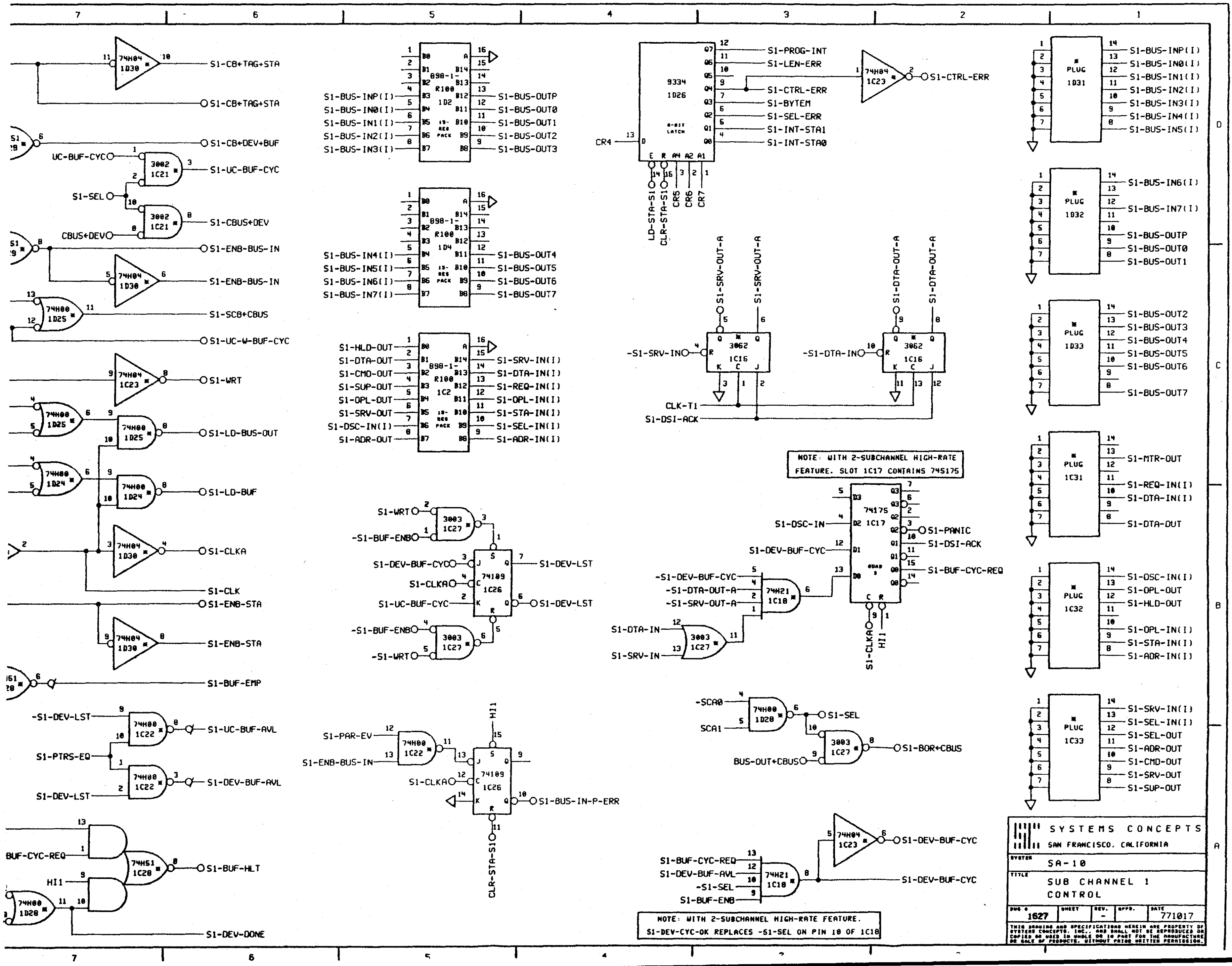
SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM: SA-10

TITLE: SUB CHANNEL 1 DRIVERS

DWG #	SHEET	REV.	APPD.	DATE
1626		-		770311

THIS DRAWING AND SPECIFICATIONS HEREIN ARE PROPERTY OF SYSTEMS CONCEPTS, INC., AND SHALL NOT BE REPRODUCED OR COPIED OR USED IN WHOLE OR IN PART FOR THE MANUFACTURE OR SALE OF PRODUCTS, WITHOUT PRIOR WRITTEN PERMISSION.



NOTE: WITH 2-SUBCHANNEL HIGH-RATE FEATURE. SLOT 1C17 CONTAINS 74S175

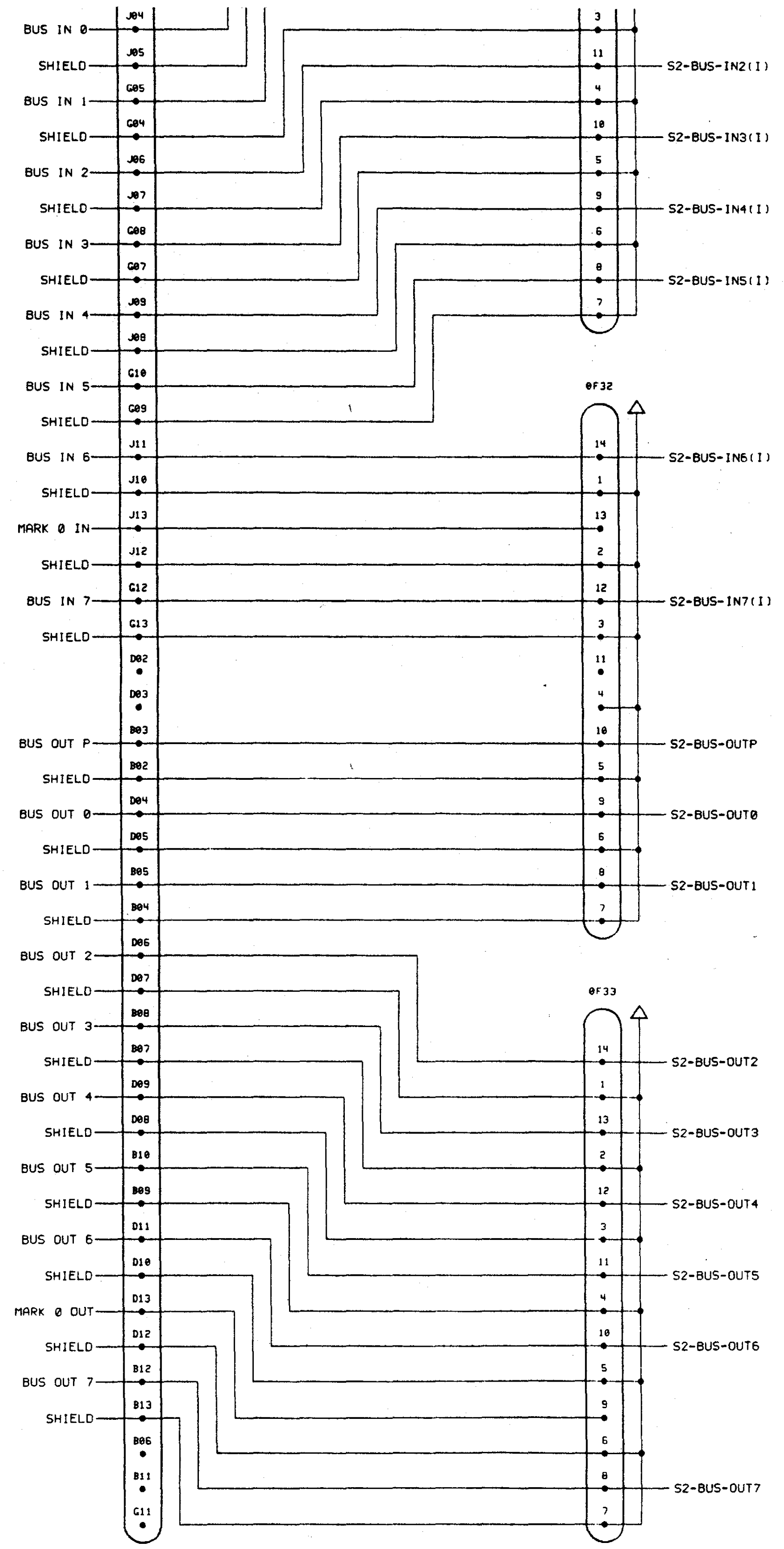
NOTE: WITH 2-SUBCHANNEL HIGH-RATE FEATURE. S1-DEV-CYC-OK REPLACES -S1-SEL ON PIN 10 OF 1C18

SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM: SA-10

TITLE: SUB CHANNEL 1 CONTROL

DWG #	SHEET	REV.	APPD.	DATE
1627		-		771017

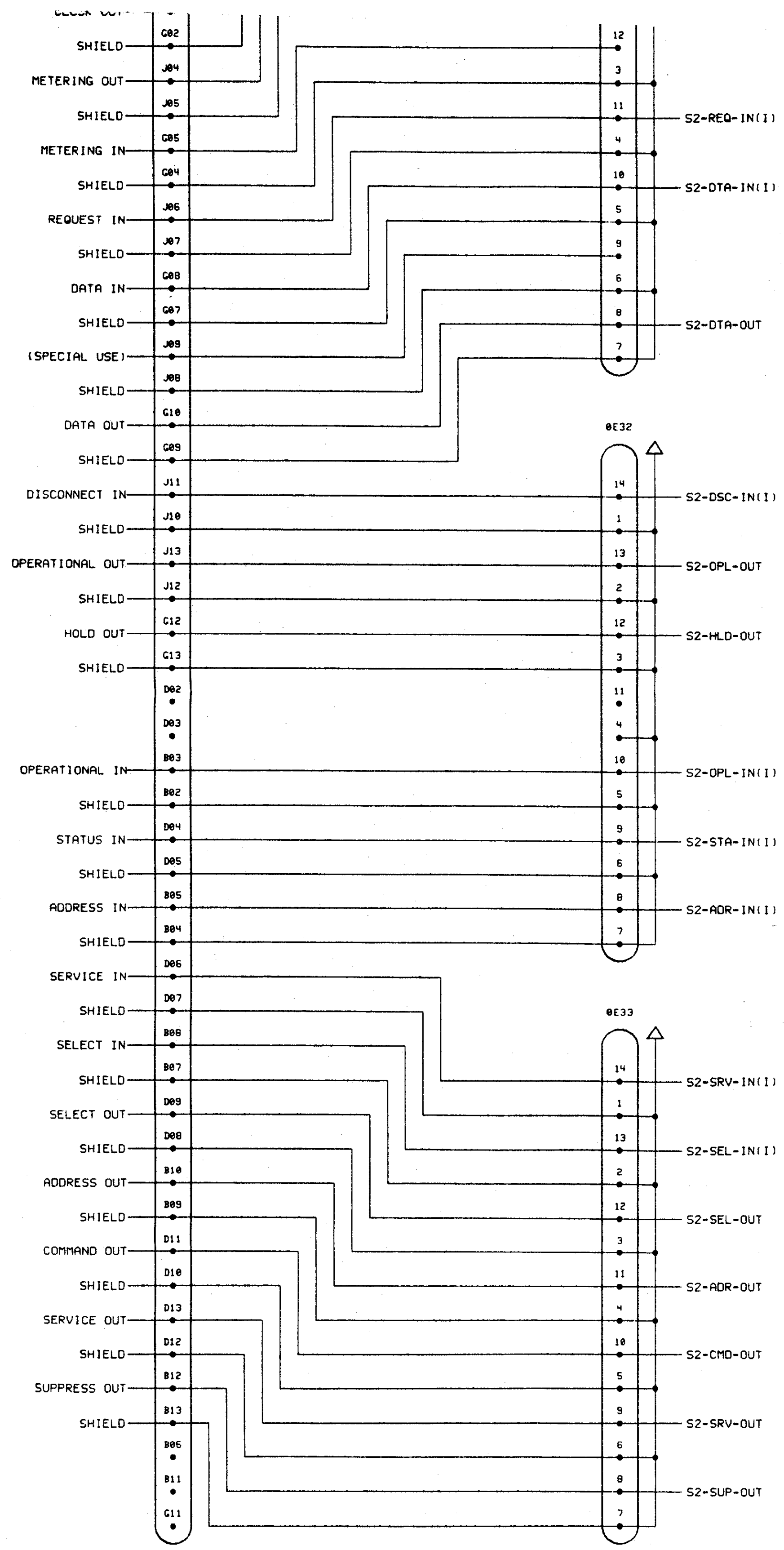


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM SA-10

TITLE SUB CHANNEL 2 IBM BUS INTERFACE

DWG #	SHEET	REV.	APPD.	DATE
1620		-		760914



SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

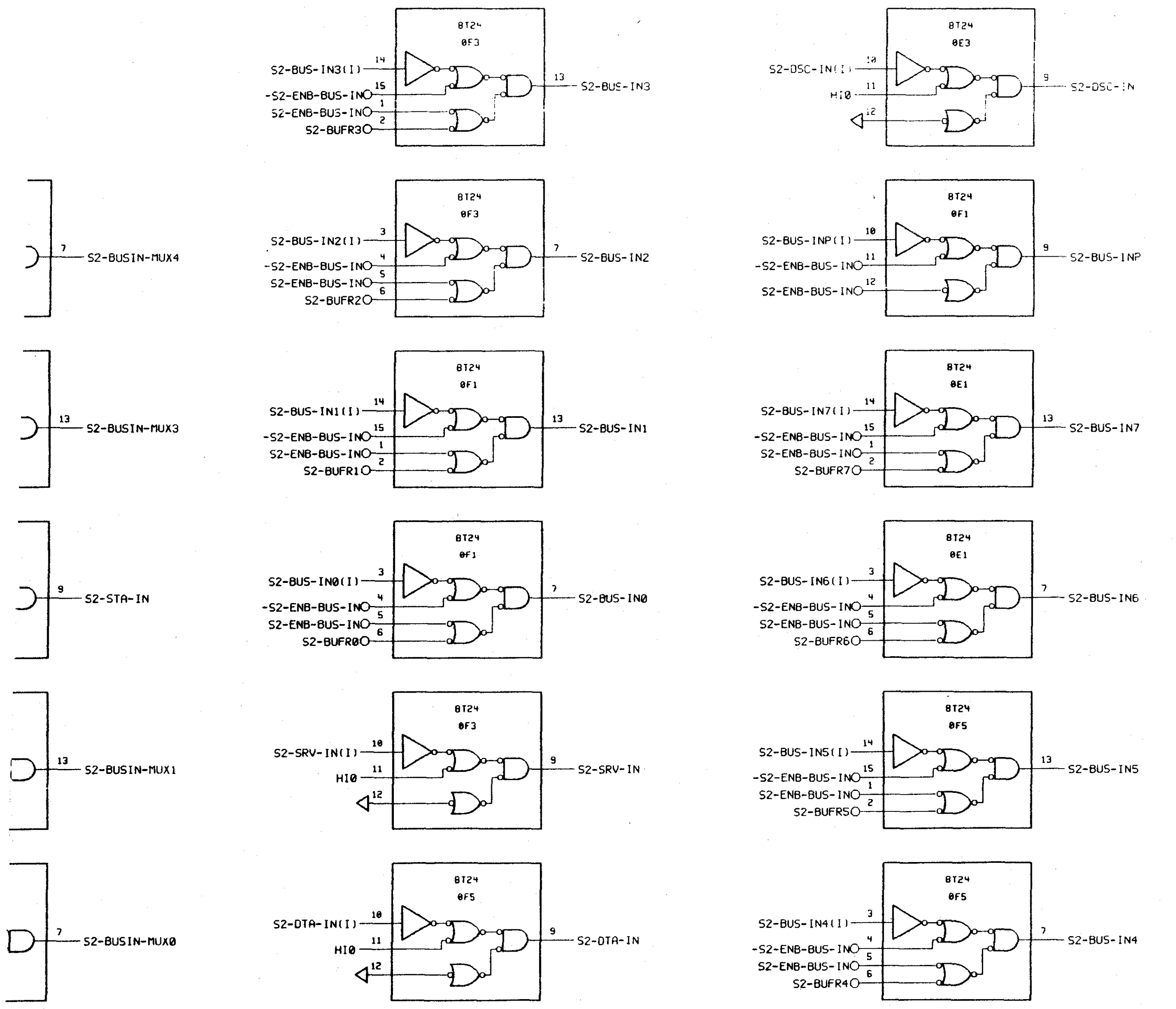
SYSTEM SA-10

TITLE SUB CHANNEL 2
IBM TAG INTERFACE

DWG #	SHEET	REV.	APPD.	DATE
1629		-		760917

7 6 5 4 3 2 1

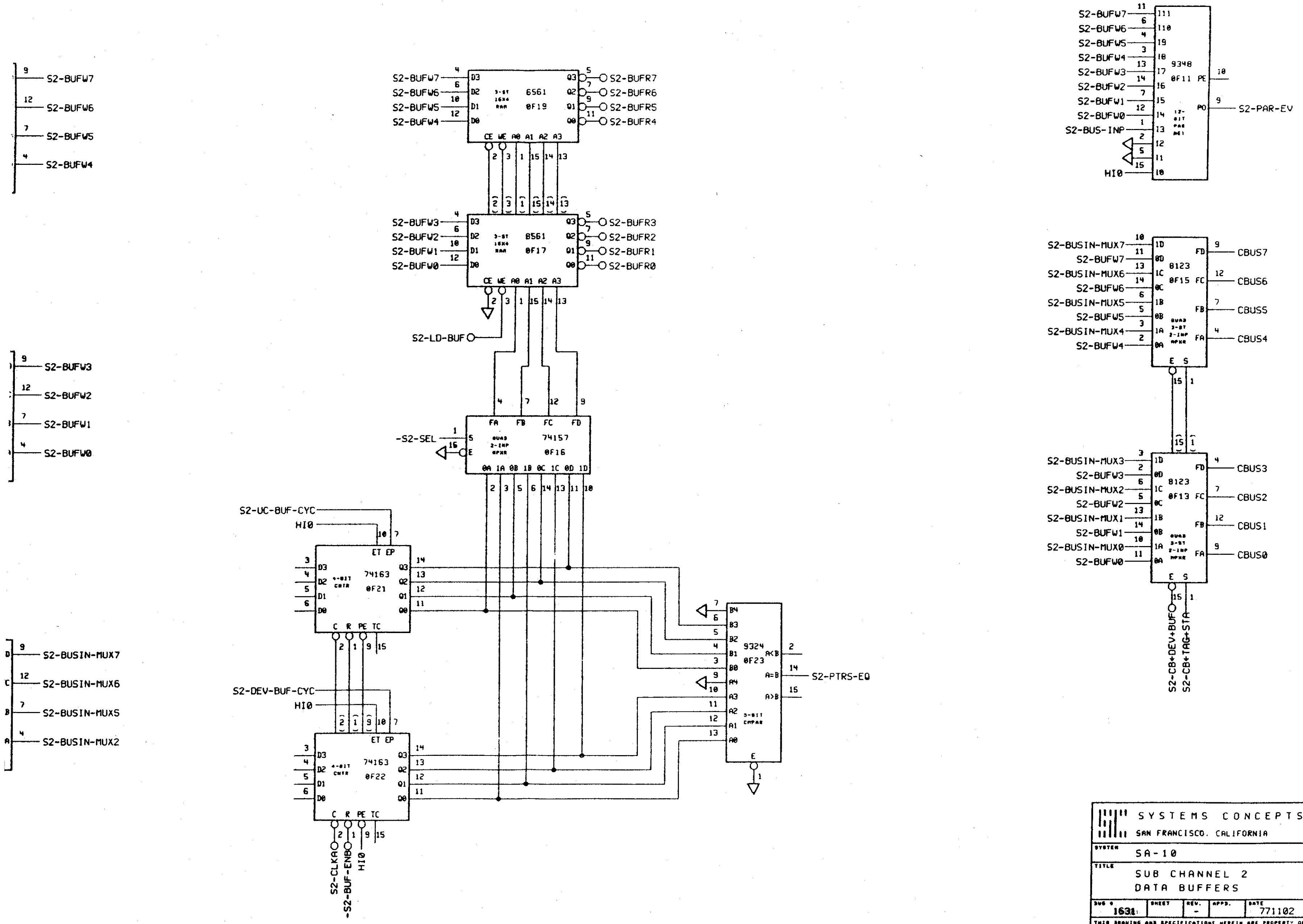
A B C D



SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM	SA-10
TITLE	SUB CHANNEL 2 RECEIVERS

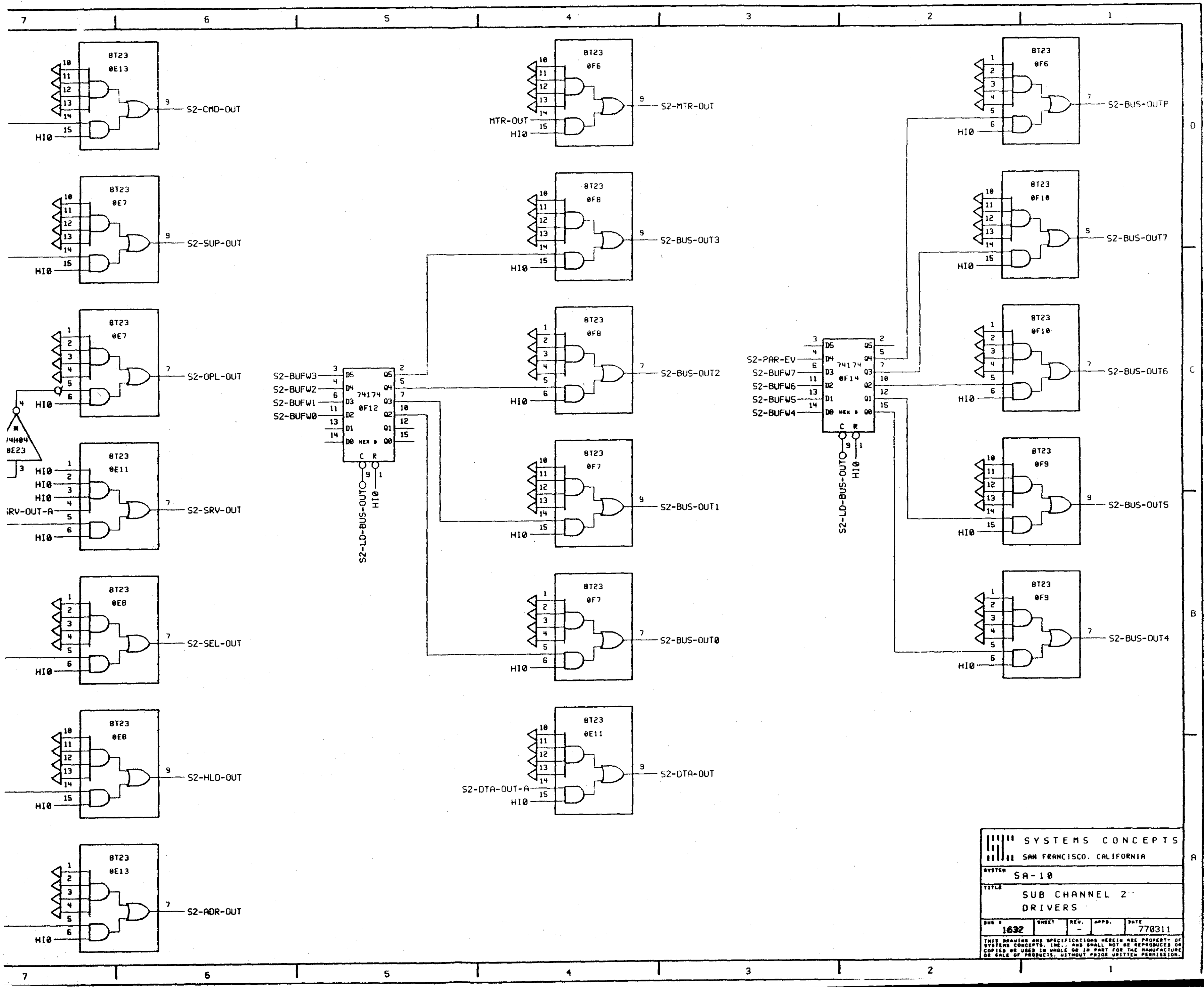
DWG #	SHEET	REV.	APPD.	DATE
1630		-		770321



SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM	SA-10
TITLE	SUB CHANNEL 2 DATA BUFFERS

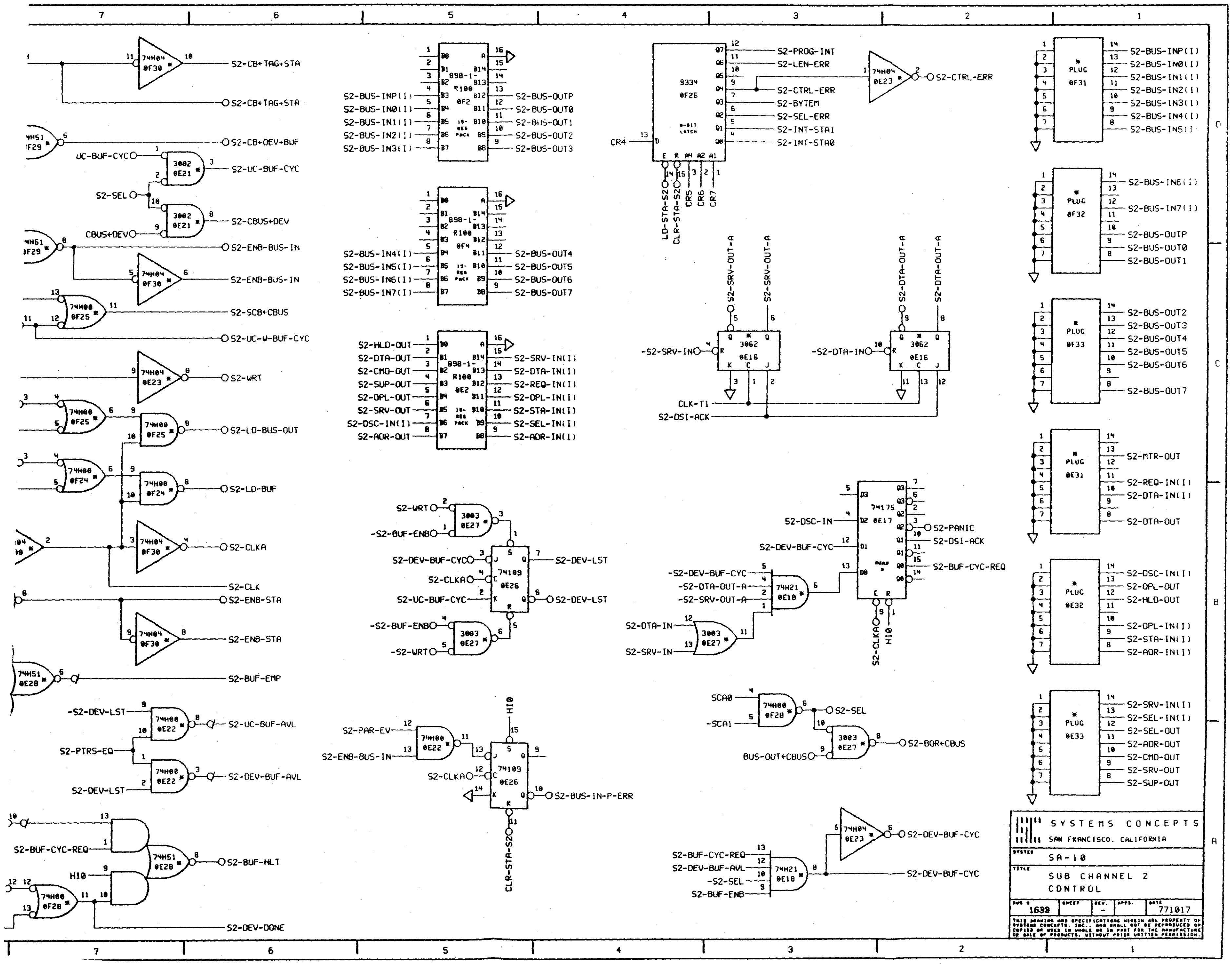
DWG #	SHEET	REV.	APPD.	DATE
1631		-		771102



SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM	SA-10
TITLE	SUB CHANNEL 2 DRIVERS

DWG #	SHEET	REV.	APPD.	DATE
1632		-		770311

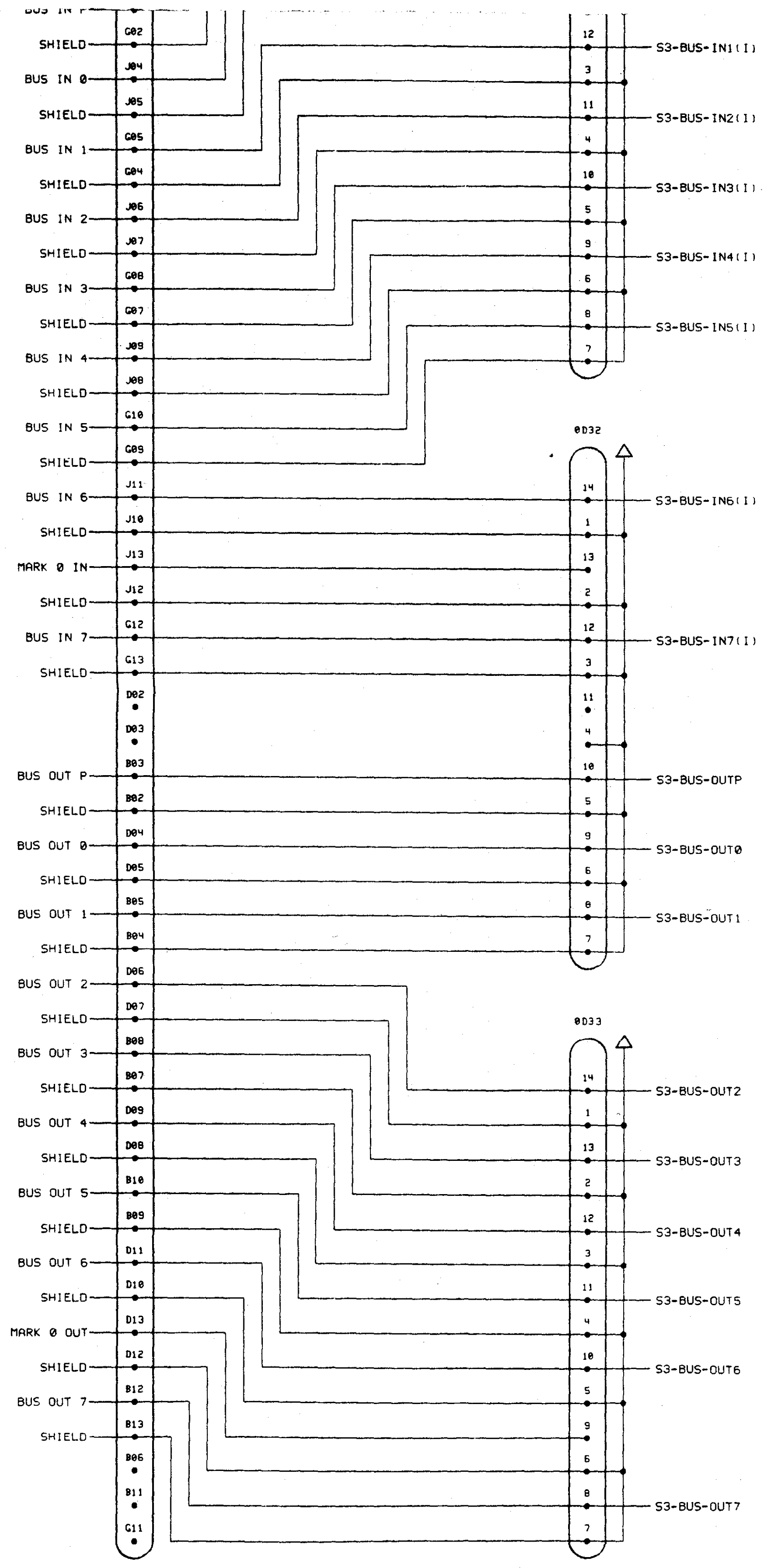


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM: SA-10

TITLE: SUB CHANNEL 2 CONTROL

DWG #	SHEET	REV.	APPD.	DATE
1633		-		771017

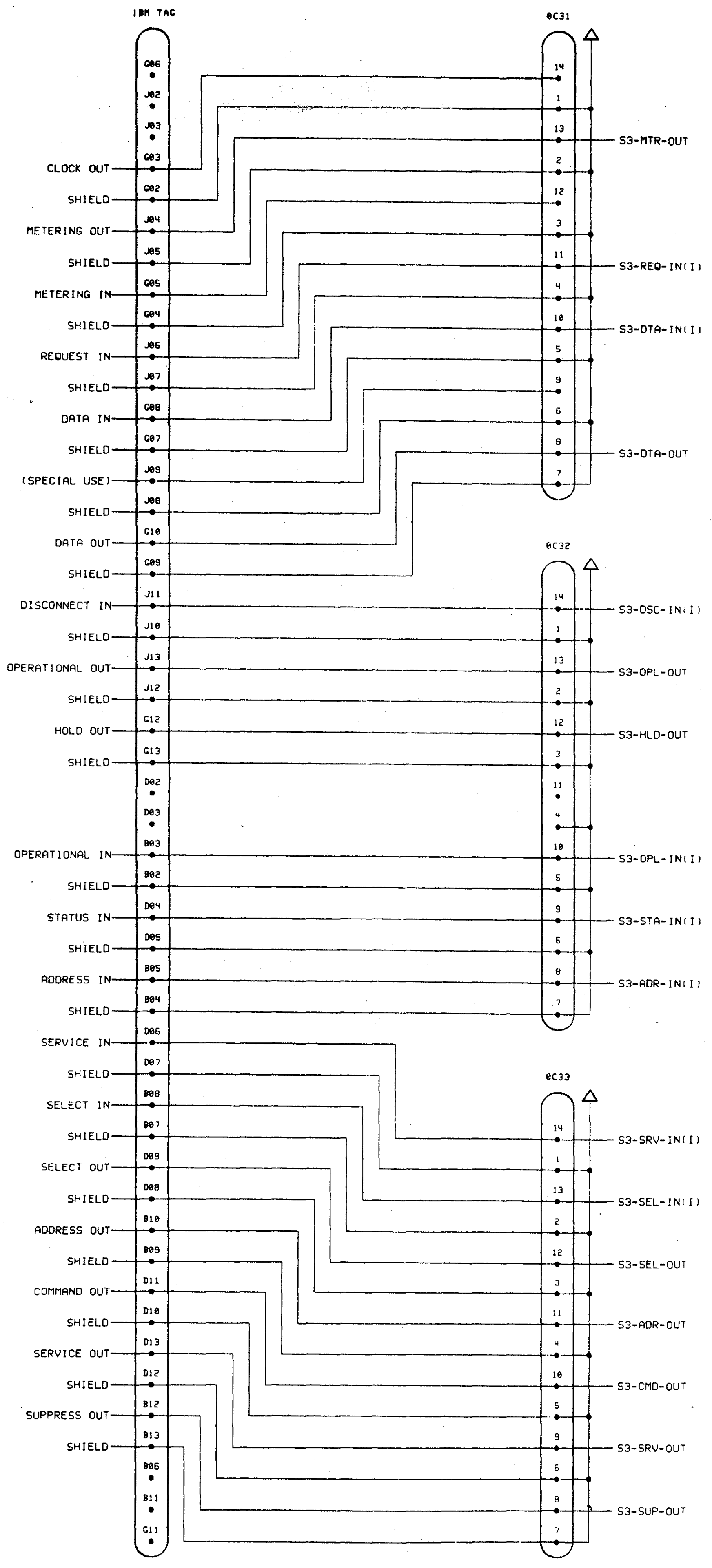


SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM SA-10

TITLE SUB CHANNEL 3
IBM BUS INTERFACE

DWG #	SHEET	REV.	APPD.	DATE
1634		-		760914



SYSTEMS CONCEPTS
SAN FRANCISCO, CALIFORNIA

SYSTEM SA-10

TITLE SUB CHANNEL 3
IBM TAG INTERFACE

DWG #	SHEET	REV.	APPD.	DATE
1636		-		760917

THIS DRAWING AND SPECIFICATIONS HEREIN ARE PROPERTY OF SYSTEMS CONCEPTS, INC., AND SHALL NOT BE REPRODUCED OR COPIED OR USED IN WHOLE OR IN PART FOR THE MANUFACTURE OR SALE OF PRODUCTS, WITHOUT PRIOR WRITTEN PERMISSION