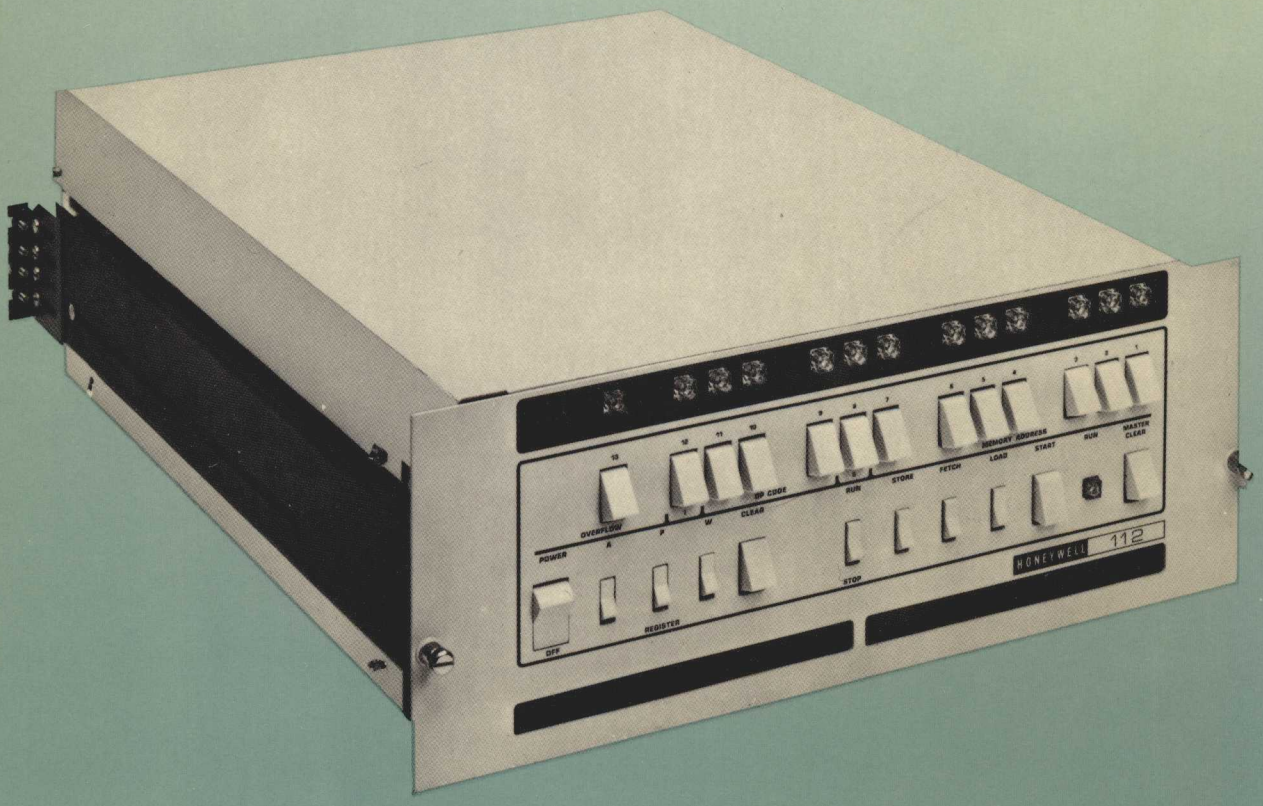
INTERFACE AND INSTALLATION



H112 Digital Controller

Honeywell

Doc. No. 70130072243A
M–1165

H112 DIGITAL CONTROLLER

INTERFACE and INSTALLATION

Instruction Manual

October 1969

Honeywell

COMPUTER CONTROL DIVISION

Printed in U.S.A.

Published by the Publications Department,
Honeywell Inc., Computer Control Division

TABLE OF CONTENTS

CONTENTS (Cont)

ILLUSTRATIONS

TABLES

SECTION I
INTRODUCTION

SCOPE OF MANUAL

This manual describes the characteristics of the various interfaces between the Honeywell H112 Controller and customer-designed external equipment. A functional description is given for each method of H112 input/output with individual requirements specified. Also defined are control lines, timing requirements, and interface drive and load requirements.

APPLICABLE DOCUMENTS

Title	Doc. No.
H112 Programmers Reference Manual	70130072242
H112 Central Processor Description	70130072244

DESCRIPTION

The Honeywell H112 Controller has four types of input/output: Programmed I/O; Load Mode Transfers; Machine Control Interface; and optionally available Direct Data Channel.

The standard programmed input/output structure in the H112 is a single cable, duplex, "party line" system. Six I/O instructions, plus reassignable, programmed priority interrupts, make the controller's parallel transfer I/O bus capable of a wide variety of applications.

Programmed I/O data transfers take place under software control. The priority interrupt line may be utilized during these transfers to identify device ready status; or alternately, device ready status may be determined by repeated testing either by an SKS command or by the skip test incorporated into the data transfer instructions. Twelve bit parallel transfers are made to and from the accumulator via the programmed I/O bus.

Load Mode Transfers also occur over the I/O bus but are entirely under the control of the H112 hardware. The direction of load mode transfers is always into the controller.

The Machine Control Interface makes available certain functions (run, stop, load, clear, start, and single instruction execution) to the system, whether or not the control panel is present. This feature allows the H112 to operate without the control panel. Using the plug-in panel feature reduces unit cost and prevents unauthorized personnel from interfering with H112 operation.

The Direct Data Channel option provides an I/O path for high-speed data transfers between the controller and I/O devices. Transfers between the channel located in the controller and subchannels located in the peripheral devices utilize both the programmed

I/O bus and the DDC bus. All transfers are under subchannel control; however, a transfer may be initiated by a command from the program to the peripheral device over the I/O bus. The address of the memory location to be transferred as well as the direction and time of transfer are under subchannel control. This option gives the H112 a direct memory transfer capability.

GENERAL INTERFACE CHARACTERISTICS

The input/output characteristics of the H112 have been carefully tailored to provide maximum interface flexibility, with the widest possible interface tolerances. I/O signals are true in the ground states and relatively immune to noise. Timing requirements give adequate time for a device to respond; and wide strobe pulses eliminate capacitive losses.

All I/O instructions have the same length, making timing consistent for programmed I/O transfers. Also, included as part of the standard instruction complement, is the stall instruction for convenient synchronization (with very low latency time) to external devices.

SECTION II
PROGRAMMED I/O TRANSFERS

This section is divided into the following parts.

a. Input/Output Instructions
b. Input/Output Circuits
c. Programmed I/O Bus Signals
d. Signal Timing
e. List of Signals, Mnemonics, and Pin Numbers
f. Interfacing the Programmed I/O Bus
g. Interrupts with Recommended Gating
h. Example of a Programmed I/O Interface

I/O INSTRUCTIONS

The following list of instructions are used to communicate between the program and the external system. Part of the hardware to execute these instructions is in the mainframe. This hardware causes the various signals to be propagated on the I/O bus. The remaining hardware to execute the instructions is in the interface unit. This hardware receives and transmits data and answers interrogations with the proper pulses. See Appendix B for device address and mask bit assignments.

OCP - Output Control Pulse

This instruction is used by the program to command an action; e.g., start the reader, reset the clock, etc. The 6-bit address field of this instruction specifies the action and device.

OTA - Output Transfer From Accumulator, If Ready

This instruction first determines whether the interface is ready for an output data transfer. If the interface is ready, a 12-bit parallel transfer from the accumulator to the interface takes place and the next sequential instruction is skipped. If the interface is not ready, the transfer does not take place and the next sequential instruction is executed. The 6-bit address field of the instruction specifies the recipient of the data.

INA - Input Transfer to Accumulator, If Ready

As in the OTA, this instruction first determines whether the interface specified by the instruction's address field is ready to transfer data to the mainframe. If ready, the accumulator is cleared and 12 bits (or less) of data is transferred to the accumulator in parallel. If the interface is not ready, the accumulator is cleared, but the transfer does not take place.

SKS - Skip If Set

This instruction is used to test a condition, line, or flip-flop in the interface. The 6-bit address field of the instruction specifies the condition to be tested. The program is informed of the status of the condition by having the next sequential instruction following the SKS skipped or not skipped depending on the status.

SMK - Set Interrupt Mask

This instruction is used to set and reset mask flip-flops in the various interfaces. An interface is allowed to generate an interrupt only if its mask bit is set and is prohibited from interrupting if its mask bit is reset. Each interface is assigned one bit of the accumulator which is copied into its mask bit upon execution of the appropriate SMK instruction. There are two SMK instructions, differentiated by one bit in the address field. Thus, there can be up to 24 mask flip-flops.

STL - Stall On Line

This instruction is used to synchronize the controller with the external system. When the STL instruction is executed, the controller will stall or wait until the appropriate line in the bus is made true. The program will then continue from the instruction immediately following the STL. By feeding an external clock or other synchronization signal into the bus, the controller can be slaved to the external source.

I/O BUS CIRCUITS

All signals into and out of the controller are considered active or true when at ground and passive or false when at +6 volts. All levels are μ-PAC (0 to +6 volts) levels.

The controller will drive on output lines 16 receiver loads consisting of a standard μ-PAC gate input in parallel with a 6800-ohm resistor to +6 volts, and a diode with anode to ground. Each input line to the controller appears as one receiver load; however, peripheral devices load the input lines in parallel. Thus, each line may appear as up to 17 loads to a particular device including the device's own load and the controller's input load.

The I/O bus timings are guaranteed only when driven and received with circuits such as the CS-517 driver/receiver PAC (see Appendix). A standard cable whose total length is less than 25 feet must be between the controller and all interfaces.

Signal edges are guaranteed to fall within the specified time, but they may cross the gate threshold more than once in that period. It is a recommended practice, therefore, to design for this contingency.

The standard devices are designed such that each interface loads each line by no more than one load and drives each line with no more than one driver.

PROGRAMMED I/O BUS SIGNALS

Data Bus (KDB01- through KDB12-)

Signals KDB01- through KDB12- from the controller represent the contents of the accumulator (A-register) during OCP, SMK, and OTA commands. During INA commands the addressed peripheral device places data on this bus for eventual gating into the accumulator. The interface must leave these lines at +6 volts until a data transfer to the controller is made in response to an INA. Thus these signals are bidirectional, being outputs during OTA's, SMK's and OCP's and inputs during INA commands. A line will be at ground when the correspondingly numbered bit in the accumulator is set and +6V when reset. To transfer a ONE into the accumulator in particular bit position during an INA command, the corresponding line must be grounded.

Address Bus Lines (KAB01- through KAB06-)

The six output signals KAB01- through KAB06- are the least significant bits of the W-register and carry the address portion of the I/O instruction. KAB01- corresponds to instruction bit 1 and KAB06- to bit 6.

Instruction Lines (KOTAL-, KINAL-, KOCPL-, KSKSL-)

An output on one of the lines KOTAL-, KINAL-, KOCPL-, KSKSL- indicates that the execution of an OTA, INA, OCP, or SKS, respectively is in progress. Execution of an SMK causes the OCP line (KOCPL-) to pulse.

Set Mask (KSMKL-)

The output signal KSMKL- will be at ground during the execution of either an SMK0 or SMK1 instruction. The peripheral device is to use this line with KAB01+ or KAB01- to determine which instruction is being executed and the appropriate data bus bit to set or reset the mask flip-flop. (KAB01- is used for SMK0 and KAB01+ is used for SMK1).

Strobe Signal (KSTRB-)

The output signal KSTRB- indicates that an active (ground) test signal has been recognized during either an INA, OTA, or SKS instruction, that the next instruction will be skipped, and that a data transfer, if any, will take place.

During OCP and OTA commands, this signal also serves to define the time when the address and data lines are stable. During OTA, INA, and in some cases during SKS commands, STROBE is used to reset a ready flip-flop. This signal is also active, but redundant during execution of an SMK.

Interrupt Signal (KINTL-)

KINTL- is an input signal to the controller. This line is made active (ground) to interrupt the program, and held until the interrupt is recognized. The line is made +6 volts when the controller resets the interrupt via separate programming action. This is usually accomplished via data transfer or OCP instruction.

Load Signal (KLOAD-)

KLOAD- is active (ground) when the load function is activated from either the external machine control interface or the panel. It is an output from the controller and initiates load mode input transfers.

Test Line (KTSTL-)

If a peripheral device is ready for a data transfer, it responds to address and instruction signals by making the KTSTL- input signal active (ground). This signal is used during an INA or OTA instruction or if a skip is to occur during an SKS instruction. This line serves no function during OCP. The interface must leave the KTSTL- line at +6 volts until addressed.

Stall Line (KSTAL-)

KSTAL- must be made active (ground) with a pulse which must be greater than 500 ns long to guarantee restarting the controller after the execution of an STL instruction. If KSTAL- is at ground during execution of the STL instruction, the machine does not stop. This line is an input signal to the controller.

Master Clear (KXCLR-)

KXCLR- is active (ground) during the operation of the MASTER CLEAR pushbutton on the panel, during the activation of the remote clear line, and during a power on or off sequence. KXCLR- is an output signal from the controller.

Power Failure (KPWFL-)

An active (ground) level on the KPWFL- output line indicates that operation of the controller will be terminated because of a power failure or power turn-off operation. After this line becomes active, there is at least 1 millisecond of program execution time remaining before all processing will stop due to lack of power. This output signal is supplied only if the power failure interrupt option is included in the machine.

SIGNAL TIMING

Figures 2-1 through 2-5 show the timing requirements for each command. Unless timing tolerances are shown, the timing illustration is the minimum duration in the "true" state.

Table 2-1 lists the mnemonics, functions, and pin assignments for the lines in the I/O bus.

Figure 2-6 shows I/O cable routing within controller and cabling coming out of the machine.

Table 2-1.
Programmed I/O Bus Signals

Mnemonics	Function	Pin No.
KDB01-	Data Bus Bit 1	1
KDB02-	Data Bus Bit 2	2
KDB03-	Data Bus Bit 3	3
KDB04-	Data Bus Bit 4	4
KDB05-	Data Bus Bit 5	5
KDB06-	Data Bus Bit 6	6
KDB07-	Data Bus Bit 7	7
KDB08-	Data Bus Bit 8	8
KDB09-	Data Bus Bit 9	9
KDB10-	Data Bus Bit 10	10
KDB11-	Data Bus Bit 11	11
KDB12-	Data Bus Bit 12	12
KAB01-	Address Bus Bit 1	13
KAB02-	Address Bus Bit 2	14
KAB03-	Address Bus Bit 3	15
KAB04-	Address Bus Bit 4	16
KAB05-	Address Bus Bit 5	17
KAB06-	Address Bus Bit 6	18
KOTAL-	OTA Instruction	19
KINAL-	INA Instruction	20
KOCPL-	OCP Instruction	21
KSKSL-	SKS Instruction	22
KSMKL-	SMK Instruction	23
KSTRB-	Strobe Signal	24
KINTL-	Interrupt Signal	25
KLOAD-	Load Signal	26
KTSTL-	Test Line	27
KSTAL-	Stall Line	28
	SPARE (Unassigned)	29
KXCLR-	Master Clear	30
KGND-	Ground	31
KPWFL-	Power Failure	32

The I/O bus originates in mainframe slot C18. Figure 2-6 shows the mainframe drawer configuration, available option area, and I/O cable routing.

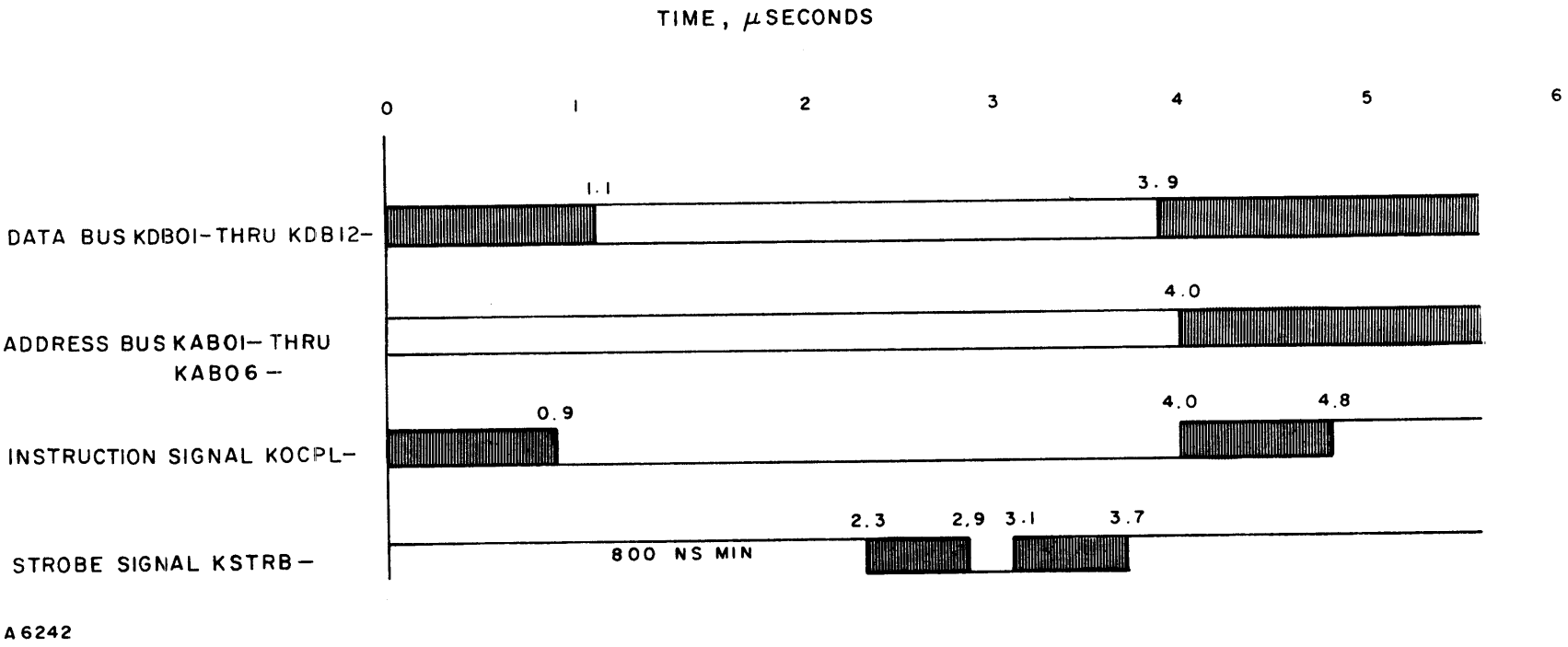


Figure 2-1. OCP Timing

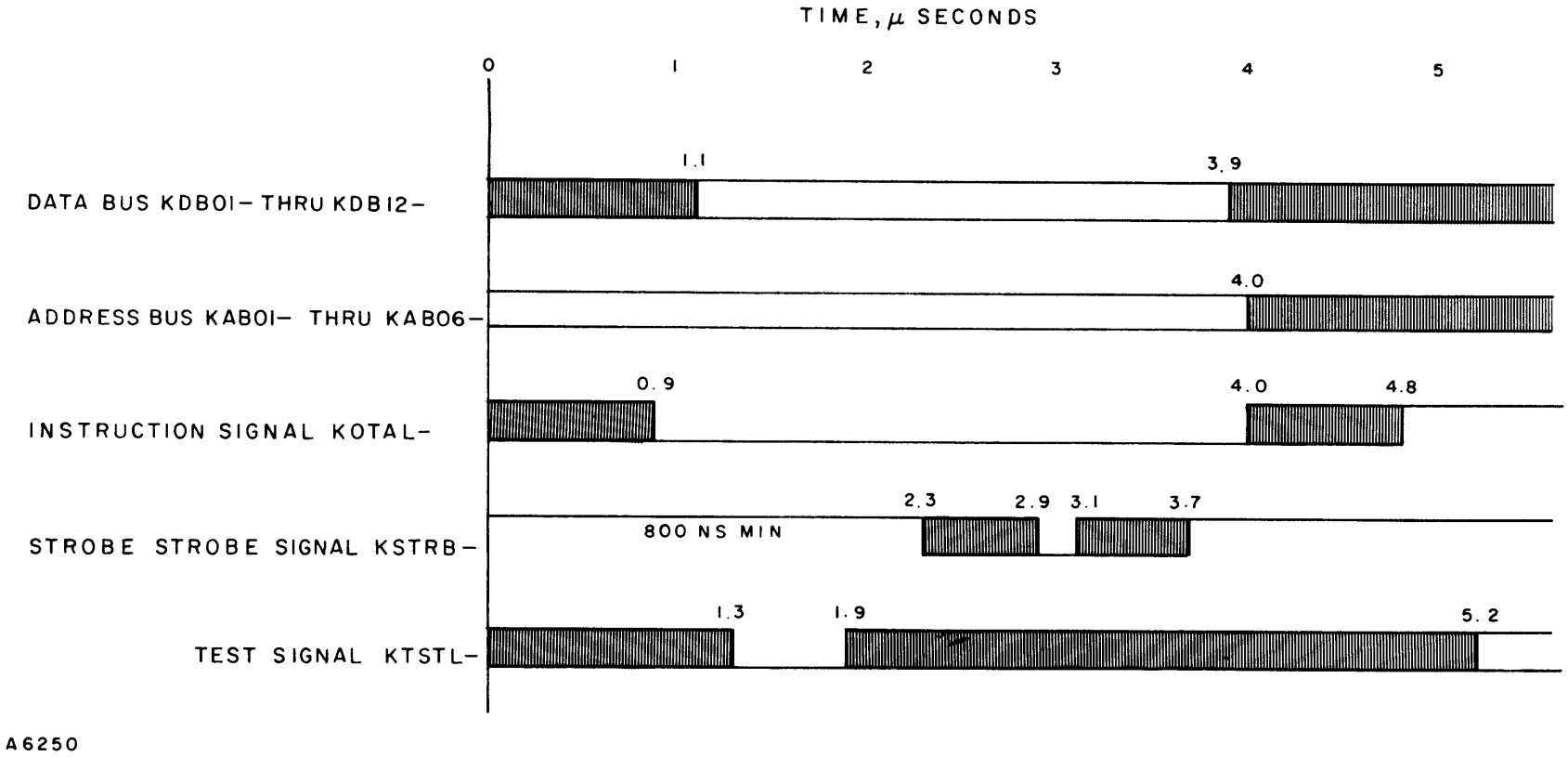


Figure 2-2. OTA Timing

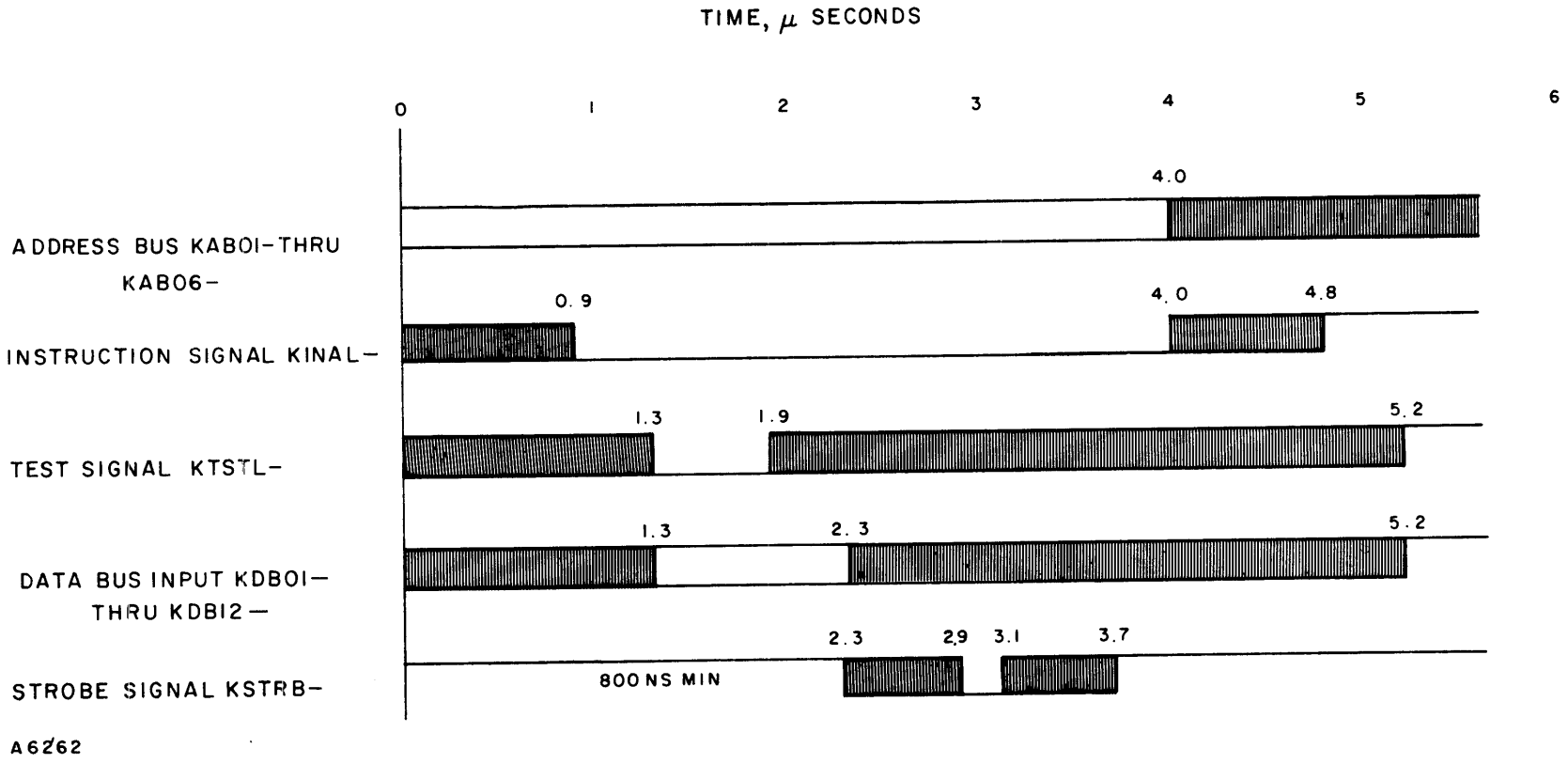


Figure 2-3. INA Timing

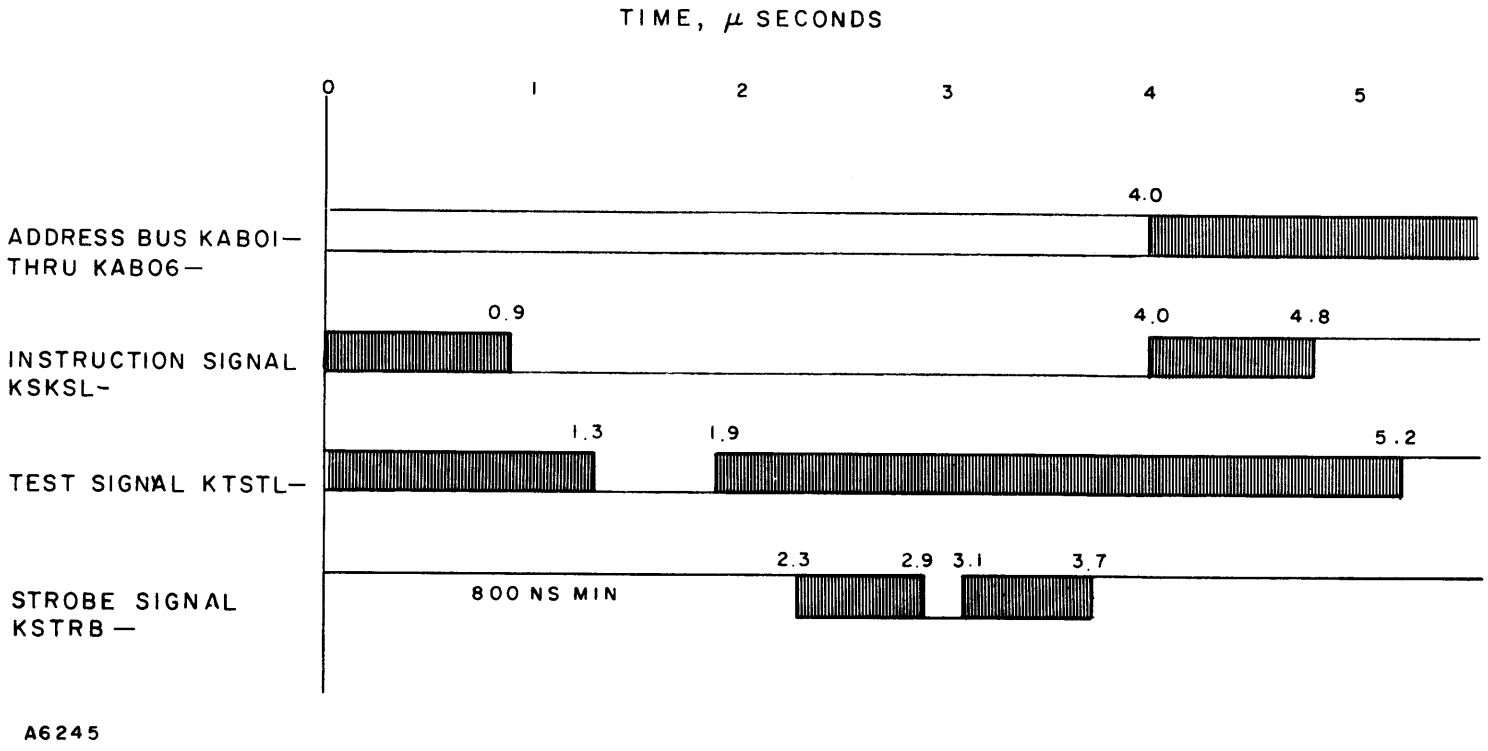


Figure 2-4. SKS Timing

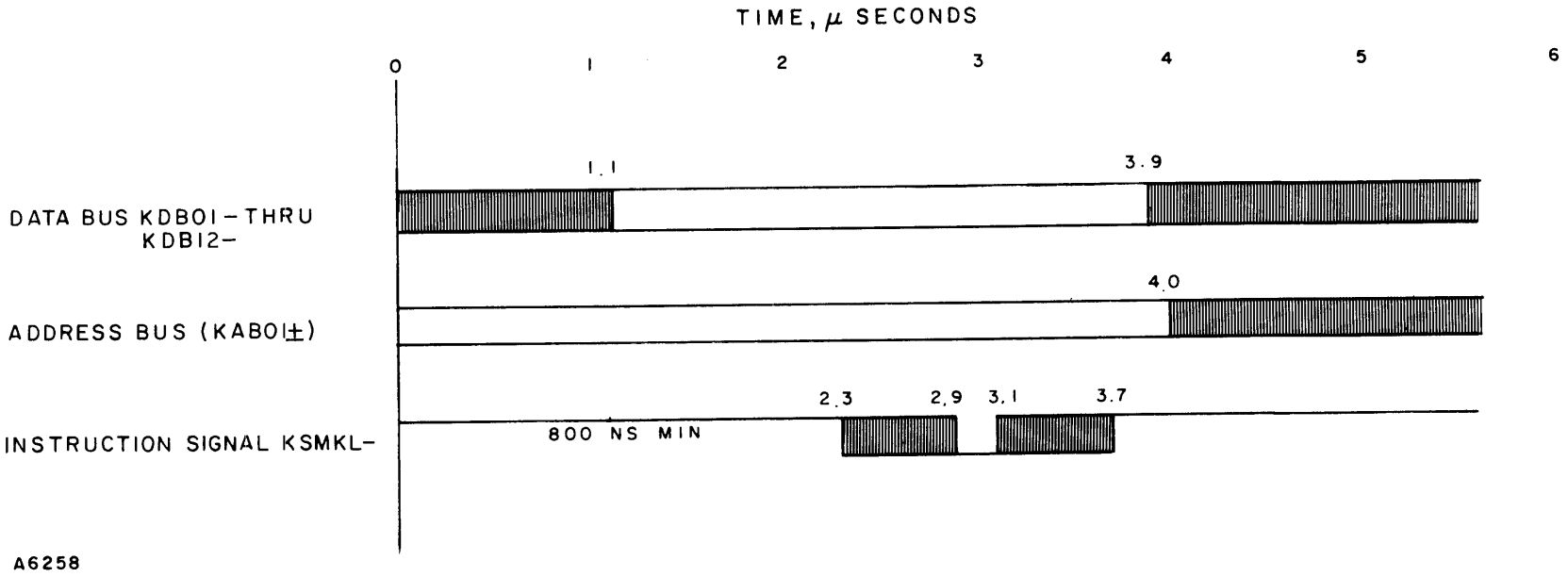


Figure 2-5. SMK Timing

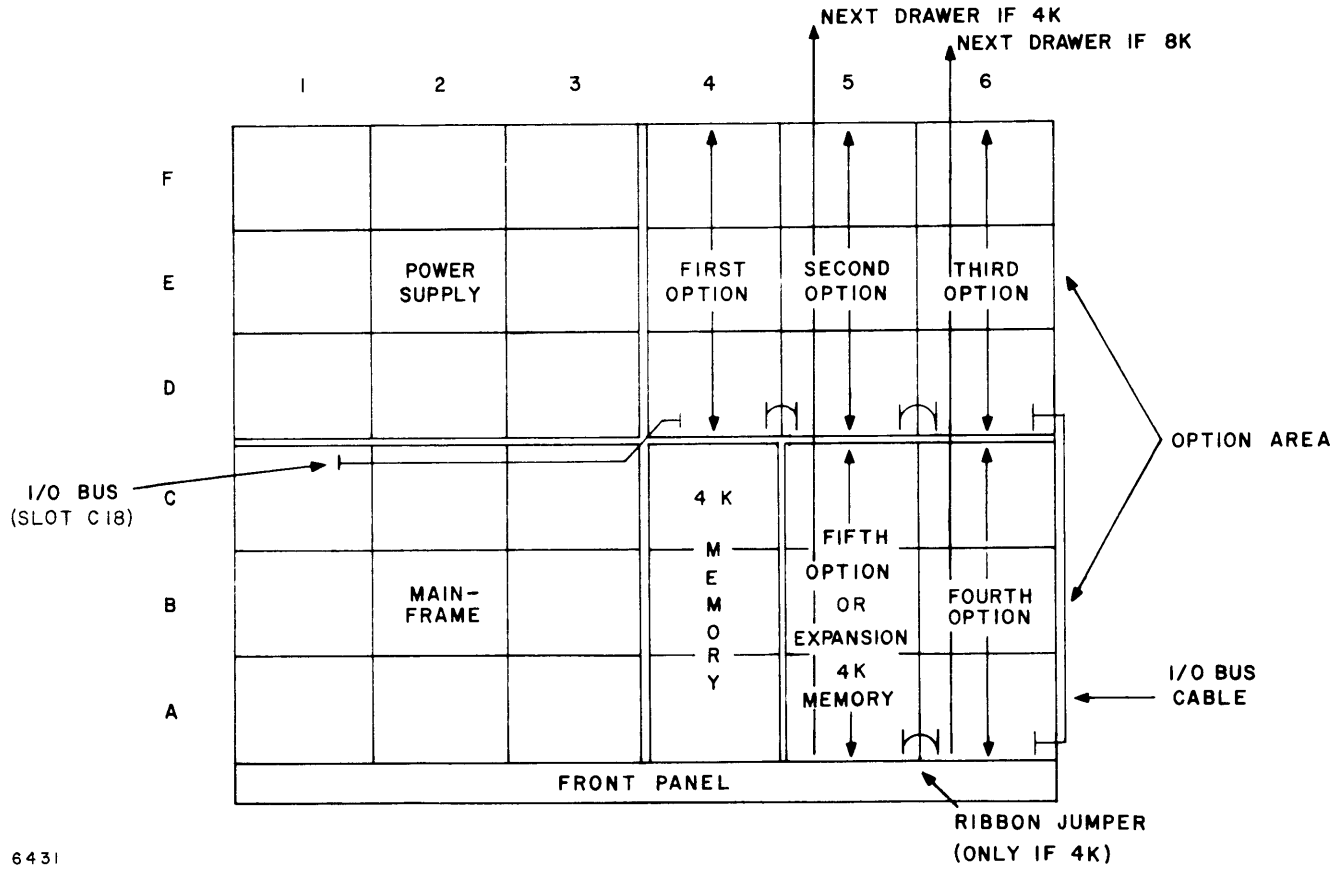


Figure 2-6. I/O Bus Cable Routing, PAC View

INTERFACING THE PROGRAMMED I/O BUS

The following section gives examples of the hardware which must be added in the interface for each instruction used in the H112 programmed I/O. For discussion purposes, hardware which can be time-shared in the interface will be presented as independent components.

When interfacing the programmed I/O bus, three steps must precede actual hardware design. The recommended steps are as follows:

a. Determine the functional or subsystem design.
b. Define the instructions needed to execute the above design.
c. Develop an interface to the external device.

Determining the functional or subsystem design involves defining, in general terms, how the CPU and external device will interact. Points to consider are: the paths for blocks of data; inherent synchronization problems; and control over the direction of data transfers.

In defining the instructions, the designer and programmer must agree on the following: which commands cause data transfer; what events cause interrupts; what resets interrupts; what conditions will cause OTAs and INAs to skip or not skip; and what SKS instructions are required.

Developing an interface with the external device involves a consideration of its signals, voltage levels, logic levels, and timing requirements.

After executing these steps, the actual design of the interface begins. To facilitate this task, an explanation of the design used to interface standard options is helpful. These are the recommended methods. Other designs can be used and function properly; however, care must be taken to avoid interfering with devices using standard methods.

All signals from the programmed I/O bus to the interface and from the interface to the bus must go through the recommended driver and receiver circuits. The design of these circuits guarantees proper termination and loading of the wires. Other circuits may cause spurious signals and faulty timing. PACs containing these circuits are available from Honeywell CCD. Descriptions of these circuits are presented in the appendix. See Appendix B for allocation of H112 device address assignments.

LOGIC DESIGN TO COMPLETE INSTRUCTIONS

OCP - Output Control Pulse

Upon execution of this instruction, a command pulse is to be delivered to the device (or device function) specified by the address field. The execution of this command is not contingent upon a device ready response and will never skip. Address field 00 and 01 are not available for OCP instructions.

To implement an OCP command, the address bus is terminated in line receivers and the outputs generated are then fully decoded with the equivalent of a 6-input gate.

If the address decode indicates that the least significant half of the W-register has a bit configuration which is an instruction to be mechanized in the interface, the output is gated with the received KOCPL- line as an assertion level and the KSTRB- as an assertion level.

Figure 2-7 shows OCP gating. The output from this gate is used to initiate the action assigned to the OCP command.

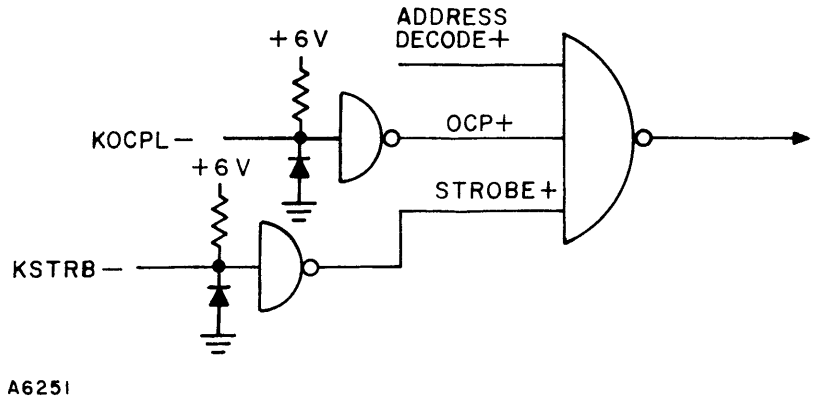


Figure 2-7. OCP Gating

OTA - Output Transfer from Accumulator

Implementing an OTA requires gating all the bits of the address bus together as in the OCP case. When the least significant half of the address bus (W-register) has a device address assigned to the interface, a +6 volt output is produced. If the OTA is contingent upon a ready flip-flop being set, the decoded address signal is gated with a positive OTA signal and the skip criteria (set side of the ready flip-flop). If, however, the interface is always ready for the OTA, the READY+ term is omitted. Figure 2-8 shows OTA gating with skip criteria.

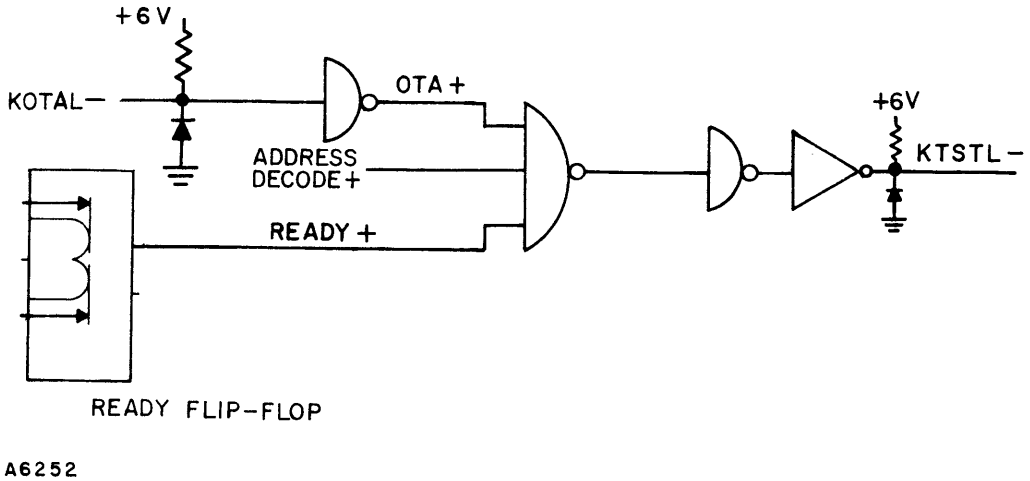


Figure 2-8. OTA Gating with Skip Criteria

The output of the gate drives the test line, KTSTL-. When the test line is grounded, the CPU will skip the next instruction and generate the strobe signal KSTRB-. If the test line is not grounded, the CPU executes the next instruction with no strobe signal generated.

Assuming a transfer condition, the interface grounds the test line and gates the data on the data bus (KDB01- to KDB12-) into a register. The terms in the gate are: OTA+; ADDRESS+; and STROBE+.

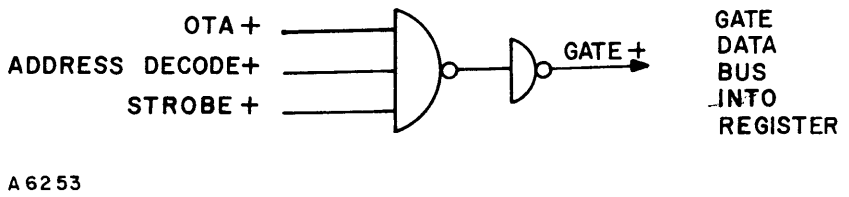


Figure 2-9. Data Gating on OTA Command

Figure 2-10 shows two typical methods for gating in data. The ready flip-flop can be reset with the leading or trailing edge of the gate pulse (GATE+), if that is desired.

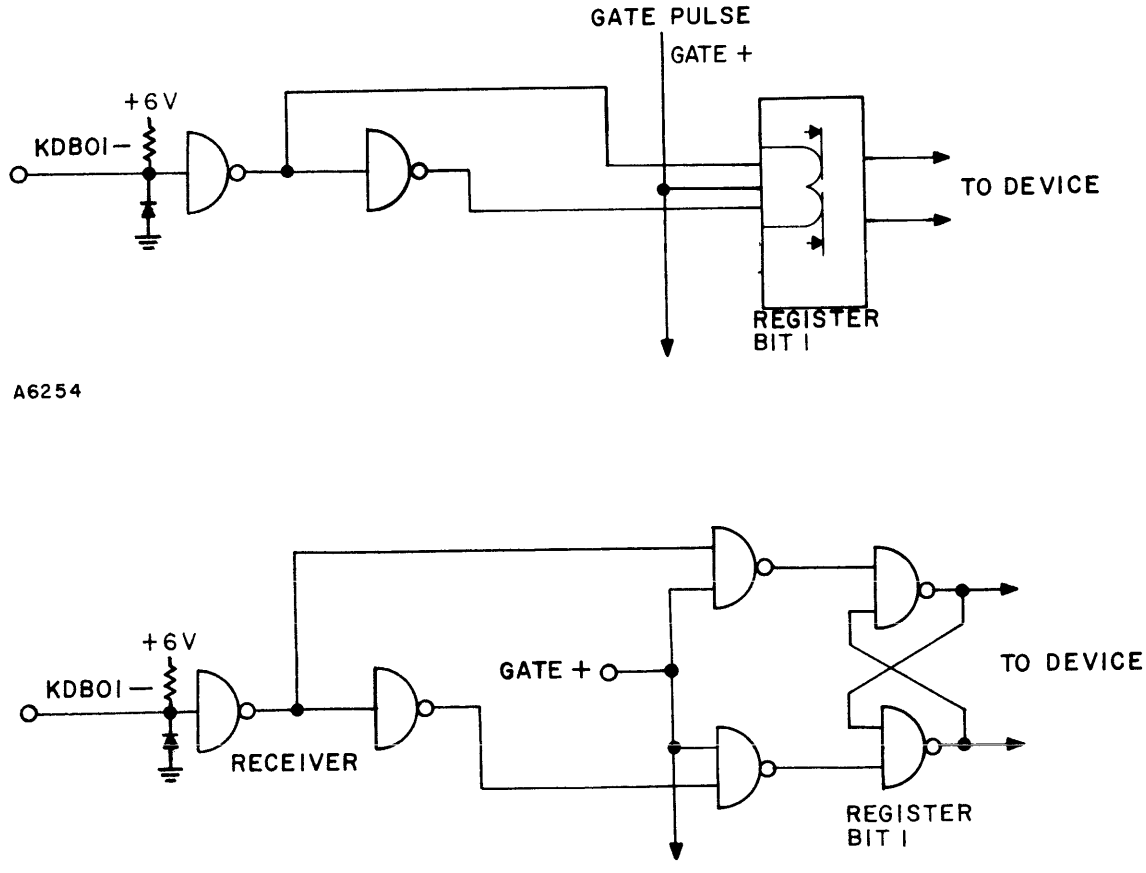


Figure 2-10. Typical Methods of Gating Data on OTA Commands

INA - Input Transfer to Accumulator

After address decoding, the skip condition is gated in the same manner as in the OTA instruction. If the test line (KTSTL-) is grounded, the CPU performs the following operations: it clears the accumulator and takes in data; issues the strobe signal KSTRB-, and skips the next instruction. If KTSTL- is not grounded, the CPU clears the accumulator and goes to the next instruction. Data is not taken into the accumulator and the strobe signal is not generated.

Figure 2-11 shows a skip gate being implemented with a decoded address, an INA signal, and skip criteria from the ready flip-flop.

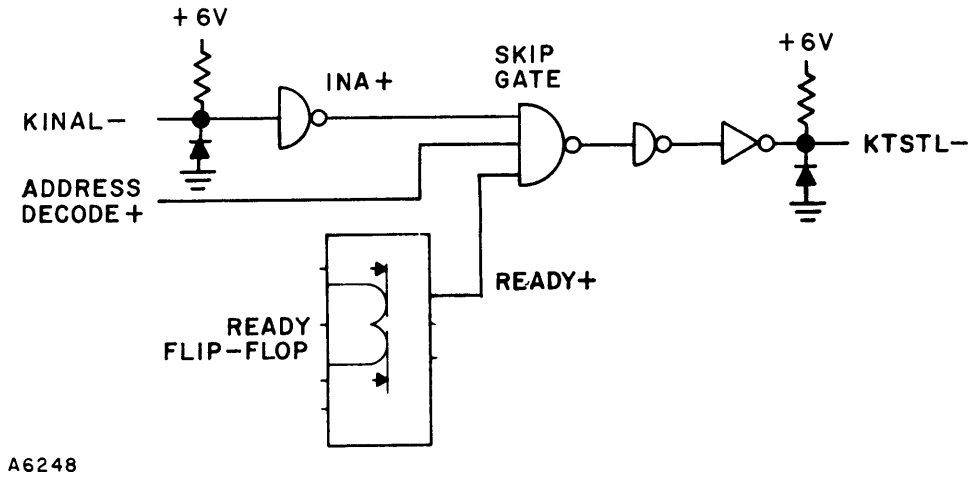


Figure 2-11. INA Gating

Because the data is transferred into the accumulator before the strobe pulse occurs, the data must be gated onto the data bus when the address decode and INA+ are true. See Figure 2-12.

NOTE

Even though a data transfer is not to take place, the address decode and INA+ may be true during transient conditions. This causes data to gate onto the bus. The same is also true for the KTSTL- test line. These conditions do not create a problem, however, because they will not occur while the lines are stable during the actual input time. Generally, in machines that place many spurious pulses on the bus, error-free data transfers are insured by gating critical lines. This is performed by the controller mainframe.

The leading or trailing edge of the strobe signal, generated by the CPU and gated with the address decode and INA+, may be used to reset the ready flip-flop. When the flip-flop is reset, data may also be removed from the data bus. It is not necessary to remove data, however, until the device address or INA goes false.

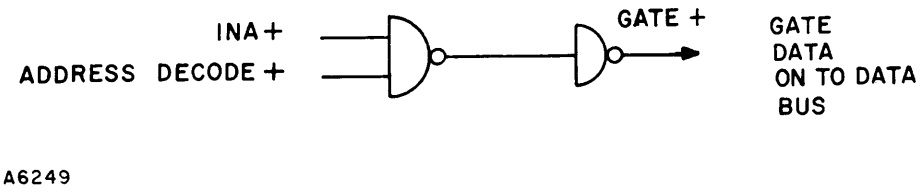


Figure 2-12. Data Gating to Data Bus on INA Commands

SKS - Skip if Set

If the device condition specified in the address field is true (set) and the device responds by grounding KTSTL-, the next sequential instruction will be skipped. If the condition is not true (not set) or the device does not respond, the next sequential instruction will be executed.

Implementing an SKS instruction requires a full decode of the address bus and a received SKS signal. These signals are gated with a signal indicative of the skip condition. The resulting signal is then used to drive the KTSTL- test line. Figure 2-13 shows SKS gating.

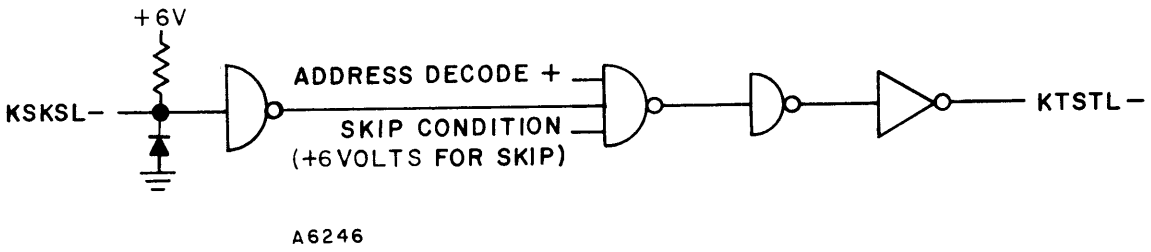


Figure 2-13. SKS Gating

SMK - Set Mask

To implement an SMK instruction, the SMK line is gated with address bit 1 (assertion or negation). The resulting signal from the gate determines whether the command was SMK0 or SMK1. If both the SMK and address bit are true, the assigned A-register bit (data bus) is copied into the mask flip-flop. If the A-register bit is a ONE, the mask bit is set and interrupts are permitted; if a ZERO, no interrupts are permitted. The truth table in the interrupt section (refer to page 2-18) illustrates the relationship.

Figure 2-14 shows a mask flip-flop assigned to bit 5 of SMK1. (Additional information on SMK is presented in the subsection on Interrupts.)

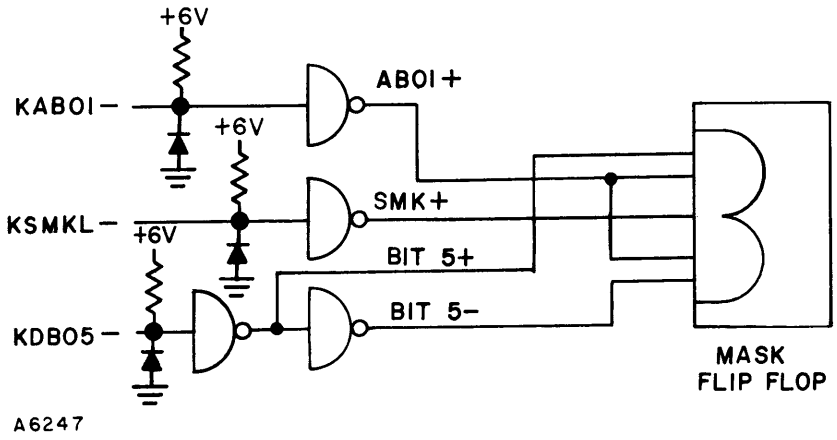


Figure 2-14. SMK1 Using Bit 5 to Control Mask Flip-Flop

STL - Stall Instruction

Although the hardware for the stall instruction is quite simple, the application may be varied. In the example in Figure 2-15 when the program needs to synchronize with the external system clock, the controller resets the flip-flop by separate programming action and then executes an STL instruction. When the system clock drops from +6V to ground, the flip-flop sets, and the stall breaks.

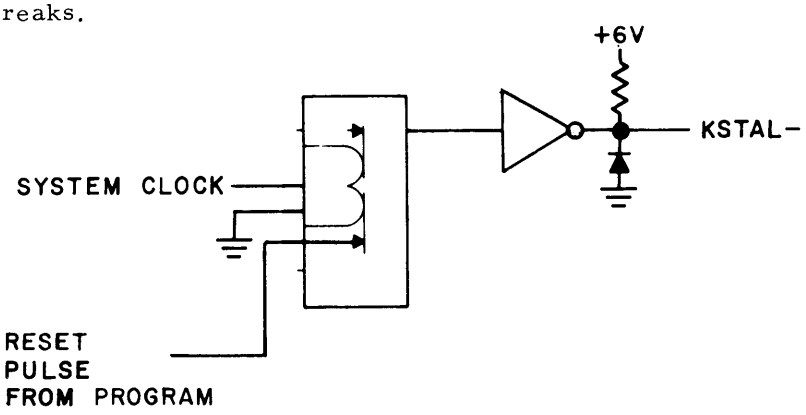


Figure 2-15. STL Gating

INTERRUPTS

An external device may create a program interrupt to request a data transfer or to inform the program of an event in the external system. The external device activates the interrupt by grounding interrupt line KINTL-. All devices share this line, therefore, in cases of simultaneous interrupts, more than one device may ground the line. A device must keep KINTL- at ground until the program informs the device that its particular interrupt has been recognized.

When interrupts are enabled in the controller and a device interrupts, the program does not execute the next sequential instruction. Instead, the program takes the next command from dedicated location 00002 and the controller inhibits further interrupts. Then, because there is only one interrupt line, the program must determine which device caused the interrupt.

Since the controller locates an interrupt by performing a series of SKS-JMP commands, each device must have a mechanized SKS (skip if device not interrupting) instruction. The SKS is designed to skip the JMP command if the device polled is not interrupting. This polling continues until the SKS detects the interrupting device. At this time, the JMP command is executed, transferring control to a subroutine which services the device.

Servicing an interrupting device involves several events. First, the data transfer or action called for by the device must take place. Secondly, the interrupting flip-flop in the device must be reset. This may be done by the data transfer command (OTA or INA) or by implementing a reset command (i. e., an OCP).

In this type of system, the order of SKS commands determines the priority assigned to simultaneous interrupts because the first interrupting device receiving an SKS is serviced. The requirements of some systems permit disabling of all other interrupts while servicing the first recognized interrupt. Other systems, however, have interrupts requiring immediate recognition and service when they occur. These systems need the ability to enable priority interrupts in a selective fashion. This feature allows urgent interrupts to be recognized during a long service routine of a less urgent interrupt.

The H112 can accommodate either system type with its multiple level interrupt capability. To make the multiple level priority scheme, the H112 uses a mask bit in each external device.

In a particular device, the mask bit (mask flip-flop) is gated with the interrupt flip-flop; and the interrupt line is grounded (activated) only when both flip-flops are set. This case represents an interrupt from a device in MASK ON condition. If the mask bit is reset and the interrupt flip-flop is set, the interrupt line is not grounded and the SKS (skip if not interrupting) will skip. Thus the state of the interrupt flip-flop has no effect.

The following truth table represents the conditions available with interrupt and mask flip-flop gating: (see Figure 2-17).

Interrupt Flip-Flop	Mask Flip-Flop	Interrupt Line	SKS Command
0	0	Not Active (+6V)	Skip
0	1	Not Active (+6V)	Skip
1	0	Not Active (+6V)	Skip
1	1	ACTIVE (Gnd)	Not Skip

The state of the mask flip-flop is under the control of the program via SMK (SET MASK) instructions. Each standard device has one mask bit and an associated bit in the accumulator. The mask flip-flop copies the state of the accumulator bit upon execution of the proper SMK instruction. For example, a one in an accumulator bit permits interrupts from the associated device, while a zero inhibits interrupts from that device.

There are two SMK instructions, SMK0 and SMK1. Twelve devices copy the accumulator on SMK0 and twelve additional devices copy on SMK1 for a total of 24. OCP and OTA instructions can be used if the system has more than 24 devices.

Figures 2-16 and 2-17 show the gating associated with the interrupt flip-flop, mask flip-flop, SMK instruction, and SKS instruction.

The mask flip-flop copies its assigned accumulator bit when the KSMKL- line goes active (ground) with the appropriate address bus level. Since the KSMKL- line pulses during SMK0 and SMK1, the KDB01- or KDB01+ line is used to determine which instruction is in progress. Resetting the mask flip-flop is done conventionally with Master Clear.

Generation of the interrupt and the SMK instruction is self explanatory. If the multiple level arrangement is not needed, the mask flip-flop, the hardware to set it, and its terms in the interrupt and SKS gates may be omitted.

Figure 2-16 shows simplified set and mask flip-flop logic for the SMK instruction.

Figure 2-17 shows the logic for gating with the mask flip-flop using the test line to check for interrupts.

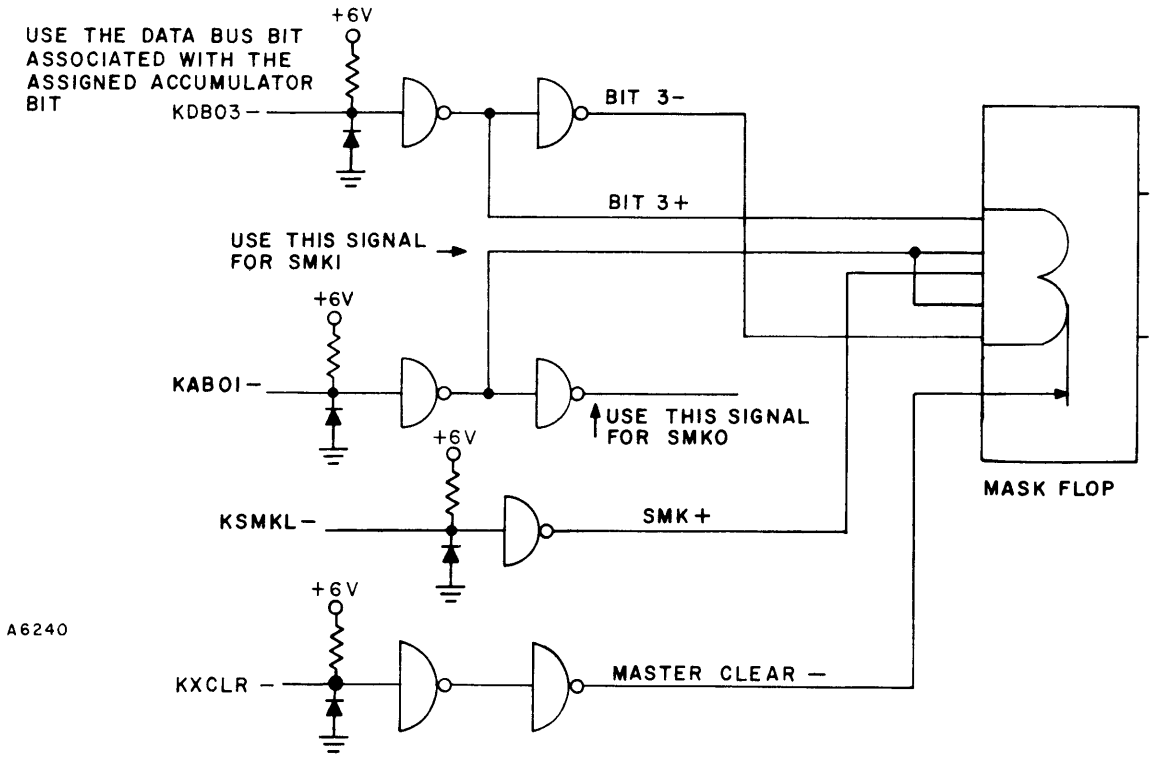


Figure 2-16. SMK Logic

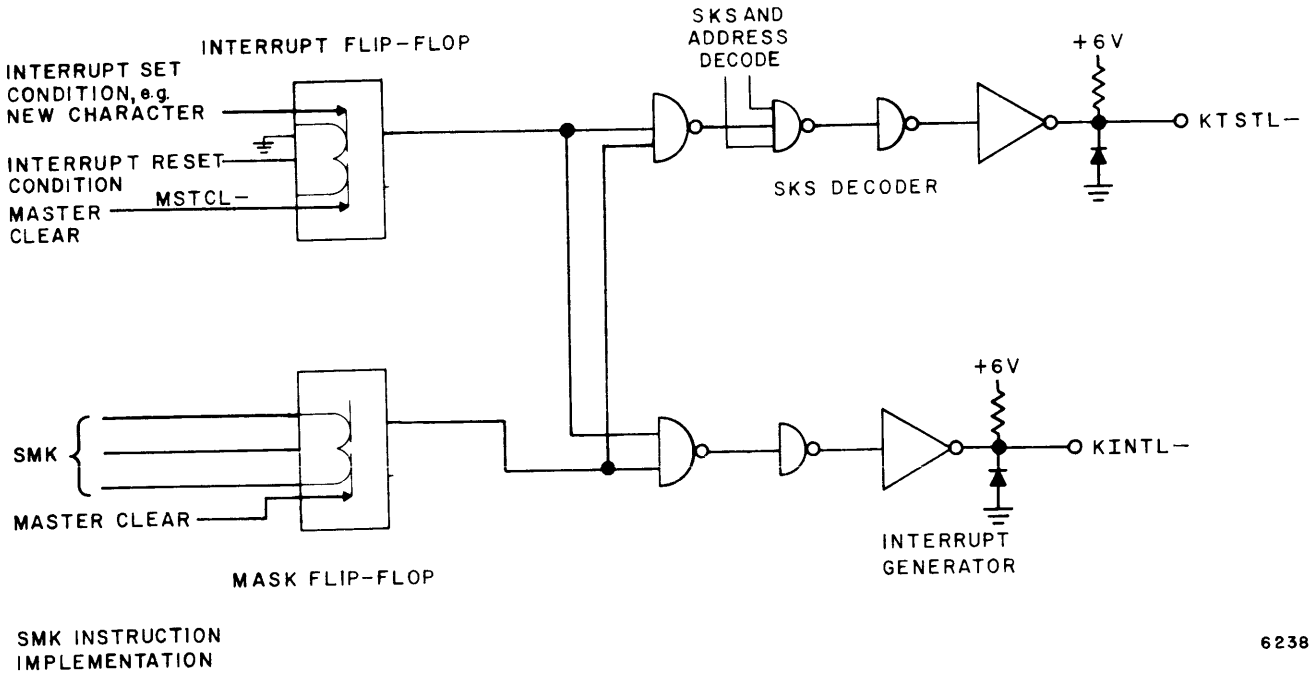


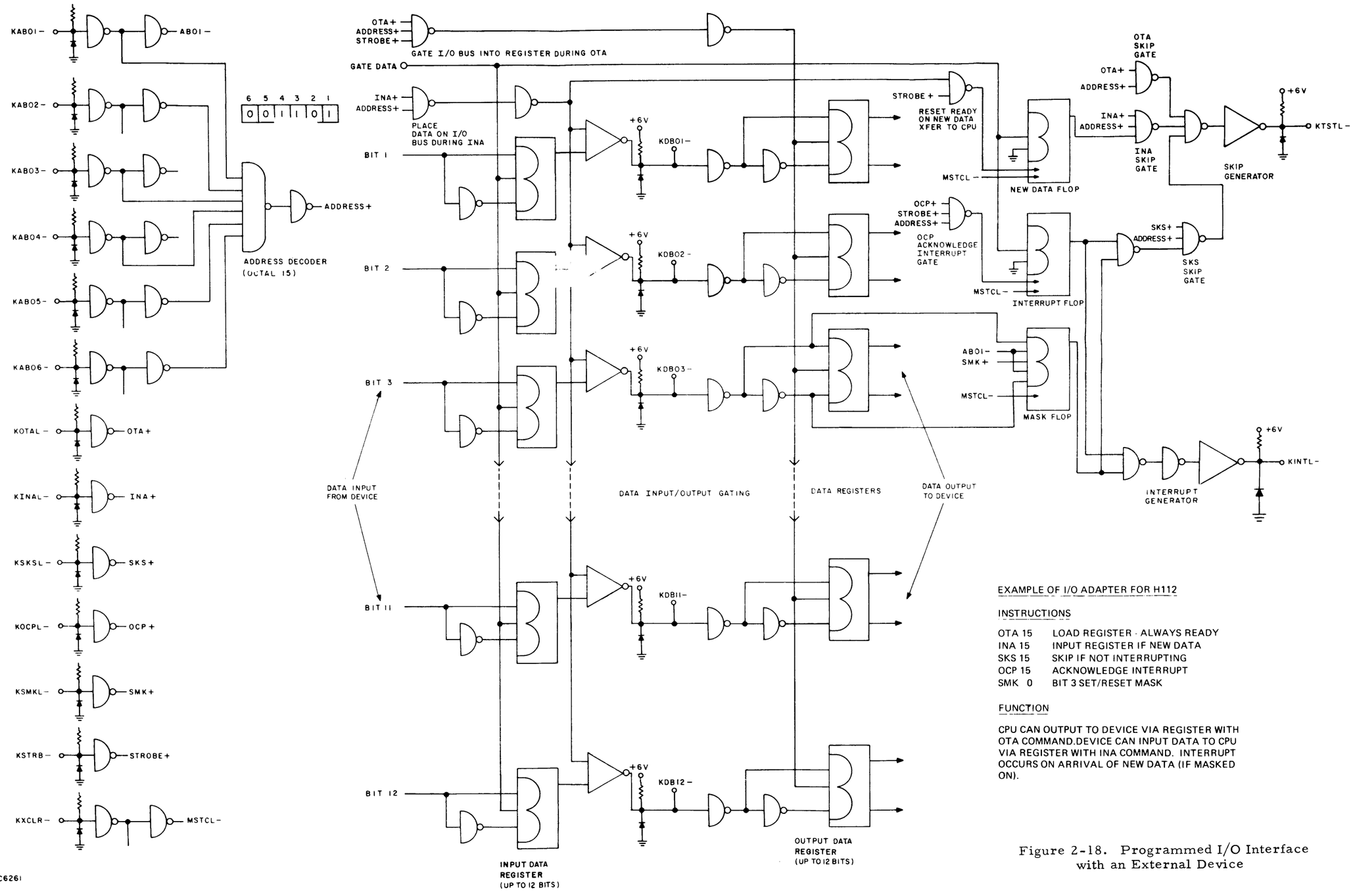
Figure 2-17. Gating with Mask Flip-Flop

Interface Example for Programmed I/O Data Transfers

The logic in Figure 2-18 is designed to transfer data between an external (remote) device and the H112 via the programmed I/O bus. The central processor can output data at any time to the device via the output register. Since the output transfer can occur at any time, the OTA skip gate contains only the device address decode (ADDRESS+) and OTA+ and has no skip term. When the strobe occurs during OTA 15, data is gated from the data bus into the output register.

When the external device sends data to the interface logic, the data is presented on the input lines to the input register. The data is loaded when the external device pulses the line marked "gate data." At the same time, the "gate data" line sets the "new data flip-flop" and the "interrupt flip-flop." Assuming that the mask flip-flop is set, an interrupt is generated. When the program interrogates the interface with the SKS 15 (skip if not interrupting) command, the SKS skip gate is blocked and the instruction does not skip. The controller may now issue an INA 15 which causes an active (ground) test signal because all three terms into the INA skip gate are at +6 volts. The data from the input register will be gated onto the data bus. The new data flip-flop (but not the interrupt flip-flop) will be reset via the strobe signal. Note that a second INA 15 issued at this point will not skip or transfer data, thus new data from the device is transferred into the controller only once. To reset the

interrupt flip-flop, an OCP 15 command is used which is implemented in the interrupt acknowledge gate. In this example, the transfer of data could also have been used to reset the interrupt flip-flop. With the resetting of the interrupt flip-flop, the interrupt line is cleared. The mask flip-flop is treated in the conventional fashion.



EXAMPLE OF I/O ADAPTER FOR H112

INSTRUCTIONS

OTA 15	LOAD REGISTER · ALWAYS READY
INA 15	INPUT REGISTER IF NEW DATA
SKS 15	SKIP IF NOT INTERRUPTING
OCP 15	ACKNOWLEDGE INTERRUPT
SMK 0	BIT 3 SET/RESET MASK

FUNCTION

CPU CAN OUTPUT TO DEVICE VIA REGISTER WITH OTA COMMAND.DEVICE CAN INPUT DATA TO CPU VIA REGISTER WITH INA COMMAND. INTERRUPT OCCURS ON ARRIVAL OF NEW DATA (IF MASKED ON).

Figure 2-18. Programmed I/O Interface with an External Device

SECTION III
LOAD MODE TRANSFERS

Load mode transfers load 3-bit characters into the H112 controller for eventual storage in core as 12-bit packed words. Transfers occur over the programmed I/O bus under H112 hardware control with no software program being executed. All load mode transfers are inputs to the controller.

Four transfers of 3-bit data characters are required to pack each location (starting location is determined by the contents of the P-register when function is initiated). Timing for transfers of data bits into the H112 is controlled by the transition of KDB09- line from +6 volts to ground. This strobe signal from the input device may have a period as short as 10 μs or as long as required by the sending device. The +6V to ground transition of KDB09- loads the three least significant bits of the I/O bus into the three most significant bits of the accumulator. The accumulator is then stored in the memory location determined by the P register. On the next KDB09- ground-going transition, the following action occurs:

a. the accumulator (with the first character) is right-shifted three places.

b. the three least significant bits of the I/O bus are loaded into the three most significant bits of the accumulator.

c. the accumulator with both characters is stored in memory at the location determined by the P-register overwriting the previously stored character.

After total of four characters are stored in this memory location, the P register is incremented and the next location is filled. Thus, to load a location with 4 3 2 1, the sequence of input characters must be 1 2 3 4. This sequence continues until a signal is recognized by the controller on KDB07- which terminates the process. The remainder of the bit configuration included with the KDB07- signal is ignored.

Figure 3-1 shows the generation of data characters and control signals in their respective timing sequences.

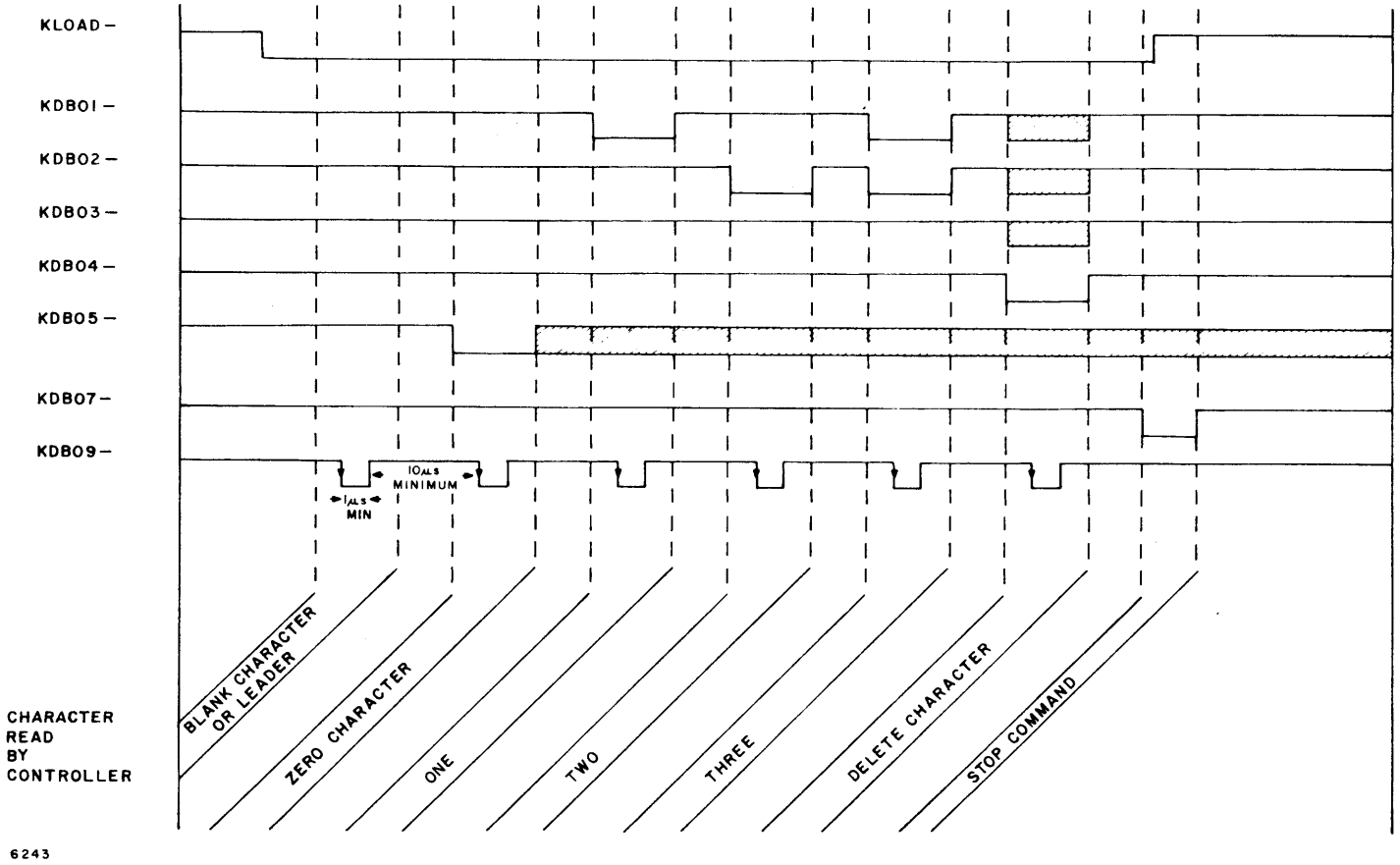
Table 3-1 lists the load mode signal mnemonics, functions and pin assignments. All the loading and circuit rules of the I/O bus apply. Standard receiver and driver circuits must be used. (See Appendix A.) Data signals need be stable only during the KDB09- ground-going transition and 3 μs afterward. The states of KDB06, KDB08, KDB10, KDB11, and KDB12 are not important.

Table 3-1.
Signal Functions

Mnemonic	Function	Pin
KLOAD-	Load mode transfers: Initiated by operation of load switch and start switch on control panel or activation of load line and start line from machine control interface. When these conditions occur, the KLOAD- signal become active (ground) and remains active until the recognition of a stop command. After recognition of the stop signal, the KLOAD- line becomes passive (+6V). This output signal from the controller is to be used by the interface unit as a command to initiate transfers.	26
KDB01-	Least significant input bit of octal character; passive (+6V) is defined as a ZERO, ground as a ONE. This signal must be passive during a blank character or leader.	1
KDB02-	Second most significant input bit of octal character; passive (+6V) is defined as a ZERO, ground as a ONE. This signal must be passive during a blank character or leader.	2
KDB03-	Most significant input bit of octal character; passive is defined as a ZERO, ground as a ONE. This signal must be passive during a blank character or leader.	3
KDB04-	When this input signal is active (ground) it is interpreted as a delete character signal, and causes the data character associated with it to be treated as a blank character or leader no matter what the character content.	4
KDB05-	This input signal must be passive (+6V) during blank character or leader; when active (ground) it differentiates a ZERO character from blank or leader.	5
KDB07-	When this input signal is made active (ground) it is interpreted as a stop command. It is not gated with the transition of KDB09- as is a data character. A pulse on this line at any time will stop the controller. This character can not be deleted.	7
KDB09-	Transition from +6V to ground on this input line gates data character into the controller. This strobe pulse from the input device also causes shifting and storage of the accumulator and after every fourth shift, advancing of the program counter.	9

The following example (Figure 3-2) shows a typical load mode interface from a paper tape reader.

Since more than one device can be operated in the load mode and the controller does not select which device is to be read, each device must have a means by which the operator can select the desired device. The standard devices which can operate in the load mode are the input/output typewriter and the high-speed paper tape reader.



NOTE: OTHER OCTAL CHARACTERS NOT SHOWN IN THIS DIAGRAM REQUIRE THE CORRESPONDING BIT PATTERN ON LINES KDB01– THROUGH KDB03–; THE STATE OF KDB05– IS NOT IMPORTANT FOR OCTAL 1–7.

Figure 3-1. Timing for Load Mode

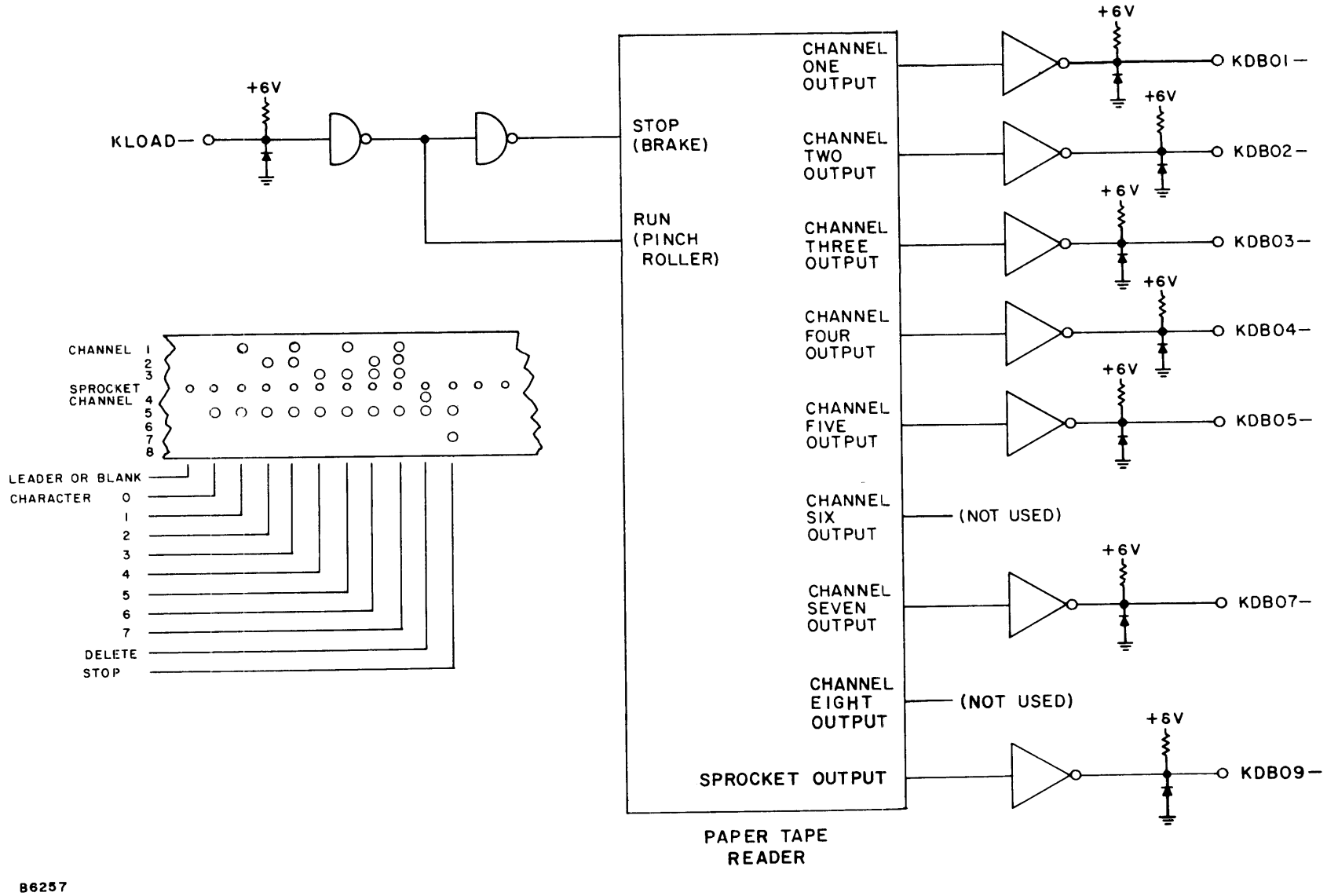


Figure 3-2. Typical Load Mode Interface from Paper Tape Reader

SECTION IV
DIRECT DATA CHANNEL OPTION

The purpose of the Direct Data Channel (DDC) is to provide an I/O path for high-speed data transfers between the controller and I/O device. Communication is via the DDC bus and programmed I/O bus.

The transfer is entirely under the control of the external device and can occur at the end of any instruction. The external device controls address of the memory location, the direction of transfer and the number of cycles to be taken at one time. The hardware in the controller performing the transfer is called a channel. There are two channels in the DDC option (A and B). The device or device adpater hardware is designated as the subchannel. This device hardware is not included as part of the DDC option. In the case of simultaneous requests, channel A has priority.

SIGNALS

All signals on the DDC bus are active, true or represent a logical ONE when at ground. They are passive, false or represent ZEROs when at +6 volts. Signal timings are guaranteed at the device adapter (subchannel) when the standard cable is used for interconnections. Each subchannel is connected to a channel by a separate cable; the cables are not daisy-chained. The length of each cable may be no longer than 10 feet.

Each subchannel may load a signal line on the DDC bus with up to 3 loads. A load is to consist of a standard gate input in parallel with a 6.8K ohm resistor to +6 volts, and the cathode of a grounded diode. The controller uses a type 932 integrated circuit power amplifier circuit to drive the output signals on the DDC bus. A subchannel must drive the channel via the DDC bus with standard drive circuits used on the programmed I/O bus. This circuit is a type 944 integrated circuit power amplifier (uncommitted collector) with a 6.8 Kilohm pull up resistor to +6 volts in parallel with the cathode of a grounded diode.

Standard driver and receiver circuits are contained on the CS-517, CS-520, CS-521 and CS-548 PACs (see Appendix A). Those signals on the programmed I/O bus which are actively used during DDC transfers must be received and driven by standard I/O circuits. All rules for the I/O bus apply.

DDC Interface

The interface between the channel and a subchannel consists of the signals shown in Figure 4-1. All of the signal names which end in "A" apply to subchannel A and those that end in "B" apply to subchannel B.

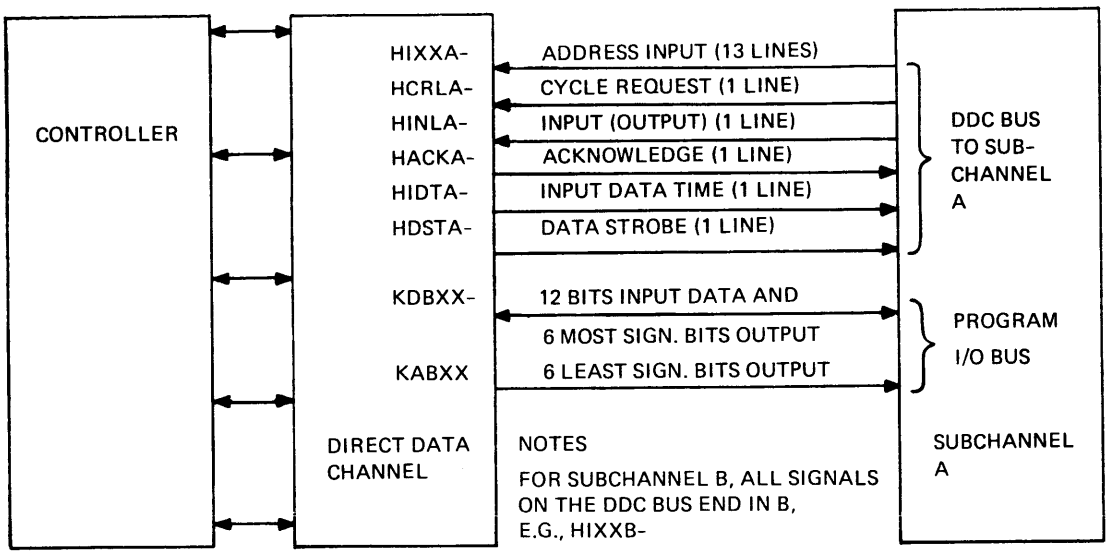


Figure 4-1. DDC Interfacing

The procedure to execute a data output DDC cycle is for the subchannel to place the address of the location in memory from which the data is to be taken on the DDC address bus (HIXXA- or HIXXB-). The address must be a 13-bit address. Input line HINLA- or HINLB-) is made passive (+6 volts) to indicate to the channel that an output cycle is required. After these two items have been accomplished, the cycle is requested by activating (grounding) the cycle request line (HCRLA- or HCRLB-). The address must be stable at the channel before the request is made. When the request is recognized, the channel will acknowledge only the appropriate subchannel by signal HACKA- or HACKB- and place the six least significant data bits on the program I/O bus lines KAB01- through KAB06- and the six most significant bits on the program I/O bus lines KDB07- through KDB12-. The channel will indicate when the data is stable and valid only to the recognized subchannel by HDSTB- or HDSTA- which is used to strobe the data into a register. The acknowledge signal (HACKX-) is used to reset the request.

To produce a DDC cycle which will input data to the controller, the address of the desired memory location is placed on the DDC address bus (HIXXA- or HIXXB-). The input line (HINLA- or HINLB-) is activated (ground) to command an input transfer. Cycle request (HCRLA- or HCRLB-) must be made only after the address lines are stable. The channel will acknowledge only the appropriate subchannel via HACKA- or HACKB-. The channel will also indicate that the input data is to be placed on the program I/O data bus, KDB01- through KDB12-, via activation of the HIDTA- or HIDTB- line (input data time). Input data is then gated on the data bus and must be valid and stable during the time indicated on the timing diagram, Figure 4-2. Only the acknowledged subchannel will be requested to gate data onto the data bus.

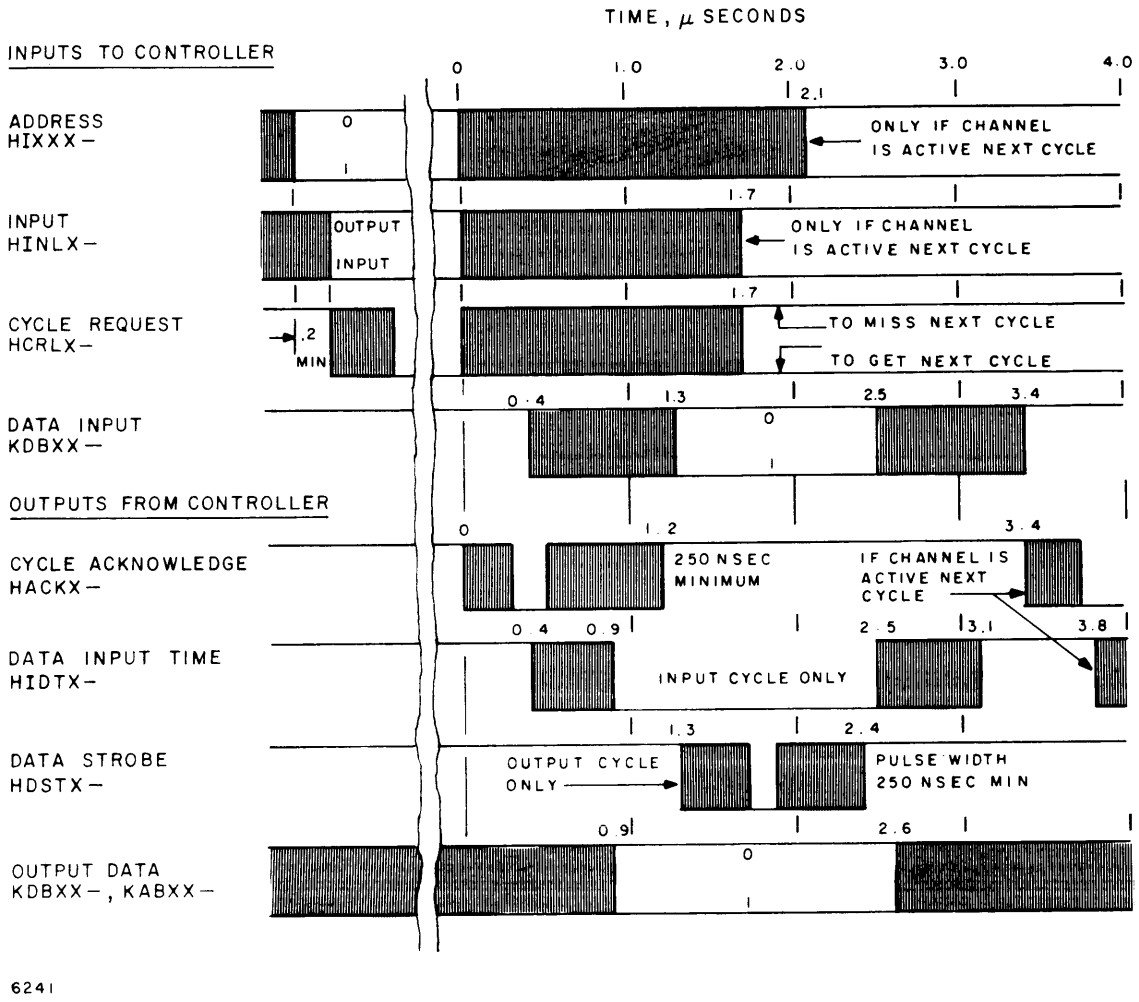


Figure 4-2. Direct Data Channel Timing (Composite)

A single DDC cycle takes 3.816 μs to execute. The same period of time is taken from the execution of the program. If a second request (from either the active or inactive subchannel) is received before the completion of the first transfer, the second DDC cycle will take place immediately following the first. The second and all subsequent DDC cycles takes 3.392 μs if all subsequent cycle requests are made prior to the completion of the previous transfer. If continued requests are made, the program execution may be locked out entirely. The DDC transfer alters no programmable registers, flip-flops or machine states (interrupt inhibits, etc.) except for the core location into which data might have been written. DDC cycles take place only when the machine is running (not halted due to a HLT instruction, or stalled due to a STL instruction). The maximum wait before a DDC cycle starts is 11 μs (the longest instruction) for Channel A. Channel B has the same maximum wait except if a Channel A request locks Channel B out.

Table 4-1 gives pin assignments, signal mnemonics, and functions for the DDC option. The location where the cable to subchannel A is plugged is in mainframe option OMNI-BLOC at mainframe drawer location D35. Cable B is plugged into D36.

Table 4-1.
DDC Bus Signals

Pin	Mnemonic Subchannel A	Mnemonic Subchannel B	Function
1	HI01A-	HI01B-	Address in Memory from which data is to be read or into which data is to be written
2	HI02A-	HI02B-	
3	HI03A-	HI03B-	
4	HI04A-	HI04B-	
5	HI05A-	HI05B-	
6	HI06A-	HI06B-	
7	HI07A-	HI07B-	
8	HI08A-	HI08B-	
9	HI09A-	HI09B-	
10	HI10A-	HI10B-	
11	HI11A-	HI11B-	
12	HI12A-	HI12B-	
13	HI13A-	HI13B-	
14	HACKA-	HACKB-	Acknowledge
15	HDSTA-	HDSTB-	Data Strobe (Output Data Valid)
16	HIDTA-	HIDTB-	Input Data Time (Gate Data In)
17	HCRLA-	HCRLB-	Cycle Request Line
18	HINLA-	HINLB-	Input Line
20			
21			
22			
23			
24			
25	SPARE		
26			
27			
28			
29			
30			
31			
32			
Location	F41	F48	

DDC Signals

Each of the following signals are unique for each subchannel, i.e., signals are not connected together at the controller.

HI01A- through HI13A- (Subchannel A) and HI01B- through HI13B- (Subchannel B).-- Thirteen input signals from the subchannel to the channel which specify the memory location the data is to be written into or out of. H113 is the most significant and H101 is the least significant. These lines must be valid before the request line is activated. A ONE is active (ground) and a ZERO is passive (+6V).

HCRLA- or HCRLB- (Cycle Request). -- One input line to the channel from the subchannel which is the request for a DDC cycle. The signal is active (requesting) when at ground. The signal may remain grounded to request multiple cycles.

HINLA- or HINLB- (Input Line). -- One input line to the channel from the subchannel which determines the direction of data transfer. The line must be stable and valid when the request line is activated. An active (ground) level will cause data transfers into the controller while passive (+6 V) levels will cause output transfers.

HACKA- or HACKB- (Acknowledge). -- One output signal to the subchannel which indicates to the subchannel that its request line has been recognized. The signal is active (ground) when acknowledging. The transitions of this signal are unambiguous (non-glitching). The signal remains passive during transfer to the other channel.

HDSTA- or HDSTB- (Data Strobe - Output). -- One output signal to the acknowledged subchannel during output transfers indicating that the data on the programmed I/O bus KAB01- through KAB06- and KDB07- through KDB12- is valid. This signal is used to strobe the data into a register in the subchannel. The signal is active (ground) when data is true. The signal remains passive (+6 V) during input transfers to either subchannel and during output transfers to the other subchannels.

HIDTA- or HIDTB- (Input Data Time). -- One output signal to the acknowledged subchannel during input transfers which command the subchannel to gate the data on the programmed I/O bus KDB01- through KDB12-. The signal is active (ground) when commanding the gating. The signal remains passive (+6 V) during output transfers to either subchannel and during input transfers to the other subchannel.

Figure 4-2 shows the composite timing requirements for the Direct Data Channel Option.

DDC SUBCHANNEL - EXAMPLE 1

The function of the subchannel shown in Figure 4-3 provides for the transfer of data to and from an external device. The sequence of events is for the device to initialize the subchannel and request the first output transfer. After the device accepts the data, the device loads the subchannel with data to be input to the mainframe. A total of 64 data transfers is required to complete the sequence: 32 output transfers alternated with 32 input transfers. The core map is shown in Figure 4-4. The table of data to be output is contained in octal addresses 07000 through 07037 and the device puts input data from 07040 through 07077. The tables are filled and read out starting with the lowest address, i.e., the first output is from location 07000, the first input to 07040, the second output from 07001, etc.

The diagram Figure 4-3 shows the logic (subchannel) required to provide DDC operation between the channel in the controller and the device. The sequence of events is as follows:

a. The subchannel receives an initializing pulse on line START+ from the external device. This command requests an output cycle (sets request flip-flop) and places address 07000 on the address bus. The input flip-flop is reset to cause an output transfer.

b. The controller executes a DDC output cycle causing data from core location 07000 to be presented on KAB01 through KAB06 and KDB07 through KDB12. The data is gated into the data output register with HDSTB-. The request flip-flop is reset with the acknowledge line HACKB-.

c. After the device has taken the output data, it then loads input data into the input register with a pulse on the DATA+ line. This line also sets the input flip-flop and the request flip-flop creating a request for an input DDC transfer. The address presented to the controller is 07040 because the input flip-flop controls address bit 6.

d. When the input DDC cycle is executed, the data to be input is gated onto the bus (KDB01- through KDB12-) by HIDTB- (input data time). The acknowledge line (HACKB-) resets the request flip-flop. The acknowledge line is also gated with the input flip-flop to increment the address register after the transfer.

e. When the device is ready for additional data from the CPU, a pulse is generated by the device on the MORE+ line. This pulse sets the request flip-flop and resets the input flip-flop thus generating an output DDC transfer request. The address is 07001.

f. The above sequence of first output then input transfers continues until the address register contains 37 at which time DONE+ is generated. The device is then signaled to stop transfer requests.

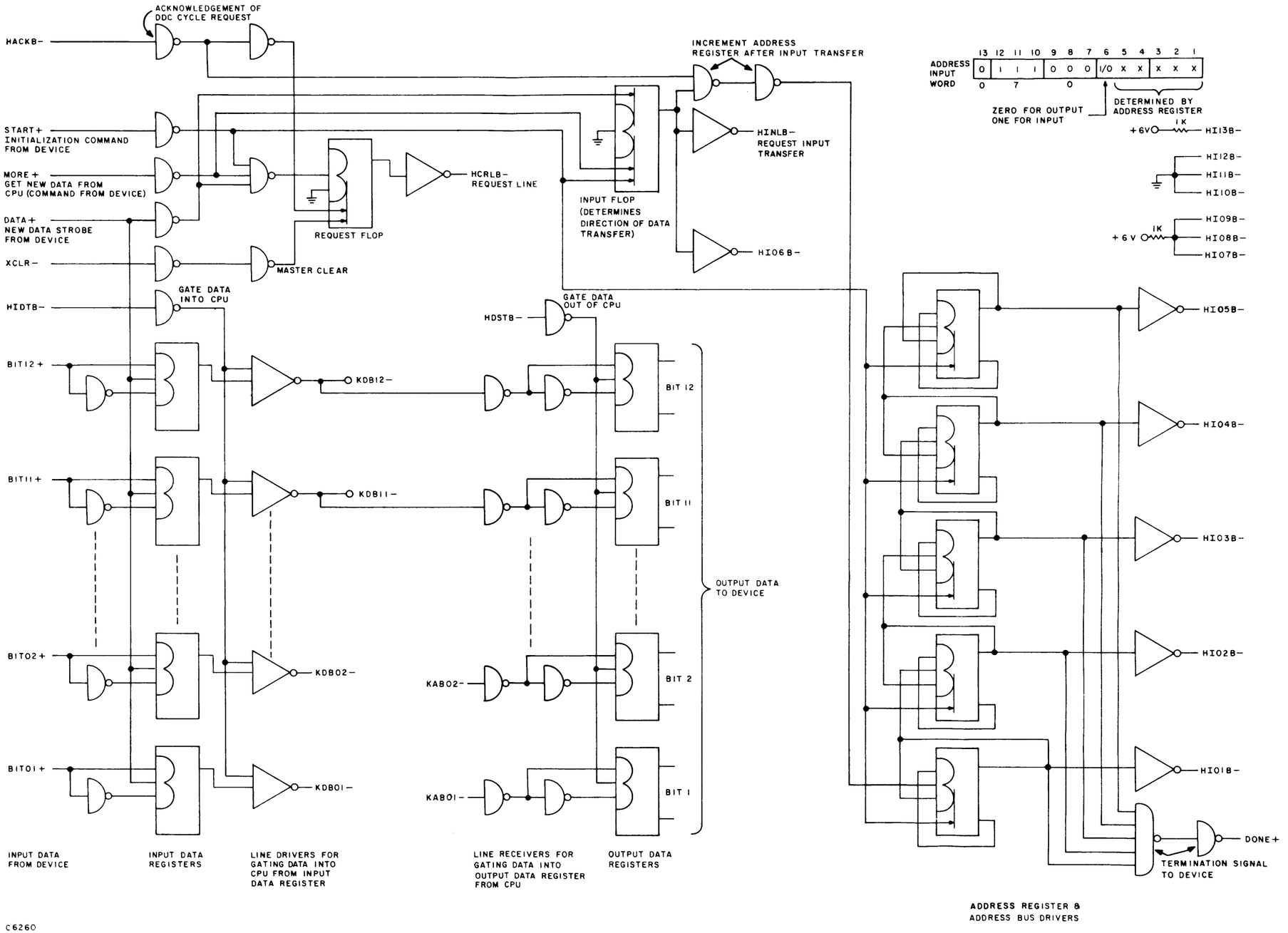


Figure 4-3. Example 1 of a DDC Subchannel

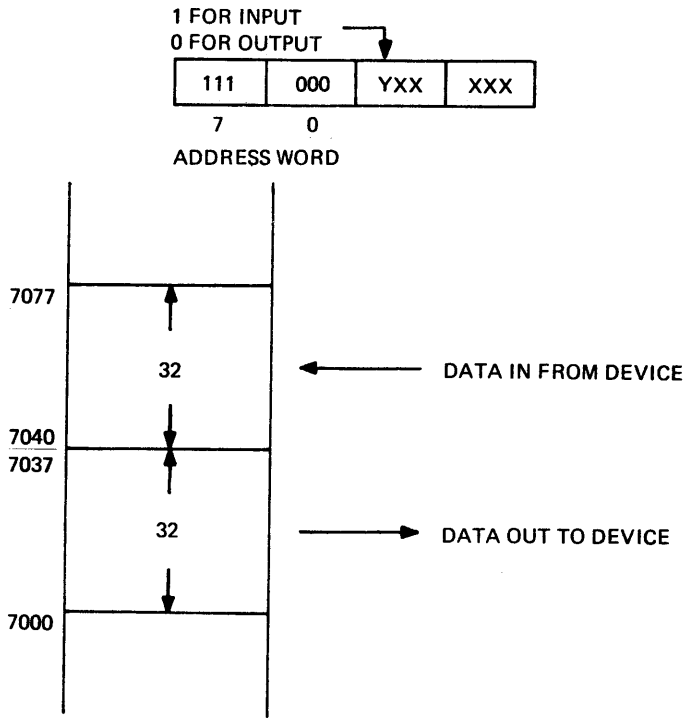


Figure 4-4. Core Map and Address Word

DDC SUBCHANNEL - EXAMPLE 2

Figure 4-5 shows the block diagram of a DDC subchannel whose function is to transfer the contents of a number of sequential memory locations from the controller to a peripheral device or in the opposite direction. The address of the first memory location, the number of locations to be transferred, and the direction of transfer are stored in the subchannel by a series of OTA commands. The address is stored in the address register and is incremented after every transfer. The number of transfers to be made is stored in the range register, which is decremented after each transfer. When the range register reaches zero, no further DDC requests are made and an interrupt is generated. An OCP command is used to reset the interrupt. The rate of data transfer is controlled by the external device. The synchronization logic is used to request a DDC cycle each time the peripheral device is ready to transmit or receive a new word.

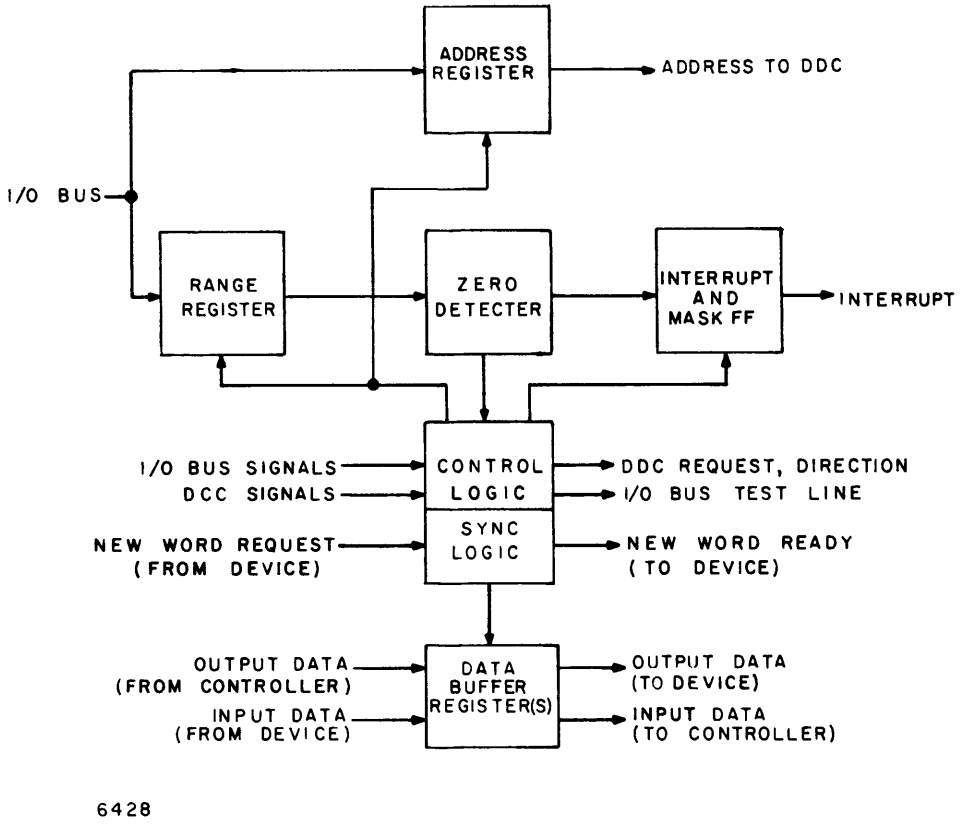


Figure 4-5. Example 2 of a DDC Subchannel

SECTION V
MACHINE CONTROL INTERFACE

Certain functions are available to the system through the machine control interface even when the full control panel is not included in the machine. These control functions are obtained by plugging a μ-PAC cable into slot C11 of the mainframe. The available functions are as follows: run, stop, load, master clear, start, and a single instruction execute.

All signals into the machine are ground true and are compatible with μ-PAC diode transistor logic. Each signal represents no more than 2 μ-loads with the exception of XCLR- which is a 100 mA load from +6 volts when the line is grounded. The output line is from a μ-PAC DTL gate.

The signals are not interlocked in the machine; therefore, if interlocking is desired it must be done in the remote panel.

SIGNALS

MASTER CLEAR (XCLR)

When this input line to the controller is activated, the P-register, overflow flip-flop and other critical control flip-flops are reset. If this line is activated (grounded) while the machine is running or in the load mode, the effect on memory is undefined. This line can be driven by a single-pole, single-throw momentary switch.

STOP/RUN (XXSS)

When this input line to the controller is made passive (+6), and the start lines are properly manipulated, the machine will access the memory location determined by the P-register and start execution of the program. If this line is made active (ground), the machine will stop after the completion of the instruction in progress. The machine will stop if either this line is grounded or the RUN/STOP switch on the panel is in the stop position. To cause the machine to run, this line must be at +6V and the RUN/STOP switch on the panel (if present) must be in the run position. This line should be returned to +6V with a resistor of less than 100K.

LOAD (XLDSW)

When this input line to the controller is made active (ground) and the start lines are properly manipulated, the machine will go into the load mode causing the KLOAD- signal in the I/O bus to go active. If this function is activated while the machine is running, the results are undefined. This line is paralleled with the load switch on the panel and will go

to ground upon activation of that switch. If the panel is not included, this line should be returned to +6V with a resistor of less than 100K.

START LINES (XSRT-, XXSRT+, XXGND-)

When these input lines to the controller are properly manipulated and the XLDSW- line is ground or the load switch on the panel is active, the controller will go into the load mode. If the XXSS- line is +6V and the run switch on the panel is in the run position, the machine will start. If neither condition (load or run) is true, the controller will single execute an instruction and halt. If both are true, the result is undefined.

To operate these lines, the XXGND- line is permanently connected to ground. This informs the controller logic of the presence of the external control. To initiate one of the above actions, first XXSRT+ is made passive (+6V) and XSRT- is grounded. Then XSRT- is made passive (+6V) and XXSRT+ is grounded. The action is initiated when XXSRT+ is at ground.

The unit is designed to be driven from a form C momentary switch (SPDT, non-bridging) with the armature grounded, XSRT- connected to the normally open contact and XXSRT+ connected to the normally closed contact. The unit is designed to accept switch bounce where the contact opens and closes a number of times but not if the armature bounces from contact to contact.

If the machine control interface is used and the full control panel is not included, line XSRT- should be returned to +6V with a resistor of 100K or less.

RUN INDICATOR (TRUN-X)

This output line from the controller will be active (ground) when the machine is running. The line will be passive (+6V) when the machine is stopped or halted. Figure 5-1 shows the design of a typical remote control panel.

Table 5-1 shows the pin assignments, signals, and their function for the machine control interface.

Table 5-1.
Machine Control Interface Signals

Pin	Mnemonics	Function
1-18	---	Not Used
19	XCLR-	Master Clear (input)
20	TRUN-X	Run Indicator (output)
21	XXSS-	Stop Line (input)
22	XLDSW-	Load Line (input)
23	XSRT-	Start Line (input)
24	XXSRT+	Start Line Reset (input)
25	XXGND-	Ground to Start Logic (input)
26-32	---	Not Used

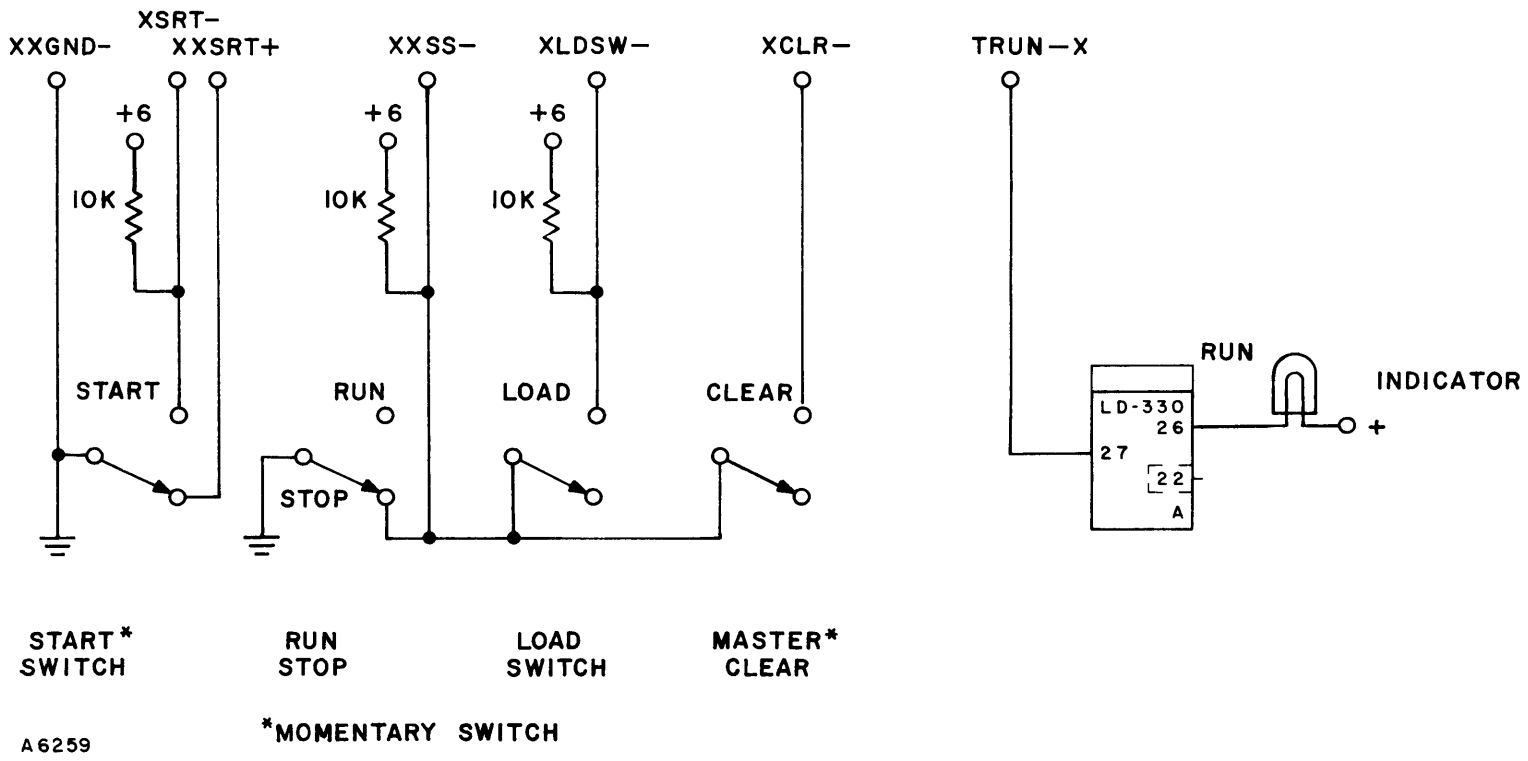


Figure 5-1. Typical Remote Control Panel Wiring

SECTION VI
INTERFACE UNIT CONSTRUCTION

This section covers the nonelectronic aspects of interface unit construction. The μ-PACs used in interfacing are described in the appendix as an aid in design. The following topics are presented:

a. Circuits
b. Packaging
c. Power supplies
d. Coding
e. Cables
f. Configuration
g. Rack requirements

Honeywell supplies a complete line of logic modules and accessory hardware for constructing interface units. The unit should be housed in the mainframe drawer or in an extension drawer (Model H112-10). This will facilitate cabling and provide proper cooling.

In large interface units, however, it is sometimes preferable to divide the interface hardware between the controller and external device. The interface circuitry associated with the controller I/O bus can be contained in the drawer with the mainframe. A cable arrangement is then constructed to the interface circuitry located in the external device which eliminates extending the I/O bus to the remote unit or trying to fit all the circuitry into the H112 drawer.

CIRCUITS

A wide variety of standard μ-PACs are available for implementing the design of interface units. These PACs are compatible mechanically and thermally with those used in the mainframe.

The CS-517, CS-520, CS-521, and CS-548 μ-PACs are suggested for interfacing the programmed I/O bus and DDC bus. Descriptions of these PACs are contained in the appendix.

PACKAGING

Each 1 x 3 OMNI-BLOC contains 24 slots for receiving μ-PACs. Multiple units can be assembled to provide additional slots for more complex functions. The OMNI-BLOC has jumper wires for power bussing, PAC guides, and miscellaneous mounting hardware. The 1 x 3 part number is 70 026 175 701.

Insulated wire (No. 30) especially made for solderless-wrapped connections is used for signal wiring on the OMNI-BLOC. This wire is available from Gore Associates, Newark, Delaware or from CCD, No. 70 940 061 110. The hand wire-wrapping tool (battery operated, CCD No. 70 917 200 001; Gardner-Denver No. 14R2, bit 504221, sleeve 500350) is recommended for wire installation. Also the hand wire-unwrapping tool (No. CCD 70 917 202 001; Gardner-Denver 505084-LH) is recommended for removing solderless-wrapped connections.

A standard OMNI-BLOC may be used in place of deflector μ-BLOC (filler plate) in the mainframe (subject to power limitations).

If the mainframe drawer is filled, an expansion drawer may be purchased. The expantion drawer, Model H112-10, is the same physical size as the controller drawer except the panel is blank. The unit contains fans for cooling, a line cord and barrier strip for power distribution. The drawer has room for the equivalent of twelve 1 x 3 OMNI-BLOCs. If space remains after installing the power supply and OMNI-BLOCs with μ-PACs, deflector OMNI-BLOCs (filler plate) No. 70 014 368 701 are recommended to control air flow. The remainder of the drawer is filled with deflectors.

POWER SUPPLY

The H112 power supply, Model PB-421, located in the mainframe drawer with the capacity to drive some interface units, produces a +6, -6, and +15-volt outputs. The +15-volt output, however, is not available for customer use because it supplies power to the memory.

The overall regulation of the PB-421 (±6V) is ±5% over line, load, temperature, and time. To compute the amount of power available for customer use, subtract the mainframe and option requirements from the capability of the supply. Table 6-1 shows the current computation at a given voltage.

Table 6-1.
Current Computation

	Voltage	
	+6V	-6V
Capability of PB-421	8.0A	0.75A
Requirements of Mainframe	4.0A	0.2A
Expansion Memory	0.7A	0.1A
Direct Data Channel	0.3A	
Power Failure Interrupt	0.2A	
I/O Typewriter	0.6A	
Paper Tape Reader	0.4A	
Power Restart	0.3A	0.05A

In the expansion drawer, either the PB-333 or PB-720 power supply may be used. Each is the size of two 1 x 3 OMNI-BLOCs.

SPECIFICATIONS

PB-333 with accessories: CCD No. 70 008 339 702

Input Voltage

117 Vac ±10%, 47 to 62 Hz

Output Voltage

+6V, 10A (max)

-6V, 1A (max)

Regulation

±2.5% over line, load and temperature

PB-720 with accessories: CCD No. 70 026 046 701

Input Voltage

117 Vac ±10%, 47 to 62 Hz

Output Voltage

+6V, 5A (max)

+24V, 1.75A (max)

-24V, 1.75A (max)

Regulation

±5% on +6V, ±2% on ±24V

The accessories supplied with the power supply include: a line cord with spade lugs for terminating on the barrier strip in the drawer; an insulator for isolating the power supply from the chassis and miscellaneous mounting hardware.

Quick Disconnect connectors (Thomas & Betts, Elizabeth, N.J. No. RA257, CCD 70 937 200 001) on the power supply and OMNI-BLOCs distribute low voltage power. Sheet metal bussing included with the OMNI-BLOCs further distributes the power to individual PACs. The mounting rails in the drawer, the frame on the OMNI-BLOCs, and the sheet metal bussing to the individual PACs distributes logic ground to the system. Jumpers with the quick disconnect connector also provide a redundant ground path.

The logic ground is isolated from the outside sheet metal of the drawer. The drawer is connected to safety common in the power cord. Some power supplies, however, have safety common connected to their frame, requiring the power supplies to be isolated from the rails on which they mount. Insulators and insulated screws are used in this case between the power supply and mounting rails.

The line cord, which plugs into a standard three prong outlet, distributes ac power to the mainframe drawer. If the mainframe drawer has a panel, the on-off switch controls power to the power supply, fans, and the outlet on the drawer. The on-off switch is a 15 amp circuit breaker. When the drawer is not equipped with a panel, the system supplying power should provide the on-off function and overload protection. In either case, the expansion drawer, equipped with a line cord, can plug into the outlet on the mainframe drawer.

CODING (Location Information)

The location of a connector pin or PAC slot in the drawer is identified by the system coding diagram, Figure 6-1. The drawer is defined as one zone. The mounting rack for the drawer is defined as the unit. Figure 6-1 also shows the breakdown within the zone. Additional equipment and components in the drawer can be described by defining more columns and rows. By convention, a 1 x 3 OMNI-BLOC containing interface circuitry (locatable in any row, front or back) is coded with the corresponding ABC and row 1 location. Also, by convention all undefined parameters (unit and zone) are designated X. For example, slot XXA11 defines the first slot in row 1, column A, in a controller which is not part of a larger system.

CABLES

Interface units are normally connected together and to the mainframe with μ-PAC to μ-PAC cables. All standard OMNI-BLOCs such as the high-speed paper tape reader have the programmed I/O bus entering in slot 1 of column A with wires from every pin of that location connected to every pin of slot 8 of column A. Thus, a system can be assembled from standard OMNI-BLOCs by connecting slot 8 of the first option to slot 1 of the next with a μ-PAC to μ-PAC ribbon jumper. The ribbon jumper cable is CCD No. 70 014 998 701.

A μ-PAC to μ-PAC cable is used to pass signals from the I/O bus terminal of the mainframe (slot C18) to the first option OMNI-BLOC and between OMNI-BLOCs which have to be connected together but which are not adjacent. The cable can be used between the mainframe drawer and the expansion drawer. The length of cable required for this purpose is 9-1/2 feet. Below are listed several lengths available and the corresponding part numbers.

Cable Length	CCD Part Number
1 foot	70 013 826 705
1-1/2 feet	70 013 826 709
2 feet	70 013 826 701
3 feet	70 013 826 706
5 feet	70 013 826 703
9-1/2 feet	70 013 826 716

The μ-PAC for terminating a cable is available by CCD No. 70 024 600 703. This number supplies the μ-PAC with accessory pieces but with no cable. The μ-PAC is useful for attaching a cable to drive a device which is remote from the controller. The μ-PAC to μ-PAC cable may also be used.

CONFIGURATION

The controller will be shipped from the factory with the mainframe and factory-installed options located as shown in Figure 6-2.

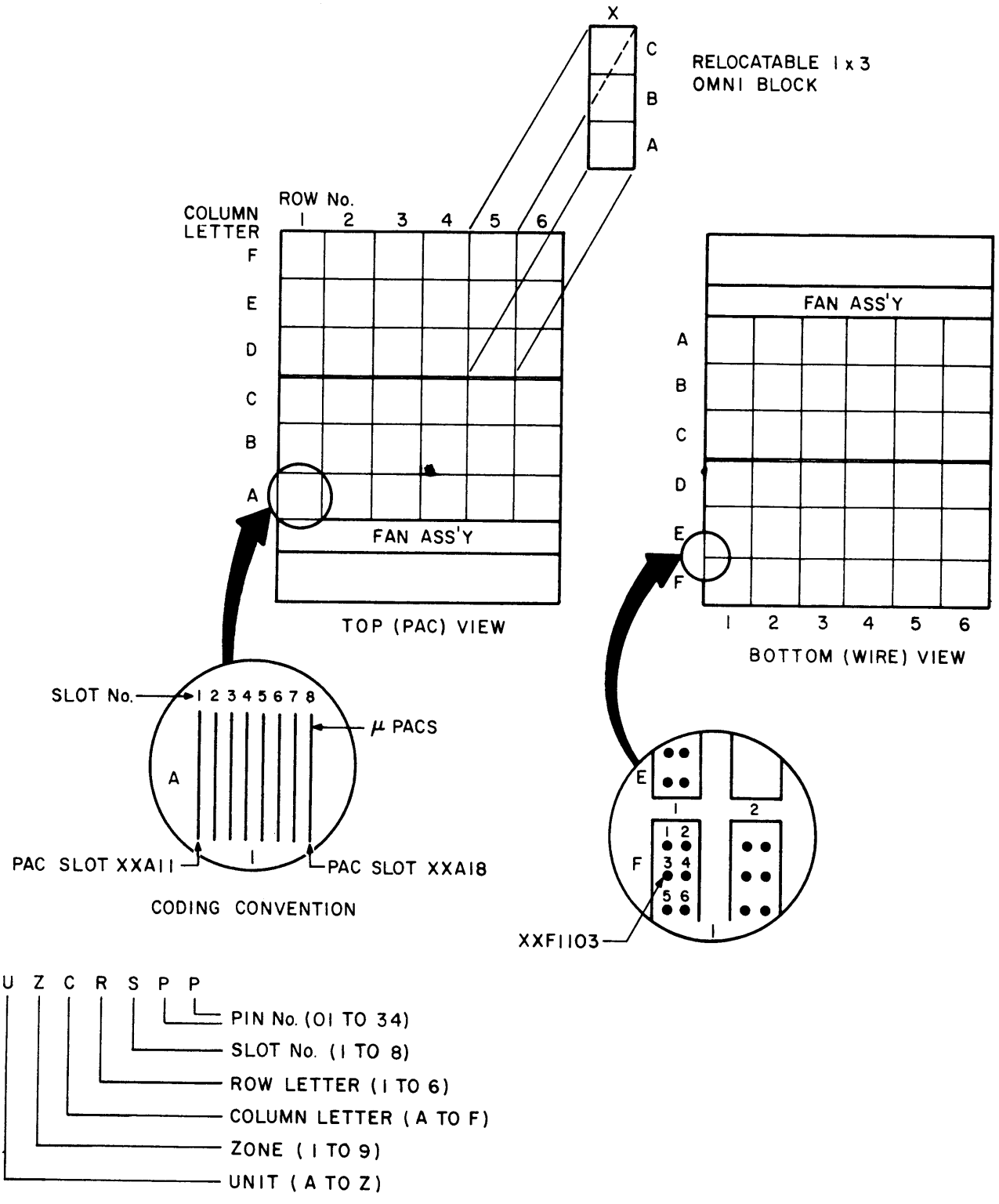


Figure 6-1. System Coding Diagram

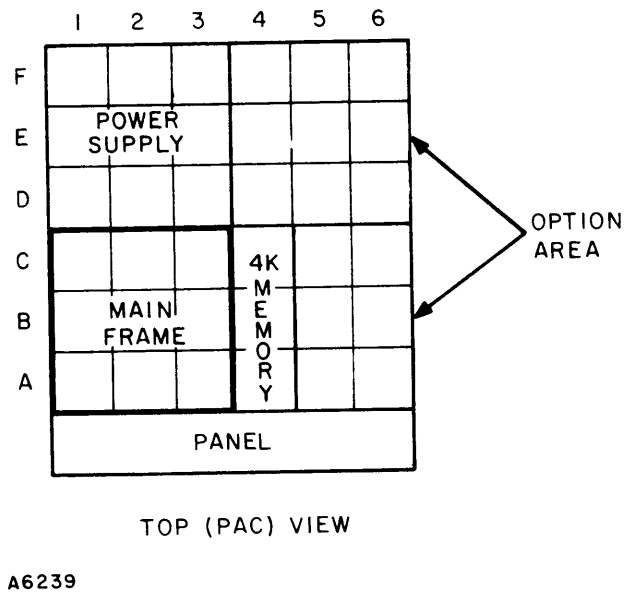


Figure 6-2. Controller Configuration

Options are added with the following priority:

a. Expansion Memory is always in ABC5

b. Direct Data Channel 1 x 3 is always in DEF4

c. First Factory Installed Option
d. Second Factory Installed Option

} These options are installed in the locations starting at the top of the list and using as many as necessary.

DEF4
DEF5
DEF6
ABC6
ABC5

RACK DIMENSIONS

The 19 in. wide x 24 in. deep x 7 in. high controller drawer requires a 19-in. rack. The actual dimensions of the rack are 24 in. wide x 29 in. deep. The unit will fit in the cabinet system listed as follows:

Piece	Number	Weight
Rack (56" panel height)	41F10	100 lbs
Right side panel	F0RK	25 lbs
Left side panel	F0LK	25 lbs
Rear door-louvered	F1AP	25 lbs
Top panel	10TH	8 lbs
Bottom panel	10PH	15 lbs
Frame base	10CA	20 lbs

The above items are not available from Computer Control Division. They are part of a comprehensive cabinet system available from:

Honeywell, Inc.
Apparatus Control Div.
2701 4th Ave. South
Minneapolis, Minnesota 55408

or

Honeywell, Inc.
Bond Street
Wabash, Indiana 46992

SECTION VII
INSTALLATION

This section contains information for the installation of the H112 controller and control panel.

CONTROLLER DIMENSIONS

Mounting

The controller drawer is 19 in. wide x 24 in. deep x 7 in. high. It mounts in a standard equipment rack, with a 19-inch panel width, and 29-inch depth. Actual dimensions of the 19-inch rack are 24 in. wide x 29 in. deep. See the installation drawings in Appendix C.

Weight

75 pounds (max)

Environment

Room ambient (less I/O device)

Temperature (operating)

0°C to 50°C (32°F to 122°F).

Humidity

0% to 90% RH

Power

Voltage

117 Vac ±10%

Frequency

47 to 63 Hz (exclusive of peripheral devices)

Consumption

200W (max)

NOTE

The line cord, which plugs into a standard three-prong outlet, distributes ac power to the mainframe drawer. If the mainframe drawer has a panel, the ON-OFF switch (15-amp circuit breaker) controls power to the power supply fans and the outlet on the drawer. When the drawer is not equipped with a panel, the system supplying the power should provide the ON-OFF function and overload protection. In either case, the expansion drawer (equipped with a line cord) can plug into the outlet on the mainframe drawer.

PORTABLE CONTROL PANEL (optional)

The portable panel, Model H112-19, provides a removable control panel for maintenance or program debug of a controller without a panel. The portable panel physically snaps onto the blank panel on the front of the mainframe drawer. The unit consists of a panel and one DK-320 μ-PAC which plugs into the 3 x 3 controller mainframe at B27. Cables from the panel connect by plugging into the appropriate panel connectors located in the H112 controller.

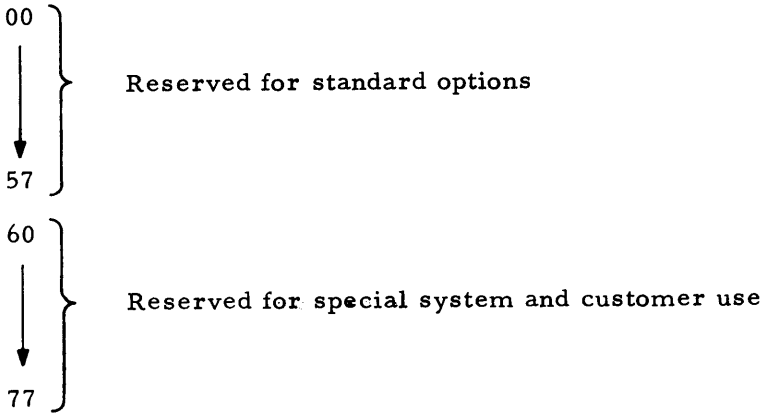
APPENDIX A
SPECIAL μ-PACS

The following special u-PAC descriptions are contained in this appendix.

Model	Description
CS-517	Driver and Receiver PAC
CS-520	Receiver PAC
CS-521	Driver PAC
CS-548	Line Receiver Assembly PAC

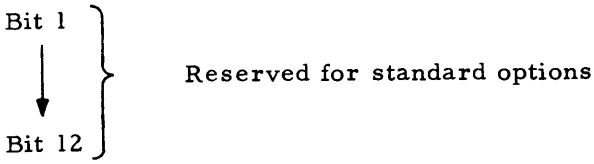
APPENDIX B
DEVICE ADDRESS AND SMK BITS ASSIGNMENT

Device Addresses Assignments

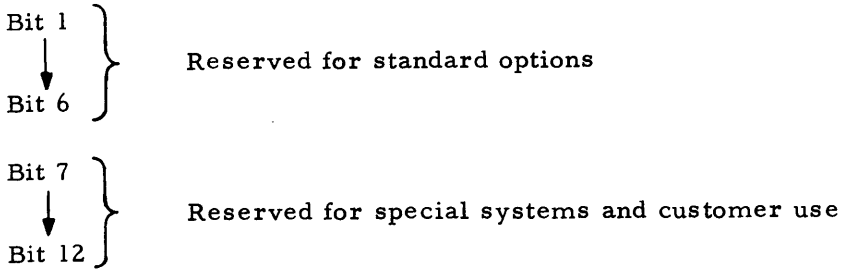


Set Mask Bit Assignment

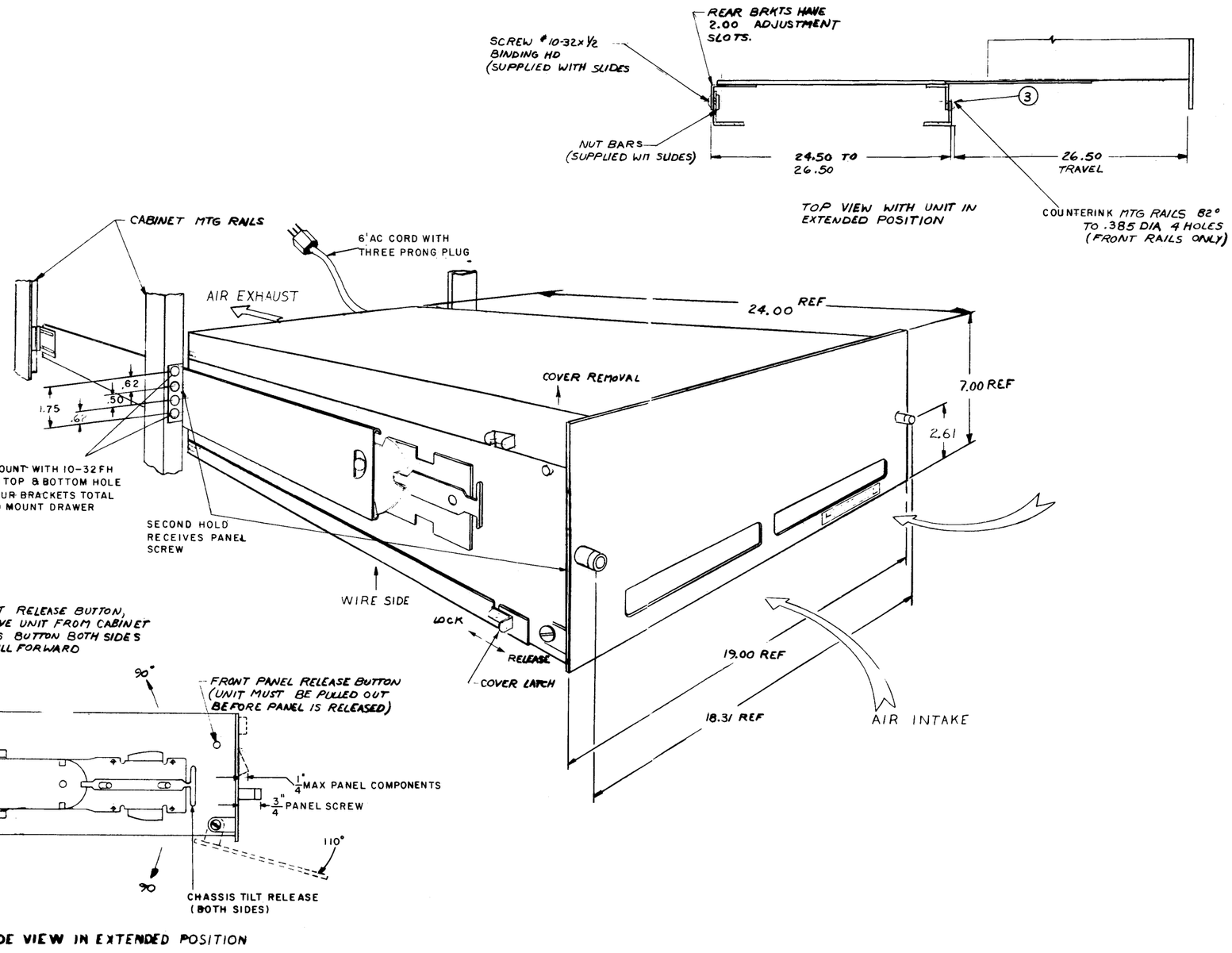
SMK0



SMK1



APPENDIX C
H112 INSTALLATION AND ACCESSORIES
DRAWING



DRIVER AND RECEIVER PAC, MODEL CS-517

The Driver and Receiver PAC, Model CS-517 (Figure CS-517-1), contains 6 line drivers and 6 line receivers for driving and receiving signals on the I/O bus. Driver rise and fall times are capacitor controlled. The 6 gated receiver outputs receive signals from the I/O bus. All circuits have built-in terminators for each signal. Pins 9 and 23 provide points for strobe signals.

SPECIFICATIONS

Receiver Section

Input Loading

1 unit load on pins 4, 6, 14, 16, 26, and 28.

Output Drive Capacity

7 unit loads on pins 5, 1, 15, 11, 25, and 19.

8 unit loads on pins 7, 3, 17, 13, 27, and 21.

Circuit Delay (1.5V average over 2 stages)

75 ns

Current Requirements

+6V: 100 mA (max)

Driver Section

Input Loading

1 unit load on pins 10, 8, 22, 18, 29, and 30.

Output Drive Capacity

20 loads on pins 4, 6, 14, 16, 26, and 28.

Circuit Delay (1.5V average over 2 stages)

150 ns (max)

Rise and Fall Time

Rise time: 50 to 250 ns at 0 to +6V.

Fall time: 50 to 100 ns at +6V to 0.

Current Requirements

+6V: 100 mA (max)

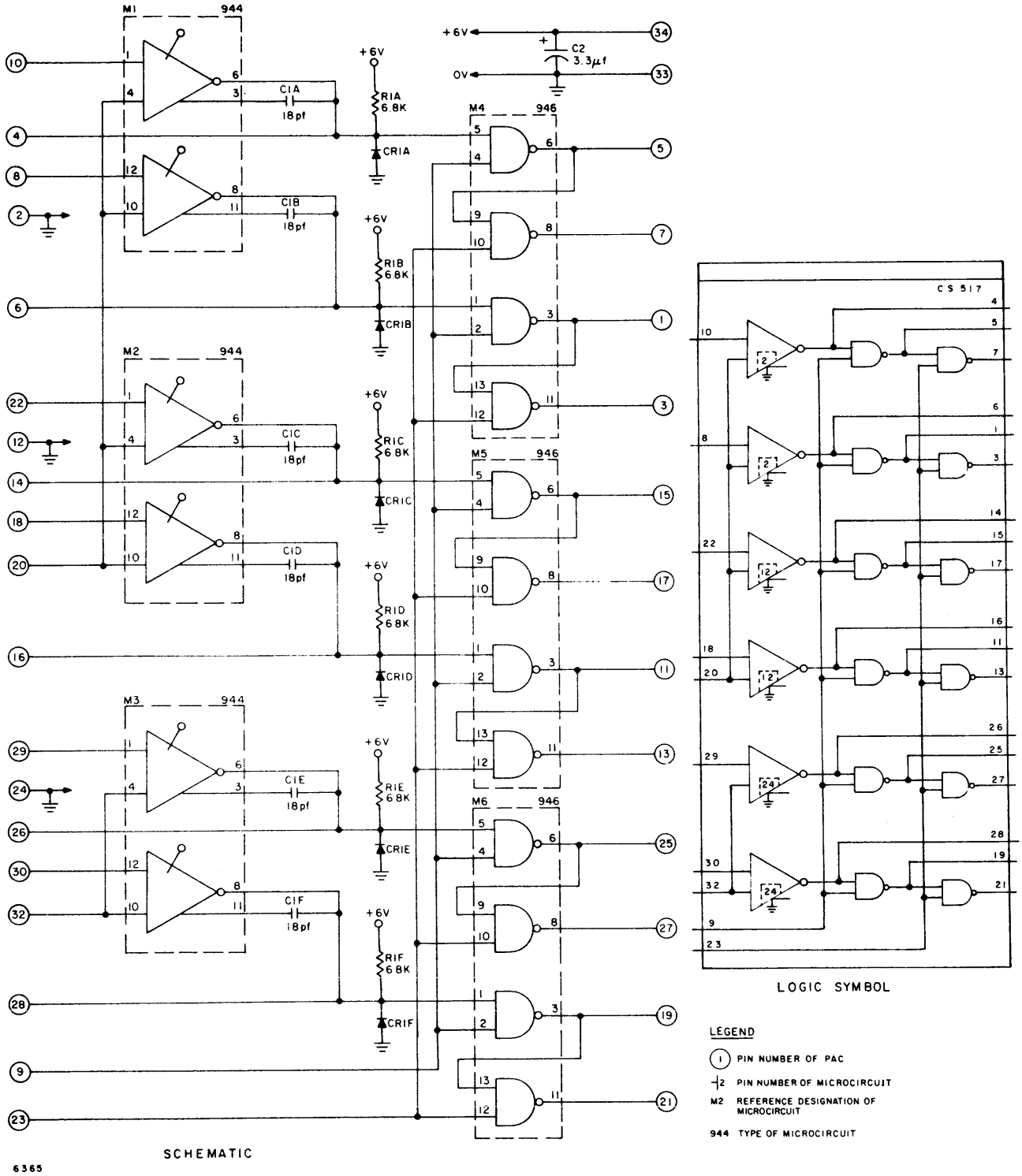


Figure CS-517-1. Driver and Receiver PAC, Schematic Diagram

Electrical Parts List

Ref. Desig.	Description	CCD Part No.
C1A-C1F	CAPACITOR, FIXED, CERAMIC, DIELECTRIC: 18 pF ±10%, 100 Vdc	70 930 173 206
C2	CAPACITOR, FIXED, ELECTROLYTIC:	70 930 230 011
CR1A-CR1F	DIODE, SI	70 943 083 001
R1A-R1F	RESISTOR, FIXED, FILM: 6.8K ±2%, 1/4W	70 932 114 069
M1, M2, M3	MICROCIRCUIT: 944 Type, Intergrated Circuit	70 950 105 008
M4, M5, M6	MICROCIRCUIT: 946 Type, Quad NAND Gate, Integrated Circuit	70 950 105 002

RECEIVER PAC, MODEL CS-520

The Receiver PAC, Model CS-520 (Figure CS-520-1), is a line receiver with 6 gated outputs used to receive signals from the I/O bus on pins 4, 6, 14, 16, 26 and 28. The circuit has a built-in terminator for each signal.

Pins 9 and 23 provide points for strobe signals.

SPECIFICATIONS

Input Loading

1 unit load on each input pin.

Output Drive Capacity

7 unit loads on pins 5, 1, 15, 11, 25, and 19.
8 unit loads on pins 7, 3, 17, 13, 27, and 21.

Circuit Delay

75 ns (at 1.5V average over two stages)

Current Requirements

+6V: 75 mA (max)

Electrical Parts List

Ref. Desig.	Description	CCD Part No.
C1	CAPACITOR, FIXED, ELECTROLYTIC:	70 930 230 011
CR1A-CR1F	DIODE, SI	70 943 083 001
R1A-R1F	RESISTOR, FIXED, FILM: 6.8K ±2%, 1/4W	70 932 114 069
M4, M5, M6	MICROCIRCUIT: 946 Type, quad NAND gate, integrated circuit	70 950 105 002

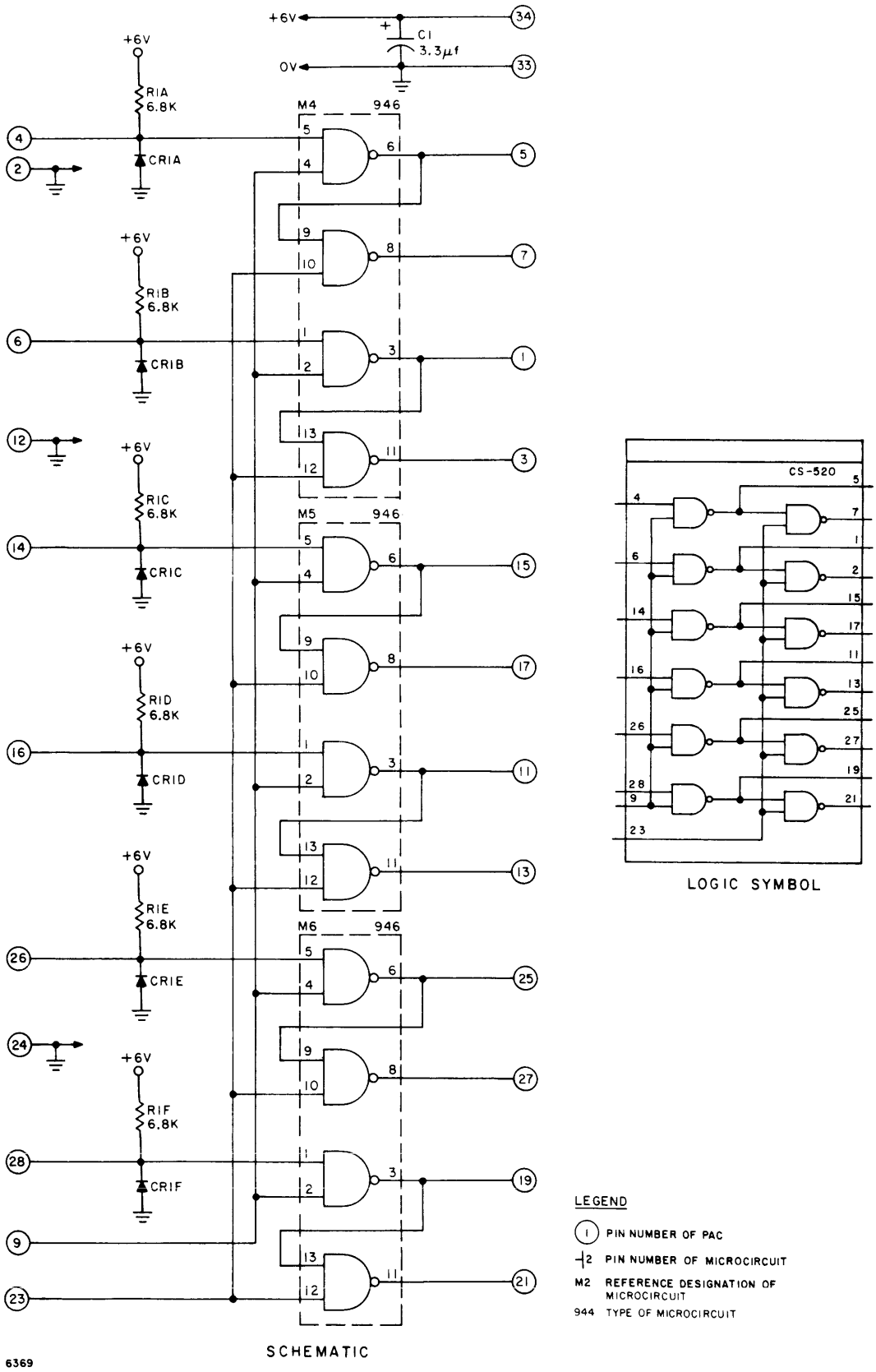


Figure CS-520-1. Receiver PAC, Schematic Diagram

DRIVER PAC, MODEL CS-521

The Driver PAC, Model CS-521 (Figure CS-521-1) contains 6 line drivers used to drive the I/O bus. Rise and fall times are capacitor controlled and each signal has a built-in terminator.

SPECIFICATIONS

Input Loading

1 unit load on pins 10, 8, 22, 18, 29, and 30.

Output Drive Capacity

20 loads on pins 4, 6, 14, 16, 26 and 28.

Circuit Delay

150 ns (max)

Rise and Fall Times

Rise time: 50 to 250 ns at 0 to +6V.
Fall time: 50 to 100 ns at +6V to 0.

Current Requirements

+6V: 36 mA (max)

Electrical Parts List

Ref. Desig.	Description	CCD Part No.
C1A-C1F	CAPACITOR, FIXED, CERAMIC, DIELECTRIC: 18 pF ±10%, 100 Vdc	70 930 173 206
C2	CAPACITOR, FIXED, ELECTROLYTIC:	70 930 230 011
CR1A-CR1F	DIODE, SI	70 943 083 001
R1A-R1F	RESISTOR, FIXED, FILM: 6.8K ±2%, 1/4W	70 932 114 069
M1, M2, M3	MICROCIRCUIT: 944 type, integrated circuit	70 950 105 008

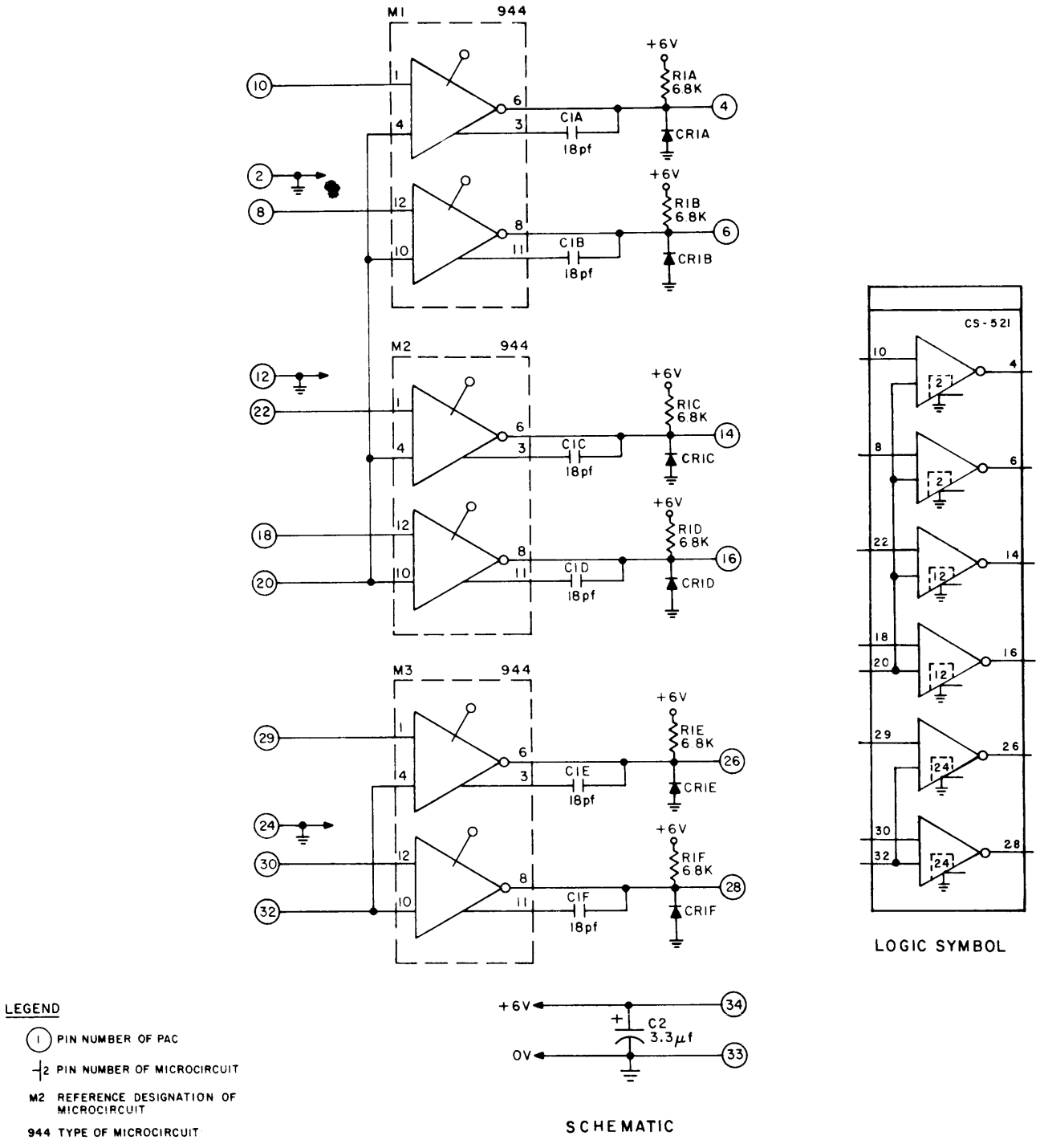


Figure CS-521-1. Driver PAC, Schematic Diagram

LINE RECEIVER ASSEMBLY PAC, MODEL CS-548

The Line Receiver Assembly PAC, Model CS-548 (Figure CS-548-1), provides 14 gated outputs used to receive signals from an I/O bus.

CIRCUIT FUNCTION

Signals are received an an I/O bus on pins 2, 6, 12, 16, 20, 26, 30, 31, 17, 21, 25, 11, 3 and 7. The circuit has a built-in terminator for each signal. Pins 8, 22, 27 and 13 provide points for gating off-strobe signals.

SPECIFICATIONS

Input Loading

1 unit load on each input pin

Output Drive Capacity

8 units loads on pins 4, 10, 14, 18, 24, 28, 32, 29, 15, 19, 23, 9, 1 and 5

Circuit Delay (at 1.5V average over two stages)

75 ns

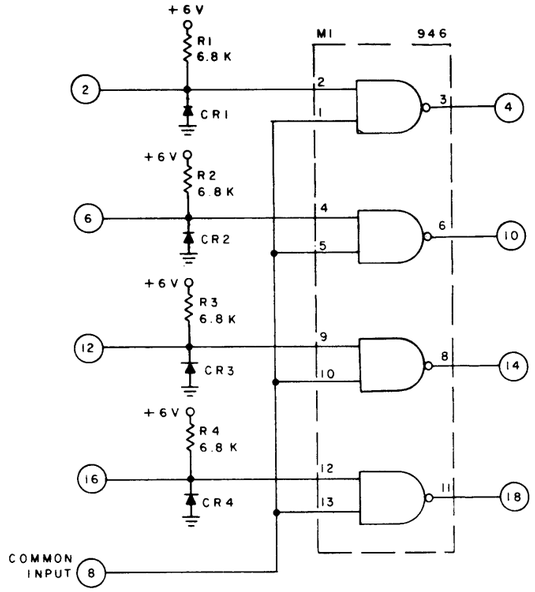
Current Requirements

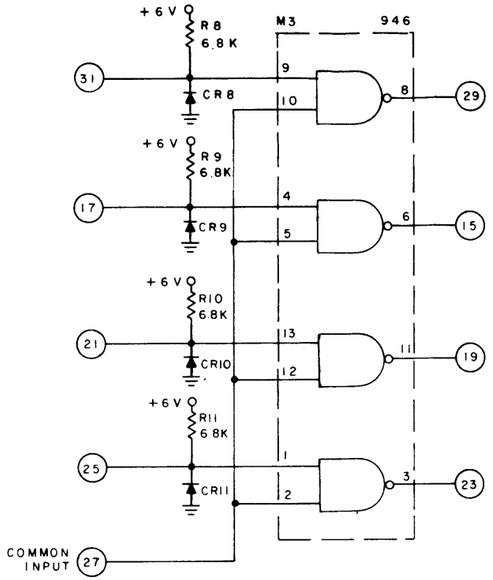
+6V: 75 mA (max)

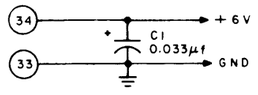
Electrical Parts List

Ref. Desig.	Description	CCD Part No.
C1	CAPACITOR, FIXED, PLASTIC DIELECTRIC: 0.033 μF ± 20%, 50V	70 930 313 016
R1 thru R14	RESISTOR, FIXED, FILM: 6.8K ± 2%, 1/4W	70 932 114 069
CR1 thru CR14	DIODE, SILICON	70 943 083 002
M1 thru M4	MICROCIRCUIT: 946, quad input NAND gate integrated circuit	70 950 105 002

SCHEMATIC

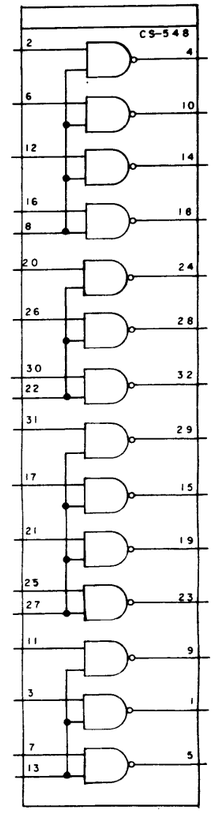






6413

LOGIC SYMBOL



LEGEND

(1) PIN NUMBER OF PAC

⊣2 PIN NUMBER OF MICROCIRCUIT

M3 REFERENCE DESIGNATION OF MICROCIRCUIT

Figure CS-548-1. Line Receiver Assembly PAC, Schematic Diagram and Logic Symbol

Honeywell

COMPUTER CONTROL DIVISION, FRAMINGHAM, MASS. 01701

Printed in U.S.A.