P850

CPU SERVICE MANUAL

PRELIMINARY VERSION

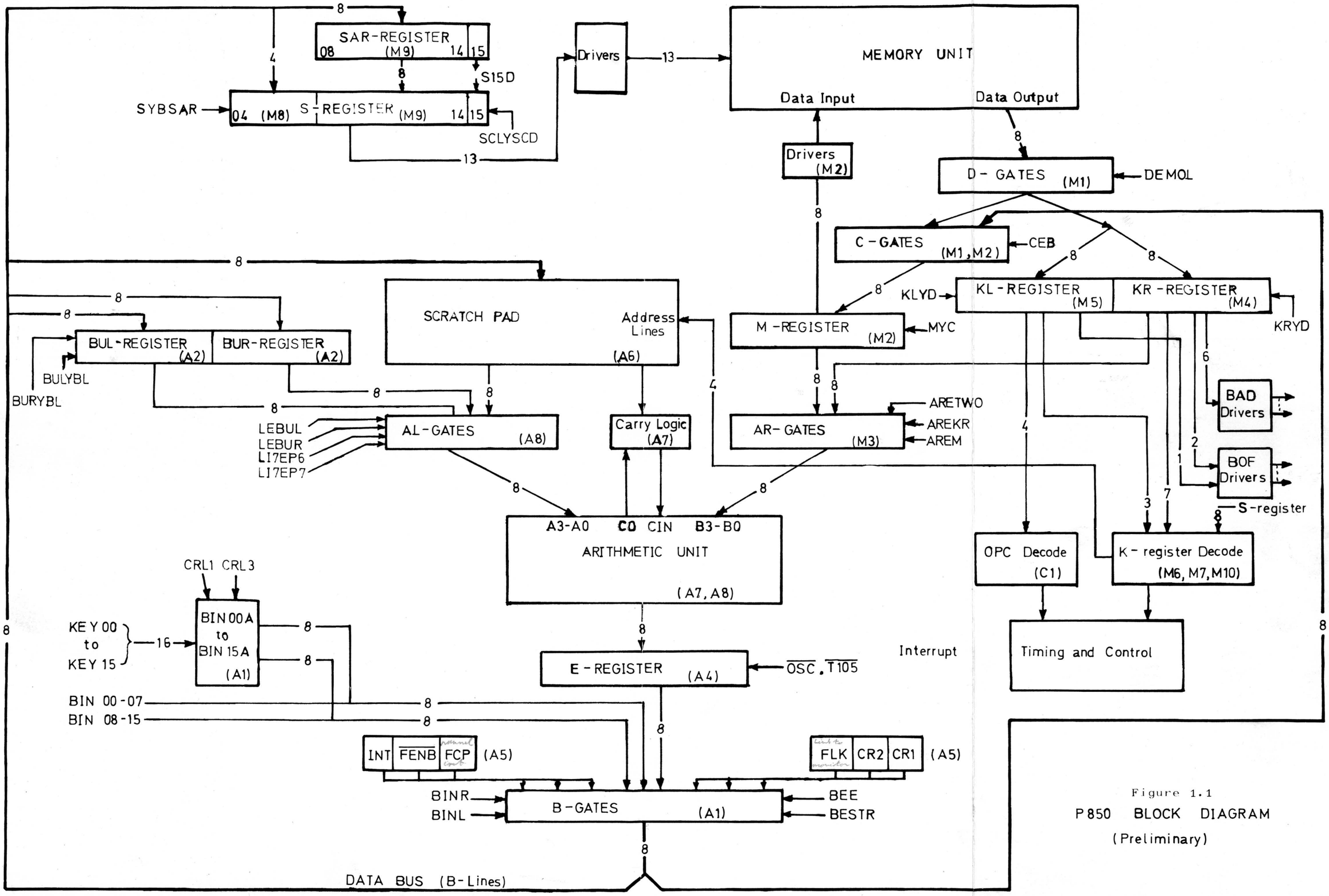


Figure 1.1
P850 BLOCK DIAGRAM
(Preliminary)

1-3/1-4

List of Effective Pages for:

P850 CPU SERVICE MANUAL PRELIMINARY

Publication number 5122 991 16881

Front pages	(i) to (iii)	original
	(iv)	REV1
	(v) to (x)	original
Section I		
pages	1-1 to 1-60	original
Section II		
pages	2-1 to 2-3	original
	2-4, 2-5	REV1
	2-6 to 2-11	original
	2-12, 2-13	REV1
	2-14 to 2-25	original
	2-26 to 2-28a	REV1
	2-29 to 2-45	original
	2-46 to 2-48a	REV1
	2-49 to 2-56	original
	2-57, 2-58	REV1
	2-59, 2-60	original
	2-61, 2-61a	REV1
	2-62, 2-63	original
Section III		
pages	3-1 to 3-15	original
Section IV		
pages	4-1 to 4-12	original

Section V

Pages 5-1 to 5-13 original

Section VI

Pages 6-1 to 6-14 original

Section VII

Pages 7-1 to 7-16 original

10 JANUARY, 1972

PRELIMINARY P850 CPU SERVICE MANUAL

The information contained in this book is based on the first version of the processor documentation and is accurate at the time of printing, 1st June 1971.

The first ten production prototypes will vary as modifications to the design are incorporated. Minor variations may be noted between these machines and the information in this book.

Additional information will be supplied when it becomes available.

A publication of
NV Philips-Electrologica
Industry Group Small Computers
Apeldoorn, Netherlands

Publication number 5122 991 16881

June 1971

The P850 central processor logic is contained on four cards with an additional card to hold the logic for the options. To make the manual easy to use during routine servicing and fault finding, it has been divided into five parts as follows:

Part 1 contains a general description of the hardware, the timing cycles and the interrupt system.

Part 2 contains logic diagrams and a description of the individual logic circuits for the basic processor.

Part 3 contains the logic diagrams and description for the option card.

Part 4 contains diagrams showing the pin numbers for the individual integrated circuit modules, the position of the modules on the circuit cards and the pin connections of the cards.

Part 5 contains an alphabetical list of signal names and abbreviations together with an explanation of their meaning.

In the pocket at the back of this book there are five full-sized logic diagrams showing the logic contained on each individual card.

The logic diagrams and description for the device controller cards are contained in a separate manual.

The basic cabinet for the P850 central processor contains the power supply unit, up to two memory units and seven positions for logic cards. When an extension cabinet is used, this can contain an additional power supply unit, two additional memory units and seven additional circuit card positions.

Four of the circuit card positions in the basic cabinet are used by the logic cards of the processor, one position is available for the circuit card that contains the hardware option logic card, and the other two positions are for device control unit logic cards. All seven positions in the extension cabinet are for device control unit logic cards.

1.1 LOGIC MODULES

The processor uses MSI/TTL integrated circuit modules to perform all arithmetic, logical, shift, load/store and addressing operations. Most of the logic is made up from the Fairchild 74 series modules. The remaining hardware units are made up of the following modules:

(1)	Arithmetic unit	comprises two 4-bit modules type 9341 wired to operate as an 8-bit unit.
(2)	Scratch pad registers	comprises four 64-bit random access memory modules (16 words - 4 bits per word), type 4103, wired to operate as thirty-two 8-bit registers.
(3)	Instruction decoder	comprises two one-out-of-ten decoder modules, type 9301.

1.2 BASIC CONSTRUCTION

It will be seen from the above and from the block diagram Figure 1.1, that the processor is basically an 8-bit orientated machine. As the instruction set is 16-bit word orientated, each instruction is performed a half-word (8-bits) at a time, the Least Significant Bits (LSBs) being performed first. This means that some of the functional cycles are used twice to operate on one instruction word.

1.3 DESCRIPTION OF MAIN HARDWARE UNITS

The following description applies to the units shown in the block diagram, Figure 1.1 . These descriptions are amplified later in the book in the section that gives the logic diagrams. At the back of the book are larger diagrams that show the logic contained on each individual card.

1.3.1 MEMORY UNIT

The basic memory module is a hybrid type that uses TTL integrated circuits. It contains a timing unit, X and Y decoding and selection circuits, inhibit drivers, sense amplifiers, an input/output data register, and the ferrite core matrix. The capacity of the module is 1K 8-bit words, the cycle time is 1.6 µs, and it has been wired to operate in READ/RESTORE and CLEAR/WRITE modes.

There are 13 address lines, 8 data input lines and 8 data output lines, and up to four modules can be connected in parallel (without additional circuitry) to give a total capacity of 4K 8-bit words.

Due to the instruction set using 16-bit words, the memory unit is organized so that each instruction word and each data word is stored in two consecutive locations. This in effect makes the unit 16-bit orientated.

1.3.2 ARITHMETIC UNIT

This unit is made up of two 4-bit arithmetic/logic modules that form an 8-bit unit with forward-looking carry facilities. All arithmetic, logical and left-shift operations are performed within the unit; any carry from an operation is included automatically in the next operation of the arithmetic unit.

There are two 8-bit operand paths into the unit, one from the AL gates and the other from the AR gates. These operands are then processed according to the type of instruction being performed. The results from the unit are input to the E-register via an 8-bit data path.

1.3.3 SCRATCH PAD

The scratch pad registers comprise four random access memory modules, each module containing sixteen 4-bit words. These modules are hard-wired to form thirty-two 8-bit registers and are always used in pairs in order to contain 16-bit operands. This in effect gives the programmer sixteen 16-bit registers.

Registers 0 and 1 are used as a program counter called the P-register, and registers 30 and 31 are used by the interrupt system as a stack pointer (also called register 15 by the programmer).

The address lines to the appropriate register are set up via four lines from the K-register decoder. Input and output data paths are both 8-bits wide.

1.3.4 SAR-REGISTER

This register is used as a buffer to hold the LSB of the memory address during updating of the S-register. The input/output path to this register is 8-bits wide.

1.3.5 S-REGISTER

This register is used to hold the current memory address prior to activating the drivers to the address lines. Bits 08 to 15 are transferred to the register from the SAR-register; bits 04 to 07 come directly from the internal data bus.

The memory address will be contained in bits 06 to 15, and bits 04 and 05 will select the appropriate memory module.

Between the thirteen output lines from the S-register and the address line pins of the memory unit are thirteen drivers which are used to activate the memory address lines.

1.3.6 K-REGISTER

This register is used to hold all sixteen bits of the instruction word. The register is split up into two 8-bit registers, KL (bits 00 to 07) and KR (bits 08 to 15). Input to the register is via the memory output lines and the D-gates.

Output from the register depend upon the instruction format. In format 0 instructions, bits 1 to 4 are output to the operation decode logic that determines the type of function to be performed by the arithmetic unit; bits 5 to 7 hold the register address that contains the first operand - these are output to the K-register decode logic; bits 8 to 15 contain a constant which is output to the arithmetic unit via the AR-gates.

Format 1 instructions can be contained in either one or two words. When two words are used, the second word contains either a memory address, constant or data but none of these is held in the K-register. The first word contains the actual instruction and bits 1 to 4 are output to the operation decode logic; bits 5 to 8 can hold either the register address that contains the first operand or, for absolute branch instructions, represents a condition - these bits are output to the K-register decode logic; bits 9 and 10 represent the addressing mode; bits 11 to 14 can hold either a register address or zero and bit 15 is the load/store bit; all these bits are output to the K-register decode logic.

When input/output instructions are used, bits 10 to 15 hold the device address and are output to the BAD drivers, and bits 8 and 9 hold the function code which is output to the BØF drivers.

1.3.7 M-REGISTER

This register is 8-bits wide and is used as a buffer register between the memory unit and the arithmetic unit and data bus lines.

Input to the register is via the D- and C-gates (when the input is from memory) or via the C-gates only when the input is from the data bus.

Output from the register is either to the memory drivers (during store operations) or to the AR-gates.

1.3.8 E-REGISTER

This is an 8-bit register that acts as a buffer between the arithmetic unit and the B-gates.

Input is from the arithmetic unit, and the result from this unit's operations is held in the register until the appropriate time for gating on to the data bus via the B-gates.

1.3.9 BU-REGISTER

This register is split up into two 8-bit registers BUL (bits 00 to 07) and BUR (bits 08 to 15). It acts as a buffer between the data bus and either the BØU lines or the arithmetic unit.

1.3.10 AL-GATES

These 8 gates are used to control data input to the arithmetic unit from either the scratch pad or the BU-register.

1.3.11 AR-GATES

These 8 gates are used to control data input to the arithmetic unit from either the M-register, the K-register or the ARETWO line.

1.3.12 B-GATES

These 8 gates control the input of data to the data bus from the E-register, the condition register (CR1 and CR2), the FLK (link-to-monitor) signal, the FENB (valid interrupt) signal, the FCP (control panel interrupt) signal and the INT (interrupt) signal.

1.4 OPERATING CYCLES

Because the data and instruction formats are 16-bit word orientated, it requires two similar cycles (one for each half-word) to perform any operation. Some of the cycles are only used once or for special functions but reference to the following text and the timing diagrams should make each cycle operation clear. The types of cycle used are:

(1)	Fetch cycles	F1 and F2.
(2)	Index cycles	X1 and X2.
(3)	Indirect addressing cycles	I1 and I2.
(4)	Execute cycles	E1, E2 and E0.
(5)	Write cycles	W1 and W2.
(6)	Last cycle	L.
(7)	Interrupt cycle	INT.
(8)	Display cycles	AF1 and AF2.
(9)	Manual read/load cycles	RL1, RL2, RL3 and RL4.

All the above cycles derive their timing either directly or indirectly from the basic timing oscillator. Some of the clock pulses shown on the diagram will be used only during certain functions although they will always occur at the times shown. The following description of the cycles is only general and more detailed explanation can be found on the logic diagrams later in this book.

1.5 FETCH CYCLES

Two cycles are needed to read an instruction out of memory.

1.5.1 F1-CYCLE

During the first half of the cycle the least significant eight bits of the instruction are read out into the KR half of the K-register. The memory address of the memory location will have been gated into the S-register during the last (L) cycle of the previous instruction.

Also during the first half cycle, the LSBs of the P-register (program instruction counter) are updated and loaded back into the LSB of the P-register. Because the programmer uses only the 15 Most Significant Bits (MSBs) of the register, to update this register effectively by 1, it requires the binary equivalent of decimal two to be added to the P-register contents by the arithmetic unit. If a carry occurs during this updating, the carry flip-flop is set and is used during the second half of the cycle.

During the second half of the cycle the MSBs of the P-register are updated and loaded back into the MSB of the P-register.

During T7 time of this and every other cycle, logic is set up ready to enter the next cycle in the sequence. The exact sequence of cycles will depend on the result of the decoded contents of the K-register. In the case of the F1 cycle, the only possible cycle that can follow it is the F2 cycle.

1.5.2 F2-CYCLE

During the first half of this cycle the MSBs of the instruction word are read out of memory into the KL half of the K-register. The K-register decode logic tests the type of instruction addressing mode etc and, if format 0 or register-to-register type of instruction, will set up logic to exit from this cycle at T4 time and enter the E1 cycle.

If the address of the second operand is contained in the word following the instruction, the contents of the P-register are loaded into the S-register. The LSBs are loaded into the SAR-register during the first half of the cycle and, during the second half of the cycle, the MSBs are gated on to the data bus(B-lines) and at T6 time the contents of the SAR-register and the B-lines are gated into the S-register.

During T7 time, logic is set up ready to enter either the indexing, indirect, execute or last cycles.

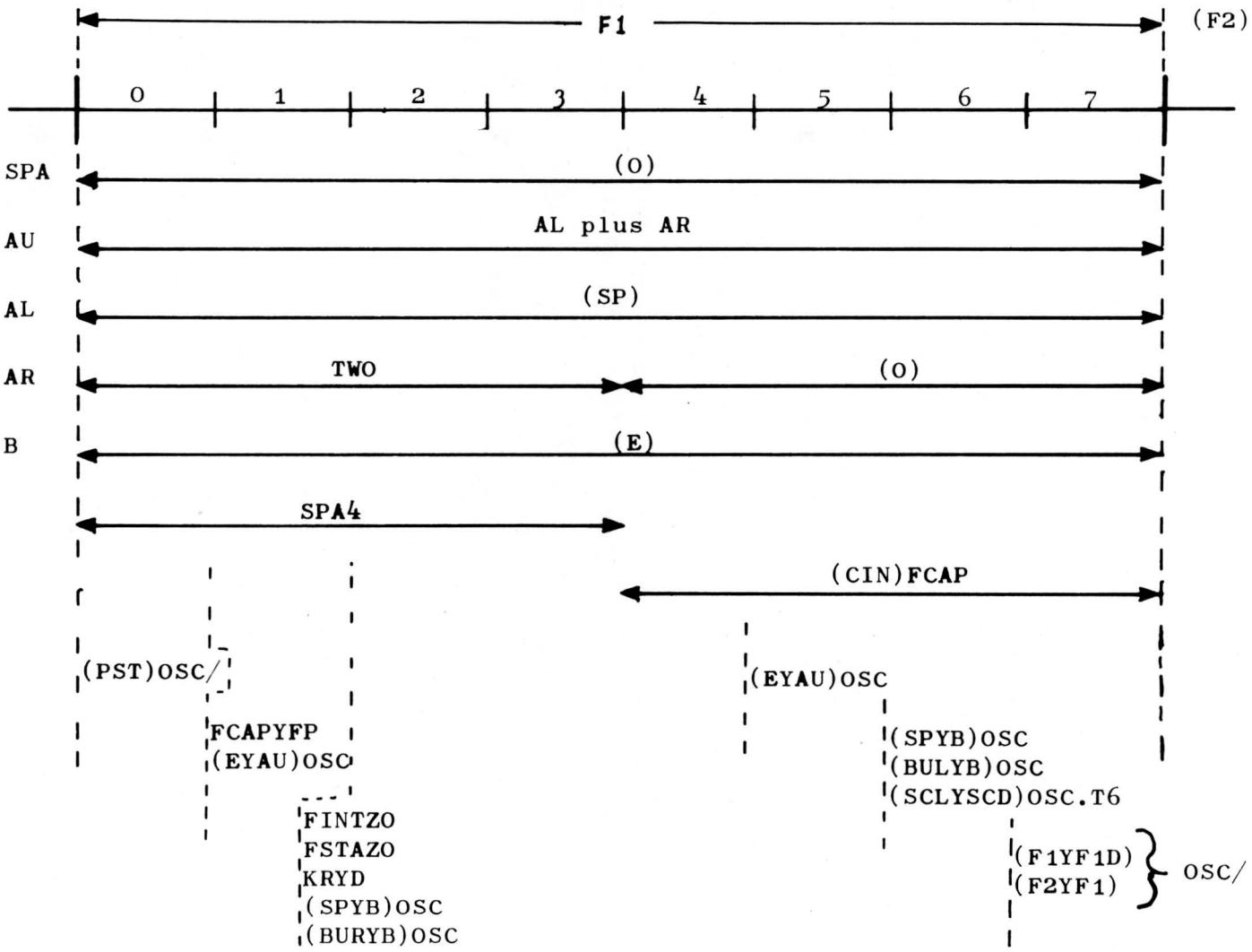


Figure H1 Fetch cycle F1

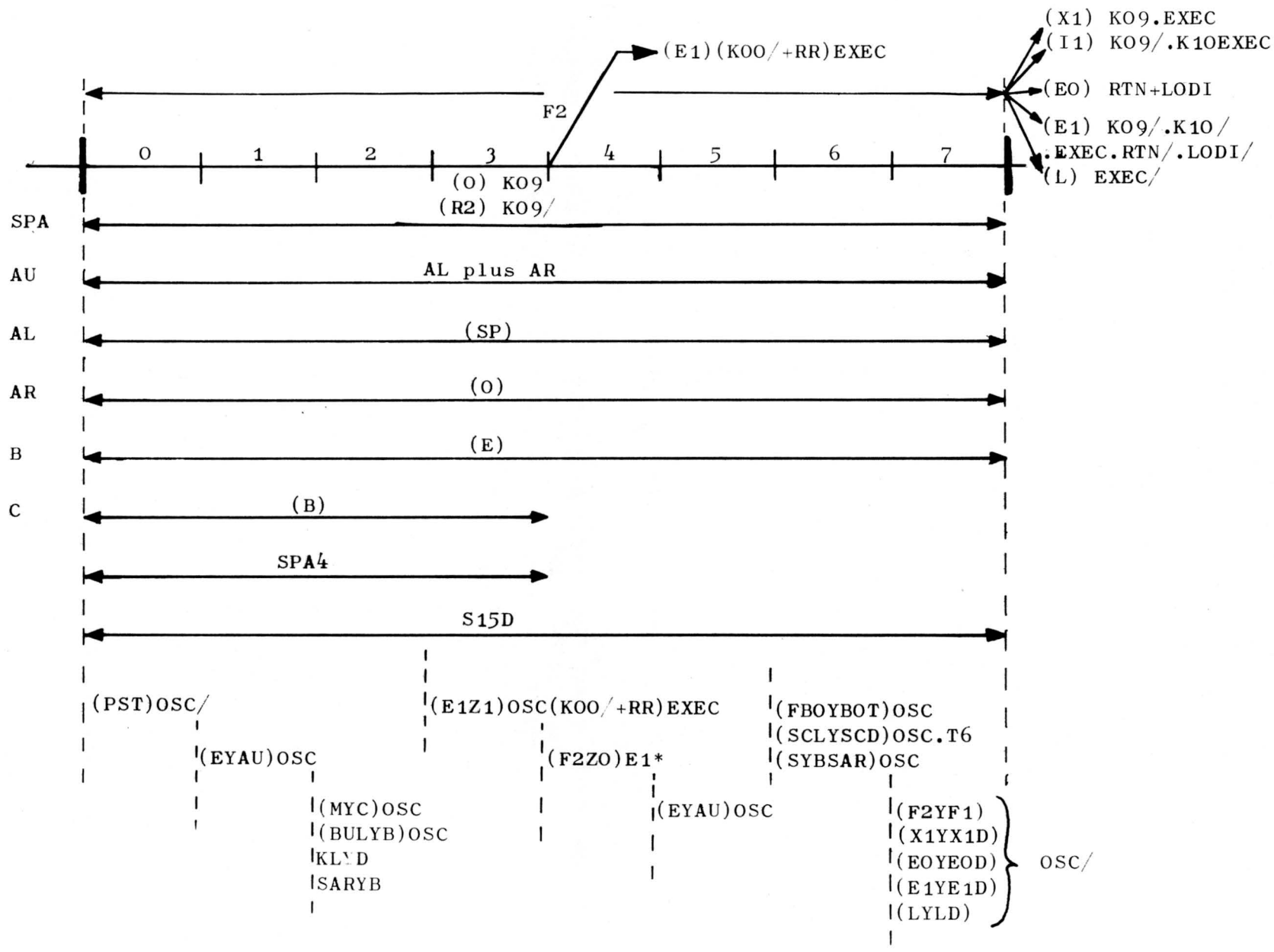


Figure H2 Fetch cycle F2

1.6 INDEX CYCLES

These cycles are used only on memory reference instructions and their use does not increase the execution time of the instruction. Basically the first half of each cycle is used to read out one half of the memory address, and the second half of the cycle is used to merge these address bits with the index bits. The resulting indexed address is then used to update the S-register.

1.6.1 X1-CYCLE

During the first half of the cycle the LSBs of the memory address are read out into the M-register. At the same time the LSBs of the index register are presented to the AL-gates.

At T4 time, the AL- and AR-gates are opened and the contents are merged in the Arithmetic Unit (AU). If a carry occurs, the carry flip-flop is set and will be used during the X2-cycle.

During T5 time, the result of the AU operation is gated to the B-lines via the E-register.

At T6 time, the B-lines contents (indexed LSBs of memory address) are gated into the SAR-register.

During T7 time, logic is set up ready to enter the X2-cycle.

1.6.2 X2-CYCLE

This cycle operates in the same way as the X1-cycle but deals with the MSBs of the address. The sequence and timing is exactly the same except that at T6 time the contents of the B-lines and the SAR-register are gated into the S-register.

During T7 time, logic is set up ready to perform either the indirect addressing cycles or the execute cycle.

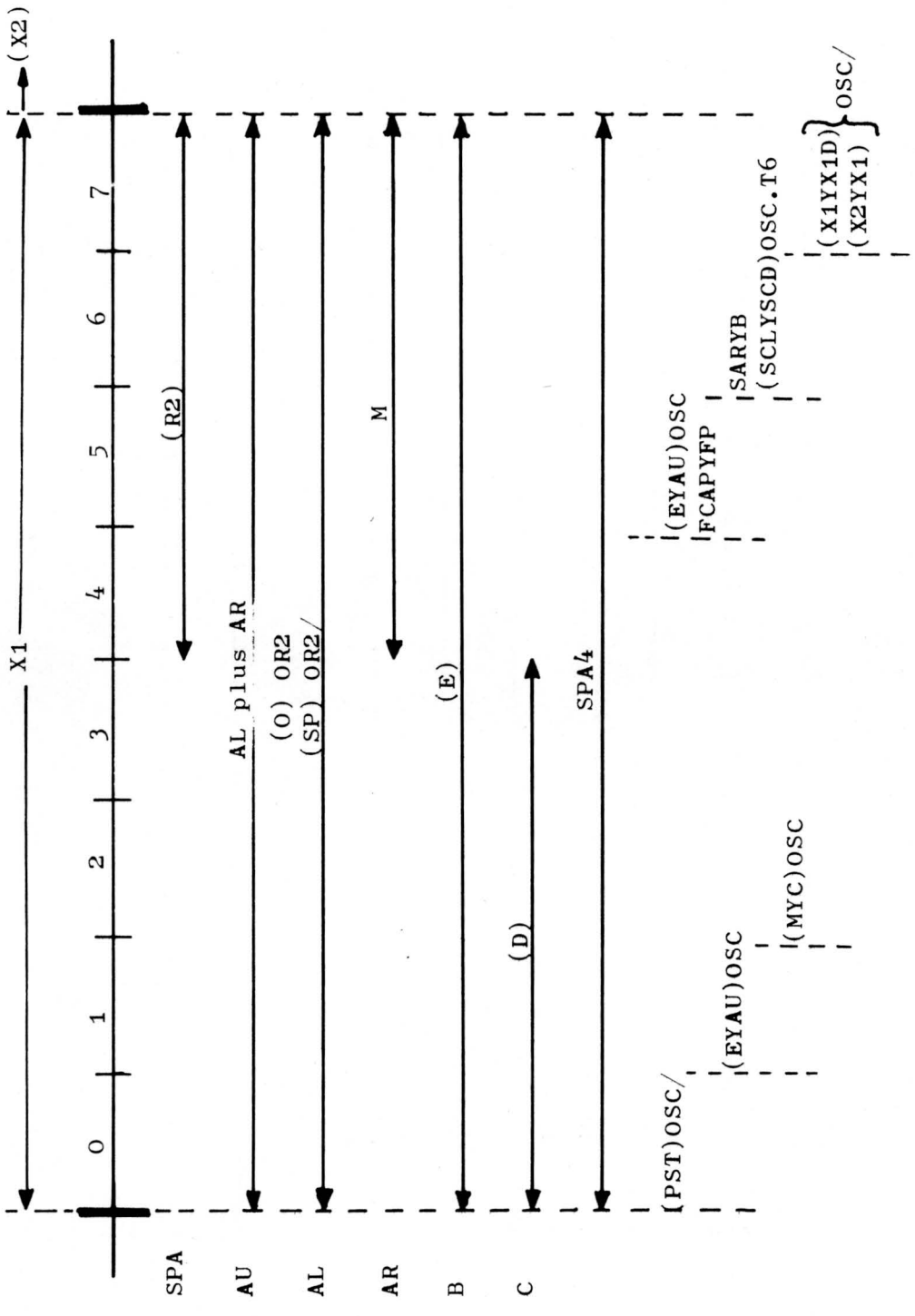


Figure H3 Indexing cycle X1

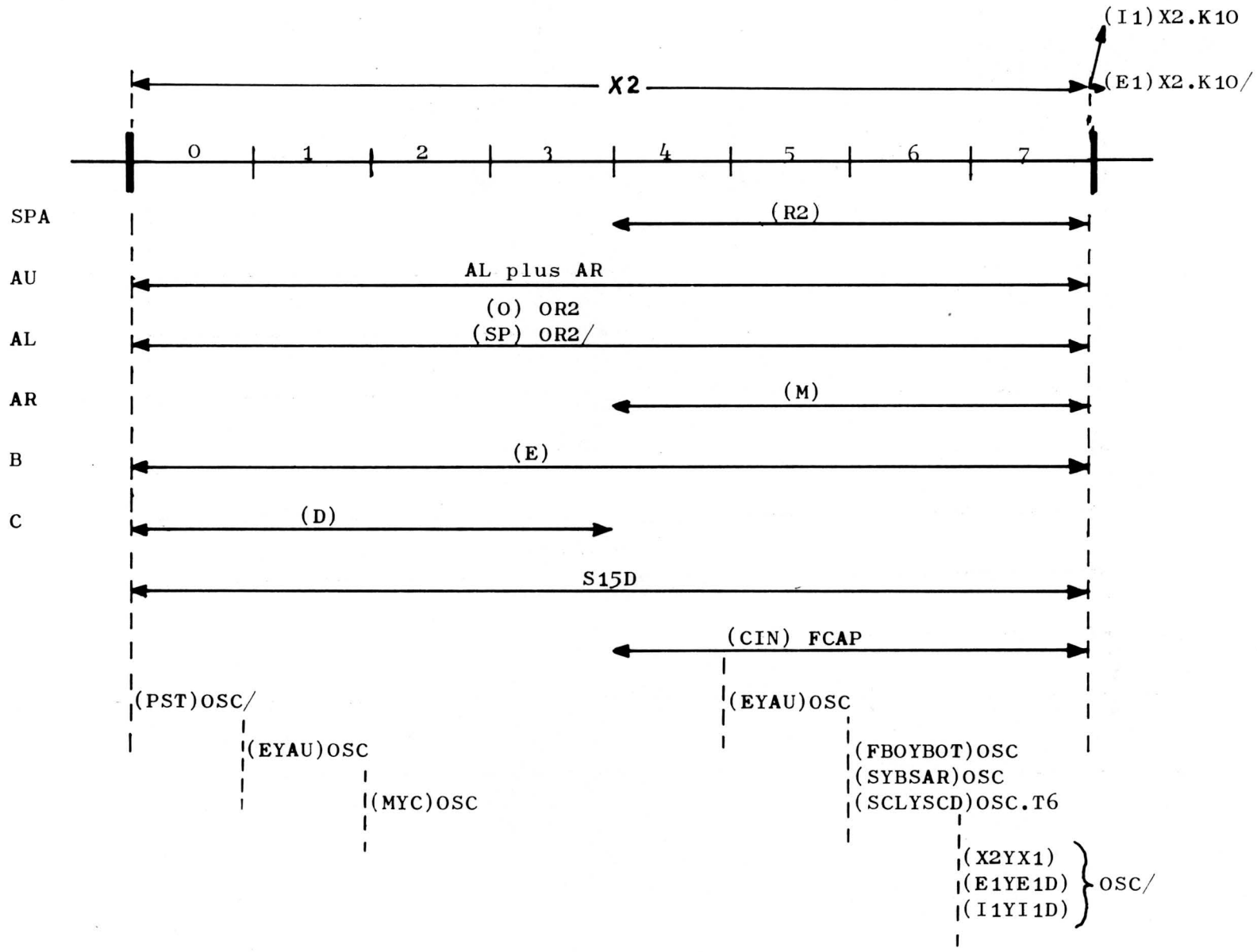


Figure H4 Indexing cycle X2

1.7 INDIRECT ADDRESSING CYCLES

These cycles operate in a similar way to that of the indexing cycles, the only difference being that no index bits are used. It must be remembered that in indirect address instructions, the contents of the second word of the instruction contain the address of a location in memory and that the contents of this memory location are the address of the operand used by the instruction.

1.7.1 I1-CYCLE

During the first half of the cycle, the LSBs of the memory address are read out into the M-register.

During the second half cycle (at T6 time), these LSBs are gated into the SAR-register.

During T7 time, logic is set up ready to enter the I2-cycle.

1.7.2 I2-CYCLE

This cycle operates in the same way as the I1-cycle except that during T6 time the contents of the B-lines and the SAR-register are gated into the S-register.

During T7 time, logic is set up ready to enter the execute cycle.

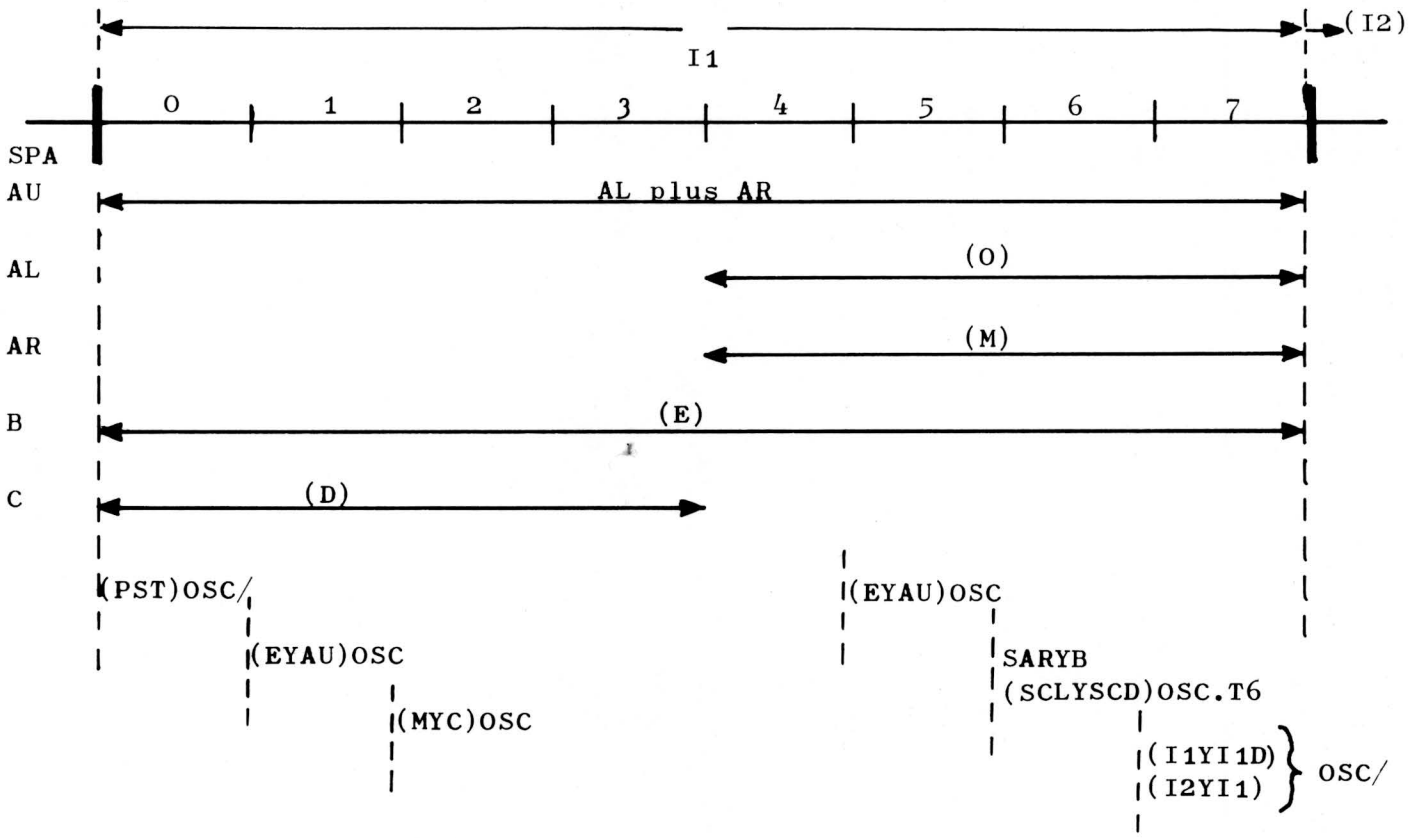


Figure H5 Indirect addressing cycle I1

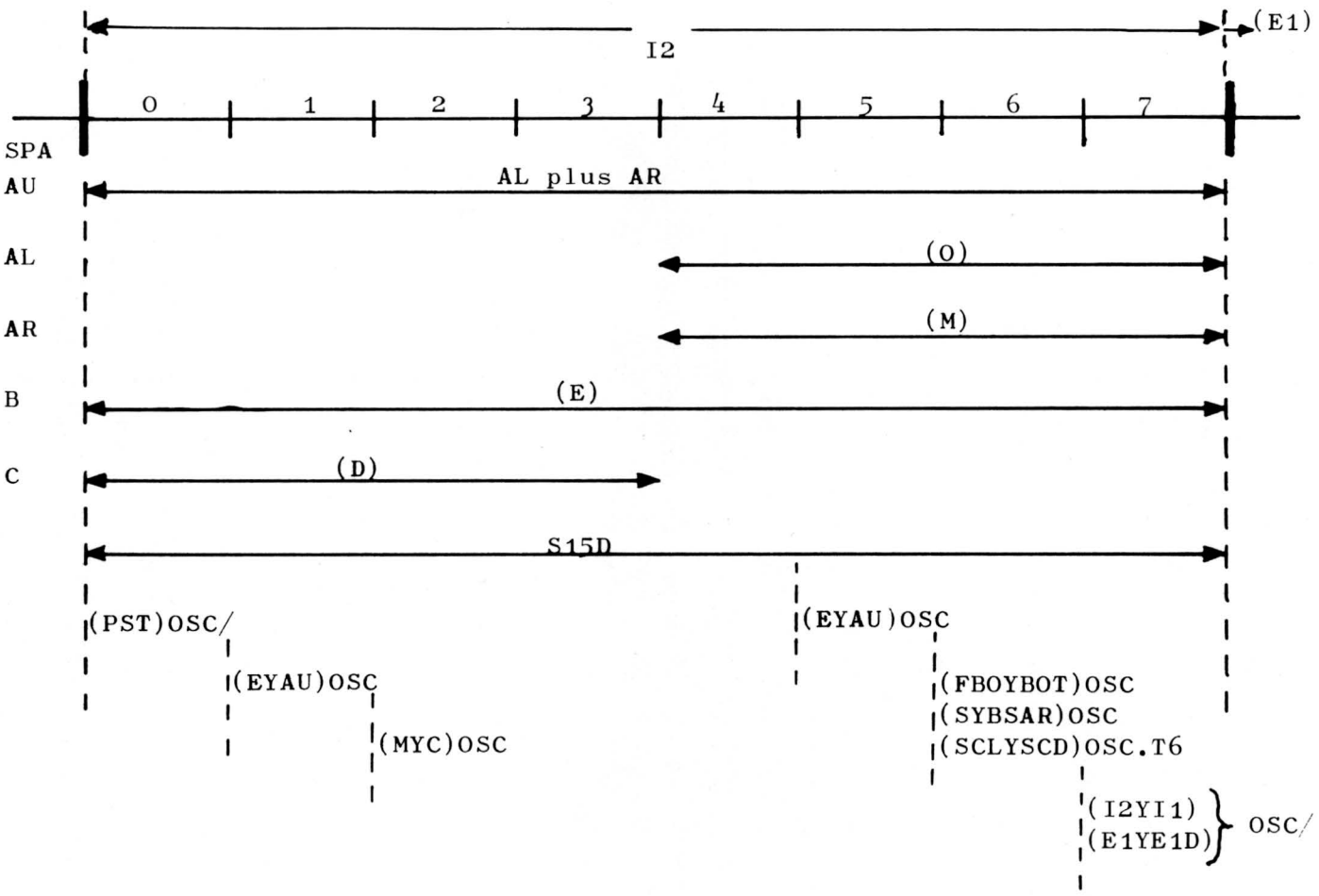


Figure H6 Indirect addressing cycle I2

1.8 EXECUTE CYCLES

There are three execute cycles; the EO-cycle is only used during call, return, load increment and hardware interrupt instructions, and the E1 and E2-cycles are used by all instructions. The first half of each cycle is used for various purposes depending on the type of instruction being executed. During the second half of each cycle the actual execution of the instruction takes place.

1.8.1 AU CONTROL DURING E-CYCLES

The function of the arithmetic unit during the E-cycles is governed by the type of instruction being executed. The following table gives the types of instruction and the mode in which the AU operates.

instruction type	AU mode
LOAD, absolute conditional branch, call function, return function, character instructions	AR
ADD, increment memory, relative branches forward, shift right instructions	AL plus AR
subtract, relative branch back and compare instructions	AL minus AR minus 1
AND	AL.AR
IØR inclusive or	AL + AR
XØR exclusive or	AL ⊕ AR
complement	AR/
shift left instructions	AL plus AL
store, character and I/Ø instructions	AL

1.8.2 E1-CYCLE

This cycle and the E2-cycle are used by every instruction and therefore the number of operations executed by these cycles is quite large. In order to simplify the explanation, only the more common operations are given. Once these have been explained the less common operations can be traced easily. This cycle operates on the LSBs of the operand. During TO time, if the instruction is a memory reference type, the memory address lines will be activated.

If the instruction is contained in two words, the P-register address (in the SP) is selected.

(1) T1 - if the P-register is being updated, the result of the AU operation is gated to the B-lines via the E-register and any carry sets the carry flip-flop.

(2) T2 - if P-register updating, the contents of the B-lines are gated into the LSB of the P-register.

If memory reference instruction, the contents of the selected memory address are gated into the M-register.

(3) T4 - if conditional branch instruction, the P-register address will be maintained on the SP address lines.

If register-to-register instruction, the appropriate SP address lines will be activated.

The AU function will depend on the instruction types (see table above).

(4) T5 - The result of the AU operation can be gated to the B-lines via the E-register.

The carry and NULL flip-flops can be set.

If input data instructions, the DAUZ1 or CRZ3 signals will be active.

(5) T6 - The AU results can be gated (via the B-lines) into either the M-register or an SP-register.

The contents of the B-lines can also be gated into the BUR register.

(6) T7 - Logic is set up ready to perform either an E2-cycle (if register-to-register instructions) or W1-cycle (if memory reference instructions).

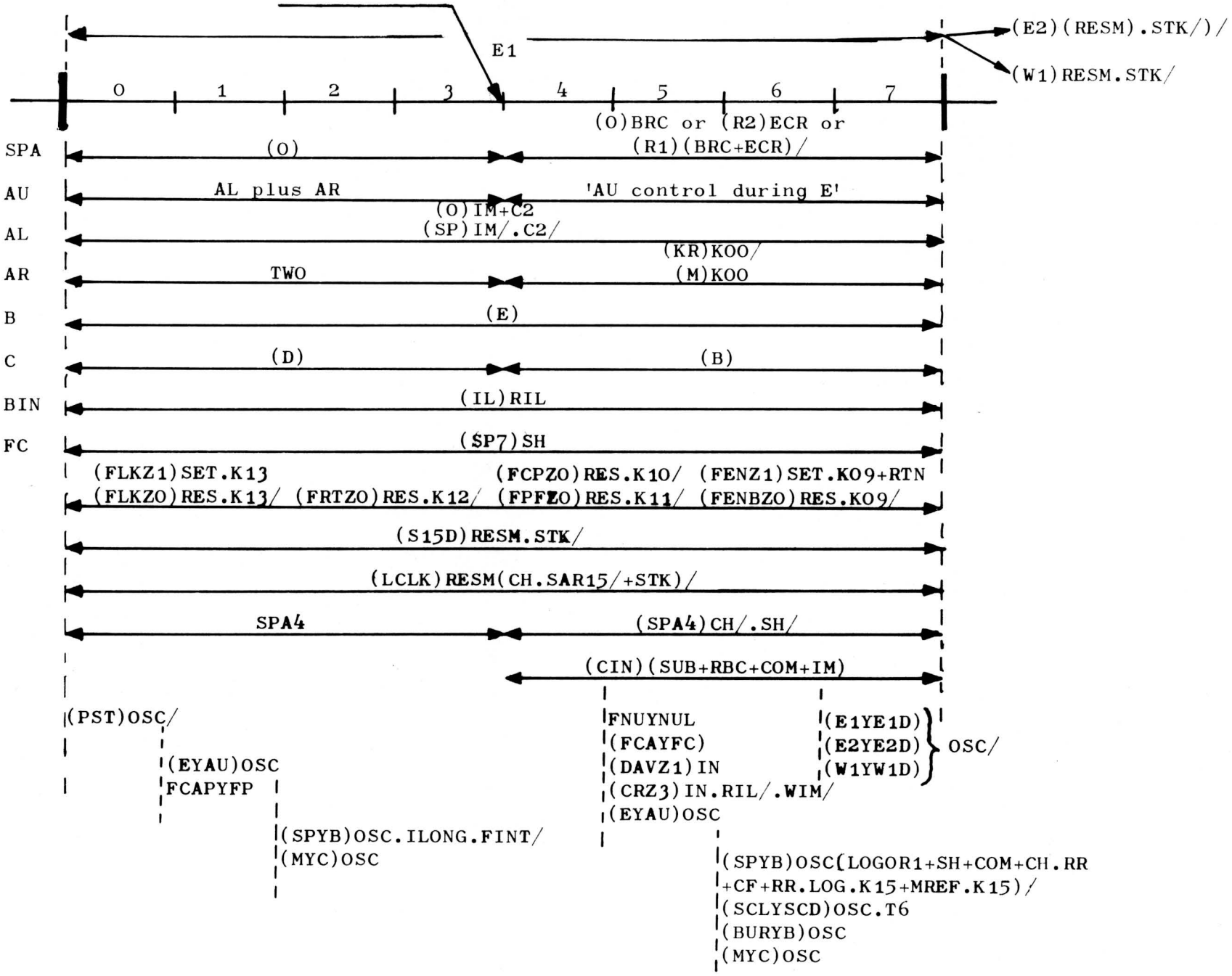


Figure H7 Execute cycle E1

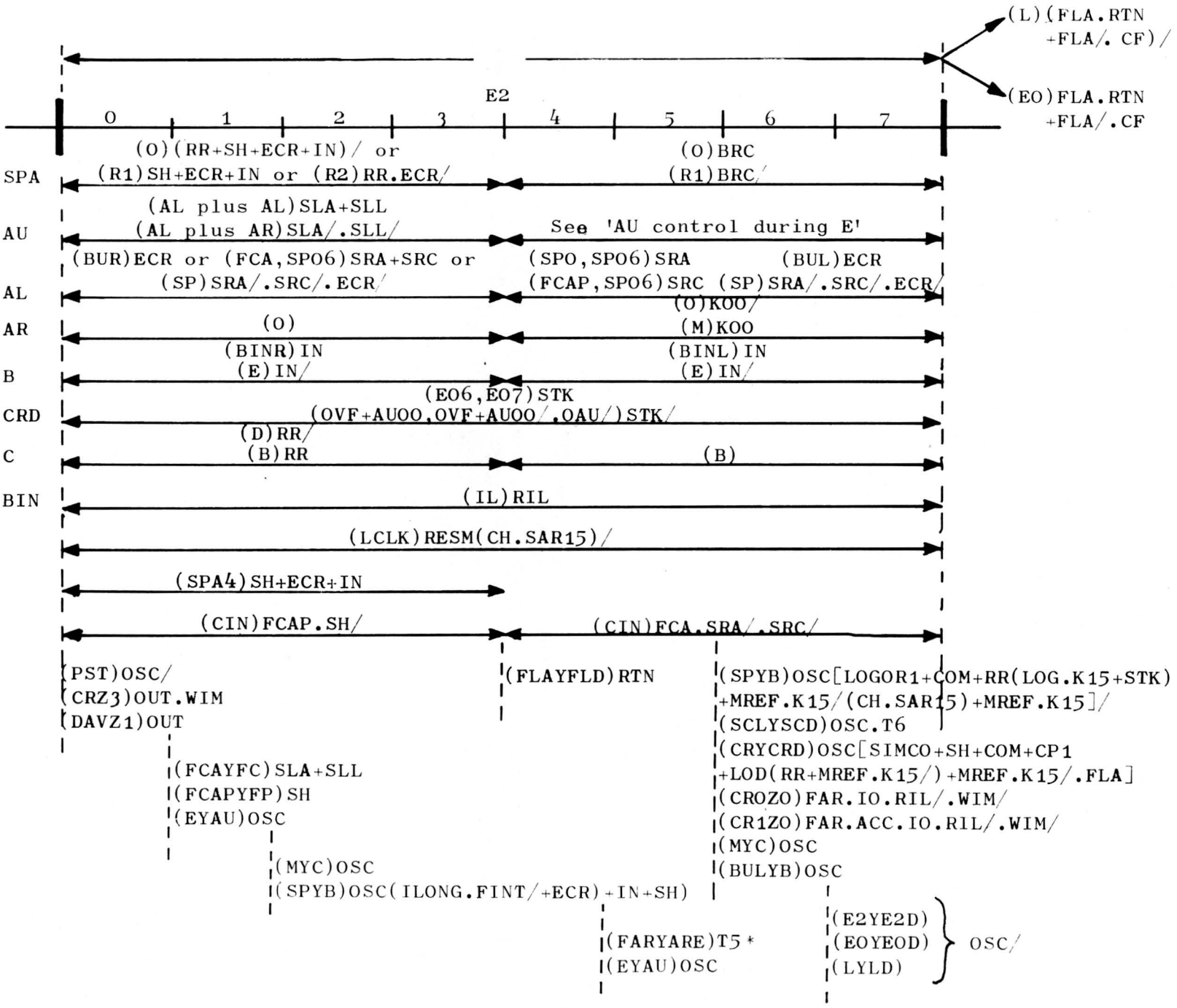


Figure H8 Execute cycle E2

1.8.3 E2-CYCLE

This cycle operates on the MSBs of the operand.

(1) T0 - if memory reference instruction, the appropriate memory address lines will be activated.

The appropriate SP address lines will be activated.

(2) T1 - the results of the AU operation will be gated to the B-lines via the E-register.

(3) T2 - the contents of the B-lines can be gated into either the M-register or an SP-register.

If memory reference instruction, the contents of the selected memory location are gated into the M-register.

If input data instruction, the contents of BINR lines are gated into the appropriate register.

(4) T4 - if return function instruction using this cycle for the first time, the repeat cycle flip-flop will be set.

(5) T5 - the result of the AU operation can be gated to the B-lines via the E-register.

If input data instruction, the contents of BINL can be gated on to the B-lines.

The address recognized flip-flop can be set.

(6) T6 - the contents of the B-lines can be gated into the appropriate SP-register.

The condition register can be set either by an input/output instruction or as a result of performing the instruction.

The contents of the B-lines can also be gated into the BUL-register.

(7) T7 - logic is set up ready to perform either an EO- or L-cycle.

1.8.4 EO-CYCLE

This cycle is used only during call, return, load increment and interrupt instructions, and deals with the incrementing or decrementing of the stack pointer (register 15) and the S-register.

The first half of the cycle either increments or decrements (depending on the type of instruction) the LSBs of the stack pointer (register 15) and gates the result into the SAR-register.

The second half of the cycle deals with the MSBs of the stack pointer and at T6 time gates the contents of the B-lines and SAR-register into the S-register. At the same time there are signals that can update the read/write flip-flop and the bootstrap flip-flop.

During T7 time, logic is set up ready to perform either a W1- or E1-cycle.

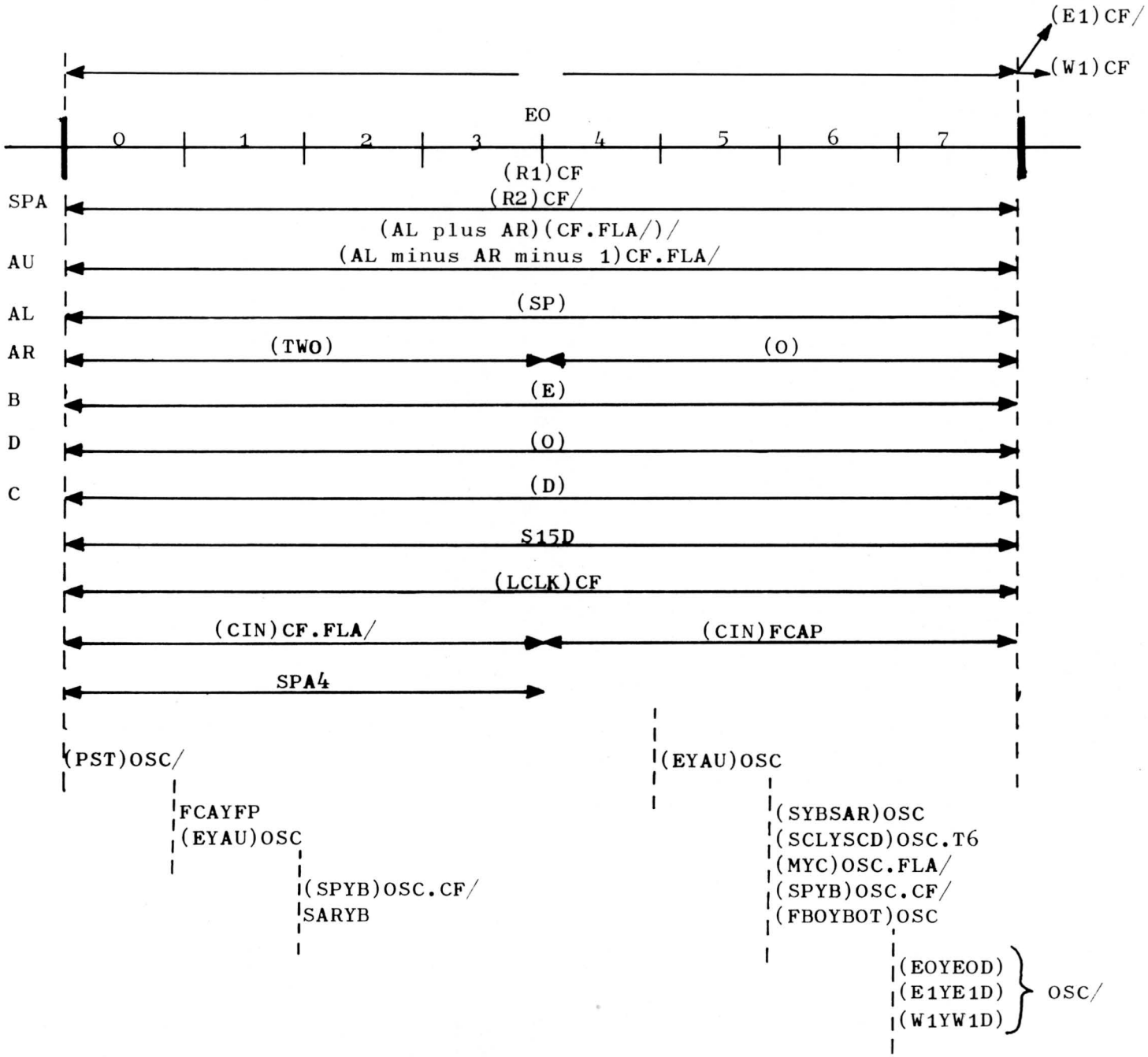


Figure H9 Execute cycle EO

1.9 WRITE CYCLES

These two cycles perform exactly similar functions except that the W2-cycle is used only during call and store decrement operations.

1.9.1 W1-CYCLE

This cycle is basically used by memory reference instructions to write the result of the E1-cycle operation into a memory location. It is also used during stack operations to write the contents of the CR into the stack areas.

During TO time, the result of the previous E1-cycle can be written into memory.

If a call function instruction, at TO time the LSB of the program status word is loaded into stack; from TO to T3 the LSBs of the stack pointer are decremented and from T4 to T6 the MSBs of the PSW are gated into the M-register.

During T7 time, logic is set up ready to perform either an E2- or W2-cycle.

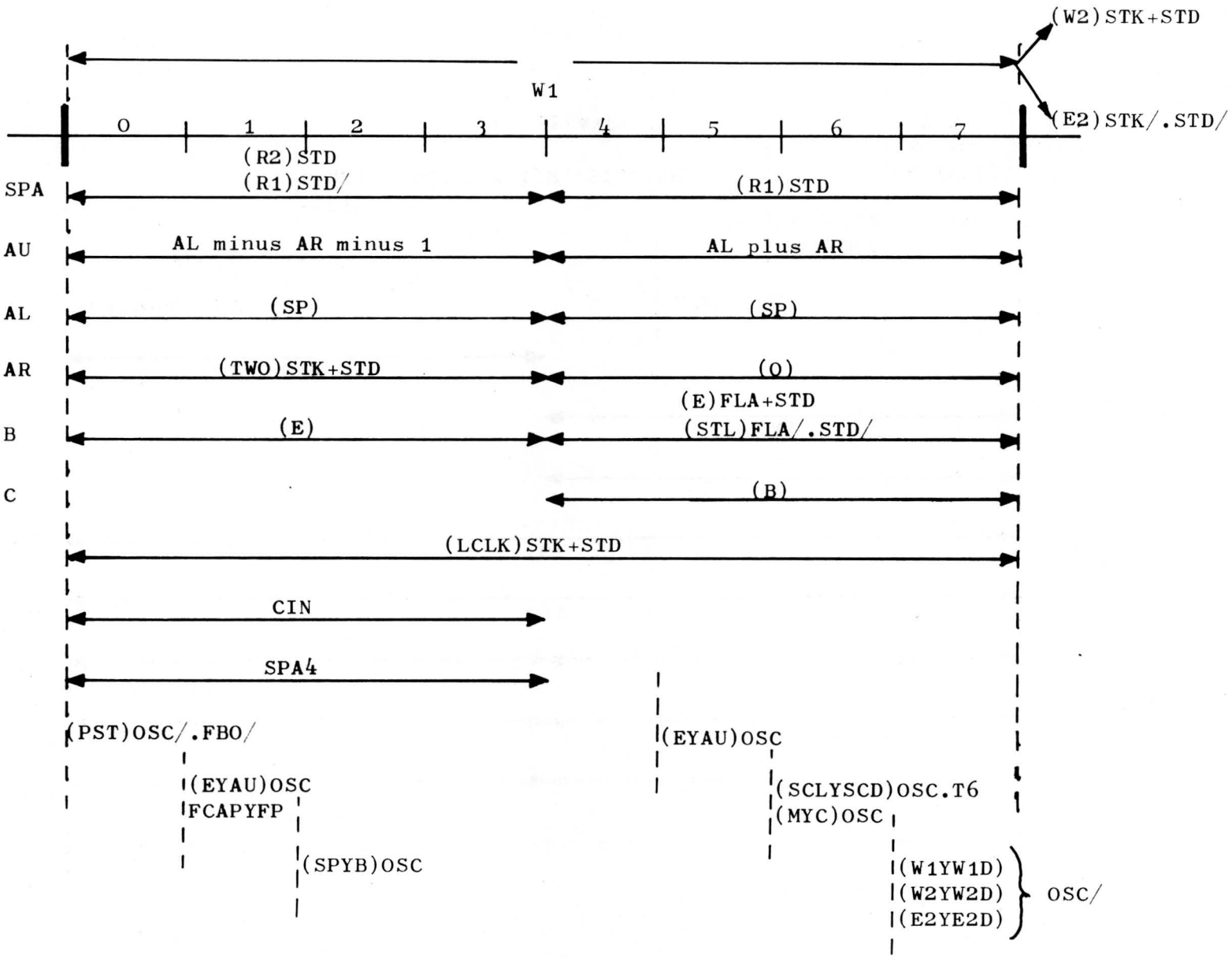


Figure H10 Write cycle W1

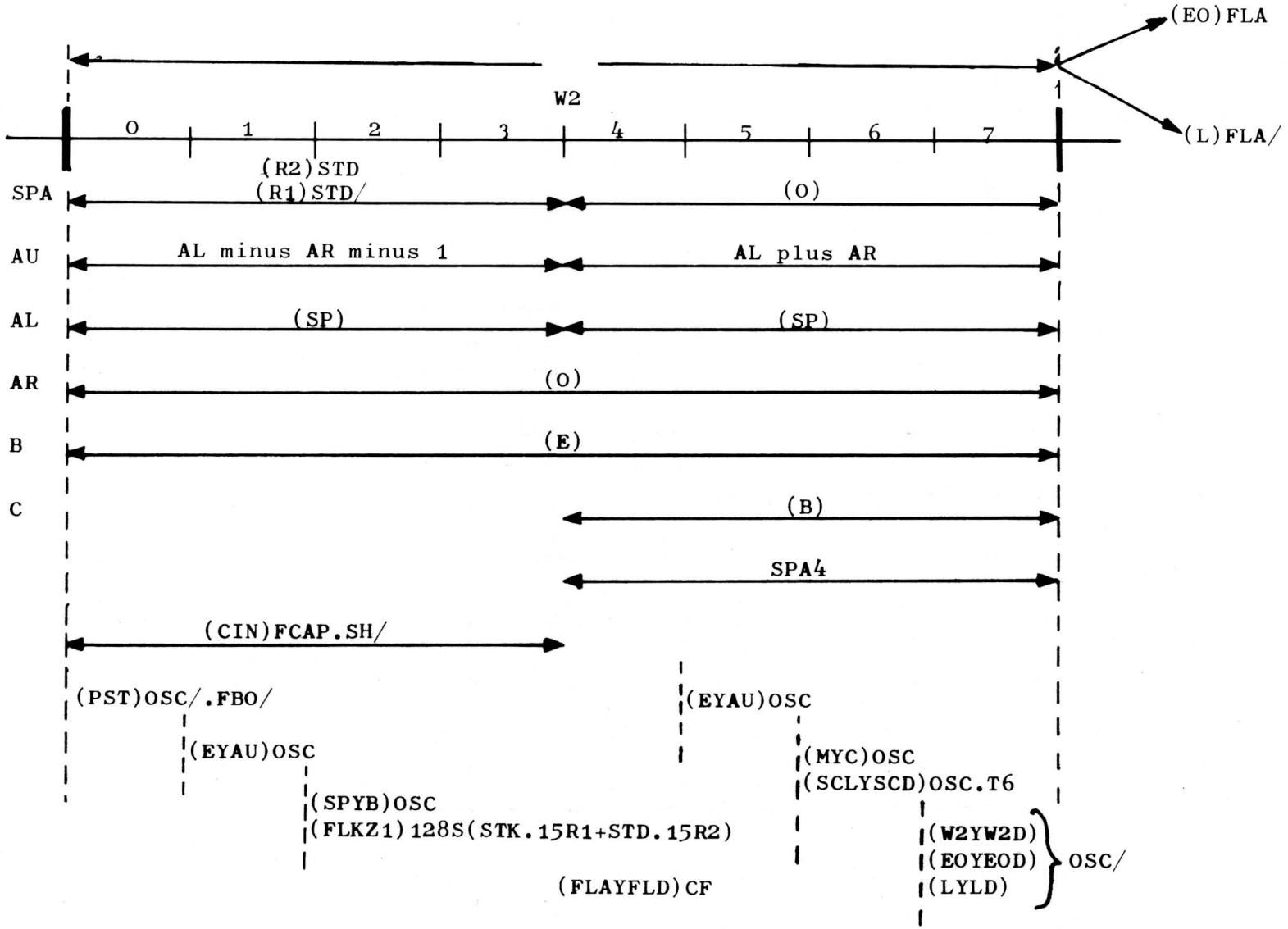


Figure H11 Write cycle W2

1.9.2 W2-CYCLE

This cycle is used only during the call and store decrement instruction. Its operation is similar to the W1-cycle.

At TO time, the contents of the M-register are written into memory.

From TO to T3, the MSB of the stack pointer has a new value loaded into it.

During T4 time, a check for stack overflow is made and an interrupt generated if this condition exists.

If this is the first time the W2-cycle has been used by the instruction, the repeat cycle flip-flop is set.

During T7 time, logic is set up ready to perform an EO-cycle.

1.10 LAST CYCLE

This cycle is used by all instructions and its main functions are to write into memory the MSB of any result obtained in the E2-cycle, to update the P- and S-registers (if necessary), and to test if an interrupt has occurred during the instruction.

At T0 time, the contents of the M-register can be written into memory.

The first half of the cycle can be used to update the LSB of the P- or S-registers. At T2 time the updated address is gated into either the P- or S-register. If a new interrupt mask pattern is required, this can be gated into the register from the BØU lines.

The second half of the cycle can be used to update the MSB of the P- or S-registers, the gating signal occurring at T6 time.

Also at T6 time a check is made to see if the interrupt-generated flip-flop needs setting.

During T7 time, logic is set up to perform either an F1-, INT-, AF1- or RL1-cycle.

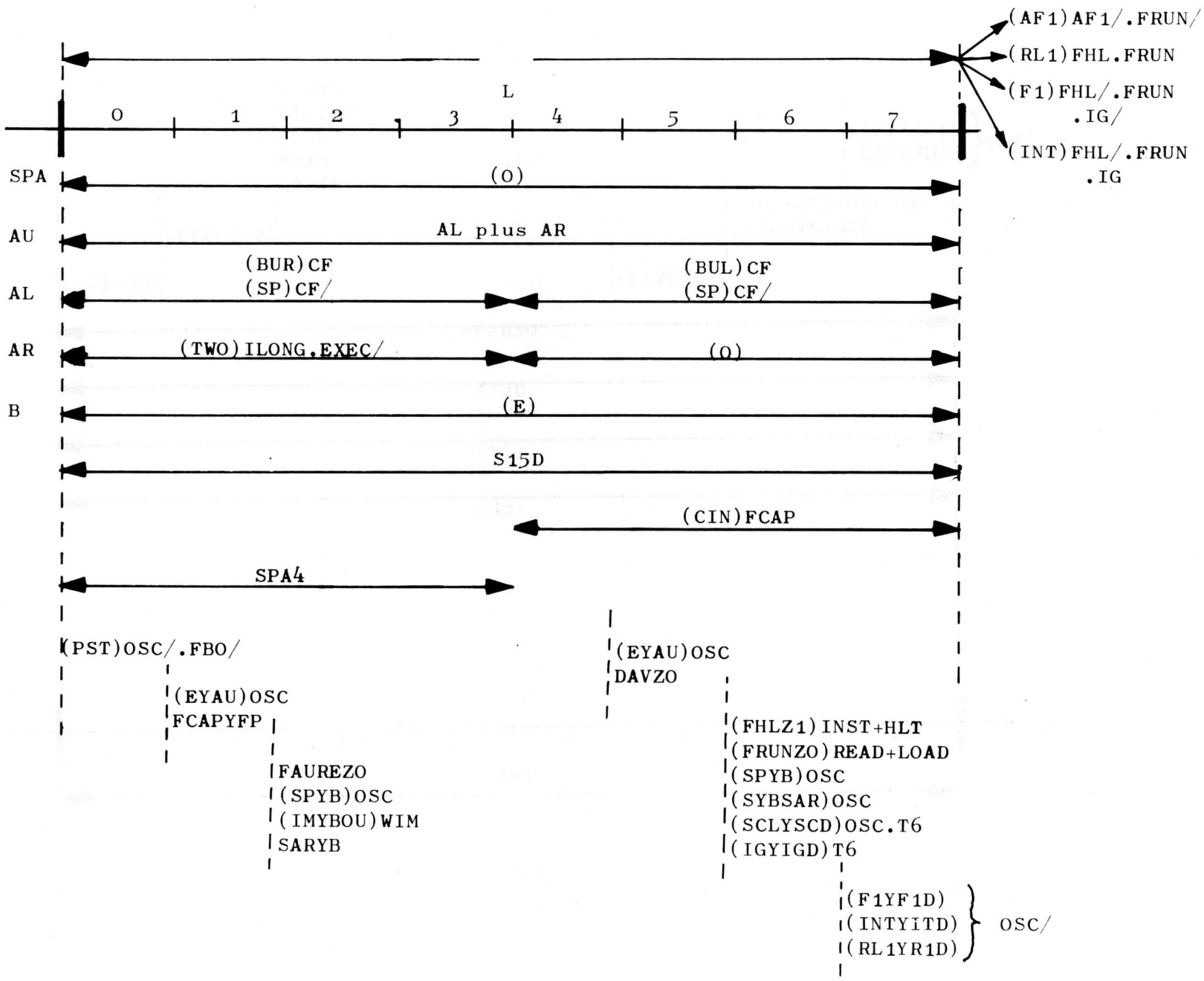


Figure H12 Last cycle L

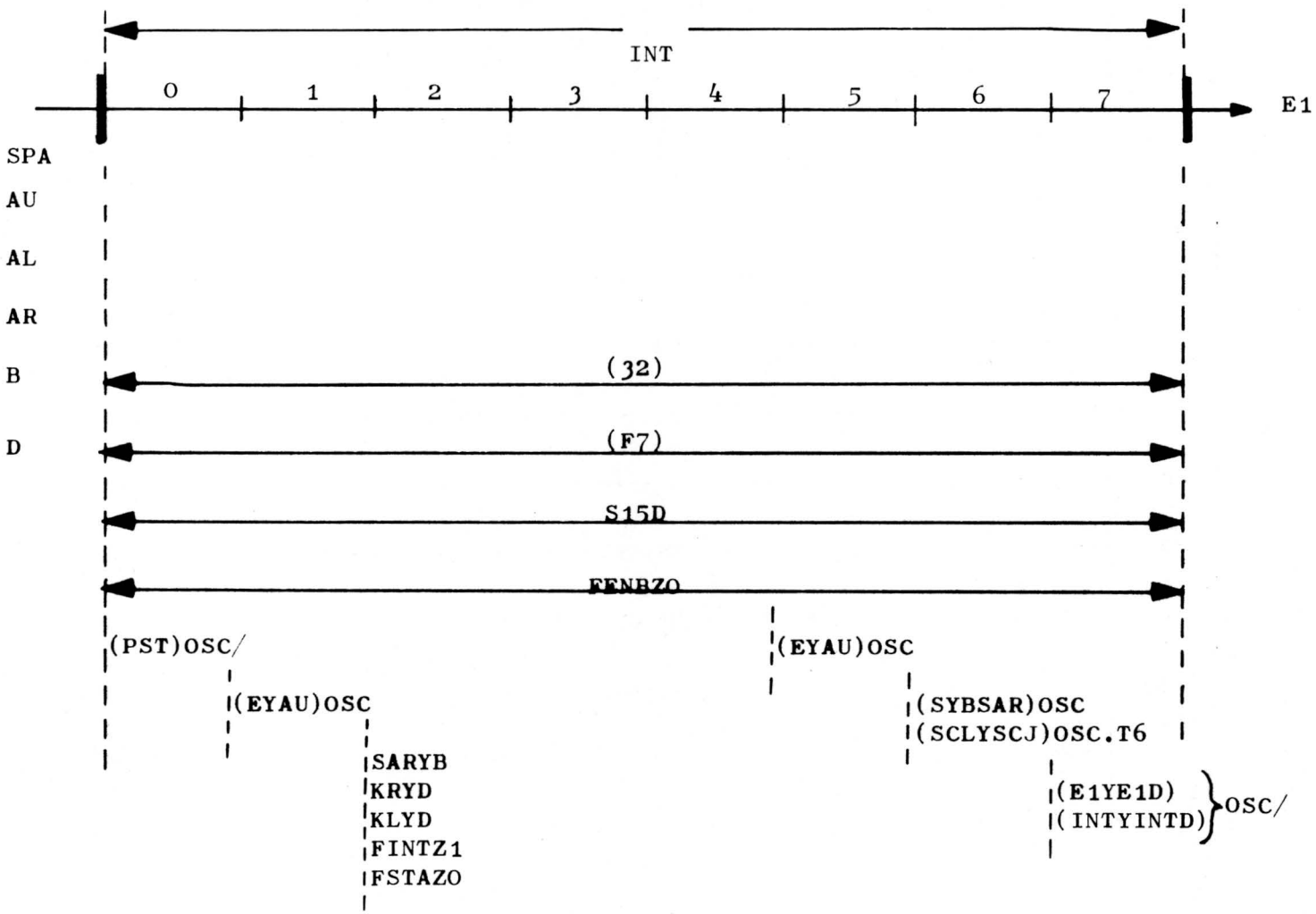


Figure H13 Interrupt cycle INT

1.11 INT-CYCLE

The main function of this cycle is to set up logic ready to perform the interrupt routine. Part of the routine is done by hardware and part by software.

During the first half of the cycle at T2 time, address 32 (start address of software routine) is gated into the SAR-register, the contents of the D-gates are gated into both halves of the K-register, and the interrupt-in-progress flip-flop is set.

During T6 time the contents of the SAR-register and B-lines are gated into the S-register. The fact that the contents of the B-lines, as well as the SAR-register, are 32 does not affect the loading of the S-register because only the 4 LSBs of the B-lines are connected to the 4 MSBs of the S-register.

During T7 time logic is set up ready to perform an E1-cycle.

1.12 DISPLAY CYCLES

Although these cycles are called display cycles they are also used when the system is first initiated, and after a power failure when the automatic restart feature is available.

They are used directly to display the contents of the SP-registers or condition register.

1.12.1 AF1-CYCLE

This cycle operates on the LSB of a word when used for display purposes.

During the first half of the cycle the selected register address is set up.

At T5 time the contents of the register or the CR register are gated on to the B-lines via the E-register.

At T6 time the contents of the B-lines are gated into the BUR-register and are available on BØU lines 08 to 15. In display mode the BØU lines are connected to the indicator lamps on the control panel.

During T7 time logic is set up ready to perform an AF2-cycle.

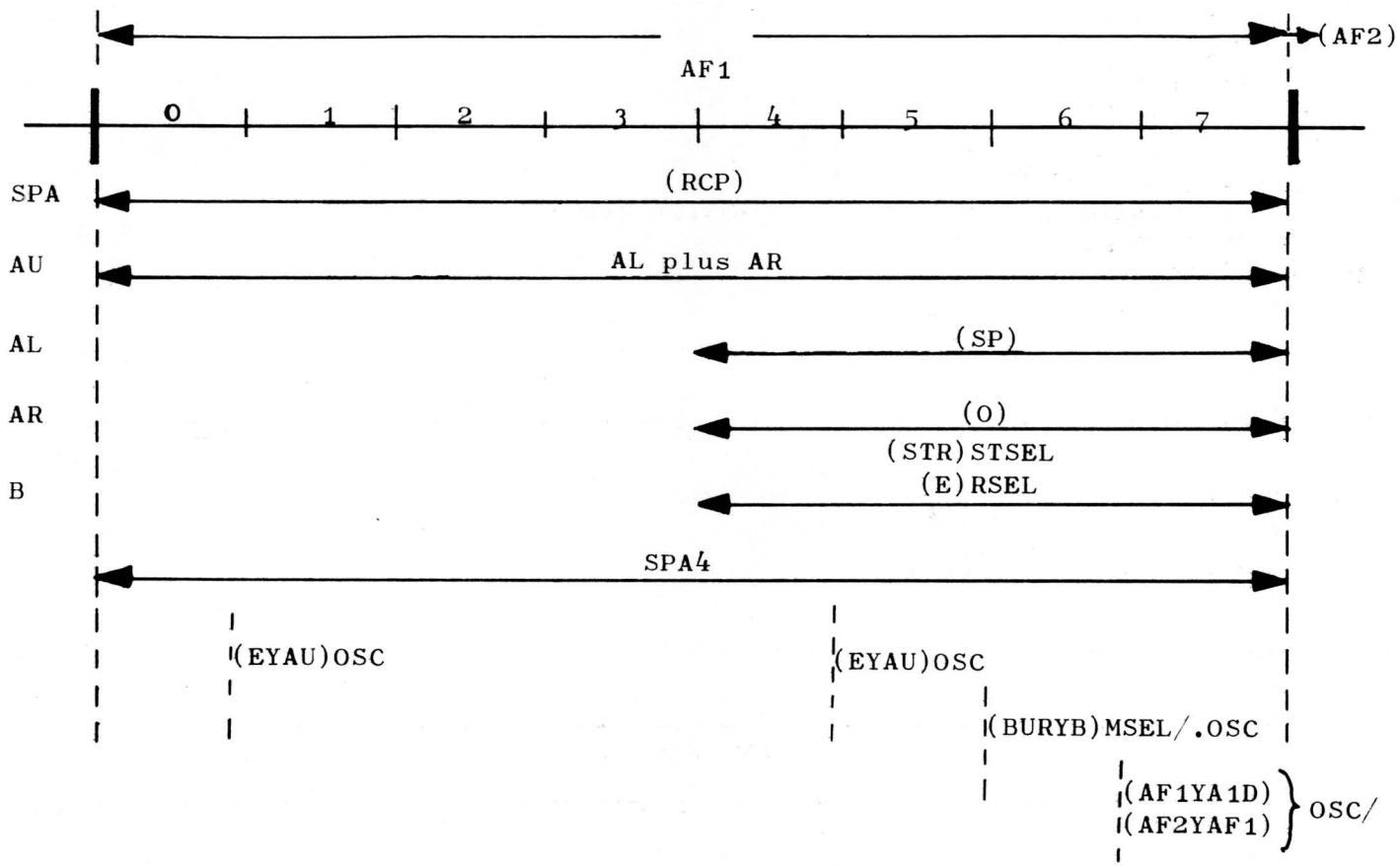


Figure H14 Display cycle AF1

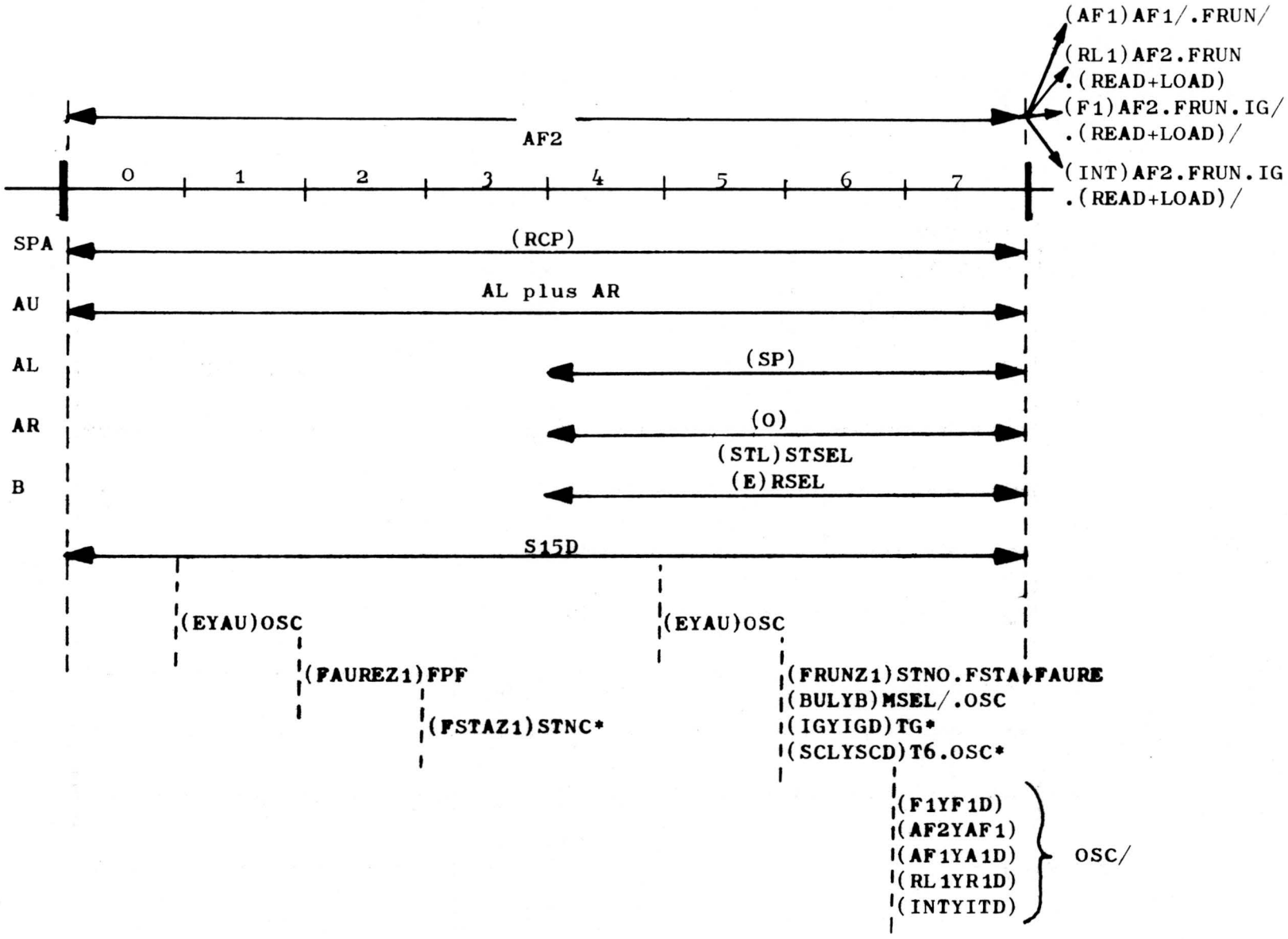


Figure H15 Display cycle **AF2**

1.12.2 AF2-CYCLE

This cycle operates on the MSB of the register to be displayed; it is also used by the automatic restart option.

During the first half of the cycle the selected register address is set up.

At T2 time the automatic restart flip-flop can be set.

At T3 time a timing signal is available which can be used if the start button is pushed.

At T5 time the contents of the selected register or the CR-register are gated on to the B-lines via the E-register.

At T6 time the contents of the B-lines are gated into the BUL-register and are available on BØU lines 00 to 07.

During T7 time logic is set up ready to perform either an RL1-, AF1-, F1- or INT-cycle.

1.13 MANUAL READ/LOAD CYCLES

These four cycles allow data either to be loaded into or read out of memory and to load the SP-registers from the control panel. The first two cycles operate on the LSB of the word and the last two cycles operate on the MSB of the word.

1.13.1 RL1-CYCLE

This cycle operates on the LSB of a memory location or register.

During load memory operations, the first half-cycle is used; it gates the contents of the BINR lines via the B-lines into the M-register.

During read memory and load register operations, the second half-cycle is used; it either gates the contents of the M-register to BØU lines 08 to 15 via the AU, B-lines and BUR-register, or gates the contents of the BINR lines into the selected register.

Note: The memory address of the required location must first be loaded into register 0 (P-register) before a read/load operation takes place.

During T7 time logic is set up ready to perform an RL2-cycle.

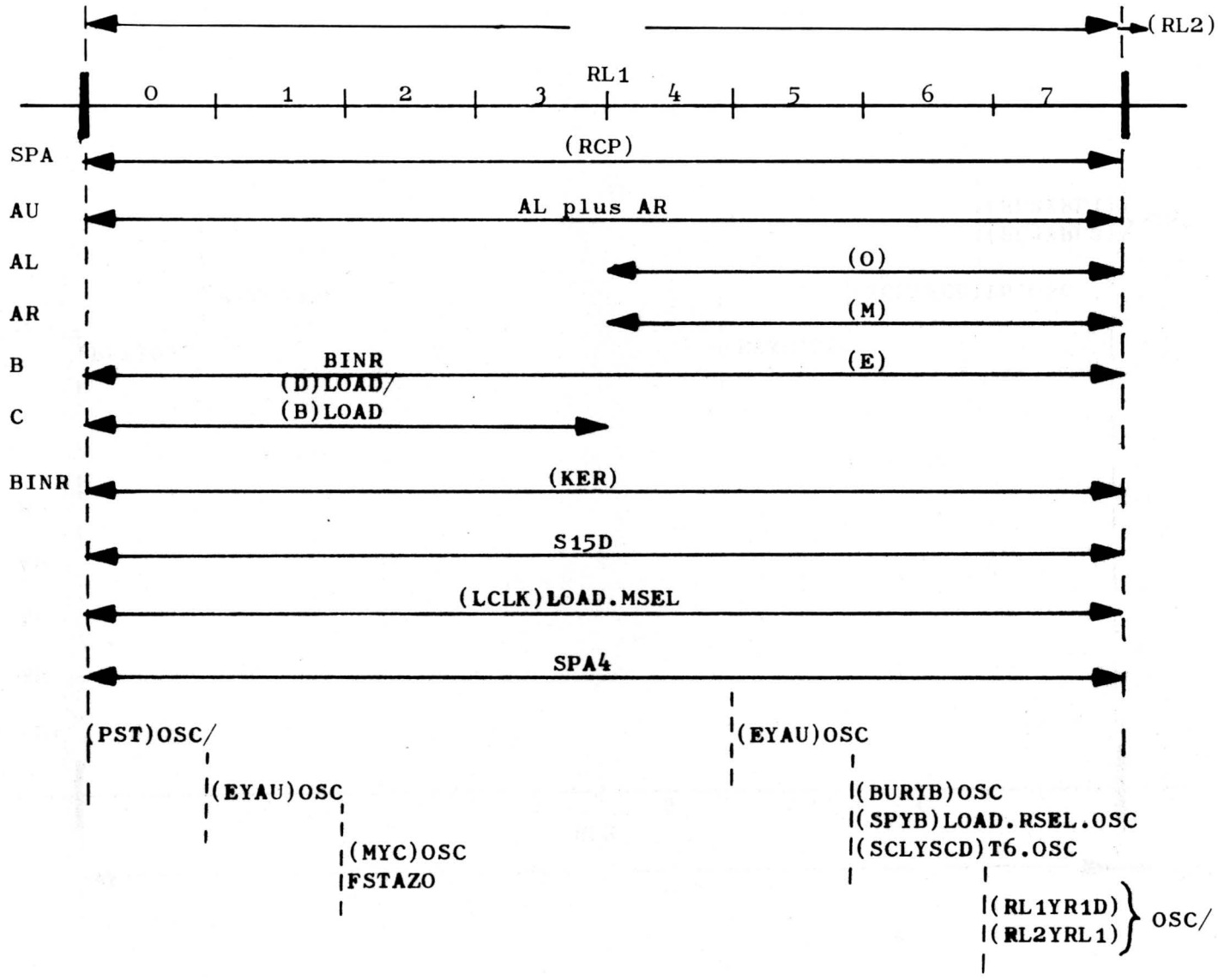


Figure H16 Manual read/write cycle RL1

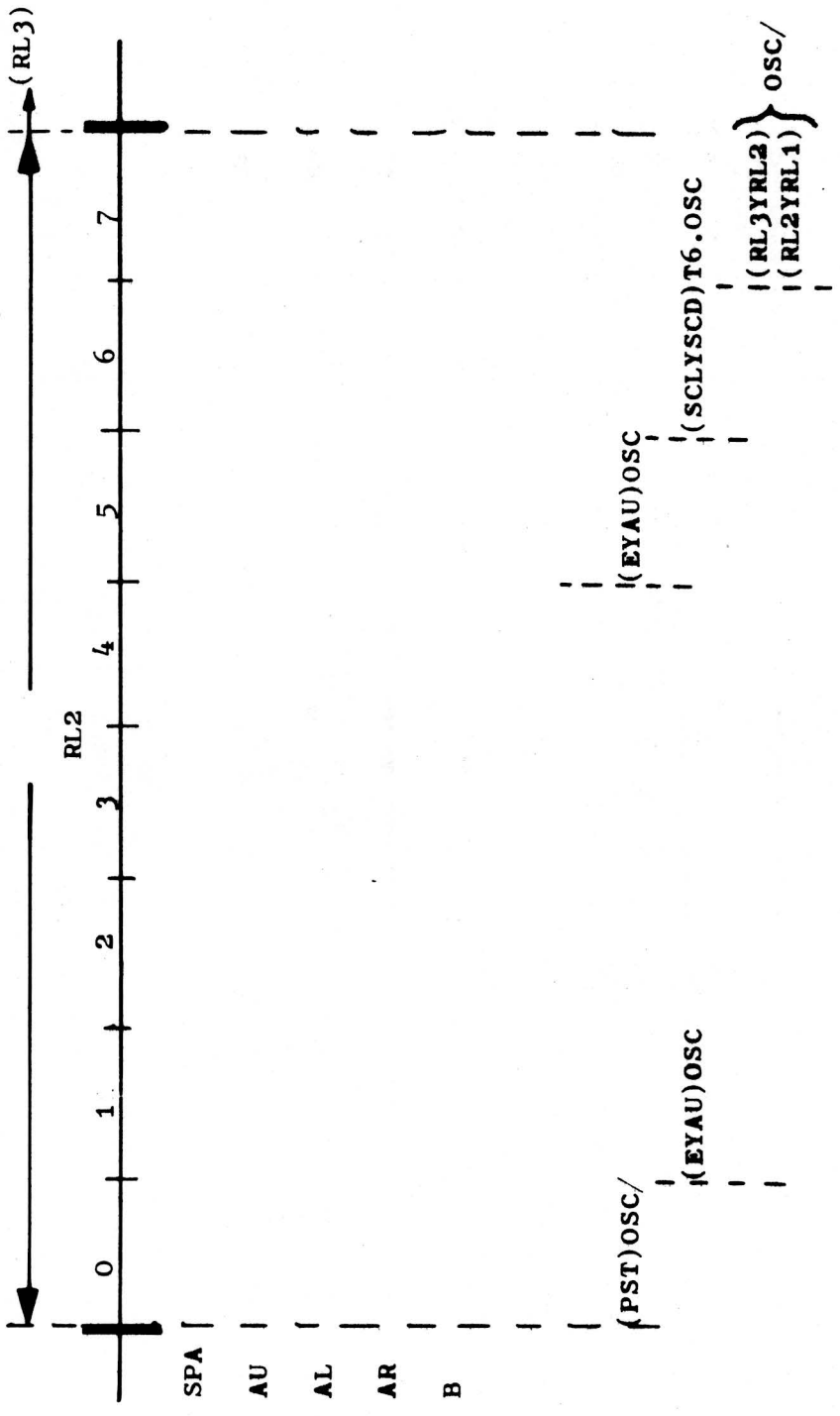


Figure H17 Manual read/write cycle RL2

1.13.2 RL2-CYCLE

This cycle operates on the LSB of a word during load operations.

At T0 time the contents of the M-register are gated into the memory.

At T6 time the read/write flip-flop can be updated.

During T7 time logic is set up ready to perform an RL3-cycle.

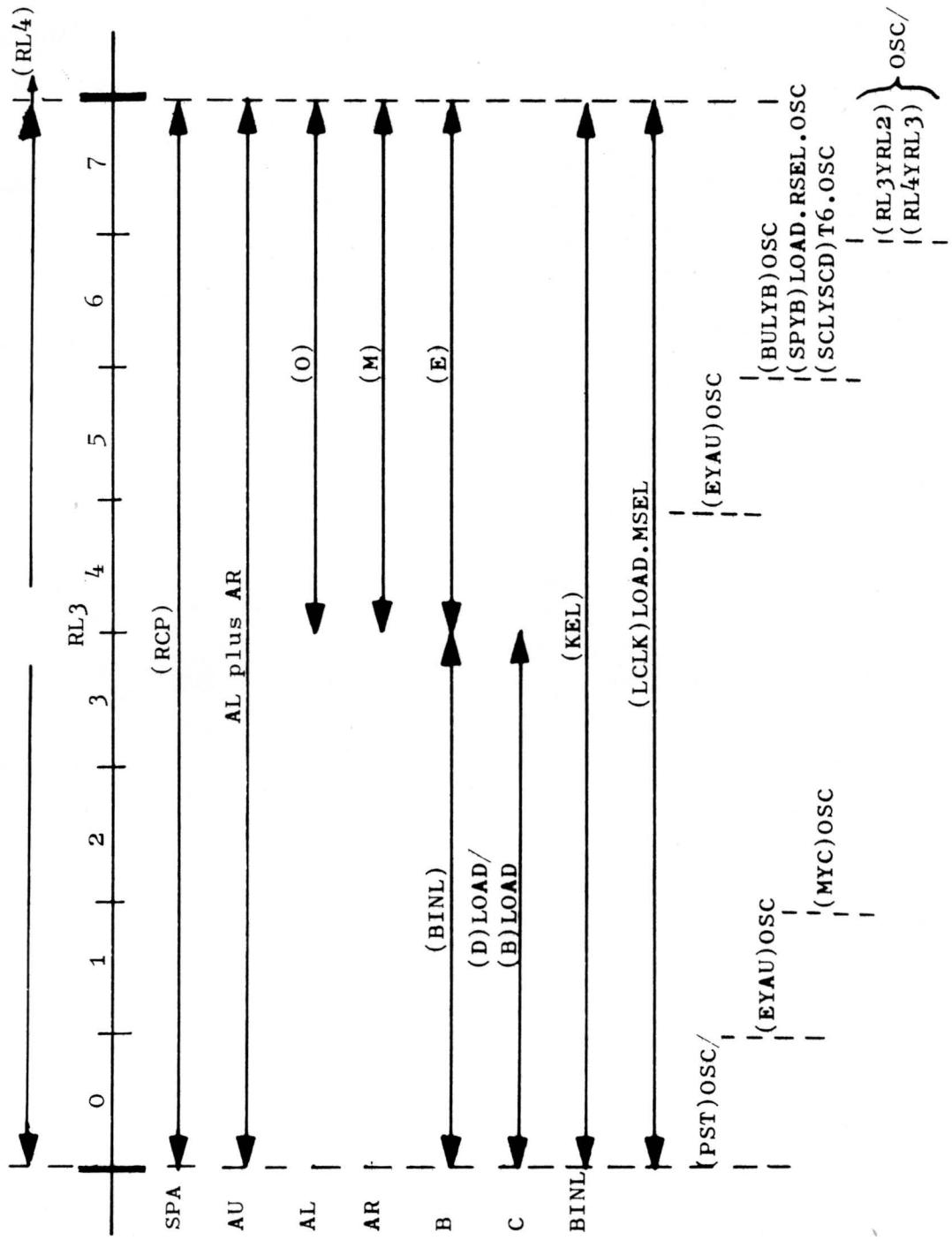


Figure H18 Manual read/write cycle RL3

1.13.3 RL3-CYCLE

This cycle operates on the MSB of the word.

During load memory operations, the first half of the cycle is used; it gates the contents of BINL in to the M-register via the B-lines.

During read memory or load register operations, the second half of the cycle is used; it either gates the contents of the M-register on to the BØU lines 00 to 07 via the AU, B-lines and BUL-register or loads the contents of the BINL lines into the selected register.

During T7 time logic is set up ready to perform an RL4-cycle.

1.13.4 RL4-CYCLE

This cycle is used for load memory operations and to update the P- and S-registers.

At TO time the contents of the M-register are loaded into memory.

During the first half of the cycle the LSBs of the P-register are updated and the updated address is gated into the LSB of the P-register and the SAR-register.

During the second half of the cycle, the MSBs of the P-register are updated and the updated address is gated into the P-register and, together with the contents of the SAR-register, gated into the S-register.

At T6 time the RUN and HALT flip-flops are reset to zero and the read/write flip-flop can be updated.

During T7 time logic is set up to perform an AF1-cycle.

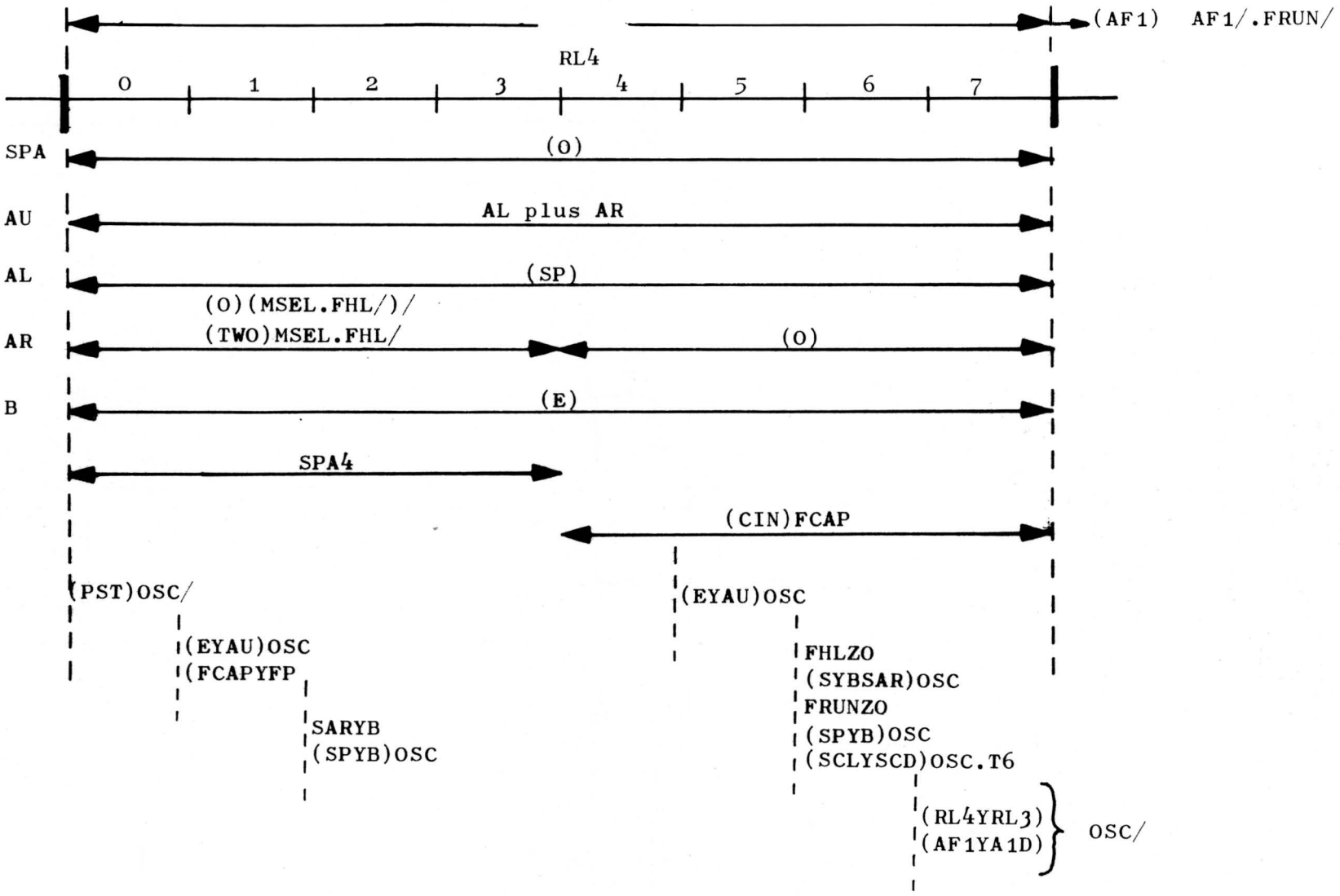


Figure H19 Manual read/write cycle RL4

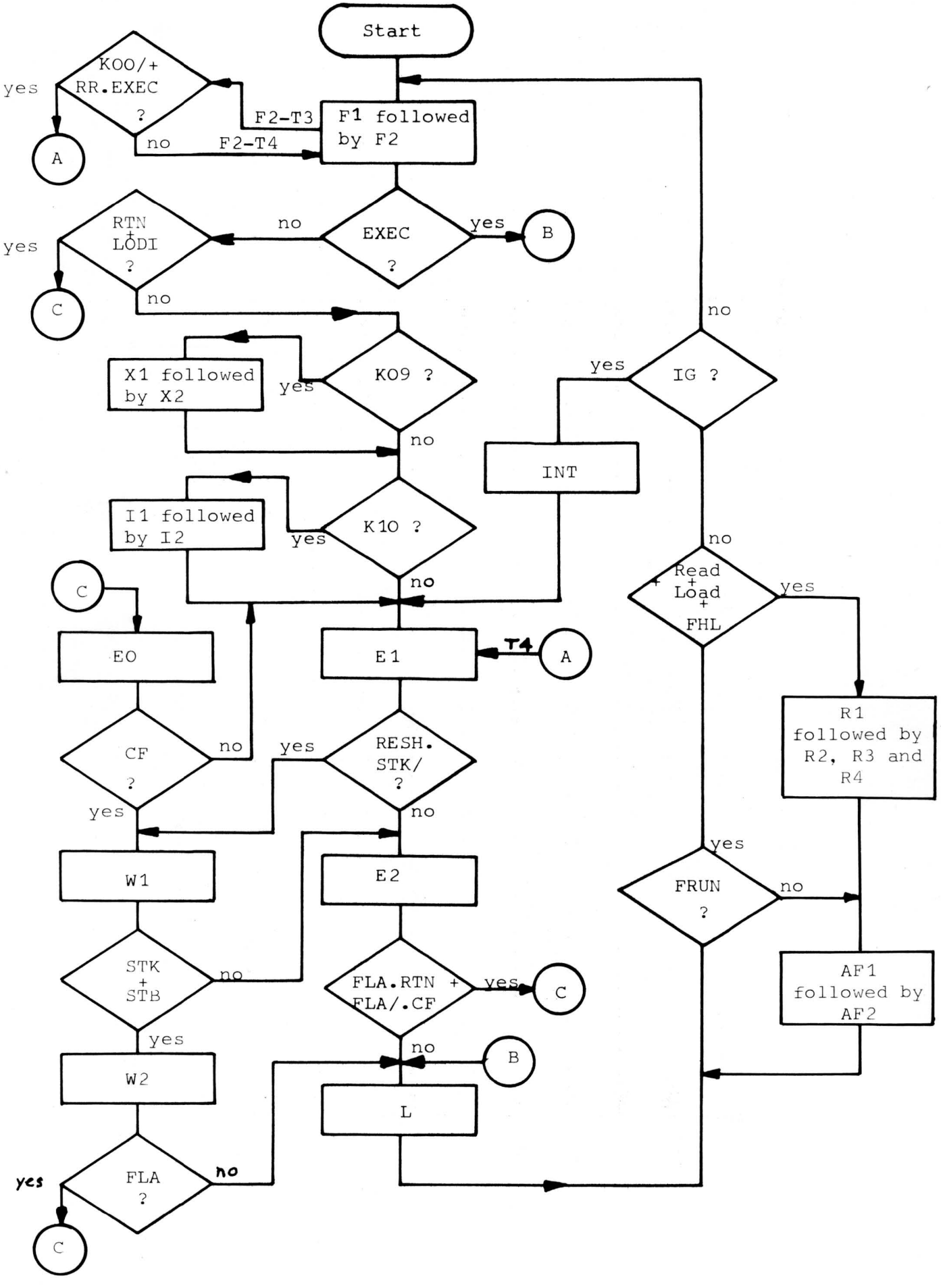


Figure 1.2 Sequence of cycles

1.14 SEQUENCE OF CYCLES

The sequence in which the cycles are used depends upon the type of instruction, the addressing mode and whether the result is stored in the memory or in a register. In the following table, the sequence of cycles is given for all instructions. The assembly language mnemonic, as well as the logic mnemonic, is given to assist in the identification of the individual instructions. (See Figure 1.2, page 1-46.)

assembly mnemonic	logic mnemonic	sequence of cycles
TW load register (MR)	LD	F1→F2→[X1→X2→I1→I2]→E1 →E2→L
TW store register (MR)	ST	F1→F2→[X1→X2→I1→I2]→E1 →W1→E2→L
TW load (register/register)	LDR	F1→F2→E1→E2→L
TW store (register/register)	STR	F1→F2→E1→W1→E2→L
TW load (constant)	LDK	F1→F2→E1→E2→L
AB absolute branch (MR)	ABI	F1→F2→[X1→X2→I1→I2]→E1 →E2→L
AB (register/register)	ABR	F1→F2[→E1→E2]→L
AB (constant)	AB	F1→F2[→E1→E2]→L
AD (MR)	AD	F1→F2→[X1→X2→I1→I2]→E1 →E2→L
	ADS	F1→F2[→X1→X2→I1→I2]→E1 →W1→E2→L
AD (register/register)	ADR	F1→F2→E1→E2→L
	ADRS	F1→F2→E1→W1→E2→L
AD (constant)	ADK	F1→F2→E1→E2→L
IM (MR)	IM	F1→F2[→X1→X2→I1→I2]→E1 →W1→E2→L
IM (register/register)	IMR	F1→F2→E1→W1→E2→L

assembly mnemonic	logic mnemonic	sequence of cycles
SU (MR)	SU	F1→F2[→X1→X2→I1→I2]→E1 →E2→L
	SUS	F1→F2[→X1→X2→I1→I2]→E1 →W1→E2→L
SU (register/register)	SUR	F1→F2→E1→E2→L
	SURS	F1→F2→E1→W1→E2→L
SU (constant)	SUK	F1→F2→E1→E2→L
CZ (MR)	CZ	F1→F2[→X1→X2→I1→I2]→E1 →W1→E2→L
CR (register/register)	CZR	F1→F2→E1→W1→E2→L
I (MR)	AN	F1→F2[→X1→X2→I1→I2]→E1 →E2→L
	ANS	F1→F2[→X1→X2→I1→I2]→E1 →W1→E2→L
I (register/register)	ANR	F1→F2→E1→E2→L
	ANRS	F1→F2→E1→W1→E2→L
I (constant)	ANK	F1→F2→E1→E2→L
IH	INH	F1→F2→E1→E2→L
HT	HLT	F1→F2→E1→E2→L
RI	RIT	F1→F2→E1→E2→L
U (MR)	ØR	F1→F2[→X1→X2→I1→I2]→E1 →E2→L
	ØRS	F1→F2[→X1→X2→I1→I2]→E1 →W1→E2→L
U (register/register)	ØRR	F1→F2→E1→E2→L
	ØRRS	F1→F2→E1→W1→E2→L
U (constant)	ØRK	F1→F2→E1→E2→L
EN	ENB	F1→F2→E1→E2→L
LM	LKM	F1→F2→E1→E2→L
X (MR)	XR	F1→F2[→X1→X2→I1→I2]→E1 →E2→L
	XRS	F1→F2[→X1→X2→I1→I2]→E1 →W1→E2→L

assembly mnemonic	logic mnemonic	sequence of cycles
X (register/register)	XRR	F1→F2→E1→E2→L
	XRRS	F1→F2→E1→W1→E2→L
X (constant)	XRK	F1→F2→E1→E2→L
LA	SLA1	F1→F2→E1→E2→L
RA	SRA1	F1→F2→E1→E2→L
LL	SLL1	F1→F2→E1→E2→L
RC	SRC1	F1→F2→E1→E2→L
CT	CIØ	F1→F2→E1→E2→L
ØT	ØTR	F1→F2→E1→E2→L
WM	WIM	F1→F2→E1→E2→L
IN	INR	F1→F2→E1→E2→L
TS	TST	F1→F2→E1→E2→L
SS	SST	F1→F2→E1→E2→L
RL	RIL	F1→F2→E1→E2→L
RF	RF	F1→F2[→E1→E2]→L
RB	RB	F1→F2[→E1→E2]→L
EC (register/register)	ECR	F1→F2→E1→E2→L
TC (MR)	LC	F1→F2[→X1→X2→I1→I2]→E1→E2→L
	SC	F1→F2[→X1→X2→I1→I2]→E1→W1→E2→L
TC (register/register)	LCR	F1→F2→E1→E2→L
	SCR	F1→F2→E1→W1→E2→L
TC (constant)	LCK	F1→F2→E1→E2→L
CW (MR)	CW	F1→F2[→X1→X2→I1→I2]→E1→E2→L
CW (register/register)	CWR	F1→F2→E1→E2→L
CF (MR)	CFI	F1→F2[→X1→X2→I1→I2]→E1→E2→EO→W1→W2→EO→W1→W2→L
CF (register/register)	CFR	F1→F2→E1→E2→EO→W1→W2→EO→W1→W2→L
CF (constant)	CF	F1→F2→E1→E2→EO→W1→W2→EO→W1→W2→L
RT	RTN	F1→F2→EO→E1→E2→EO→E1→E2→L

assembly mnemonic	logic mnemonic	sequence of cycles
C1 (MR)	C1	F1→F2[→X1→X2→I1→I2]→E1 →E2→L
	C1S	F1→F2[→X1→X2→I1→I2]→E1 →W1→E2→L
C1 (register/register)	C1R	F1→F2→E1→E2→L
	C1RS	F1→F2→E1→W1→E2→L
	TM	F1→F2→E1→E2→L
	TNM	F1→F2→E1→E2→L
	CWK	F1→F2→E1→E2→L
	LØDI	F1→F2→E0→E1→E2→L

Note: The cycles inside the [] may or may not be used depending on the addressing mode or, in the case of branch instruction, whether the branch is executed.

The following flowchart shows all the cycles and from it can be traced the sequence of the individual instructions.

1.15 INTERRUPT SYSTEM

The interrupt system is used to service both internal and external interrupts. Internal interrupts can occur for power failure, control panel, real-time clock, link to monitor and stack overflow events and indicate their presence by setting one of the bits in the GF-register. External interrupts are generated by the device control units and are used to request either a data transfer or to indicate that the device has a certain status.

With the basic P850 system there is only one interrupt signal line but this can be increased to either eight or sixteen by including the optionally-available interrupt lines. When these lines are included, they are used in conjunction with a 16-bit interrupt mask. This mask can be set to inhibit or enable the individual signal lines by the programmer.

Both internal and external interrupts are serviced by a single priority level system that uses automatic stack handling techniques.

1.16 STACK HANDLING

The automatic stack handling technique is a system which, when an interrupt occurs, stores information needed to return to an interrupted program after an interrupt has been serviced. Location 128 is used to detect stack overflow and stored in this location is an instruction that generates an interrupt if the stack tries to write beyond this location. The size and starting address of the stack is determined by the programmer and full details for this is given in the P850 User Manual, parts 1 and 2.

Part of the stack handling is done by hardware and part by a software routine and both hardware and software can access the stack. The operation of the stack handling and the sequence of cycles used is given below; Figure 1.3 shows a flowchart of the operation.

During the L-cycle, a test is made to see if an interrupt has been generated whilst executing the current instruction; if so the logic is set up ready to perform the INT-cycle at the end of the T7 pulse of the L-cycle.

INT-cycle - this will load the instruction call function into the K-register and will set up address 32 (address 32 contains the address of the first instruction in the software routine) in the S-register. It then sets up logic ready to perform the hardware routine.

1.16.1 HARDWARE ROUTINE

This routine first reads the contents of address 32 into the BU-register ready to perform an indirect branch to the software routine as soon as the hardware routine has been completed. It then proceeds to load the contents of the Program Status Word (PSW) (in actual practice this will be the state of the condition register) and the P-register into the stack updating the stack pointer so that it points to the next free location in the stack.

E1-cycle - the contents of the LSB of address 32 are read out of memory and loaded into BUR (the LSB of the BU-register). It then sets up the logic to perform the E2-cycle.

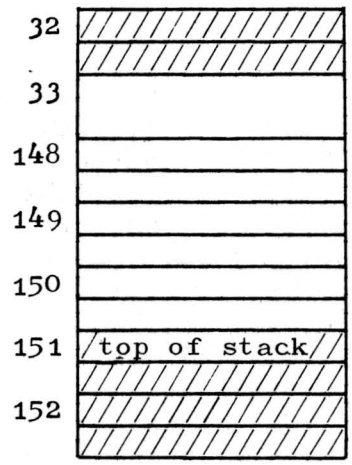
E2-cycle - the contents of the MSB of address 32 are read out of memory and loaded into BUL (the MSB of the BU-register). It then sets up the logic to perform the EO-cycle.

The remaining part of this routine deals with updating the stack pointer and loading the contents of the PSW and P-register into the stack. An unusual method of execution has been adopted so as to reduce the overall execution time of the routine; therefore, to make clear the operation of the routine, a series of block diagrams (showing the state of the stack pointer, S-register and the stack) are included in the description of the cycles which show the changes as they occur. The memory location addresses shown are all fictitious - the actual addresses will depend on the program being used.

EO-cycle - this cycle is used only during the excution of instructions which have a memory address contained in register 15. The memory address contained in register 15 is the first free location in the stack area. This address is always decremented during a call function and incremented during a return from function. In this way it always points to the next location in the stack which is to be used.

S-register contains 32

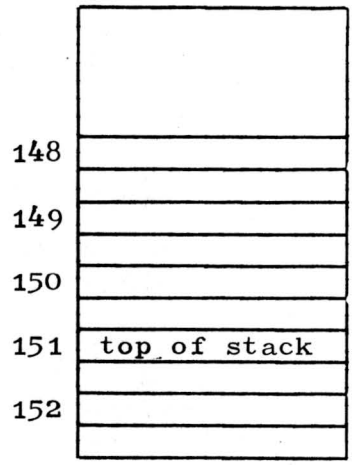
register 15 contains 150



From TO to T6 time the decremented contents of register 15 are loaded into the S-register. During T6 time the LSBs of the PSW are loaded into the M-register (the contents of the M-register will be 0) ready for loading into the memory during the W1-cycle.

S-register contains 149

register 15 contains 150



During T7 time the logic is set up ready to perform a W1-cycle.

W1-cycle - this cycle loads the LSB of the PSW into the stack and updates the LSB of the stack pointer.

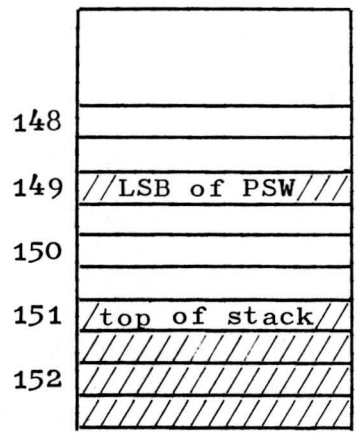
At TO time the contents of the M-register (LSB of PSW) are loaded into the least significant half word of address 149.

From TO to the end of T2 time the LSBs of the stack pointer are decremented.

From T4 to T6 time the MSB of the PSW (contents of the condition register) are loaded into the M-register ready for loading into memory during the W2-cycle.

S-register contains 149

register 15 in process of being updated



During T7 time logic is set up ready to perform a W2-cycle.

W2-cycle - this cycle loads the MSB of the PSW into the stack and loads the MSB of the stack pointer address into register 15.

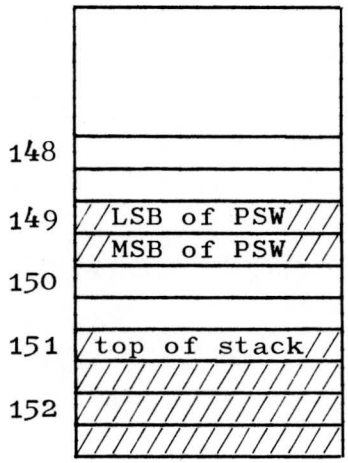
At TO time the contents of the M-register (MSB of PSW) are loaded into the most significant half word of address 149.

From TO to the end of T2 time the MSB of the stack pointer has a new value loaded into it.

During T4 time a check is made for stack overflow and an interrupt is generated if this condition exists. Because this is the first time the W2-cycle has been used, it will set up the logic ready to repeat the EO-, W1- and W2-cycles.

S-register contains 149

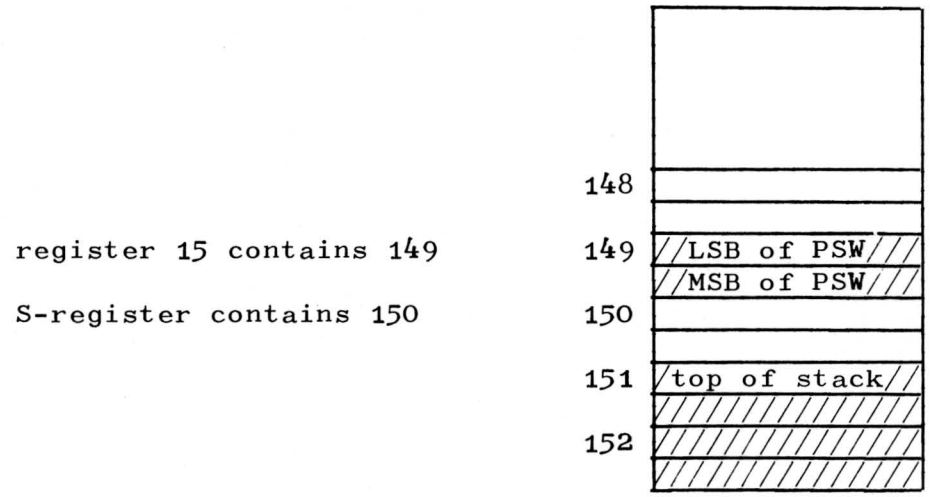
register 15 contains 149



During T7 time the logic is set up ready to perform an EO-cycle.

EO-cycle - during this cycle the contents of the S-register are incremented ready to load the contents of the P-register into the stack during the W1- and W2-cycles.

From TO to T6 time the contents of register 15 are incremented and loaded into the S-register.
During T6 time the LSBs of the P-register are loaded into the M-register ready for loading into memory during the W1-cycle.



register 15 contains 149

S-register contains 150

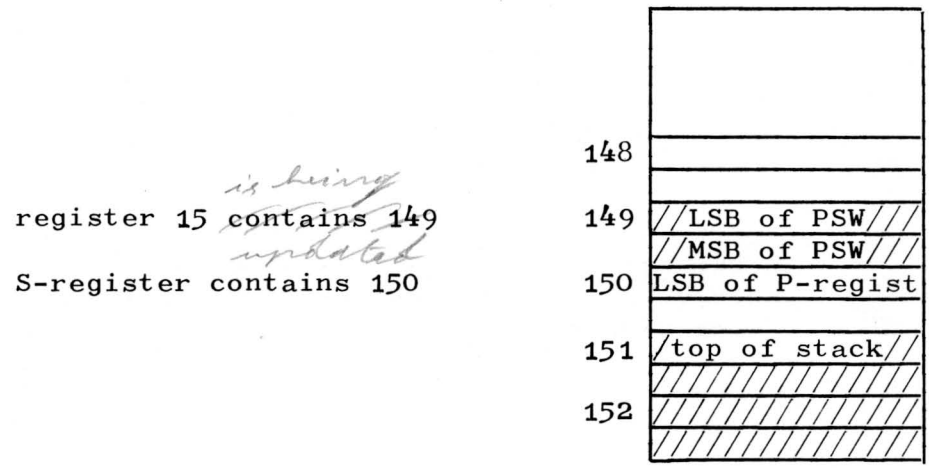
During T7 time the logic is set up ready to perform a W1-cycle.

W1-cycle - this cycle loads the LSB of the P-register into the stack area and updates the LSB of the stack pointer address.

At T0 time the contents of the M-register (LSB of the P-register) are loaded into memory.

From T0 to the end of T2 time the LSBs of the stack pointer address are decremented.

From T4 to end of T6 time the MSBs of the P-register are loaded into the M-register ready for loading into memory during the W2-cycle.



register 15 contains 149

S-register contains 150

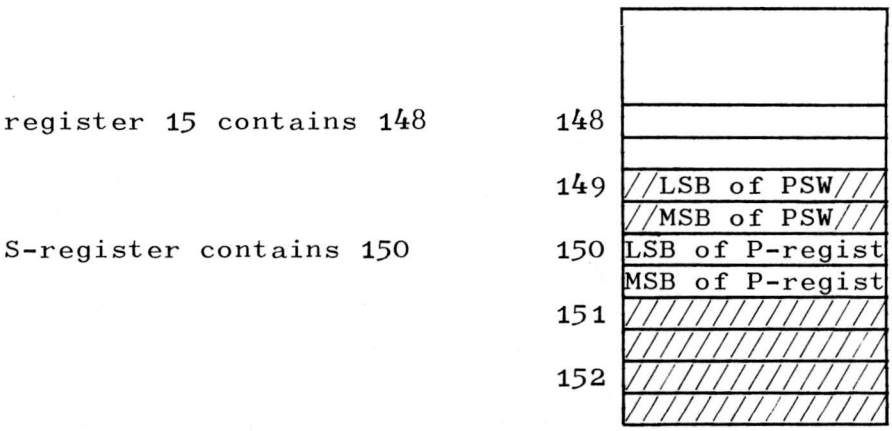
During T7 time logic is set up ready to perform a W2-cycle.

W2-cycle - this cycle loads the MSB of the P-register into the stack, loads a new value in the MSB of the stack pointer and resets the repeat cycles flip-flop.

At TO time the contents of the M-register (MSB of the P-register) are loaded into memory.

From TO to the end of T2 time the MSB of the stack pointer has a new value loaded into it.

At T4 time the repeat cycles flip-flop is reset.

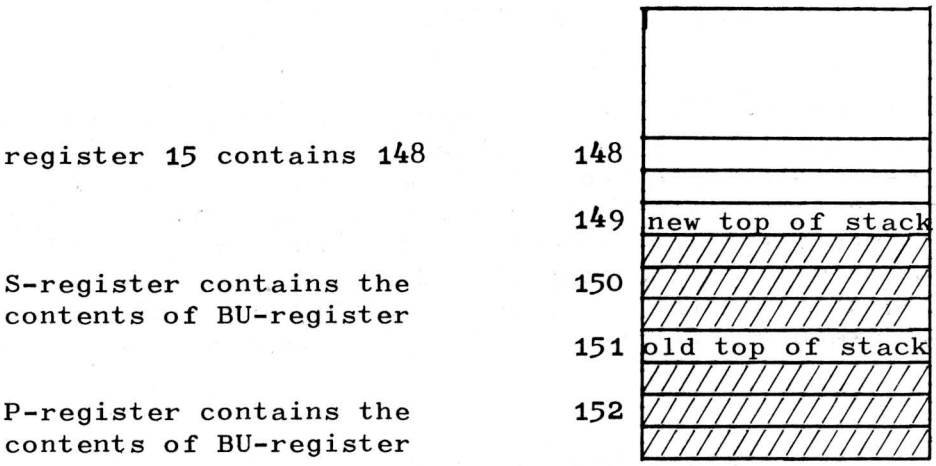


During T7 time logic is set up ready to perform the L-cycle.

L-cycle - this cycle loads the contents of the BU-register into the S- and P-registers and sets the interrupt in treatment flip-flop. The contents of the BU-register are the starting address of the software routine and were loaded into the BU-register during the E1- and E2-cycles.

From TO to the end of T2 time the contents of the BUR are loaded into the SAR-register and the LSB of the P-register.

From T4 to the end of T6 time the contents of the BUL are loaded into the MSB of the P-register and they are also loaded, together with the contents of the SAR-register, into the S-register.



At T6 time the interrupt in treatment flip-flop is set to stop any new interrupt being serviced until the essential part of the software routine has been completed.

During T7 time logic is set to enter the F1-cycle of the software routine.

1.16.2 SOFTWARE ROUTINE

The exact sequence of cycles for the software routine will depend on the type of instruction completed before the interrupt was serviced but, in general, it will follow the sequence given below.

(1) F1 and F2 will fetch the instructions starting at the address which was contained in location 32.

(2) The E- and W-cycles will be used as necessary to store the contents of any required registers and the mask register.

(3) The E-cycles are also used to test the interrupt signal lines to find out which one caused the interrupt and to reset the interrupting signal line.

(4) If more than one interrupt is expected, a new mask pattern will be loaded into the mask during the operation of one of the L-cycles and the enable interrupt flip-flop will be set.

1.16.3 INTERRUPT SERVICING

The routine for servicing the actual interrupt can now be processed. What the sequence of cycles is will depend on the type of interrupt being serviced. Once this is known the sequence can be traced from the operation timing diagrams. However, the last instructions of the routine will always be the same. The contents of the registers and the old mask register contents will be reinstated and logic will be set up to execute a return from function instruction.

1.16.4 RETURN FROM FUNCTION

The return from function instruction operates in exactly the opposite way to the call function. The contents of the stack pointer (register 15) are incremented and the contents of the P- and CR-registers are reinstated. The following sequence of cycles takes place.

E0-cycle - this cycle increments the contents of the S-register and register 15. The contents of these registers will now be the address of the location that holds the contents of the PSW.

During T7 time logic is set up to perform an E1-cycle.

E1-cycle - this is a 'do nothing' cycle for the P850 but is still included in the sequence for compatibility with the larger P800 central processors.

Its only function is to set up logic for an E2-cycle during T7 time.

E2-cycle - this will load the contents of the stack location into the CR-register and during T7 time will set up logic to perform another E0-cycle.

E0-cycle - this cycle increments the contents of the S-register and register 15. The contents of these registers will now be the address of the location that holds the old contents of the P-register.

During T7 time logic is set up ready to perform an E1-cycle.

E1-cycle - this will load the LSB of the program address from the stack into the LSB of the P-register. It will also set up the logic ready to perform an E2-cycle.

E2-cycle - this will load the MSB of the program address from the stack into the MSB of the P-register and will set up the logic ready to perform an L-cycle.

L-cycle - this will load the contents of the P-register into the S-register and will set up the logic ready to perform a fetch cycle.

If an interrupt occurs during the operation of the interrupt servicing routine or return from function, the whole procedure is started again for the new interrupt. If no new interrupt has occurred, the next program instruction will be performed.

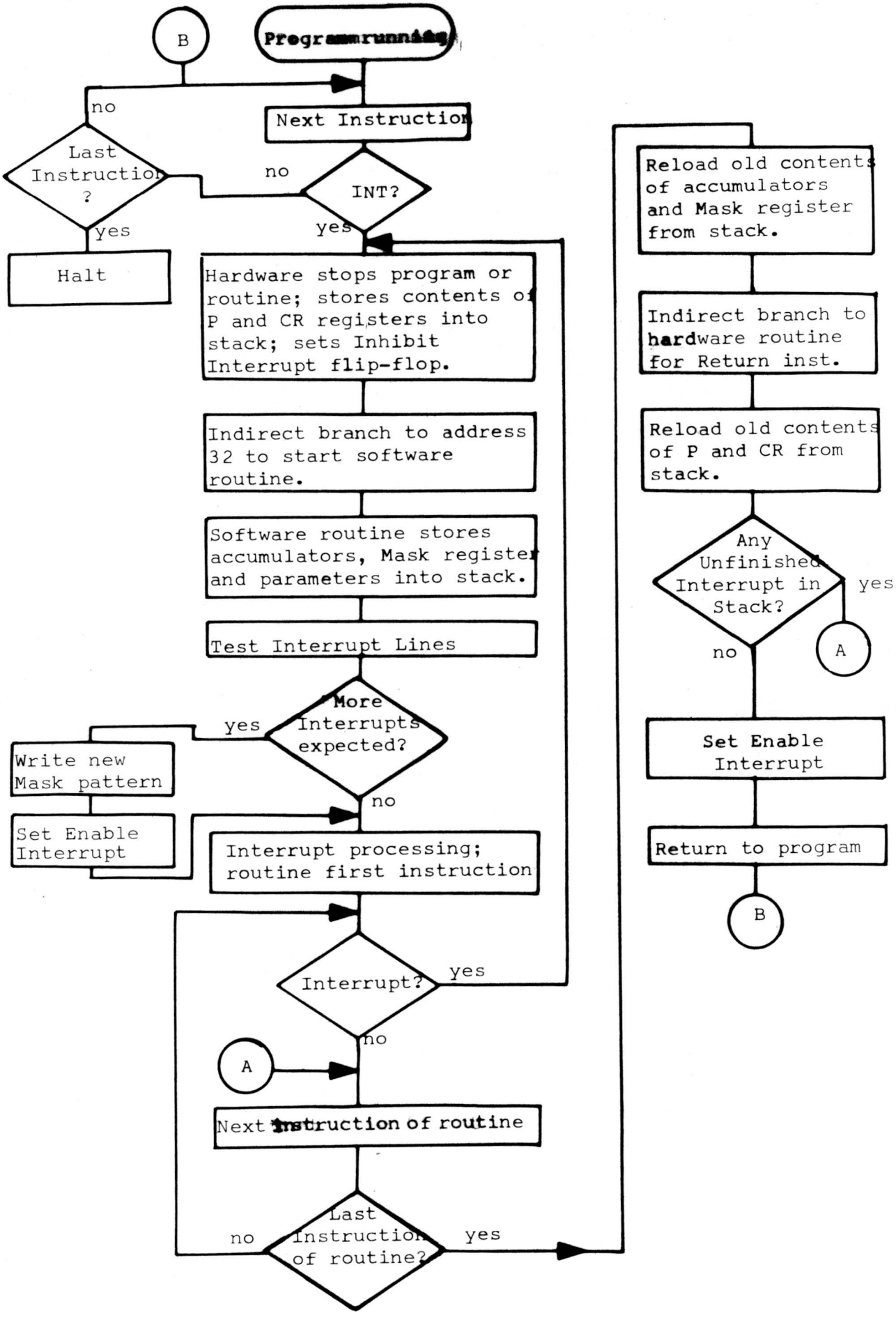


Figure 1.3 Stack handling operation

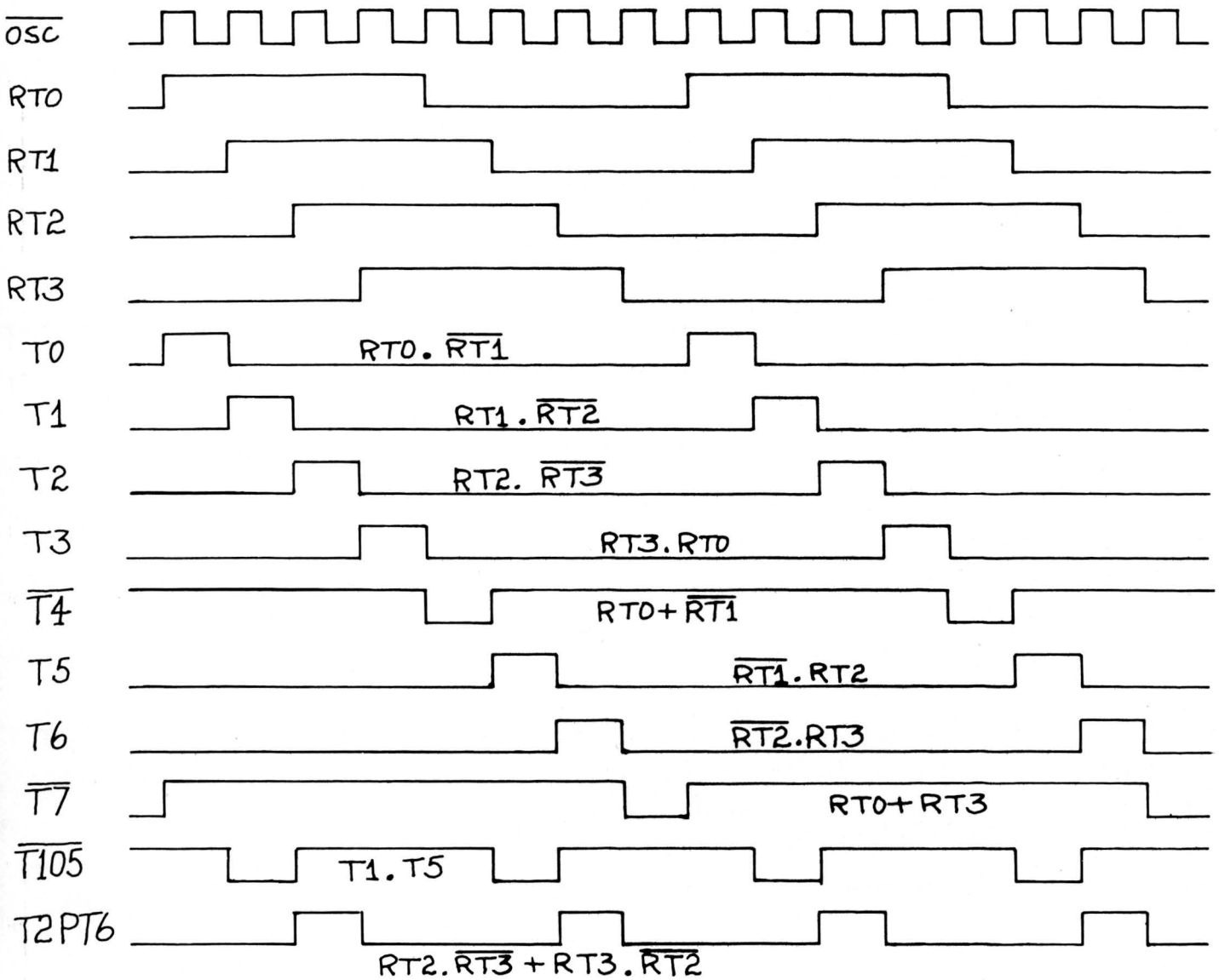
The logic for the basic processor is contained on four circuit cards. These four cards have been drawn full size and are to be found in the pocket on the back cover of this book. To simplify servicing and fault finding, these large diagrams have been broken down into individual circuits and descriptions.

In the following diagrams, the circuit card logic has the letters T, M, A and C prefixing individual diagram numbers. These letters correspond to circuit boards TIBA, MEI, AUR and CONT in the Boolean book.

To assist in using these diagrams, a list of signal names and abbreviations will be found in Part 5 of this book.

LOGIC DIAGRAMS

Logic diagrams (continued)



H 0 = RT0
H1 = RT0

| T0 | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T0 | T1 | T2 | T3 | T4 | T5 | T6 | T7 |

Figure T1 BASIC TIMING

The basic timing circuit consists of a 4-bit ring counter clocked by the oscillator signal (OSC) and reset by the master clear (MCLT1).

Outputs from the counter are decoded to give the timing signals. The equation deriving each signal is given on the timing diagram.

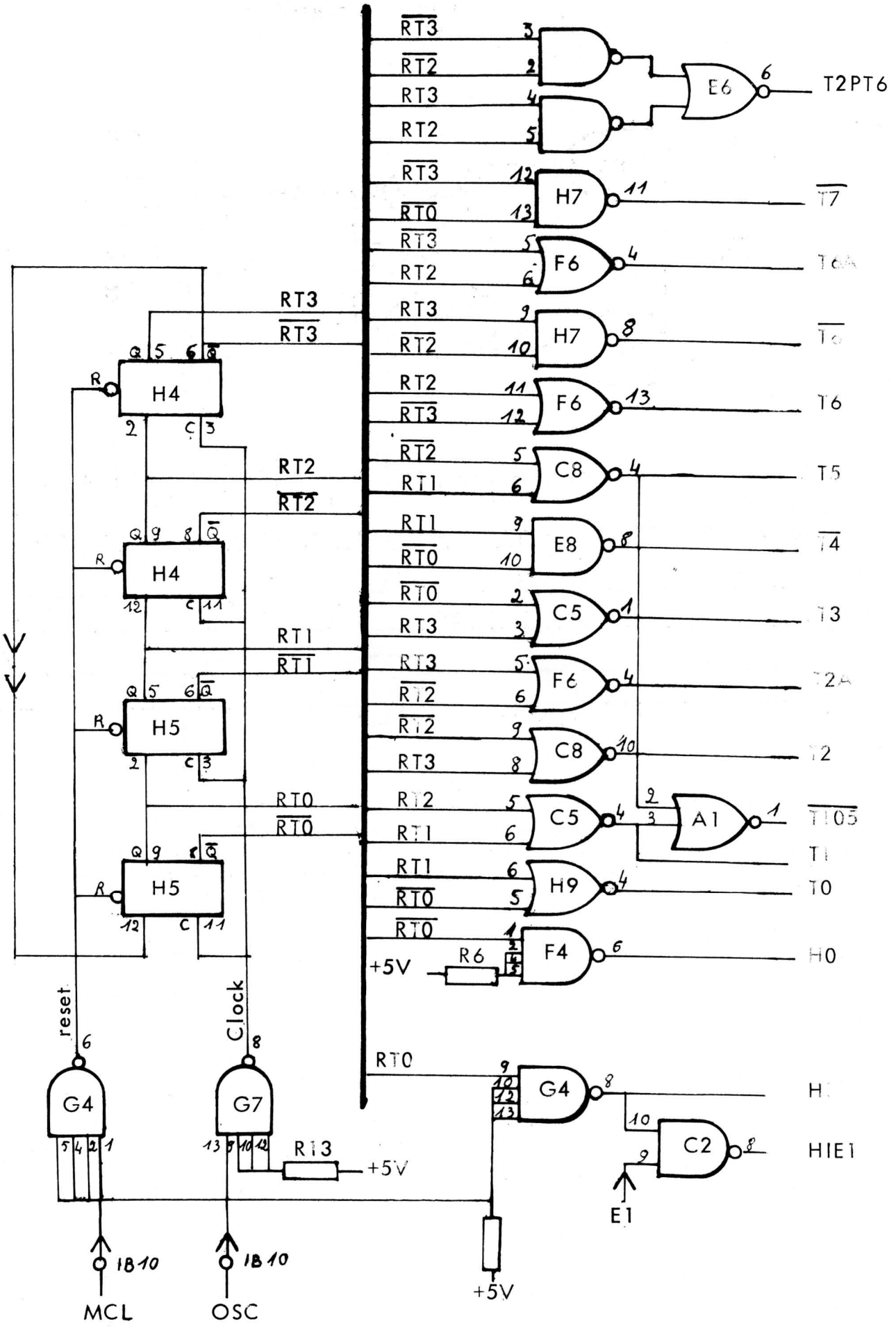
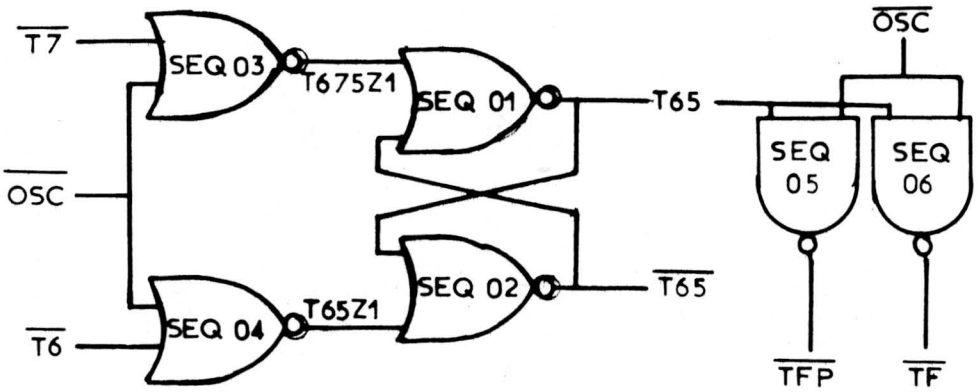
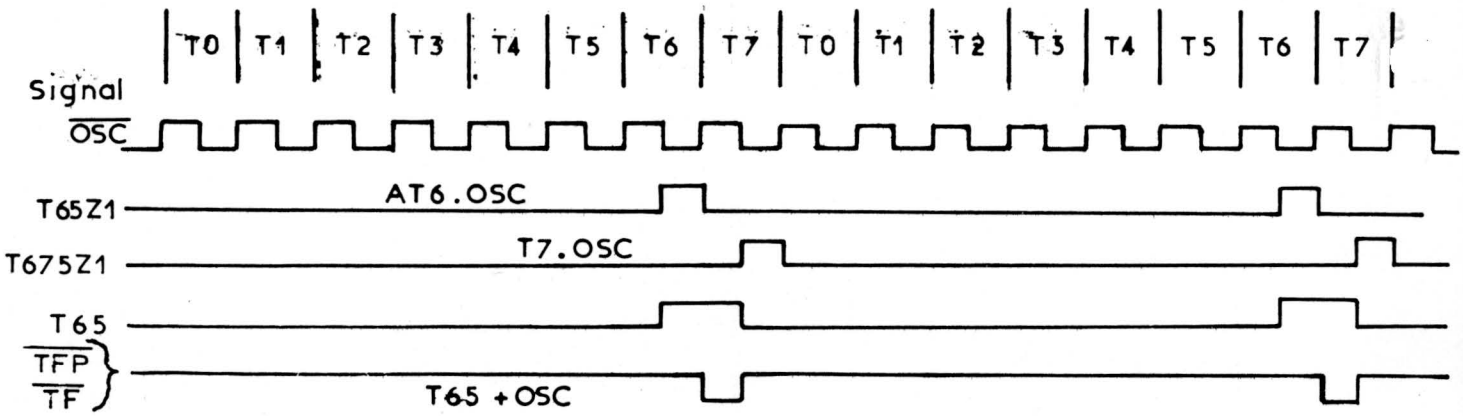


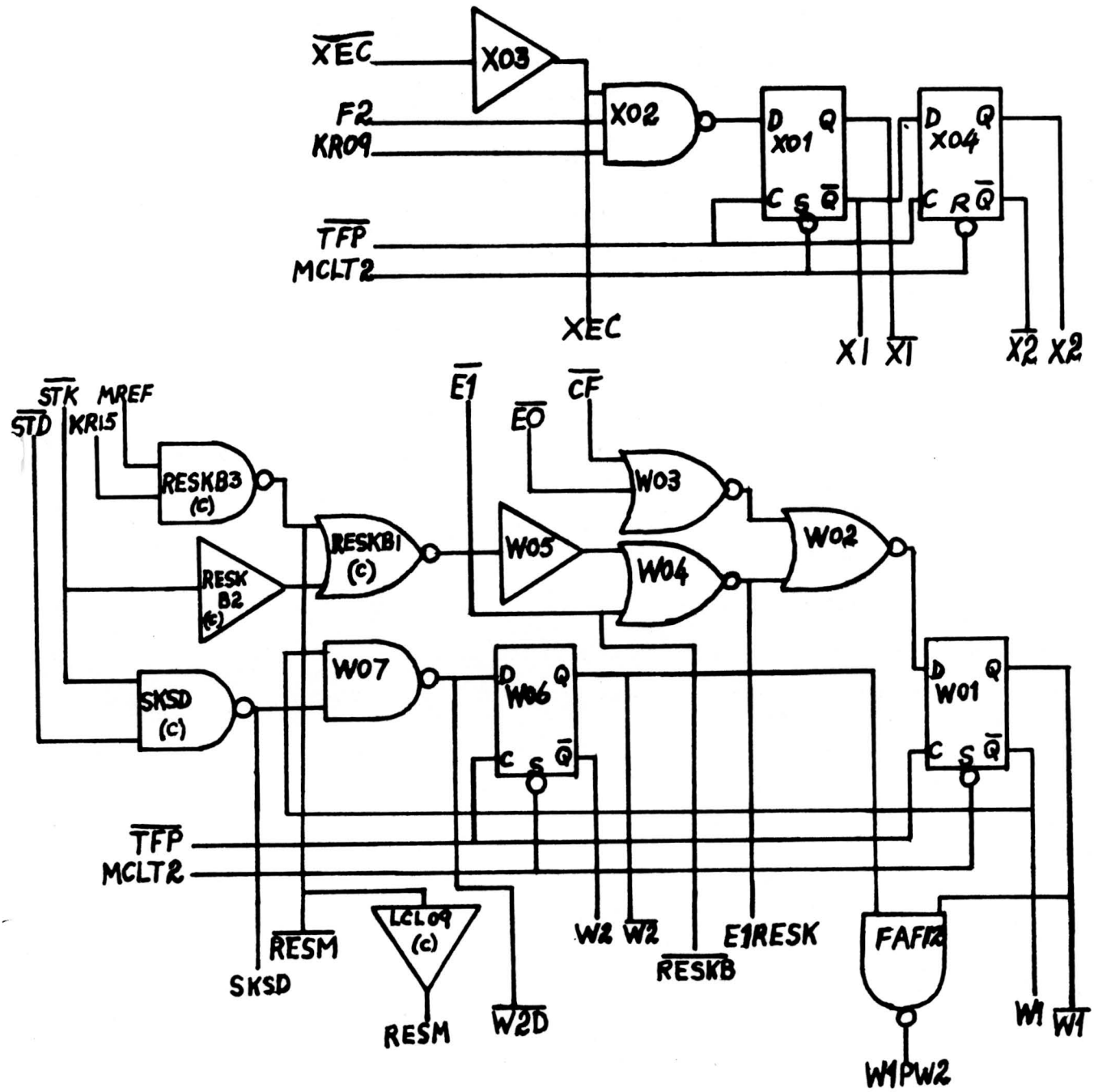
Figure T2 BASIC TIMING

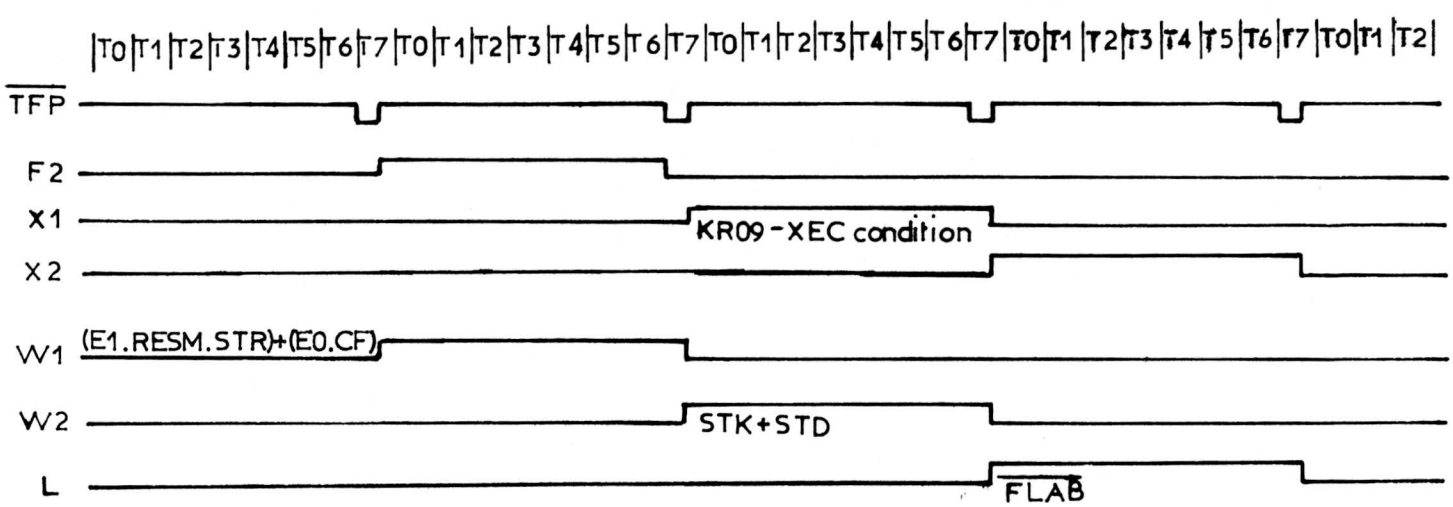




The flip-flop made of NOR gates SEQ01 and SEQ02 is set and reset by clock signals T6/ and T7/ gated by OSC/ and the flip-flop output is in turn gated by OSC/. The outputs TFP/ and TF/ give a positive-going transition between T7 and T0 which is used to clock the instruction sequence flip-flops.

X AND W TIMING PULSES

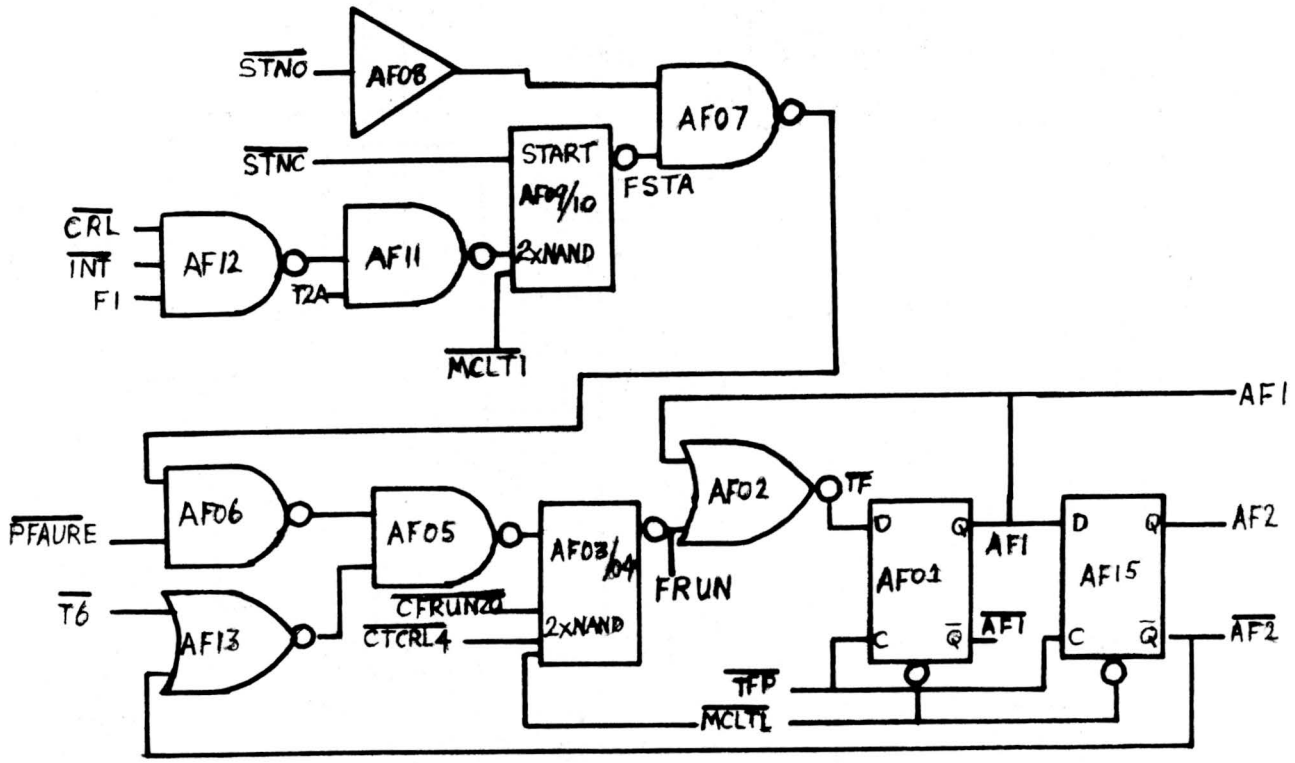


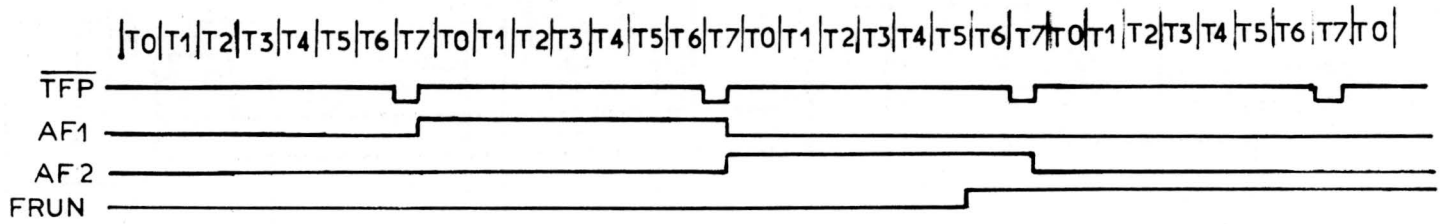


(Page T 12)

Flip-flops XO1 and XO2 clocked by TFP/ give the first and second index cycles following the fetch cycles. W1 and W2 are first and second write cycles. W1 may follow the EO or E1 cycles. W2 may follow only the W1 cycle.

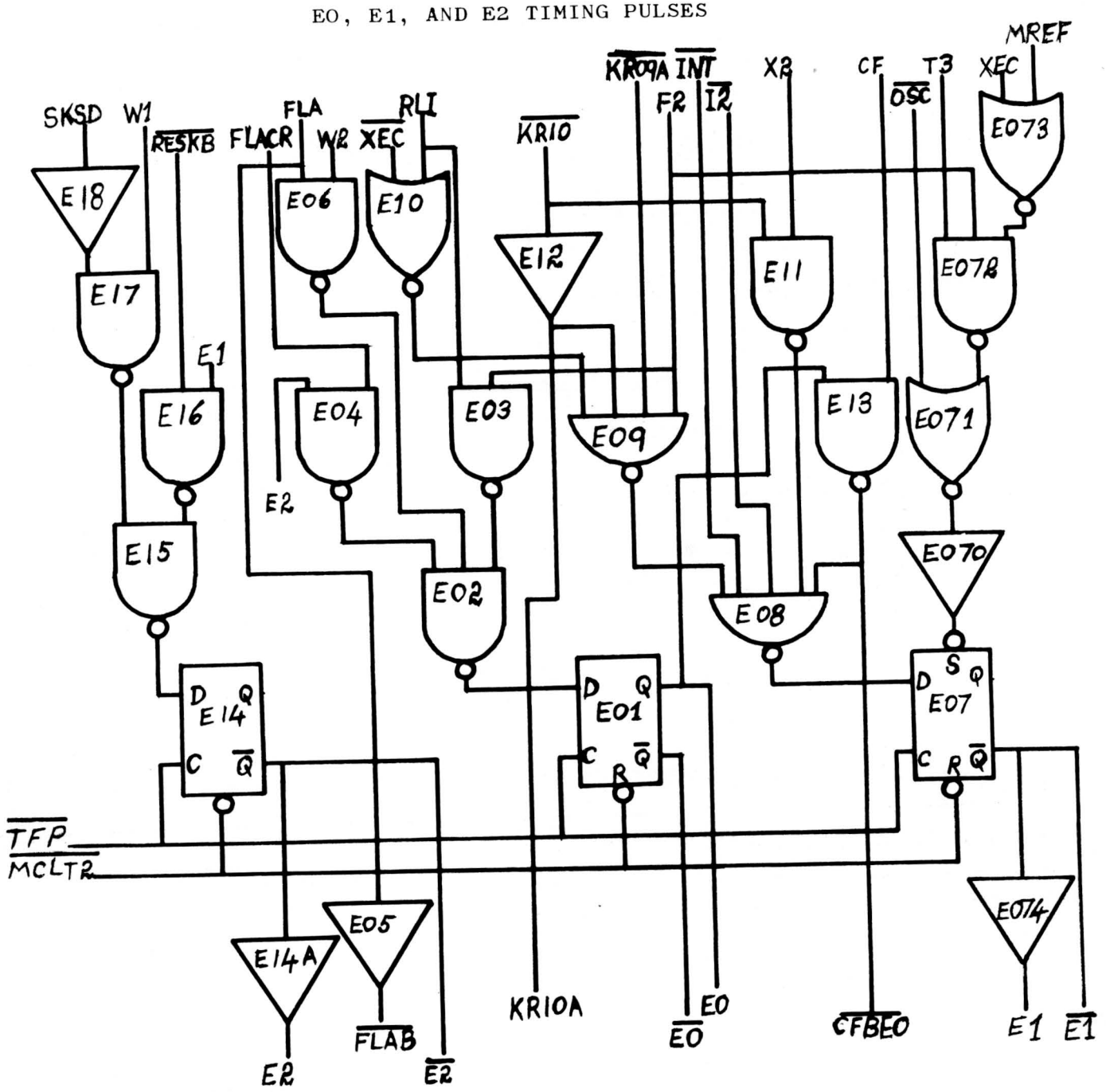
AF1 AND AF2 TIMING PULSES

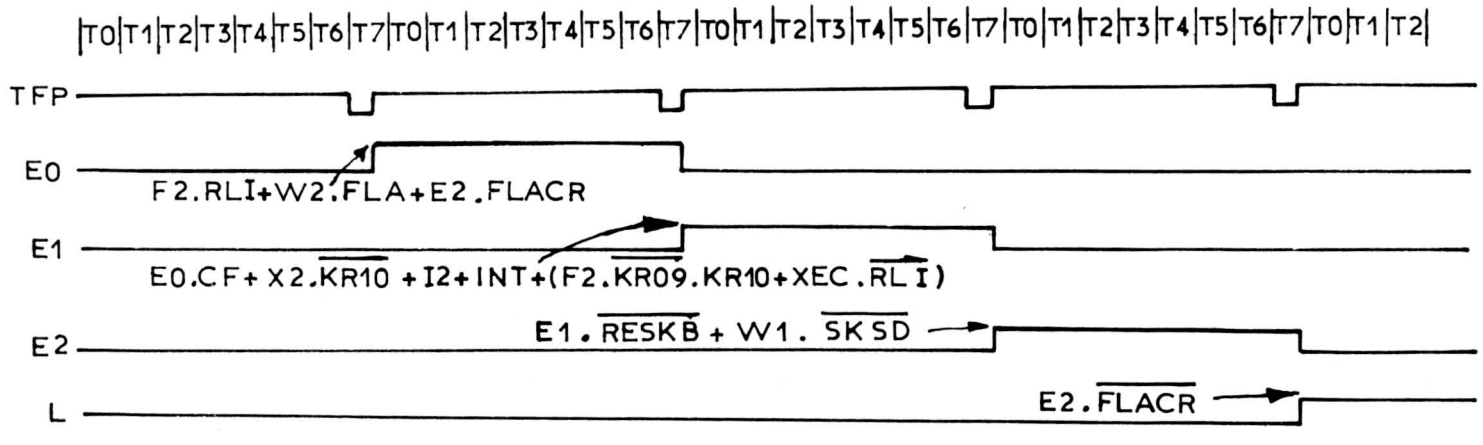




AF1, AF2 give a delay of two operation cycles timed by TFP/ after the machine is started before FRUN goes high. The FRUN flip-flop, AF03/AF04 is set by AF2.T6. FRUN set is inhibited by a power failure condition if the power failure option is fitted and by the START flip-flop, AF09/AF10, in the reset condition. In manual operations, AF1 and AF2 are used as operation cycles.

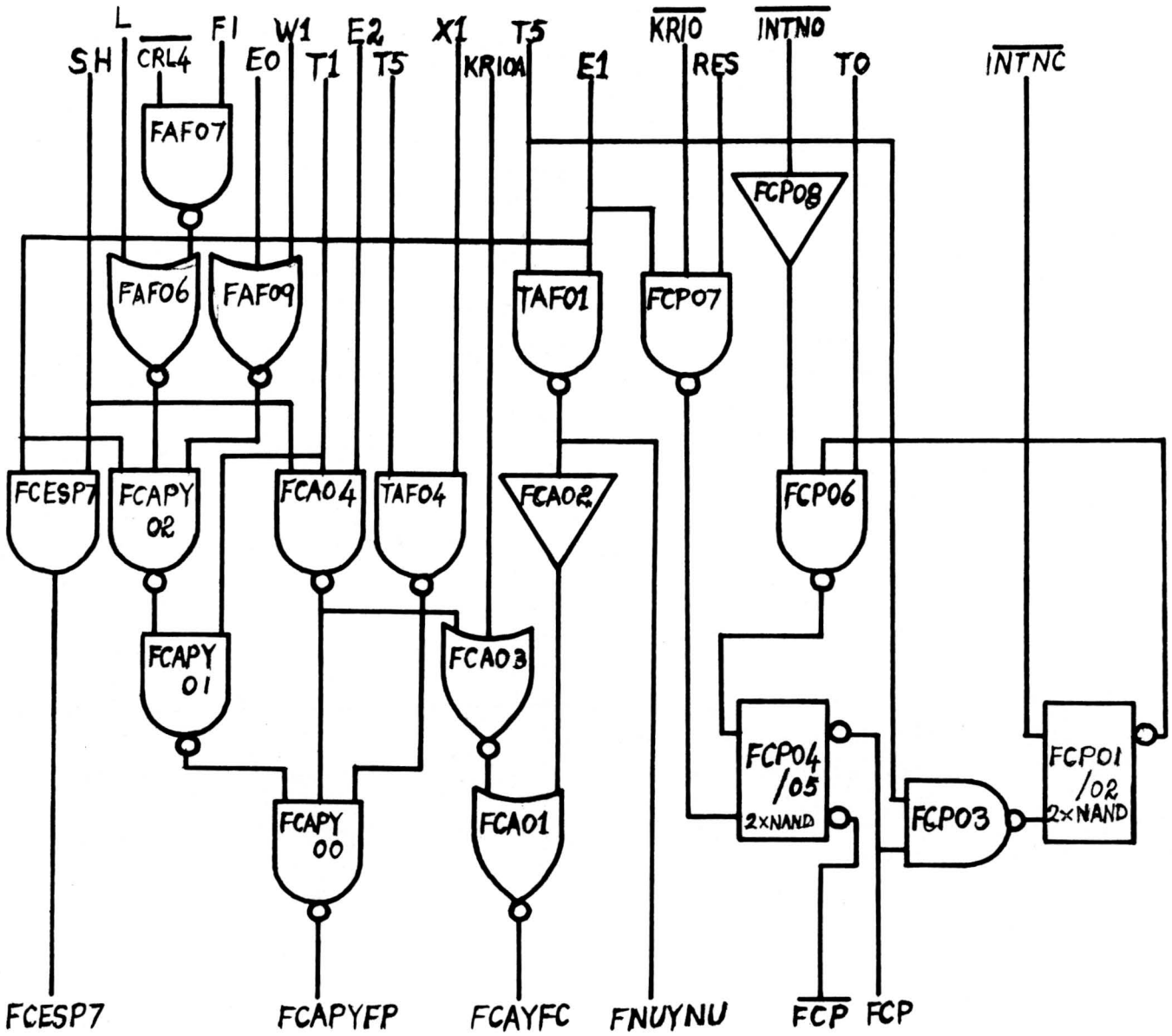
EO, E1, AND E2 TIMING PULSES





E1 and E2 are the first and second execute cycles. EO is the execute CF or RT cycle.

AU CONTROL PULSES



FCESP7 is used by shift right instructions. It is used as an input to an AND gate on diagram A7 which is associated with the AU03 flip-flop. On the timing diagram E1 it is represented as level FC which is present from T0 to T7 time.

FCAPYFP is used when the program counter is updated and during AU operations which may produce a carry bit. It occurs at T1 time during the F1, E1, E2, E0, W1, L and RL4 cycles and at T5 time during the X1 cycle, and clocks flip-flop AU04.

FCAYFC is used by shift instruction to clock flip-flop AU03. It occurs at T5 time during the E1 cycle and at T1 during the E2 cycle.

FNUYNU is used to clock the CR17 flip-flop on diagram A3. It is produced at T5 time during the E1 cycle and indicates that the result of the AU operation is NUL (A = B).

FCP is produced when the INT button on the control panel is pushed.

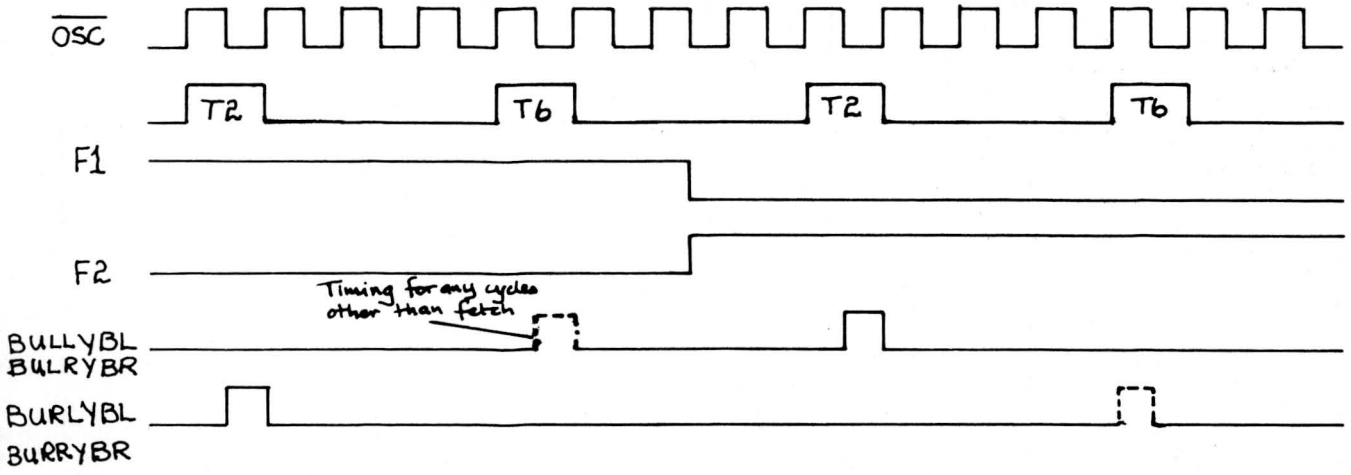


Figure T8a AU CARRY INPUT & BU REGISTER CONTROL

CINEFP reads the carry signal (FCAP diagram A7) into the arithmetic unit carry input.

BULLYBL and BURRYBR are clock inputs to the 16-bit BU register to load either BU right or BU left from the 8-bit data bus. The signals are strobed by OSC at T2 and T6 in the F1 and F2 cycles and at T6 in manual operation.

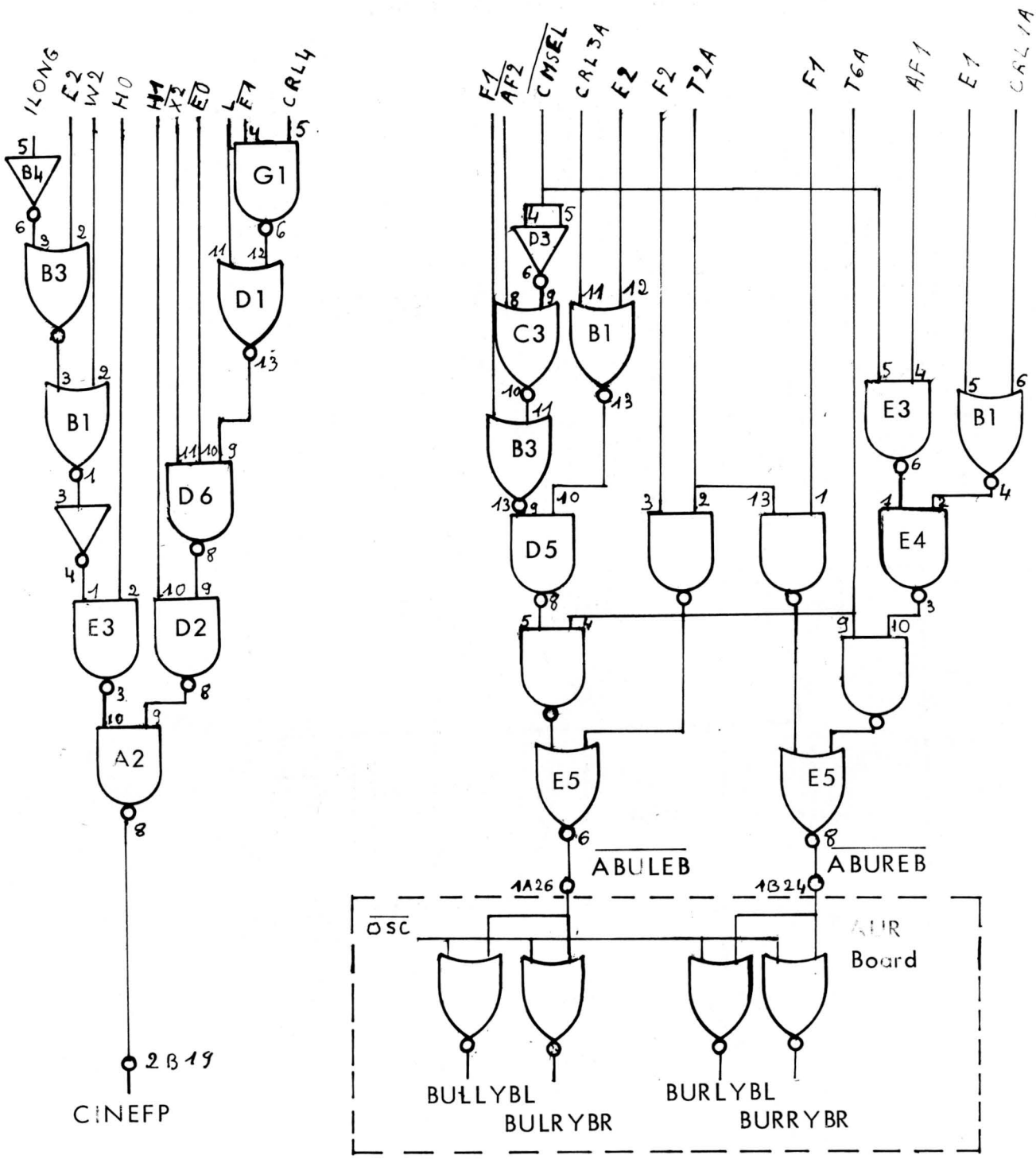
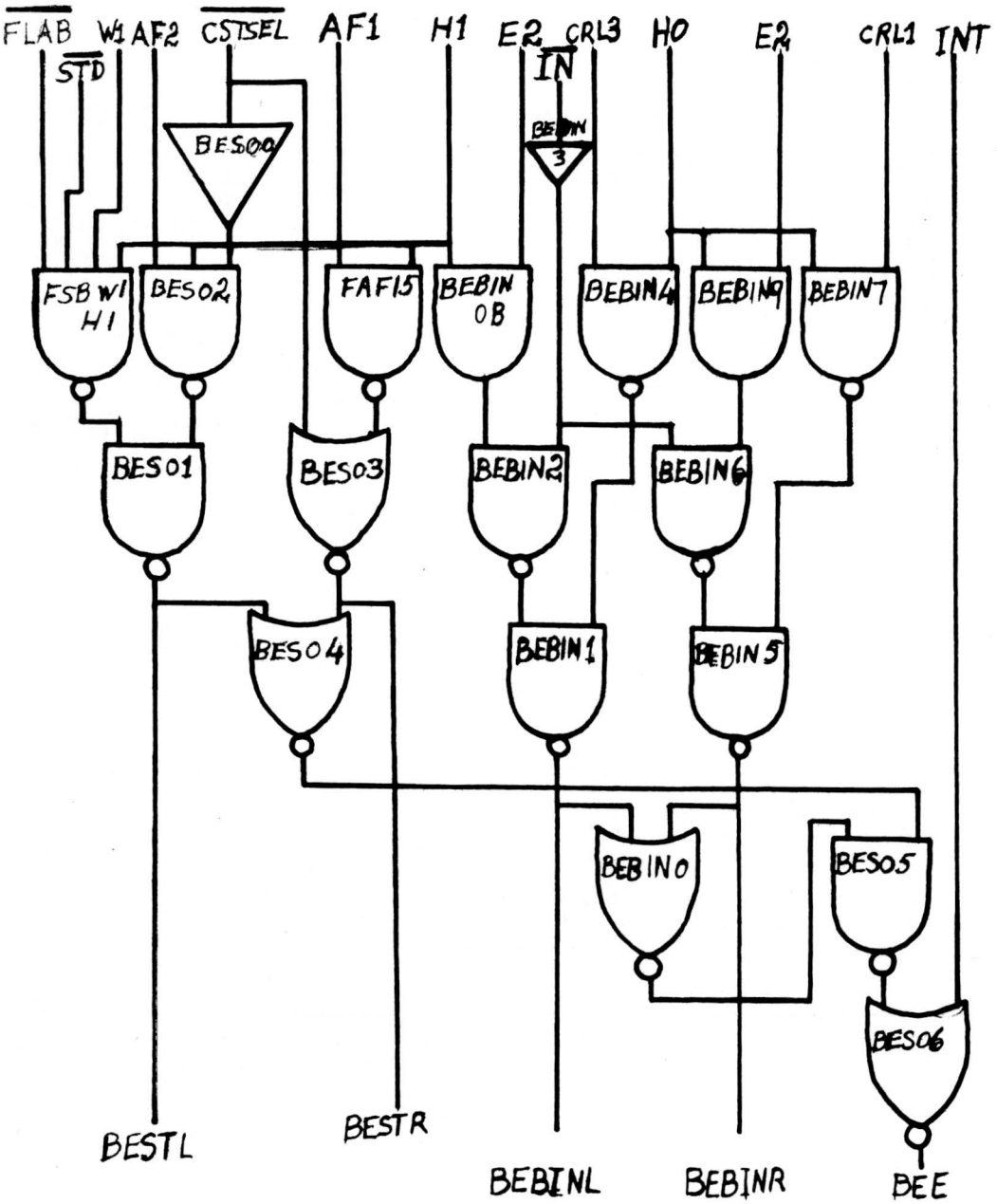


Figure 18 AU CARRY INPUT AND BU REGISTER CONTROL

B GATE CONTROL PULSES



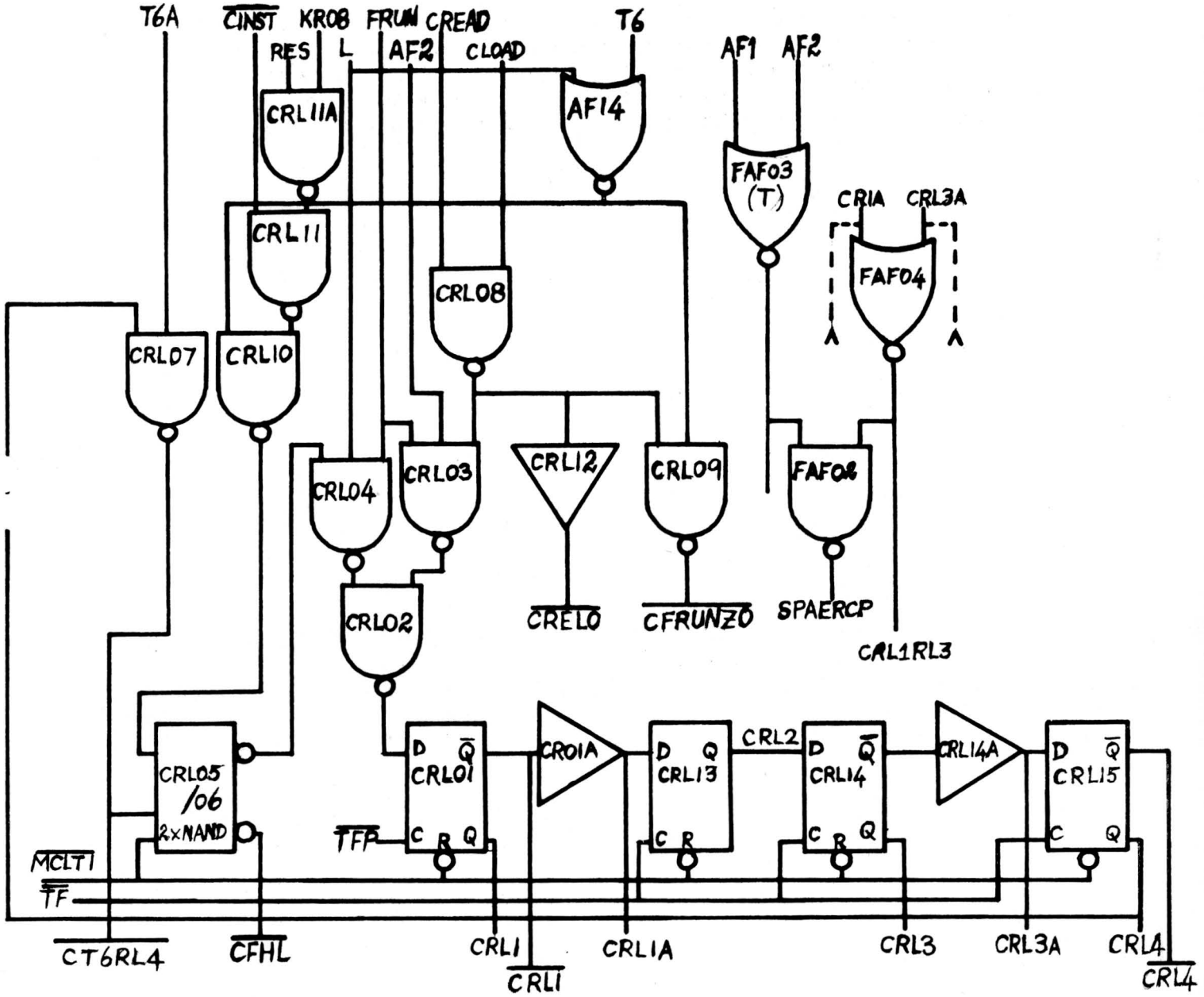
BEBINR gates BIN lines 08 to 15 on to the data bus during HO time and BEBINL gates BIN lines 00 to 07 on to the data bus during H1 time of the E2 cycle in a register input routine.

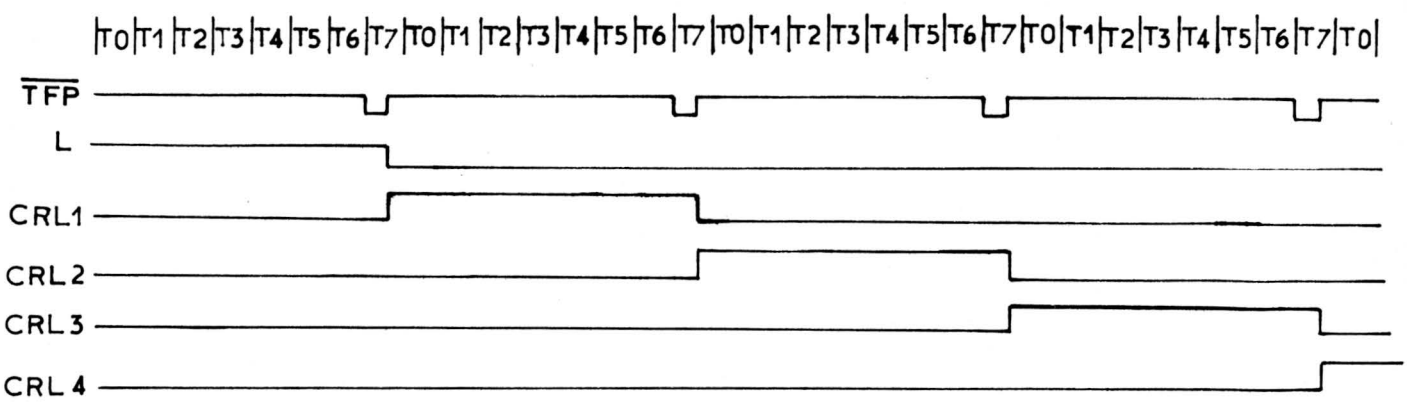
BESTL gates the condition register contents on to the data bus during H1 time in the W1 cycle.

BESTR gates the control flip-flop contents or to the data bus in manual operation at H1.AF1.

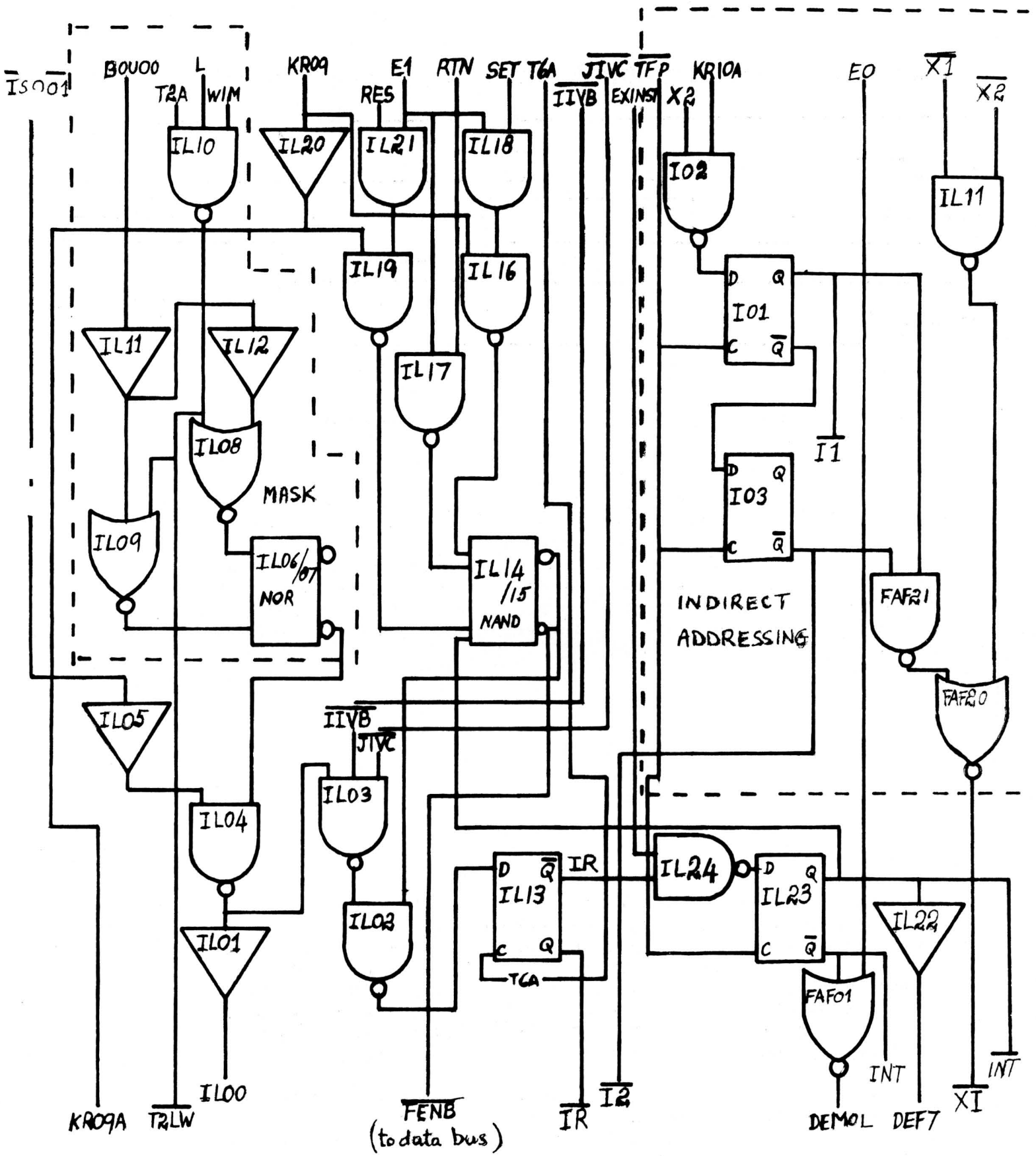
BEE gates the E register on to the data bus except when one of the other gating signals (BEBINR etc) is active or when INT is active during an interrupt routine.

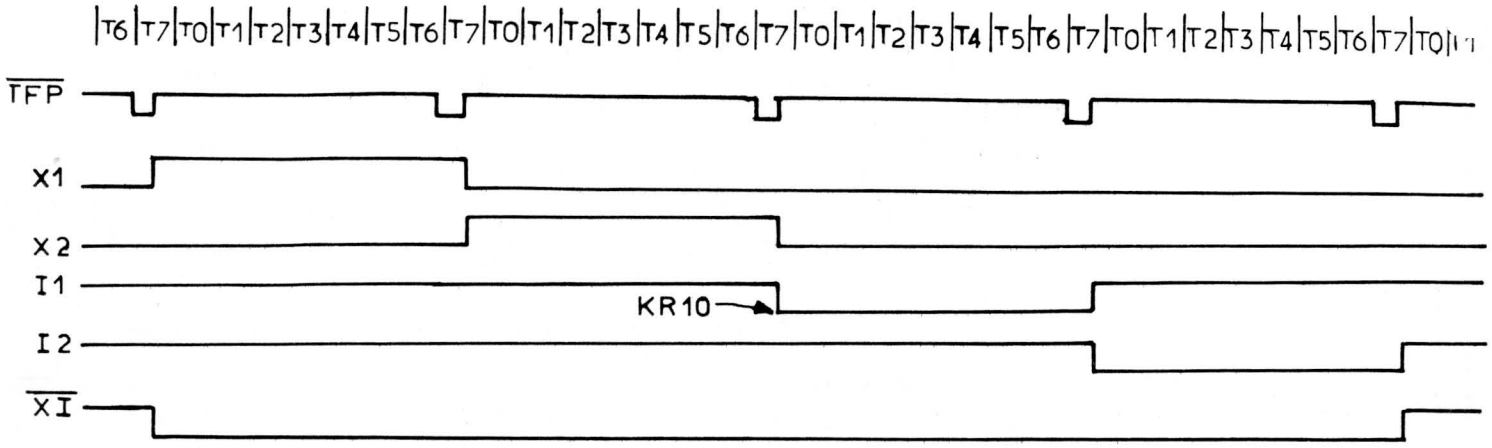
CONTROL PANEL TIMING SEQUENCE





CRL1 and CRL3 are used to gate the control panel inputs on to the data bus during manual operation. CRL1 gates keys 00 to 07 and CRL3 gates keys 08 to 15. CRL4 is combined with T6 and $\overline{OSC}$ to clock the S register thus loading the S register at T6 in CRL4 cycle (see logic on diagram C9).


ISOO1
BOUOO
L
T2A
WIM
KR09
E1
RTN
SET
T6A
JIVC
TFP
KR10A
EO
X1
X2
RES
IIVB
EXINST
X2
IL10
IL20
IL21
IL18
IO2
IL11
IL19
IL16
IO1
D
Q
C
Q
IL11
IL12
IL17
I1
IL08
MASK
IO3
D
Q
C
Q
IL09
IL06/07
NOR
IL14/15
NAND
INDIRECT
ADDRESSING
FAF21
FAF20
IL05
IIVB
JIVC
IL03
IL04
IL02
IL01
IR
IL24
IL13
IL23
IL22
TCA
FAF01
INT
INT
XI
IL00
KR09A
T2LW
FENB
(to data bus)
IR
I2
DEM0L
DEF7

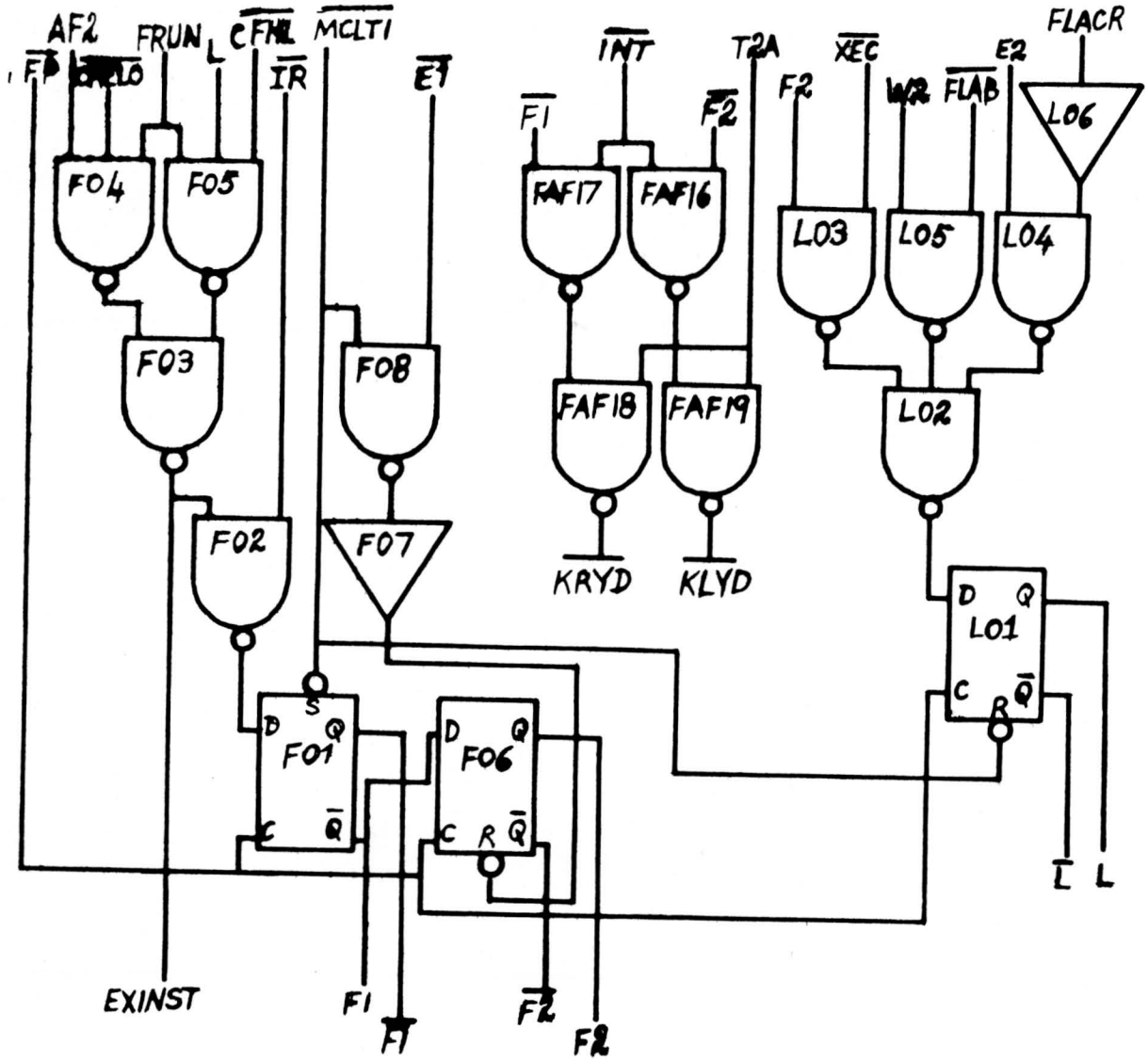


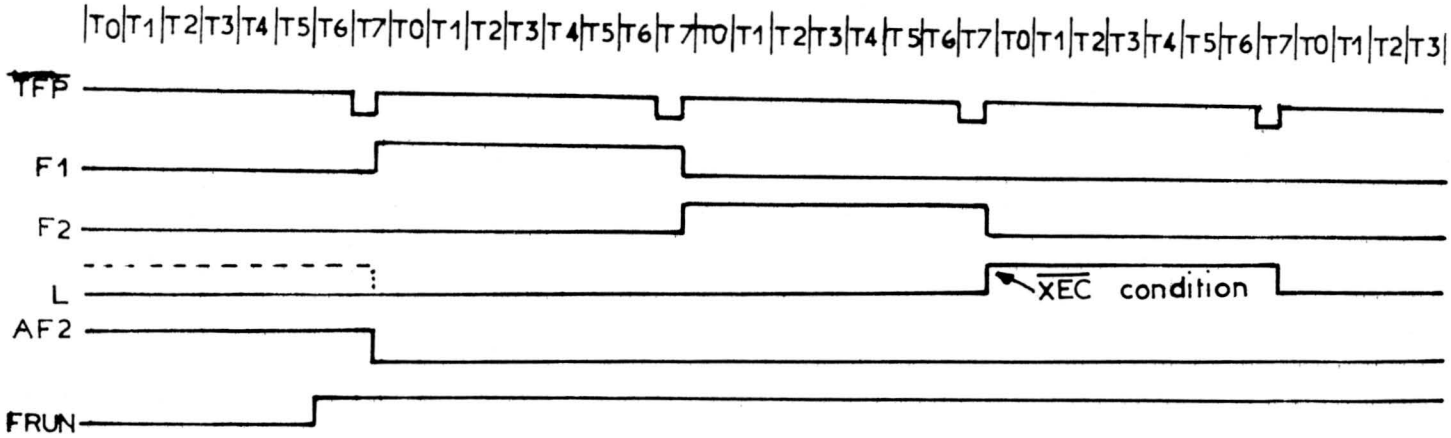
Indirect address cycles, I1, I2, follow X2 when KR10=1.
I2 is then followed by E1.

The mask flip-flop (NOR gates IL06 and IL07) is used in conjunction with mask flip-flops in the additional interrupt and mask option. Each flip-flop is set or reset by a BØU line (in this case BØU00) followed by a Write Interrupt MASK instruction. The instruction code, WIM, is gated with T2A and L in IL10 and with BØU00/ in IL08 and IL09 to set and reset the mask flip-flop. Interrupt line IS00/ is gated by the flip-flop output in IL04.

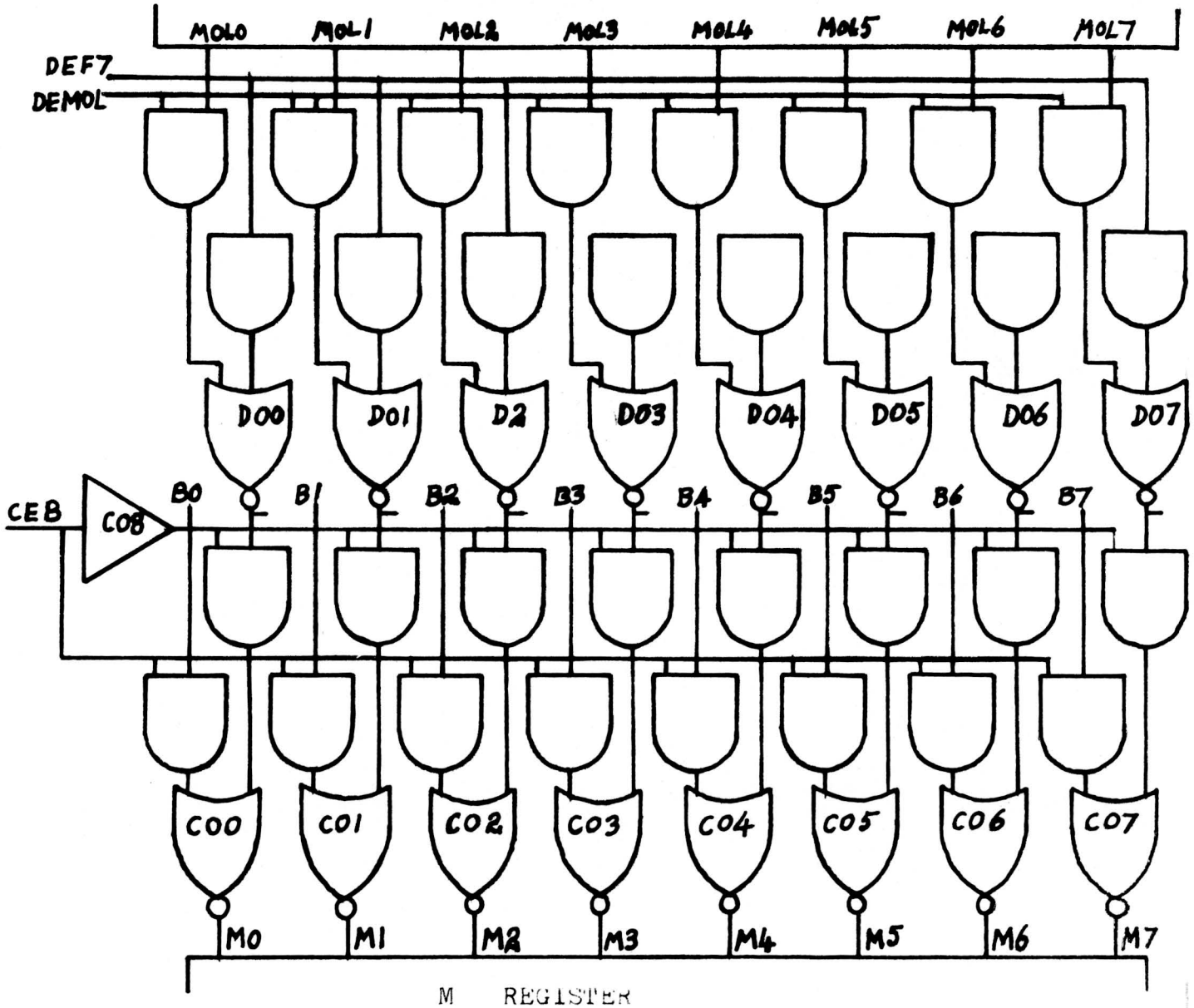
IL14/15 is the function flip-flop that, when reset, inhibits all interrupts and provides the code on the data bus which, when loaded into the S-register, addresses memory location 32. IL14/15 is set by the EMABCR INTERRUPT instruction at E1 (KR09.E1.SET) gated in IL18 and IL16, or by the RETURN FUNCTION instruction at E1 (E1.RTN) gated in IL17. The flip-flop is reset by INHIBIT INTERRUPT at E1 ($\overline{\text{KR09}}$.E1.RES) and by $\overline{\text{INTV}}$ when an interrupt has been accepted from flip-flop IL23.

IL13 is set at T6 following an interrupt request if IL04 and IL02 are enabled by the mask flip-flop and IL14/15. IL23 is then set when EXIMST enables IL24 and when clocked by TFP/ thus setting IL14/15, inhibiting IL02, so IL13 is reset at T6. TFP/ then clocks IL23 to end the interrupt cycle.





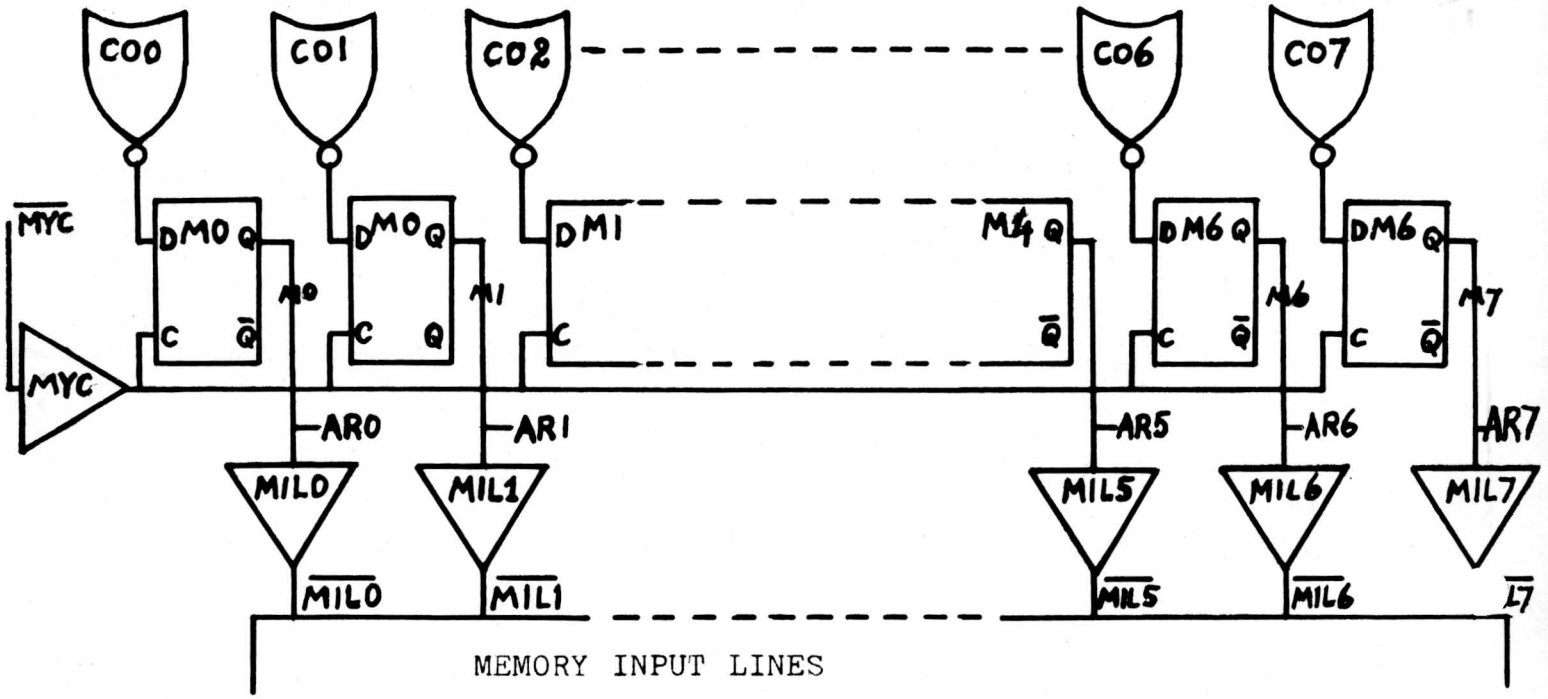
F1 and F2 enable the first and second fetch cycles. F1 is clocked by $\overline{TFP}$ when AF2 is high (first cycle after START) or when L is high (L is last cycle of the previous operation). L enables the last cycle of every operation and may follow F2, W2 or E2, depending upon the type of operation and determined by the condition of gates L03, L05 and L04.



DEF7 is produced on diagram T11 and is used during interrupts to force hexadecimal value F7 into the K-register via the D-gates.

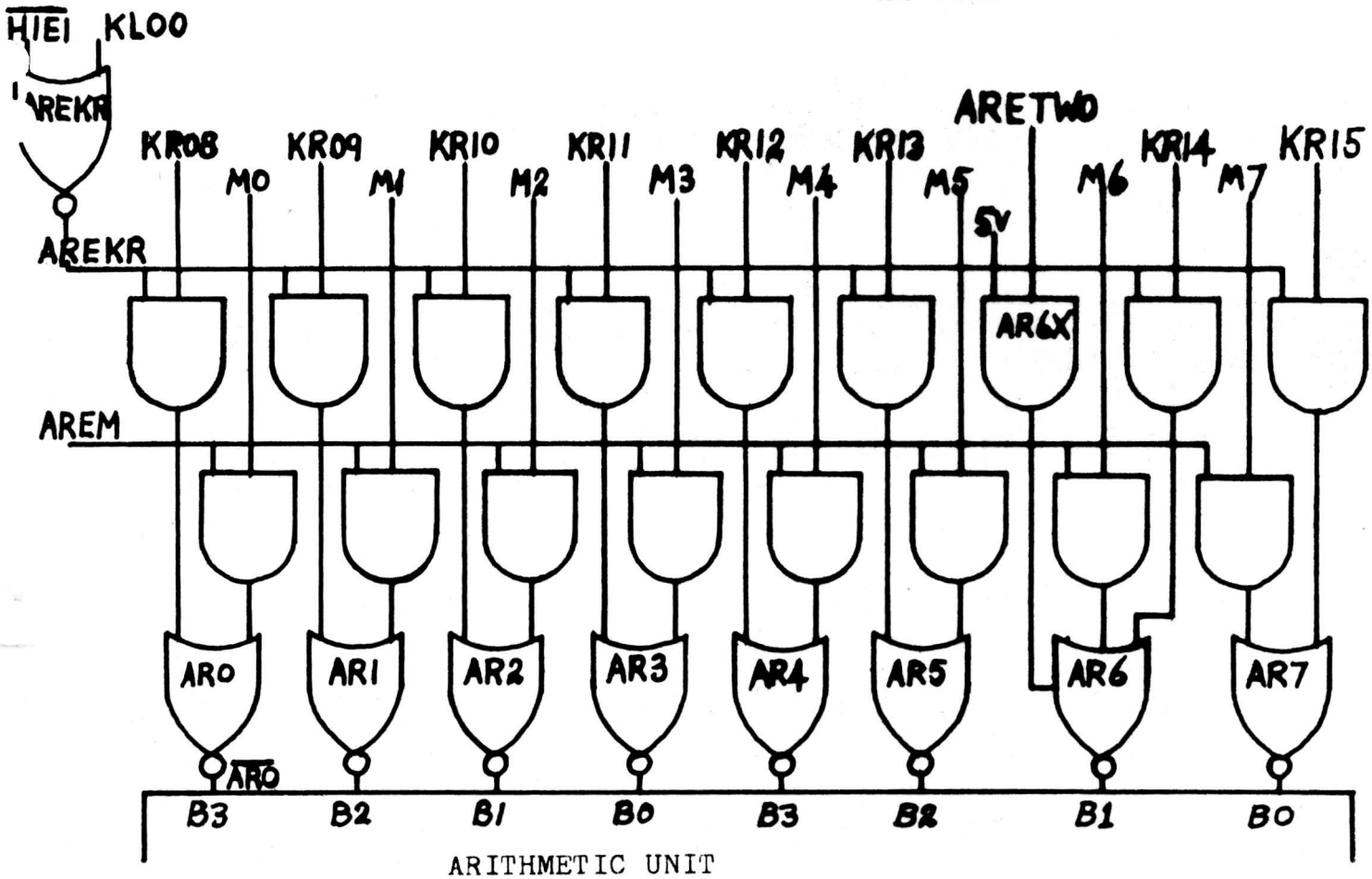
DEMOL is also produced on diagram T11 and is used during normal read-out operations from the memory.

CEB is produced on diagram C5 and is used to enable the contents of the C-gates into the M-register. The inputs to the C-gates will be from either the D-gates or the B-lines.



$\overline{\text{MYC}}$ is produced on diagram C6 and is used to clock the contents of the M-register into the memory during load operations.

MEMORY and KR REGISTERS TO ARITHMETIC UNIT
AR GATE

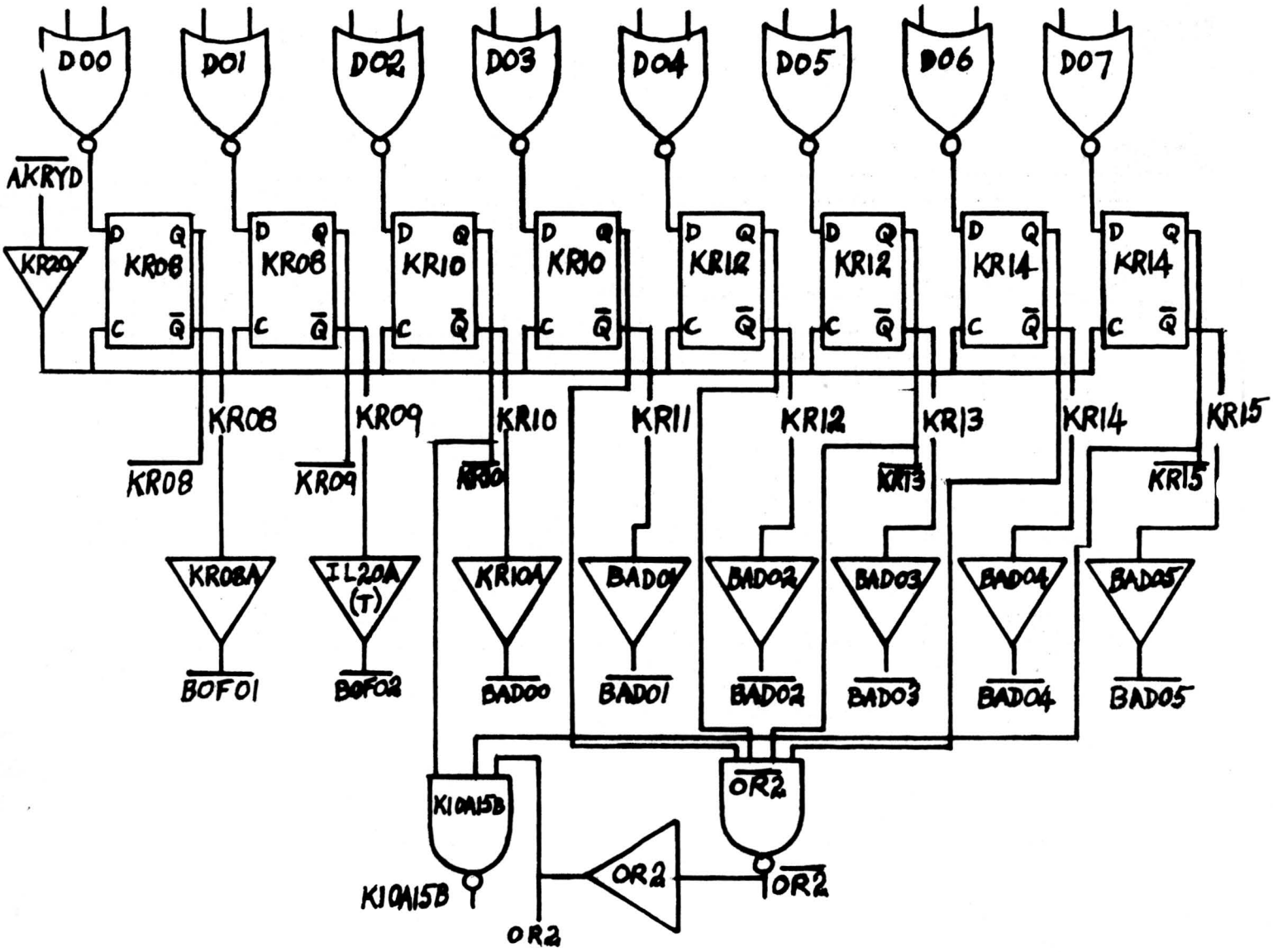


AREM is produced on diagram C3. When this signal is present it allows the contents of the M-register to be gated into the AU via the AND and NOR elements. It is available from T4 to T7 time.

ARETWO is also produced on diagram C3 and is used during updating to increment or decrement the contents of the P- or stack pointer registers. It is available from TO to T3 time.

AREKR is produced when a short constant (8-bit constants) instruction is being performed. It is available from T4 to T7 time and allows the constant to be gated into the AU. The origin of the input signals to NOR element AREKR will be found on diagrams T2 and M5.

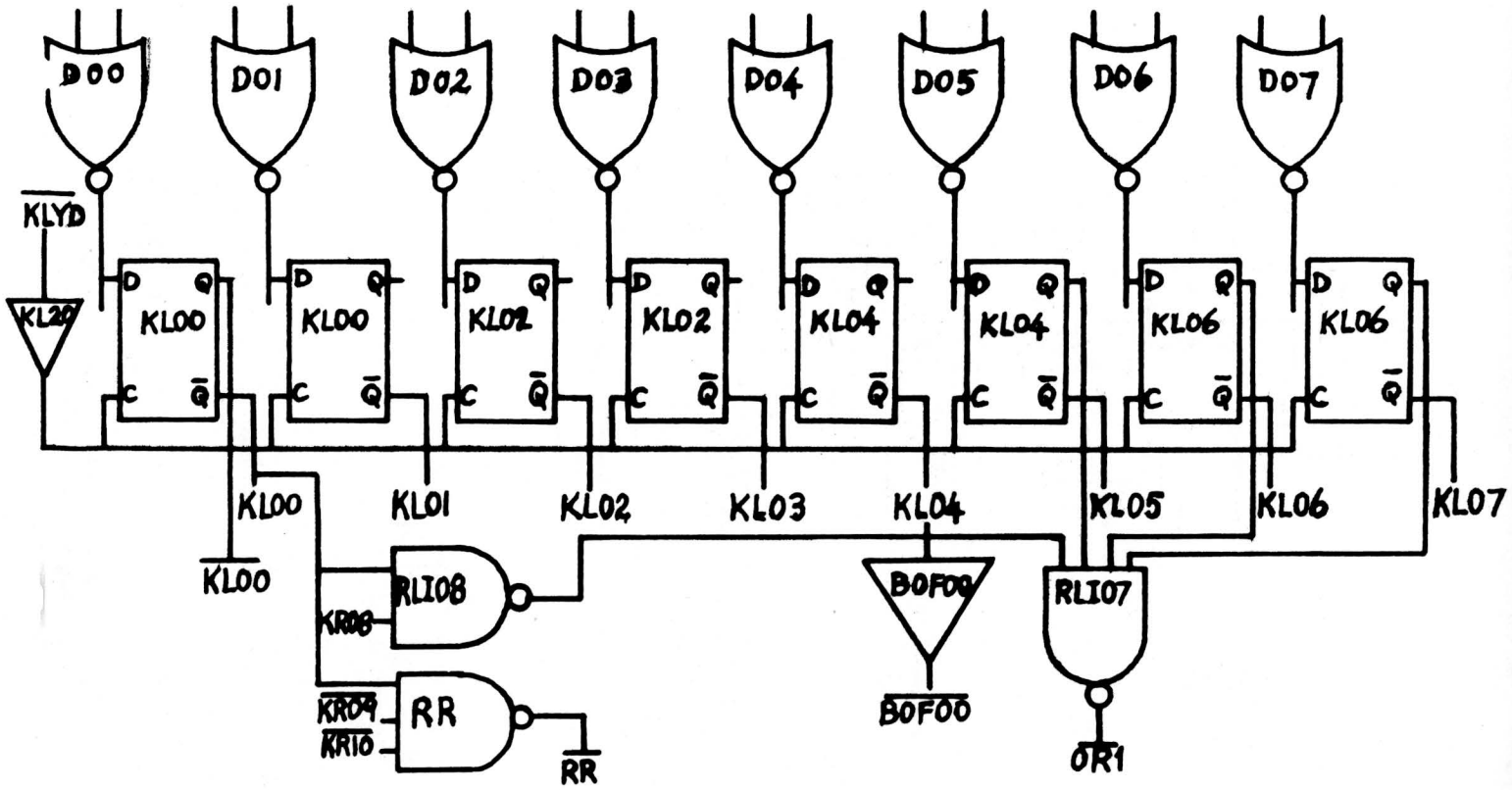
KR REGISTER and BAD LINE DRIVERS



$\overline{\text{BØFO1}}$ and $\overline{\text{BØFO2}}$ are the least significant bits of the function code (used during I/Ø transfers).

$\overline{\text{BADO0}}$ to $\overline{\text{BADO5}}$ are the bits used to specify the peripheral address during I/Ø transfers.

K10A15B is used during RIL instructions.



$\overline{RR}$ is produced when the indirect addressing mode is required by the instruction.

$\overline{BØFOO}$ is the most significant bit of the function code (used during I/Ø transfers to signify the type of transfer).

$\overline{OR1}$ is produced when register 15 is specified in the instruction word.

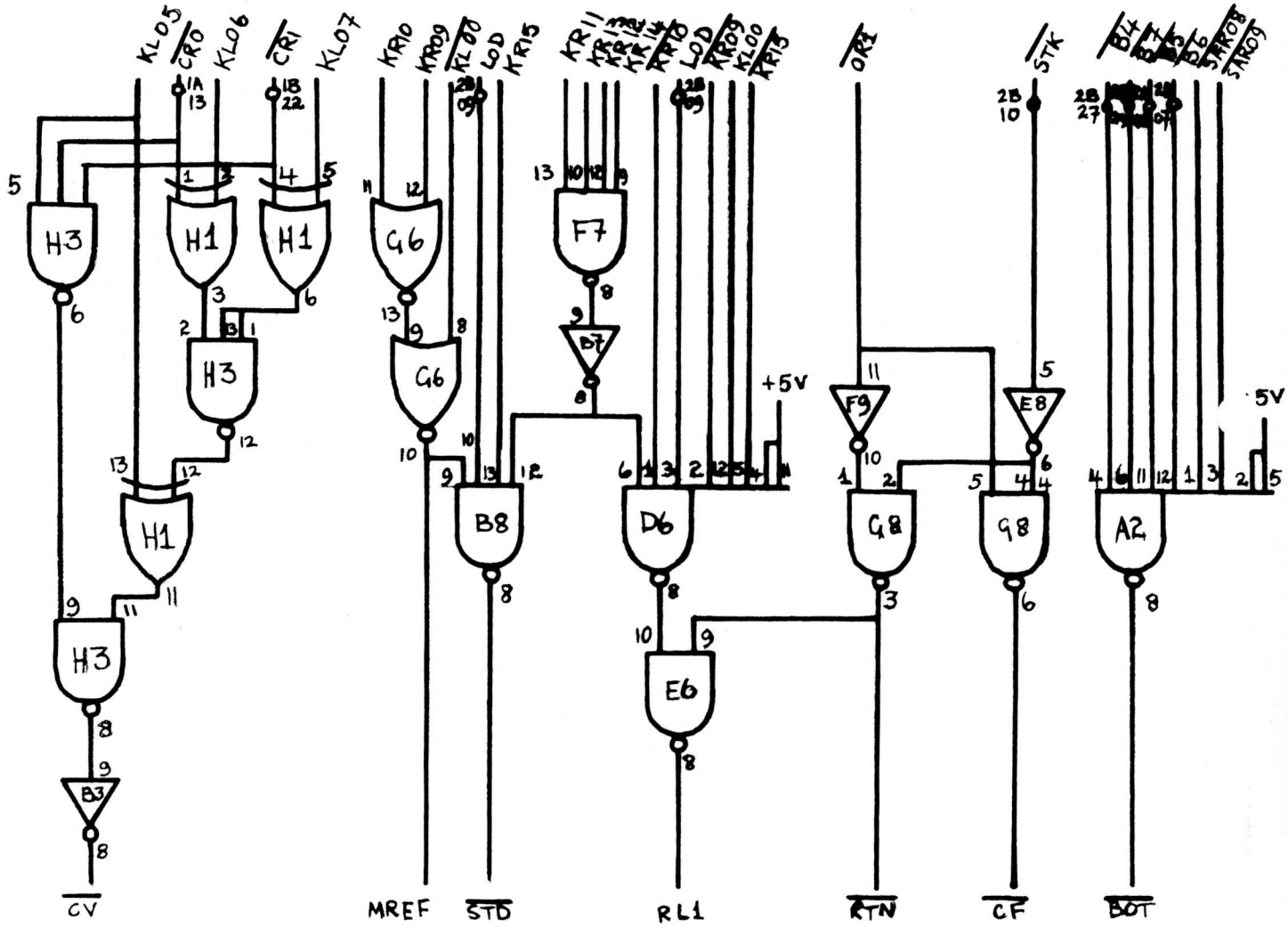


Figure M6 K REGISTER DECODING

$\overline{CV}$ is produced by branch instructions and it is used on diagram C1 to inhibit the branch when the state of the condition register is 111.

$\overline{MREF}$ is produced by memory reference instructions.

$\overline{STD}$ is produced when the W2 cycle is to be used.

BOT is used by the bootstrap logic.

$\overline{CF}$ is produced as a result of STK.$\overline{OR1}$ for the not CALL FUNCTION (see Figure M7)

$\overline{RTN}$ is produced as a result of STK.OR1 for the RETURN FROM FUNCTION (see Figure M7)

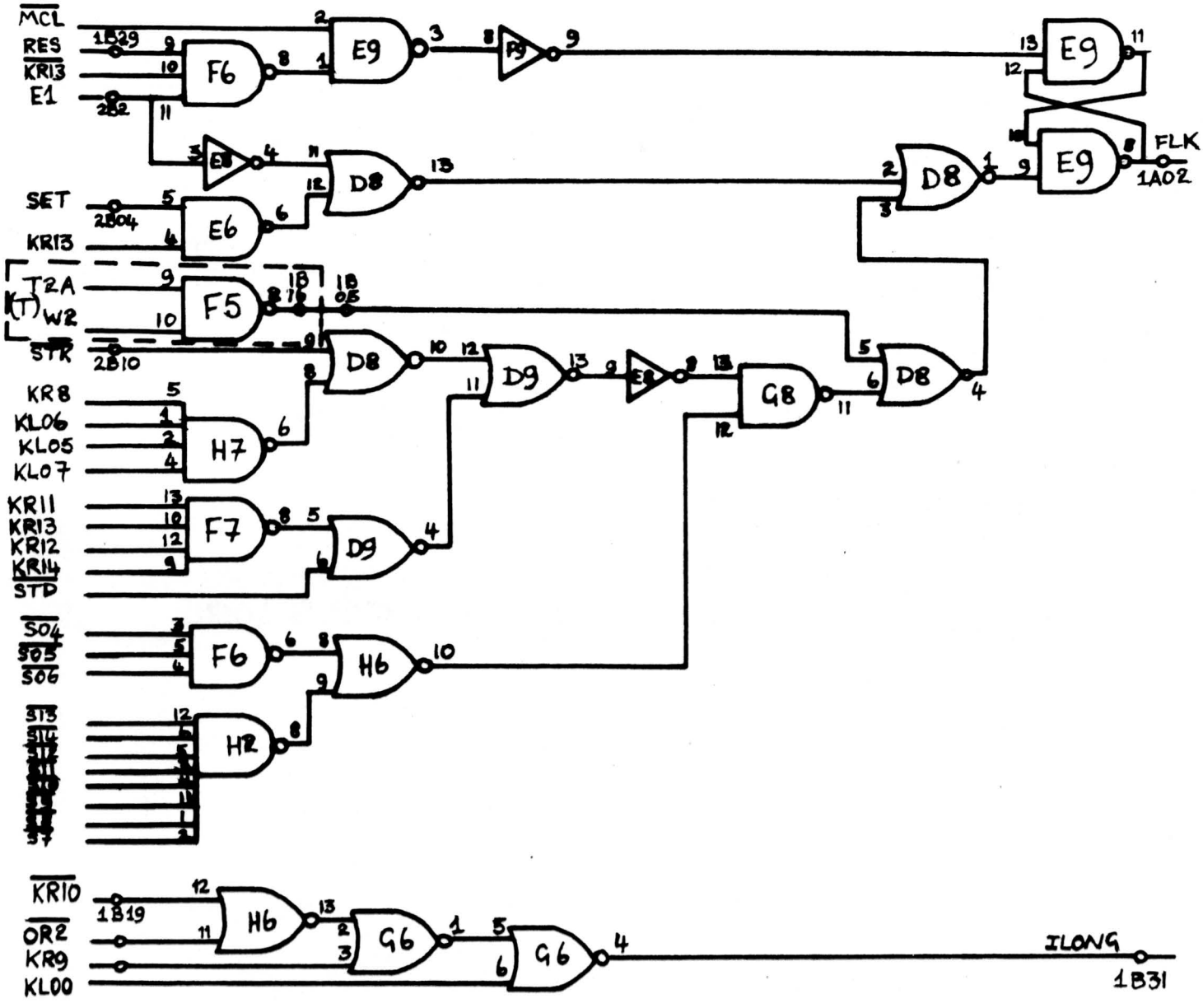


Figure M7a K REGISTER DECODE

FLK is produced when an invalid code has been decoded. Its detection will result in an interrupt being generated.

ILONG is produced when an instruction is contained in two words and is used to update the P and S registers during the L-cycle.

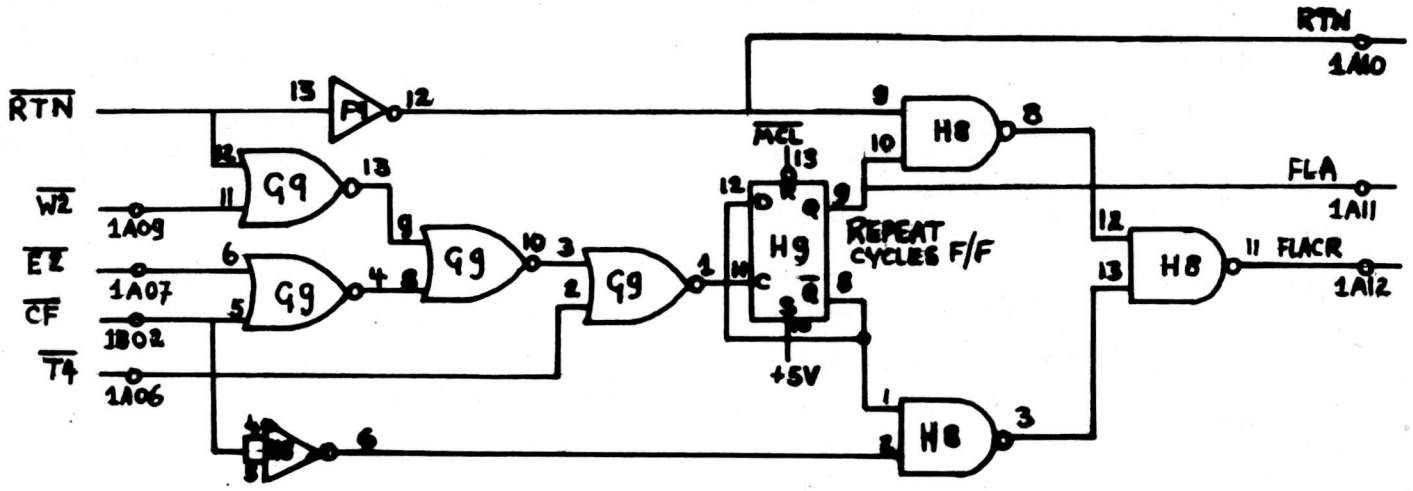


Figure M7c REPEAT CYCLES LOGIC

CF – Not CALL FUNCTION produced as a result of STK.OR1 (see Figure M6)

RTN – Not RETURN FROM FUNCTION produced as a result of OR1.STK (see Figure M6)

FLACR and FLA — Signals used to repeat some of the cycle required by the CALL and RETURN functions. Conditons are:

CALL FUNCTION

FLA.W2 = repeat E0 cycle etc.

$\overline{\text{FLAB}}$.W2 = start L cycle next.

RETURN FROM FUNCTION

FLACR.E2 = repeat E0 cycle etc.

$\overline{\text{FLACR}}$.E2 = start L cycle next.

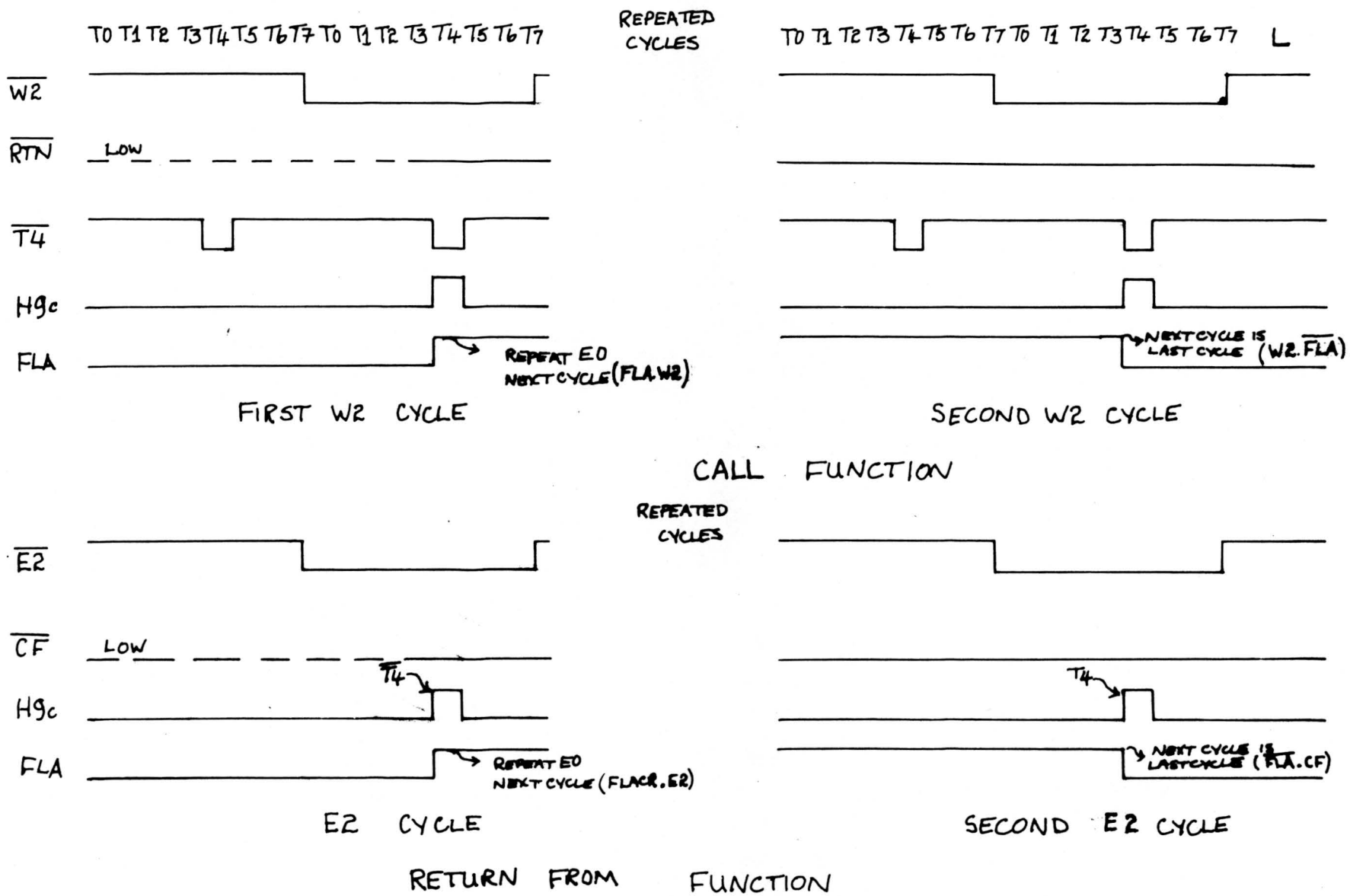
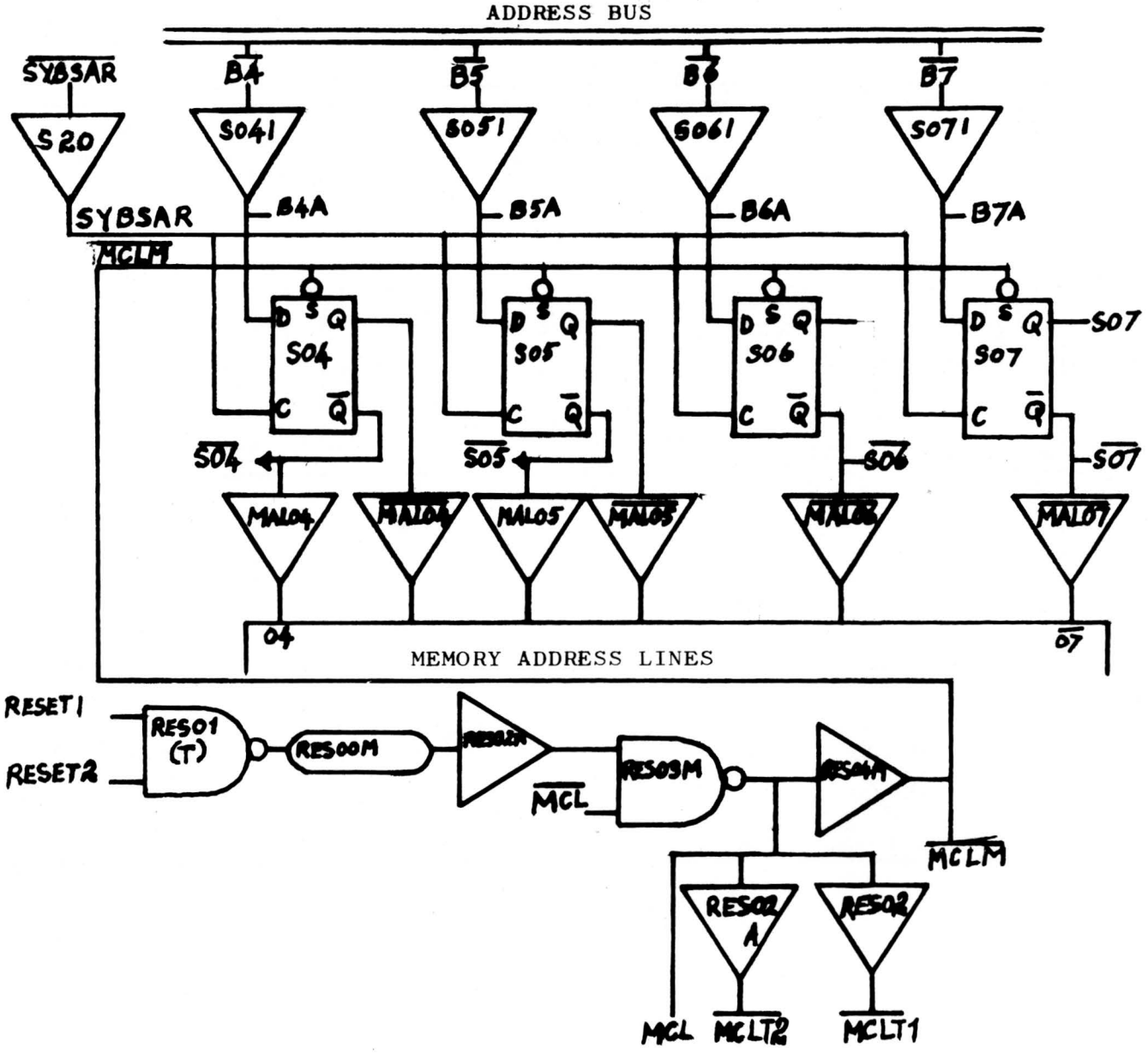
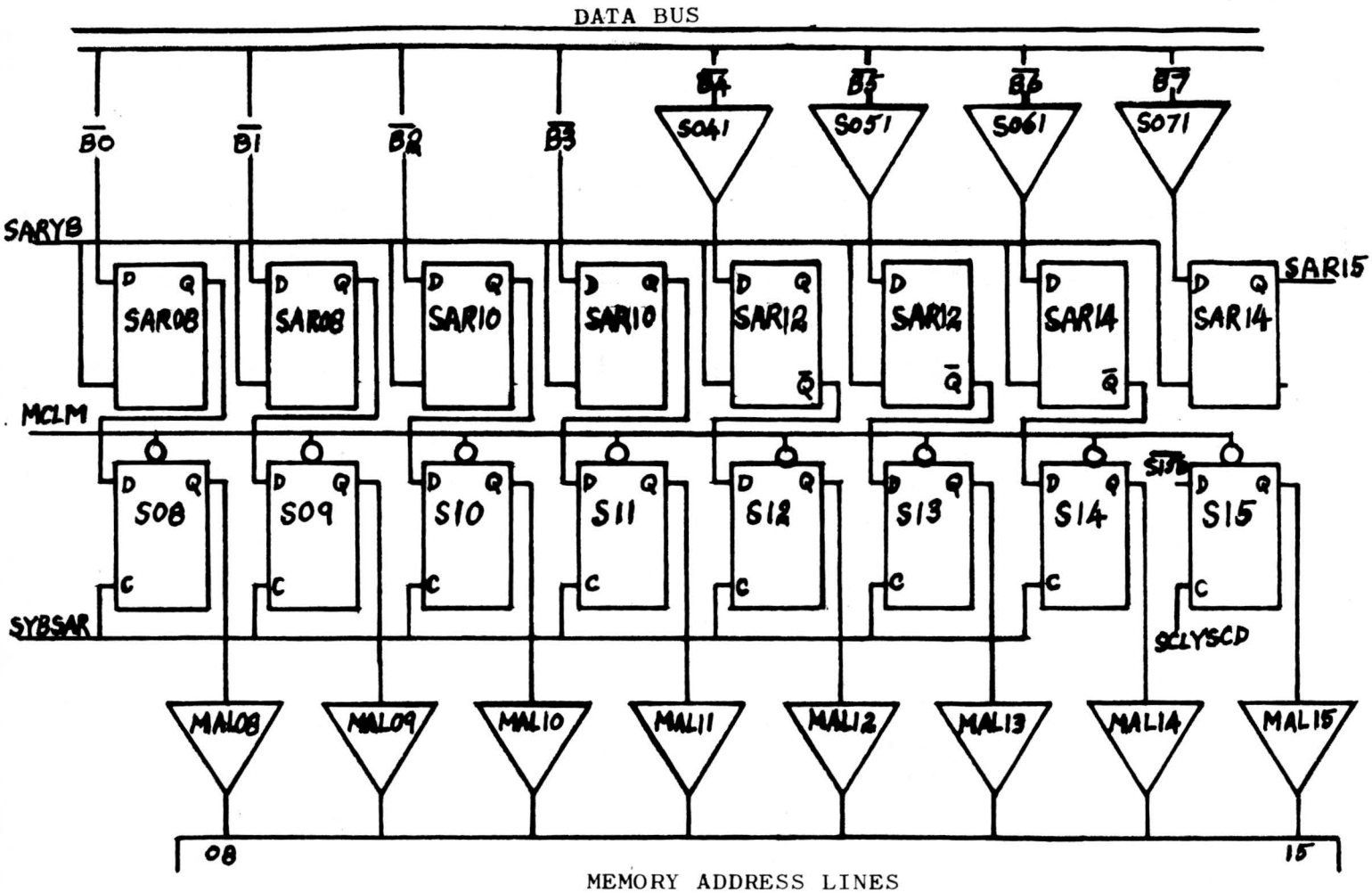


Figure M7b REPEAT CYCLES LOGIC



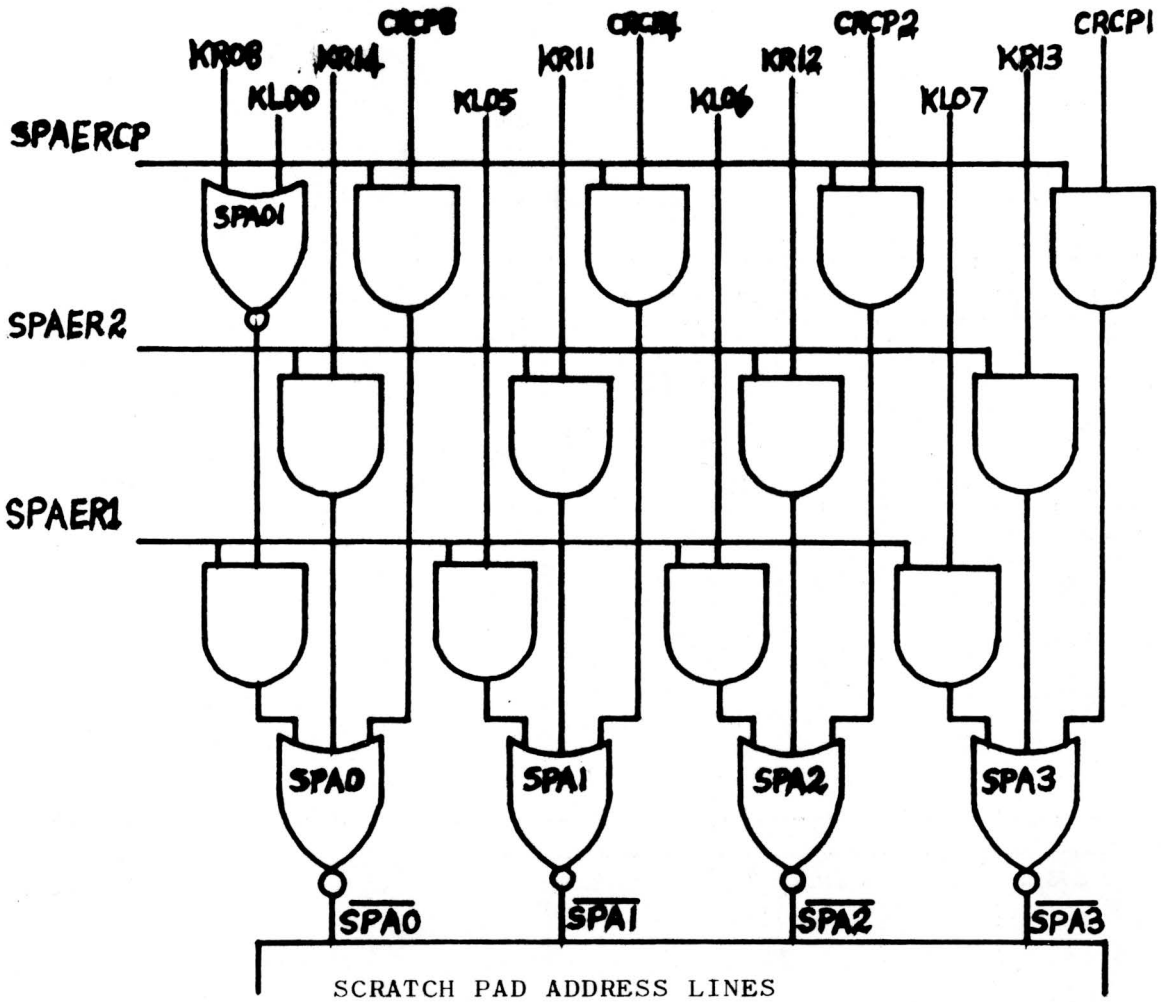
The most significant bits of the S-register are gated by signal SYBSAR. This enables bits 04 to 07 to be gated into bits 04 to 07 of the S register. Gating for the LSB of this register is shown on diagram M9.

S REGISTER BITS 08 - 15 and SAR REGISTER



SARYB, SYBSAR, SCLYSCD and $\overline{\text{SISD}}$ are produced on diagram C9 and are used to update the S-register.

MCLM is produced an diagram M8 when the master clear signal is required. It is used to clear the S-register.

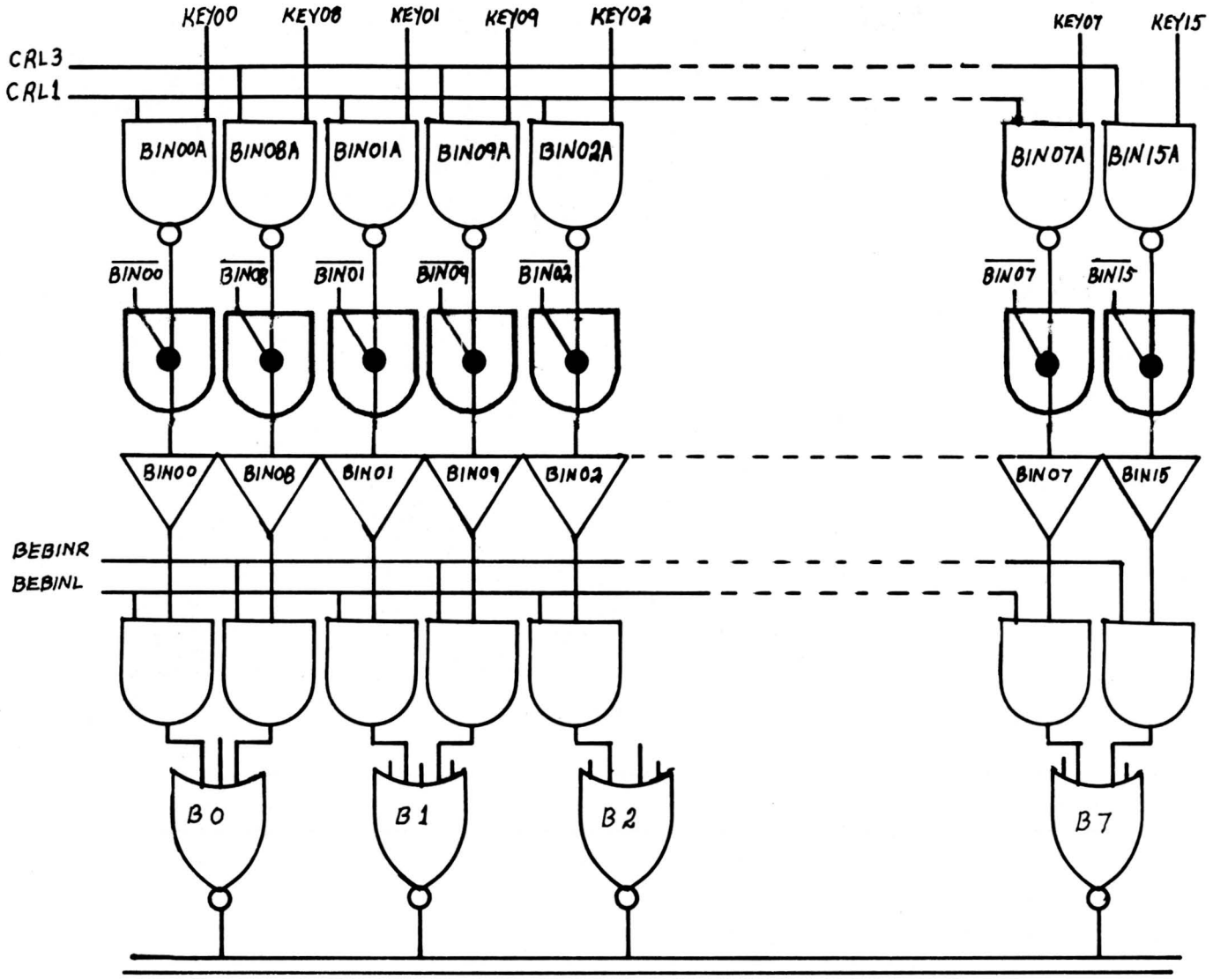


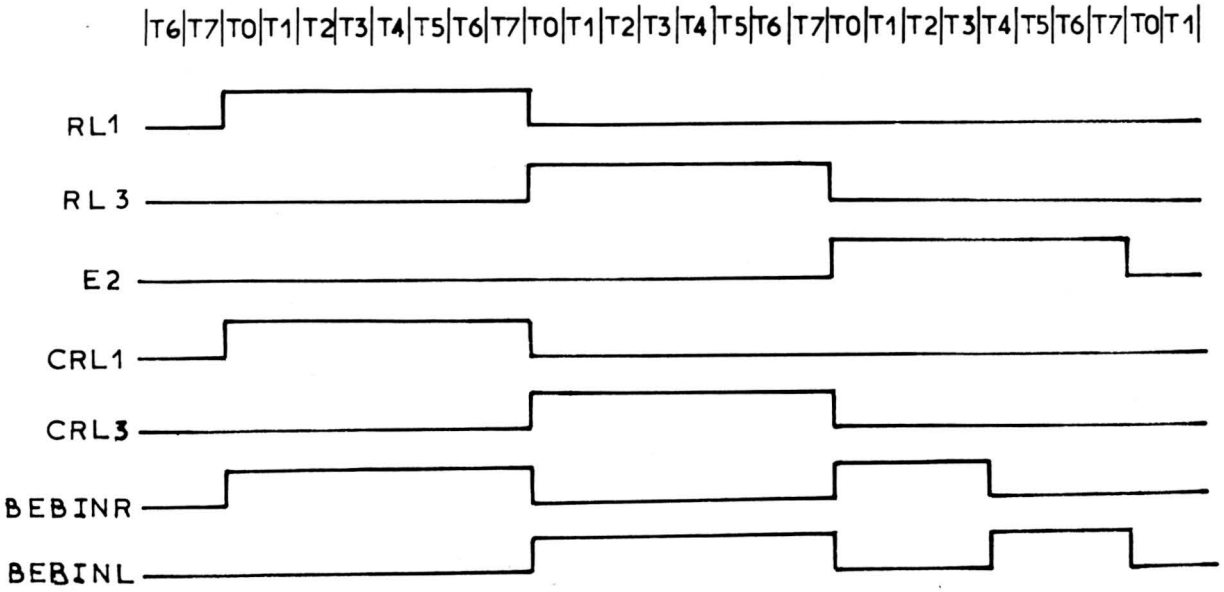
SPAER1 is produced on diagram C10 and is used to enable the selection of the address lines of the first operand (R1).

SPAER2 is also produced on diagram C10 and is used to enable the selection of the address lines of the second operand (R2).

SPAERCP is produced on diagram T10 and is used when the scratch pad address lines are selected by the switches on the control panel.

BIN LINES TO DATA BUS BIN AND B GATES

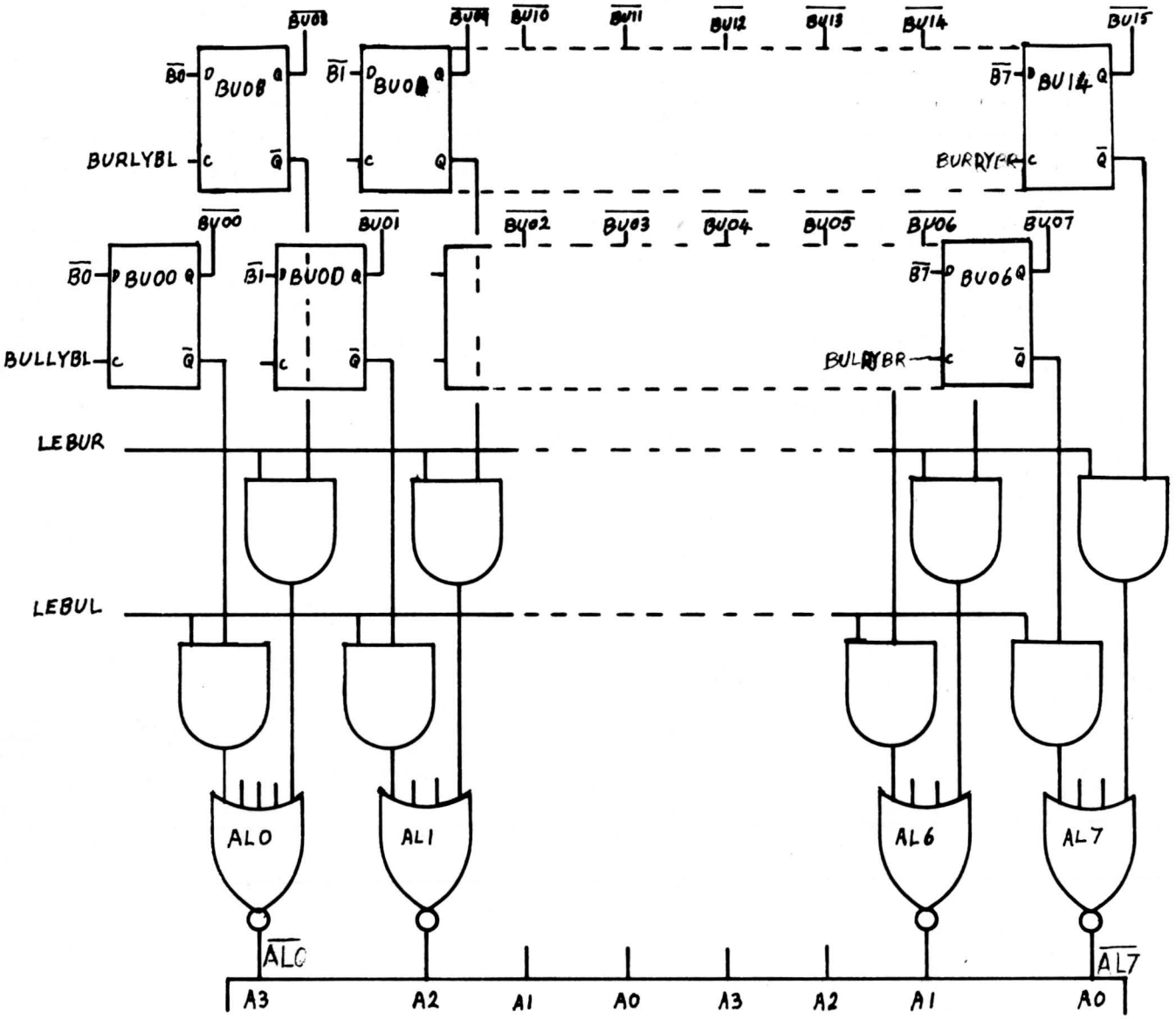




Other B-gate inputs from the E-register and condition and control flip-flops are shown on diagram A5.

Data to the B-lines are controlled by BEBINR and BEBINL. When data come the control panel keys, signals CRL3 and CRL1 are used to select the appropriate key's data for gating.

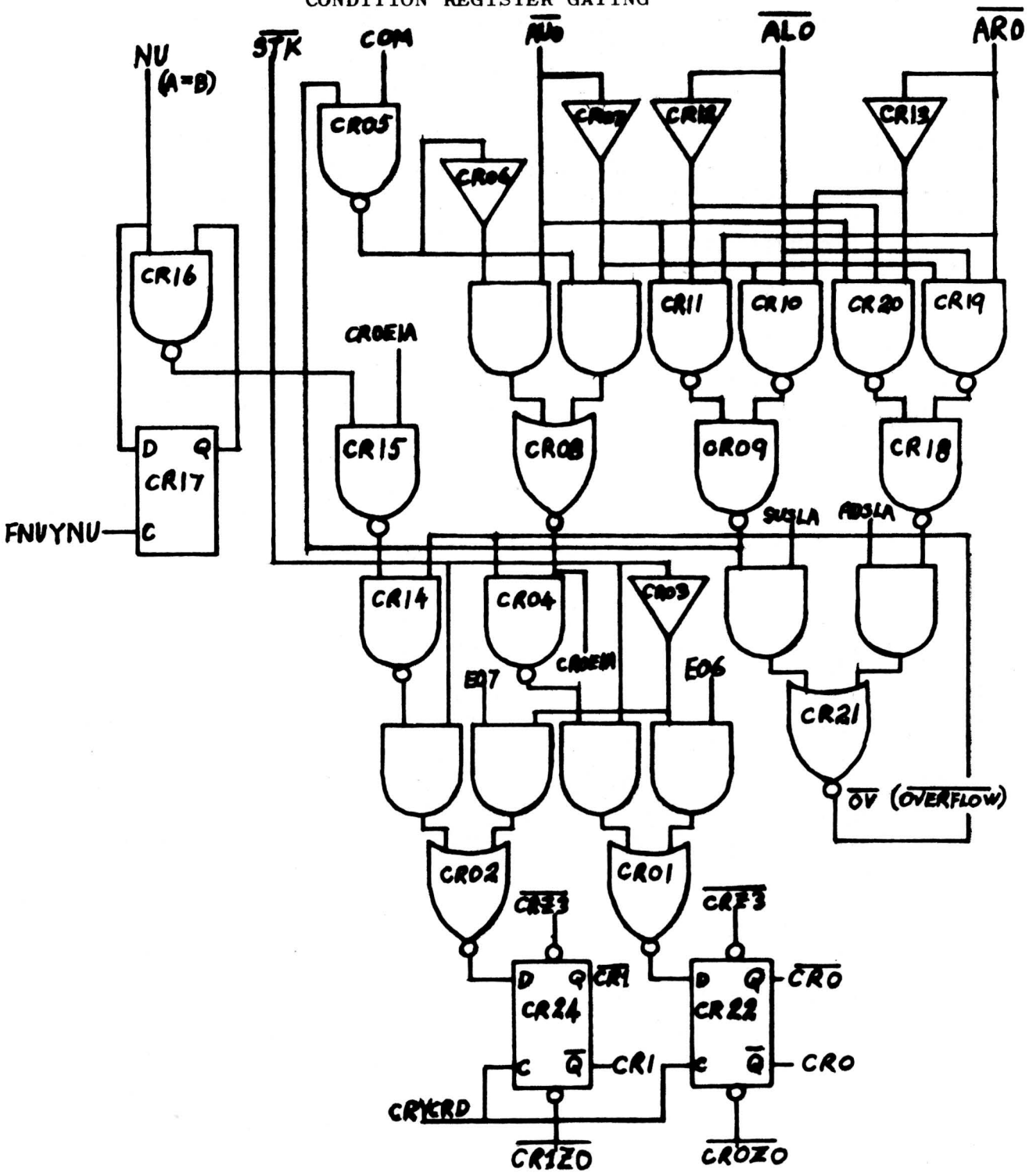
BU REGISTER TO AU



ARITHMETIC UNIT

LEBUR and LEBUL are the two signals that control the gating of the BU-register contents into the AU. These signals are produced on diagram C4 and are used during the call function instructions.

CONDITION REGISTER GATING



The condition register is set as a result of either an AU operation or an I/Ø instruction. It is clocked by CRYCRD and can be set by $\overline{CRZ3}$ or reset by $\overline{CR1ZO}$ or $\overline{CROZO}$

During I/Ø instructions, the state of the register has the following meaning:

CR = 0 command accepted (or device is in ready state).

CR = 1 command not accepted (or device busy)

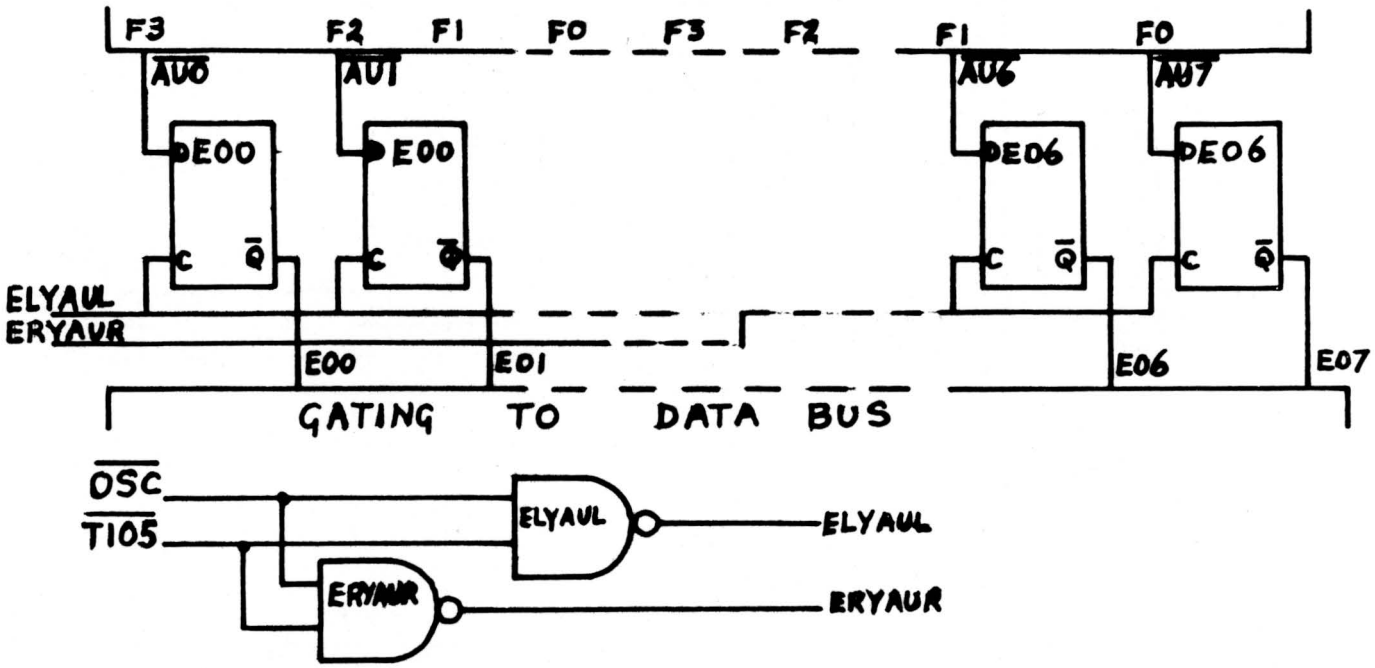
CR = 3 device address unknown

For other types of instruction it represents one of eight conditions. These conditions are usually associated with branch instructions and are:

condition	CR contents
0	= 0
1	= 1
2	= 2
3	= 3
4	≠ 0
5	≠ 1
6	≠ 3
7	unconditional

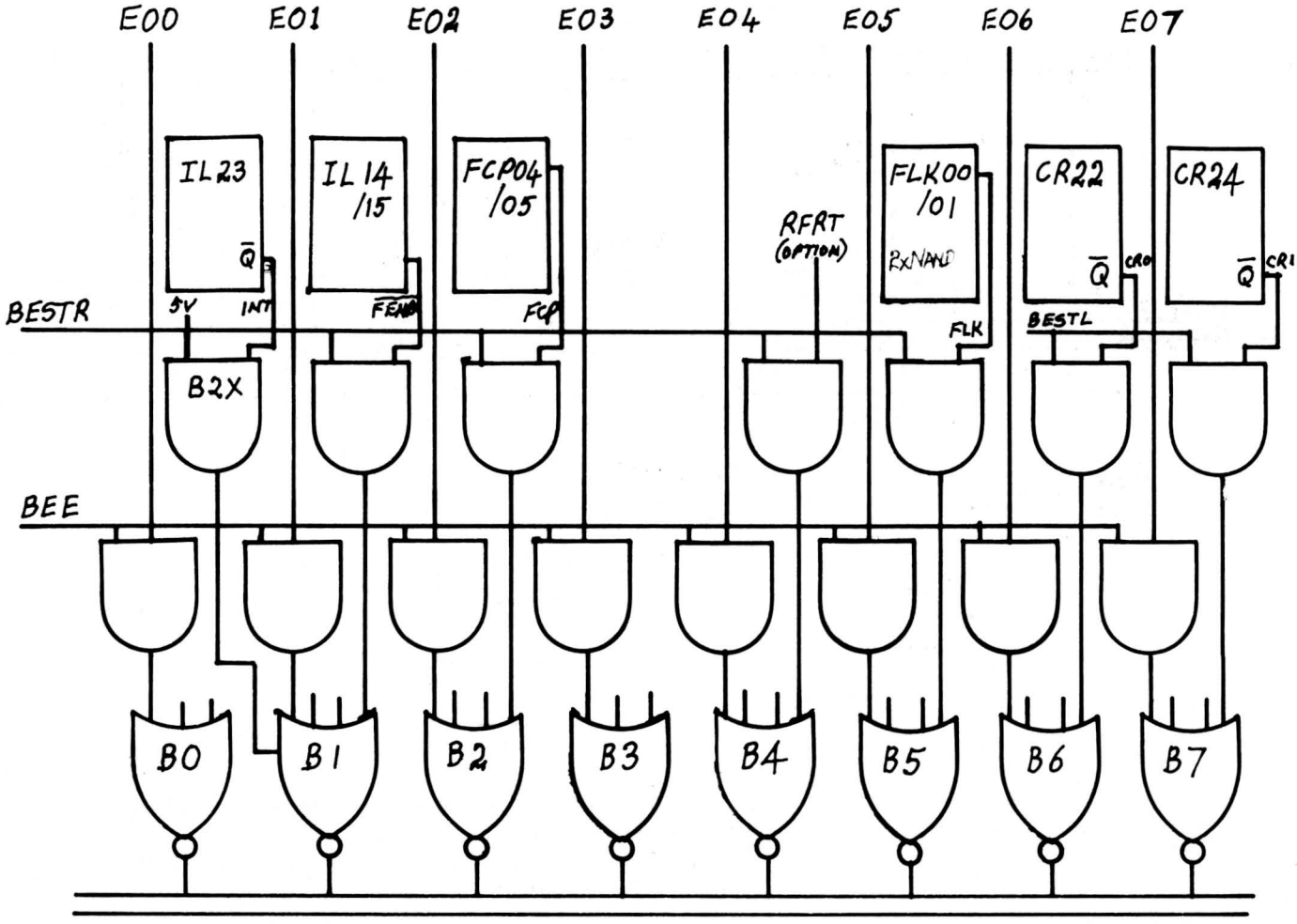
E REGISTER

ARITHMETIC UNIT



Timing signals, $\overline{T105}$ and $\overline{OSC}$, are produced on diagram T1. The contents of the AU are gated into the AU at T1 and T5 time of every cycle; whether this information is gated to the data bus depends upon signal BEE which is produced on diagram A5.

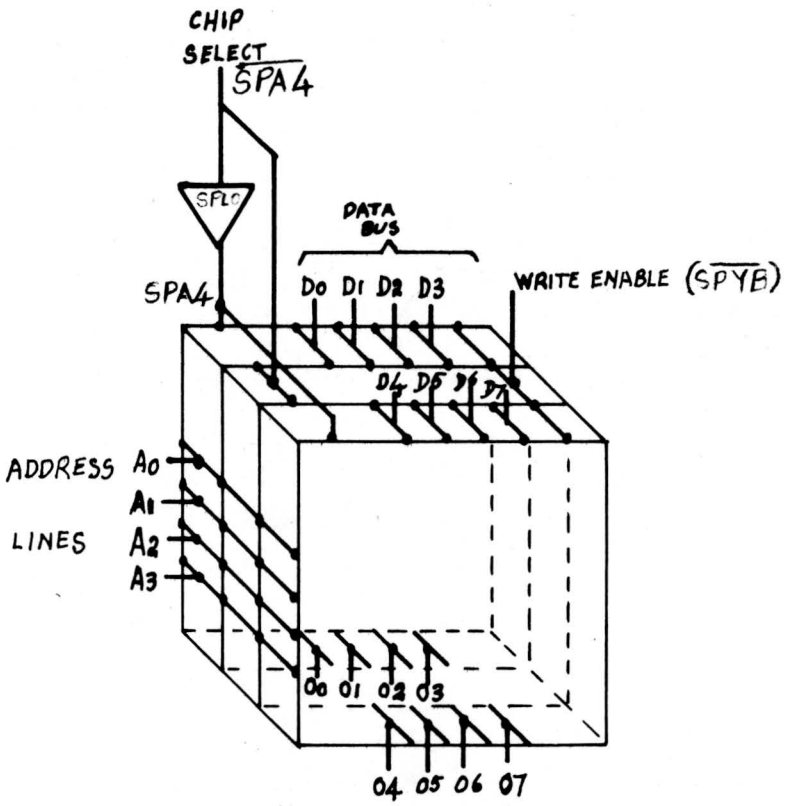
E REGISTER TO DATA BUS B GATE



IL23 is set by interrupt logic (T11) and placed on bus for loading into S-register to address memory location 32. Other B-gate inputs from BIN lines and control panel are shown on diagram A1.

Gating signals BESTR and BEE are produced and described on diagram T9.

SCRATCH PAD

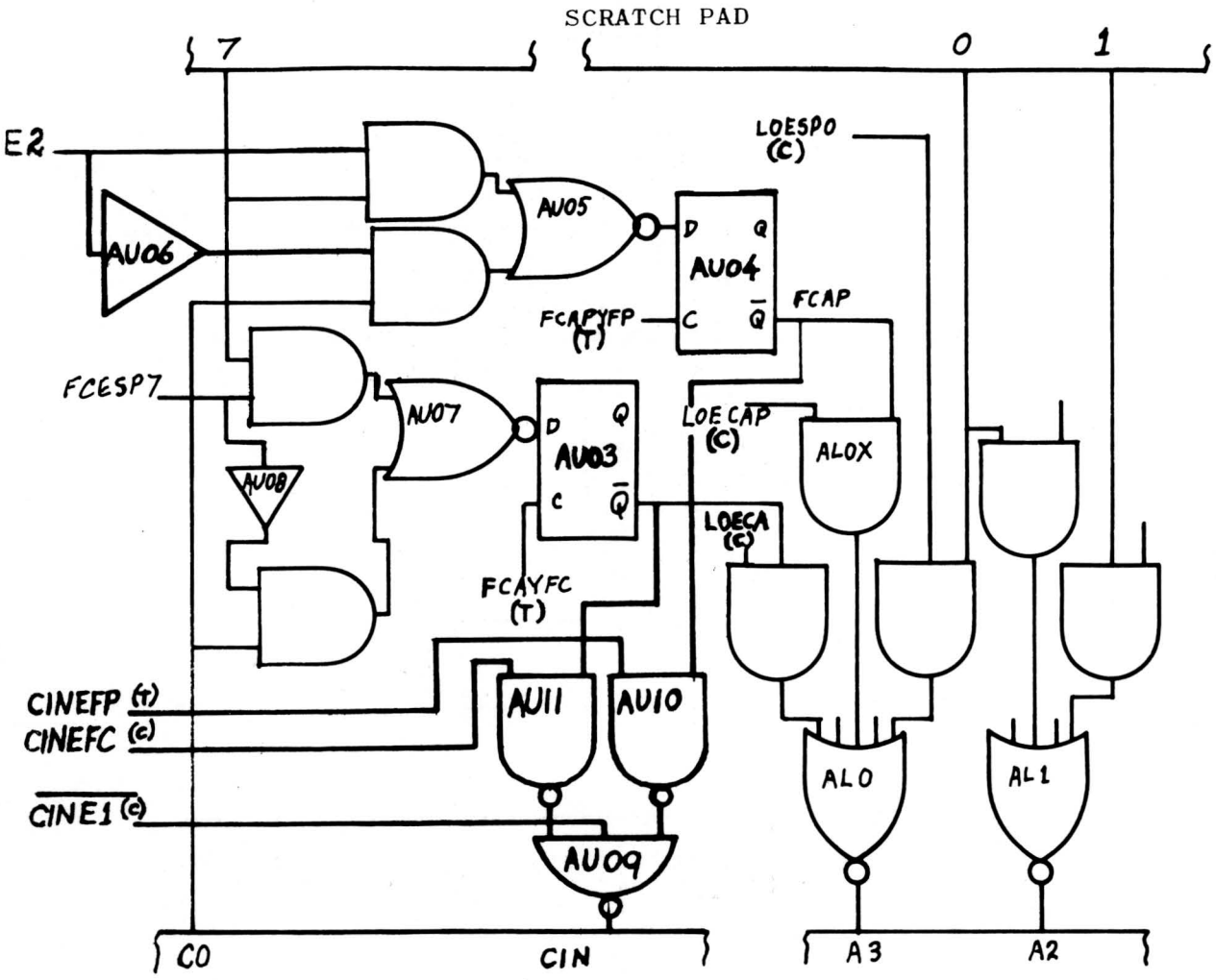


The signal SPA4 is produced on diagram C11. This signal level is used to select the chip required by the instruction cycles. When present it enables the LSBs to be used and when not present it enables the MSBs to be used.

The normal order is LSB first and MSB second but, in shift right instructions, this order is reversed during the E1 cycle.

Signal $\overline{\text{SPYB}}$ is produced on diagram C14. This signal is used to enable the write data logic.

SCRATCH PAD TO AU AND CARRY LOGIC



ARITHMETIC UNIT

FCESP7 is produced on diagram T7 and is also described on diagram C12.

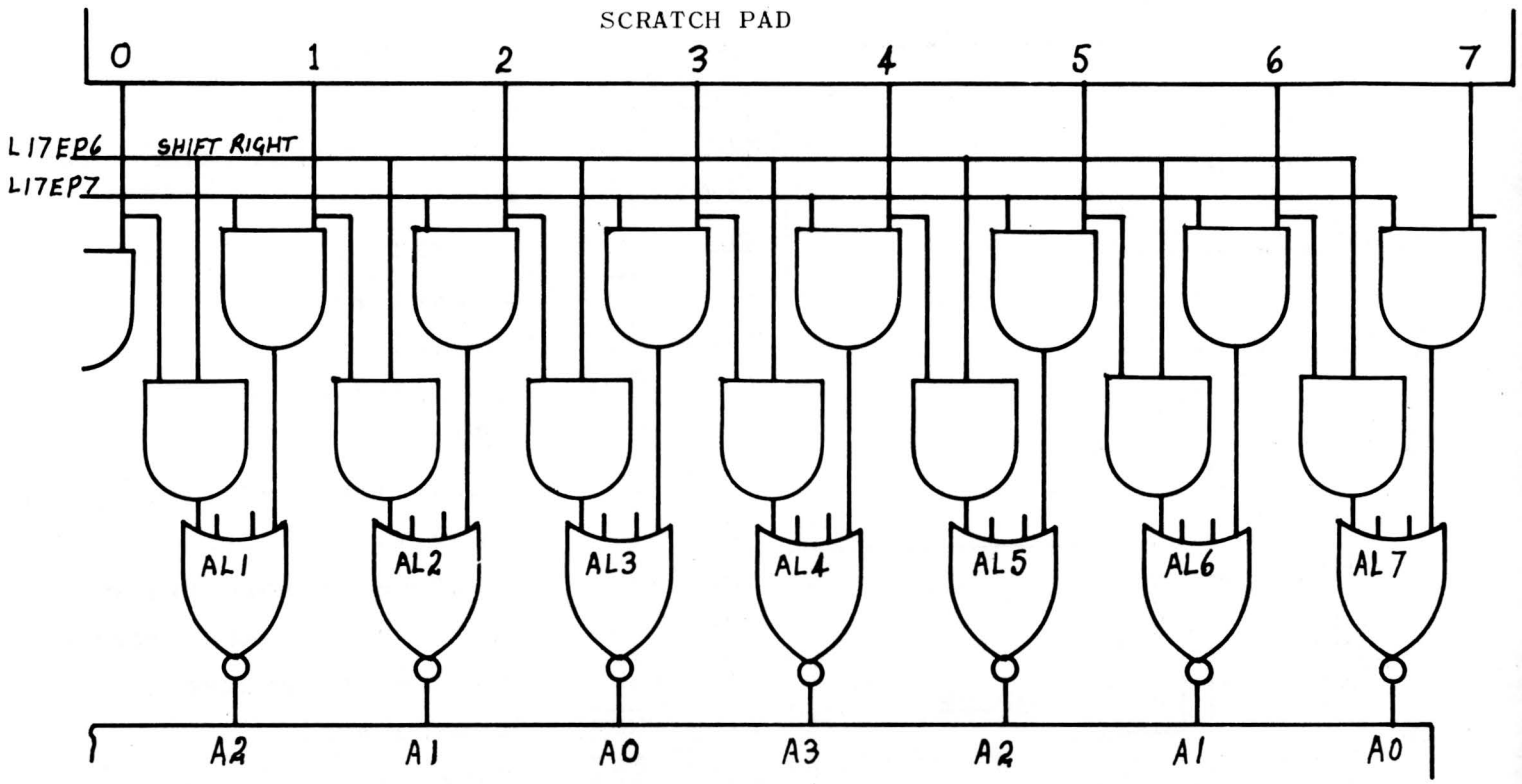
$\overline{\text{CINE1}}$ is produced on diagram C4 and is used in conjunction with CINEFP and CINEFC during normal carry functions.

CINEFP is produced on diagram T8 and is used for carry functions when updating the P-register.

CINEFC is produced on diagram C4 and is used for carry functions other than P-register updating.

Other uses of the carry logic are described on diagram C12.

SCRATCH PAD TO AU

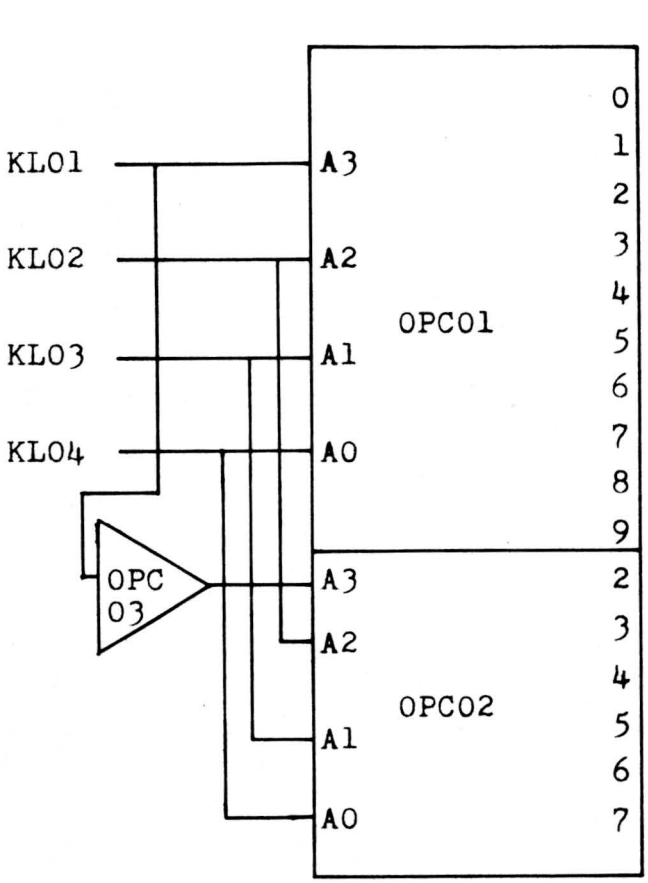


ARITHMETIC UNIT

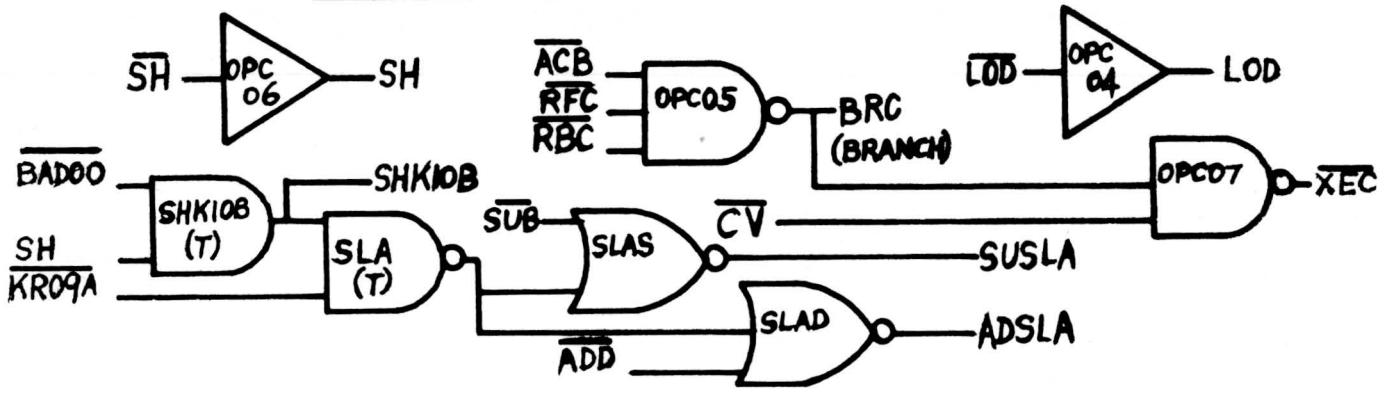
During shift right instructions, L17EP6 and the shift right carry logic control the gating of data from the scratch pad into the AU.

For other instructions the gating is controlled by L17EP7 (which controls bits 01 to 07) and LOESPO (bit 00).

OPERATION CODE DECODING



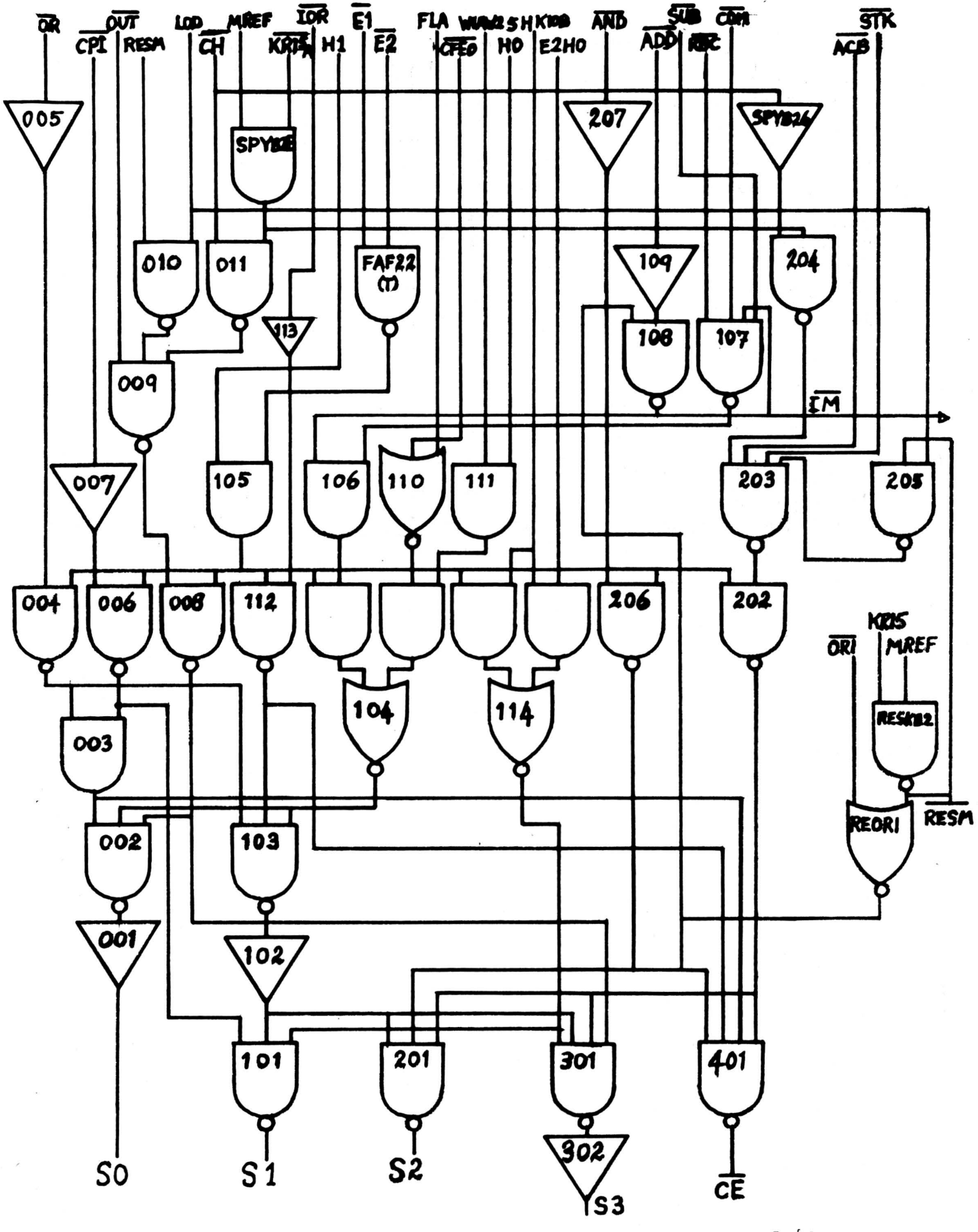
	KL 1234		INSTRUCTION
0	0000	$\overline{\text{LOD}}$	Load
1	0001	$\overline{\text{ACB}}$	Absolute conditional branch
2	0010	$\overline{\text{ADD}}$	Add
3	0011	$\overline{\text{SUB}}$	Subtract
4	0100	$\overline{\text{AND}}$	Logical AND
5	0101	$\overline{\text{IOR}}$	Logical OR
6	0110	$\overline{\text{XOR}}$	Exclusive OR
7	0111	$\overline{\text{SH}}$	Shift
8	1000	$\overline{\text{OUT}}$	Output from register
9	1001	$\overline{\text{IN}}$	Input to register
2	1010	$\overline{\text{RFC}}$	Relative forward branch
3	1011	$\overline{\text{RBC}}$	Relative backward branch
4	1100	$\overline{\text{CH}}$	Transfer character
5	1101	$\overline{\text{COM}}$	Compare word
6	1110	$\overline{\text{STK}}$	Stack
7	1111	$\overline{\text{CPl}}$	Ones complement



Operation code held in KL01 to KL04 is decoded in two type 9301, one of ten decoders, connected as shown with inverter OPC3 to give one of sixteen decoding.

xc

AU FUNCTION CODING

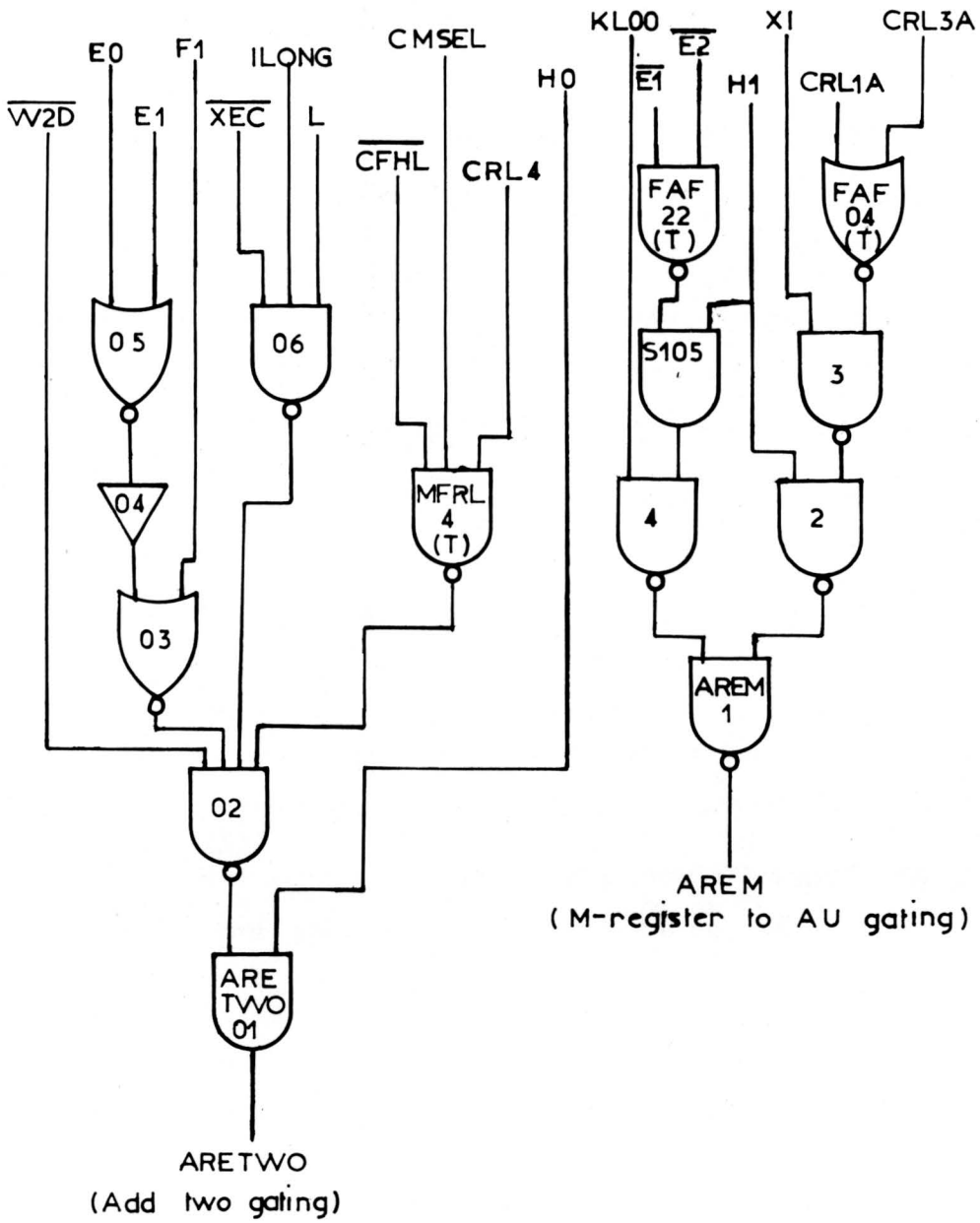


Outputs SO to S3 are function inputs to the arithmetic unit.

Inputs to the coding logic are taken from the OPC decoding networks and gated with the timing signals.

NO SIGNAL	→ $\overline{SO}.\overline{S1}.\overline{S2}.S3.\overline{CE}$ →	A plus B
XØR.H1.(E1+E2)	→ $\overline{SO}.S1.S2.\overline{S3}.\overline{CE}$ →	A ⊕ B
CPI.H1.(E1+E2)	→ $\overline{SO}.S1.\overline{S2}.S3.\overline{CE}$ →	$\overline{B}$
ØUT.H1.(E1+E2)	→ $\overline{SO}.\overline{S1}.\overline{S2}.\overline{S3}.CE$ →	A
RESM.LØD.(E1+E2)	→ $\overline{SO}.\overline{S1}.\overline{S2}.\overline{S3}.CE$ →	A
MREF.$\overline{CH}$.KR15.(E1+E2)	→ $\overline{SO}.\overline{S1}.\overline{S2}.\overline{S3}.CE$ →	A
IØR.H1.(E1+E2)	→ $SO.S1.S2.\overline{S3}.\overline{CE}$ →	A + B
$\overline{FLA}$.CFEØ.W1PW2.HO	→ $\overline{SO}.S1.S2.\overline{S3}.CE$ →	A minus B minus 1
AND.H1(E1+E2)	→ $SO.\overline{S1}.S2.S3.\overline{CE}$ →	AB
$\overline{ADD}$+($\overline{OR1}$+KR15.MREF).(SUB+RBC+CØM).H1(E1+E2)	→ $\overline{SO}.S1.S2.\overline{S3}.CE$ →	A minus B minus 1
SHKIOB.[H1(E1+E2)+E2HO]	→ $SO.S1.\overline{S2}.\overline{S3}.CE$ →	A plus A
CH.MREF.KR15A.H1(E1+E2)	→ $\overline{SO}.\overline{S1}.S2.\overline{S3}.\overline{CE}$ →	$\bar{A}B$
ACB.H1(E1+E2)	→ $\overline{SO}.\overline{S1}.S2.\overline{S3}.\overline{CE}$ →	$\bar{A}B$
STK.H1(E1+E2)	→ $\overline{SO}.\overline{S1}.S2.\overline{S3}.\overline{CE}$ →	$\bar{A}B$
(KR15+MREF)LOD.H1(E1+E2)	→ $\overline{SO}.\overline{S1}.S2.\overline{S3}.\overline{CE}$ →	$\bar{A}B$

AR GATE CONTROL



Add two function is used at the AR gate to add two to an address in memory reference instructions, thus giving the next 16-bit memory location.

$$H0.(W2D+ILONG.\overline{XEC}.L+F1+E0+E1)$$

AREM function is used to open the AR gates and allow the contents of the M-register into the AU.

Both the above signals are used on diagram M3.

LEBUR and LEBUL are used during a call function instruction to load the contents of location 32 (start address of software routine) into the S-register.

During the E1-cycle of the hardware routine the LSBs of location 32 are read out of memory into BUR and during the E2 cycle the MSBs of location 32 are read out of memory into BUL.

The BU register now contains the start address of the software routine and it is held in the BU register until the L-cycle of the hardware routine.

During the L-cycle, at T0 time, the contents of BUR are enabled to the AU with the signal LEBUR.

They are then gated via the AU, E-register and B-lines to the SAR-register and LSBs of the P-register.

At T4 time the contents of BUL are enabled to the AU with signal LEBUL and are then gated via the AU, E-register and B-lines to the MSBs of the S and P-registers.

$\overline{\text{ECR}}$ and ACR are used to enable various scratch pad logic circuits during character handling instruction. They are used on diagrams C10, C11, C13 and C14.

$\overline{\text{CFEO}}$ is produced during the E0-cycle of a call function and is used on diagram C2 by the AU function coding logic.

$\overline{\text{CINE1}}$ can be produced during an E1-cycle if the AU operation produced a carry. It is used on diagram A7 as an input to gate AU09.

CINEFC can be produced during each E2-cycle and is also used on diagram A7 as an input to gate AU11.

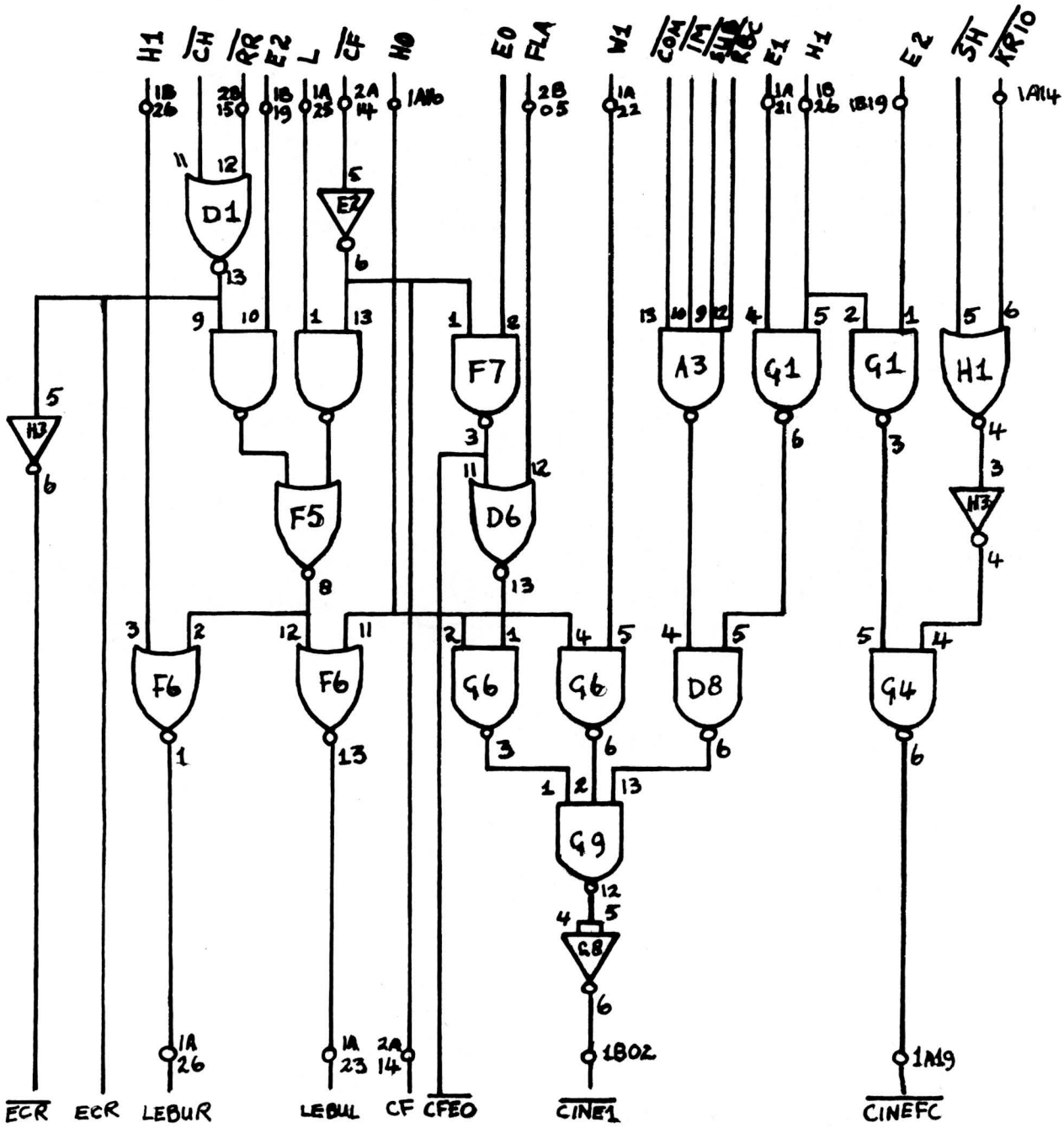


Figure C4 BU REGISTER TO AU CONTROL AND CARRY INPUT

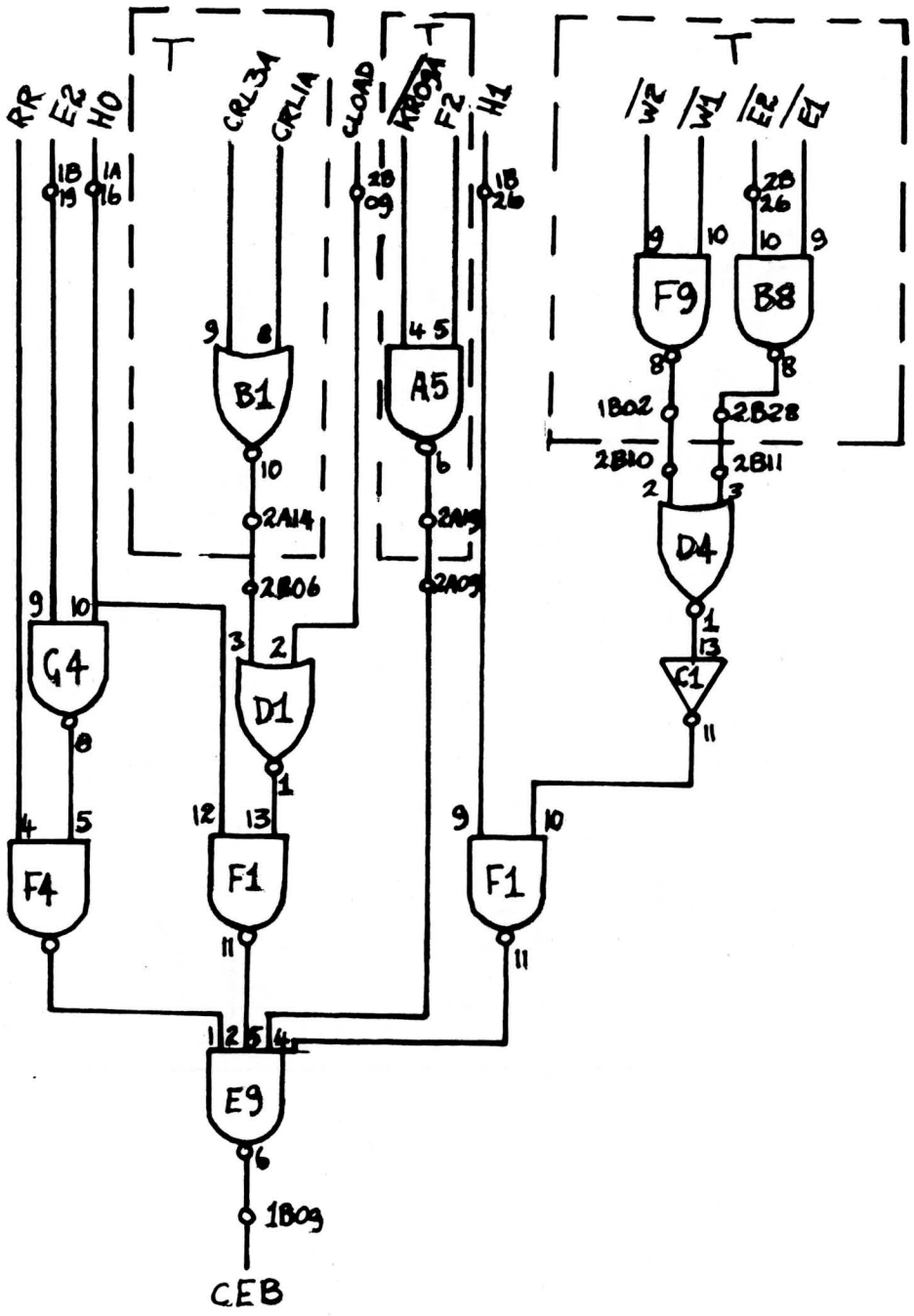


Figure C5b C GATE CONTROL

CEB is used on diagram M1 to enable the AND gate inputs to the M-register. The inputs will be either from the B-lines or D-gates depending upon the type of instruction being performed. The actual clocking of the C gates into the M-register is accomplished by the signal MYC which is produced on diagram C6.

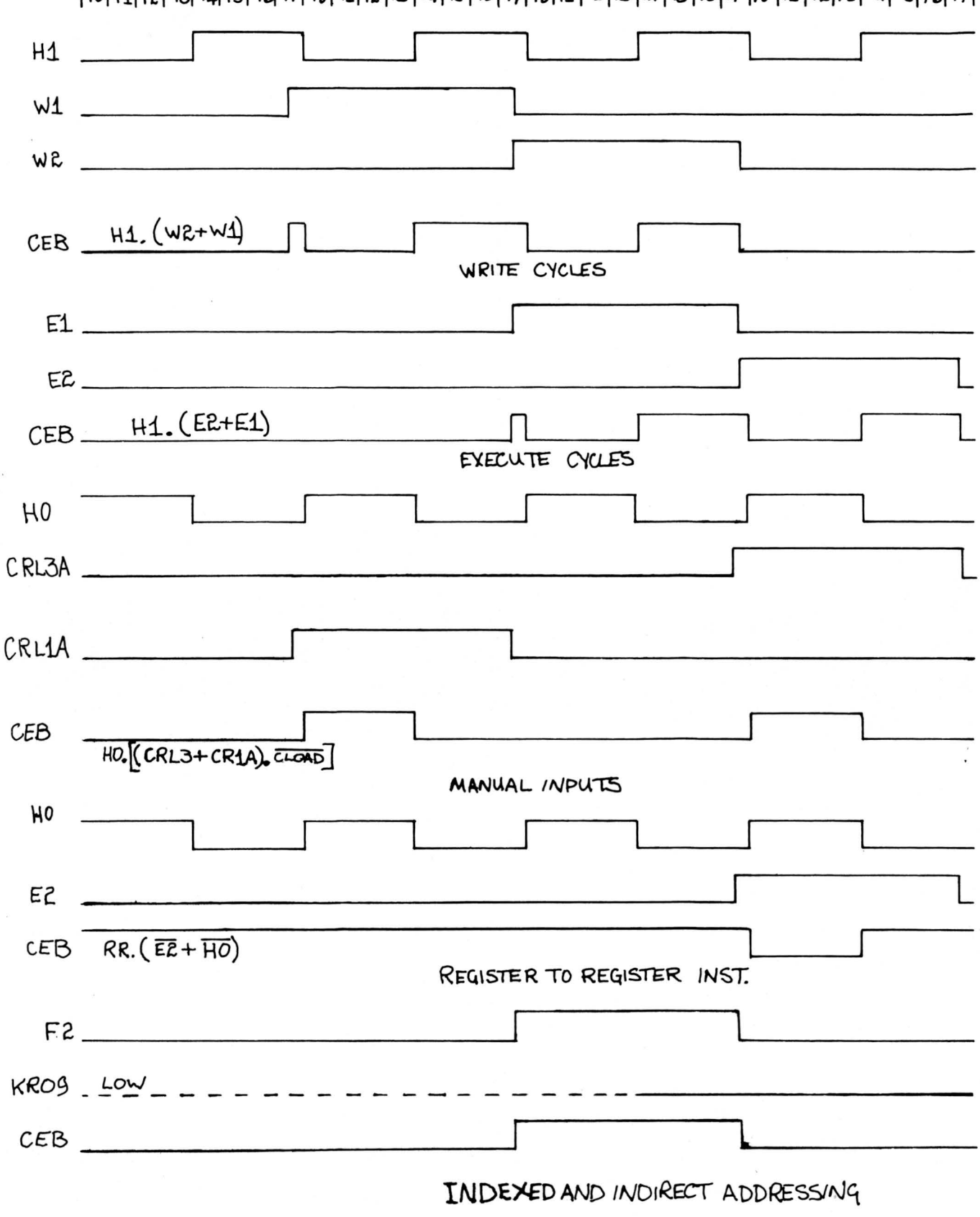
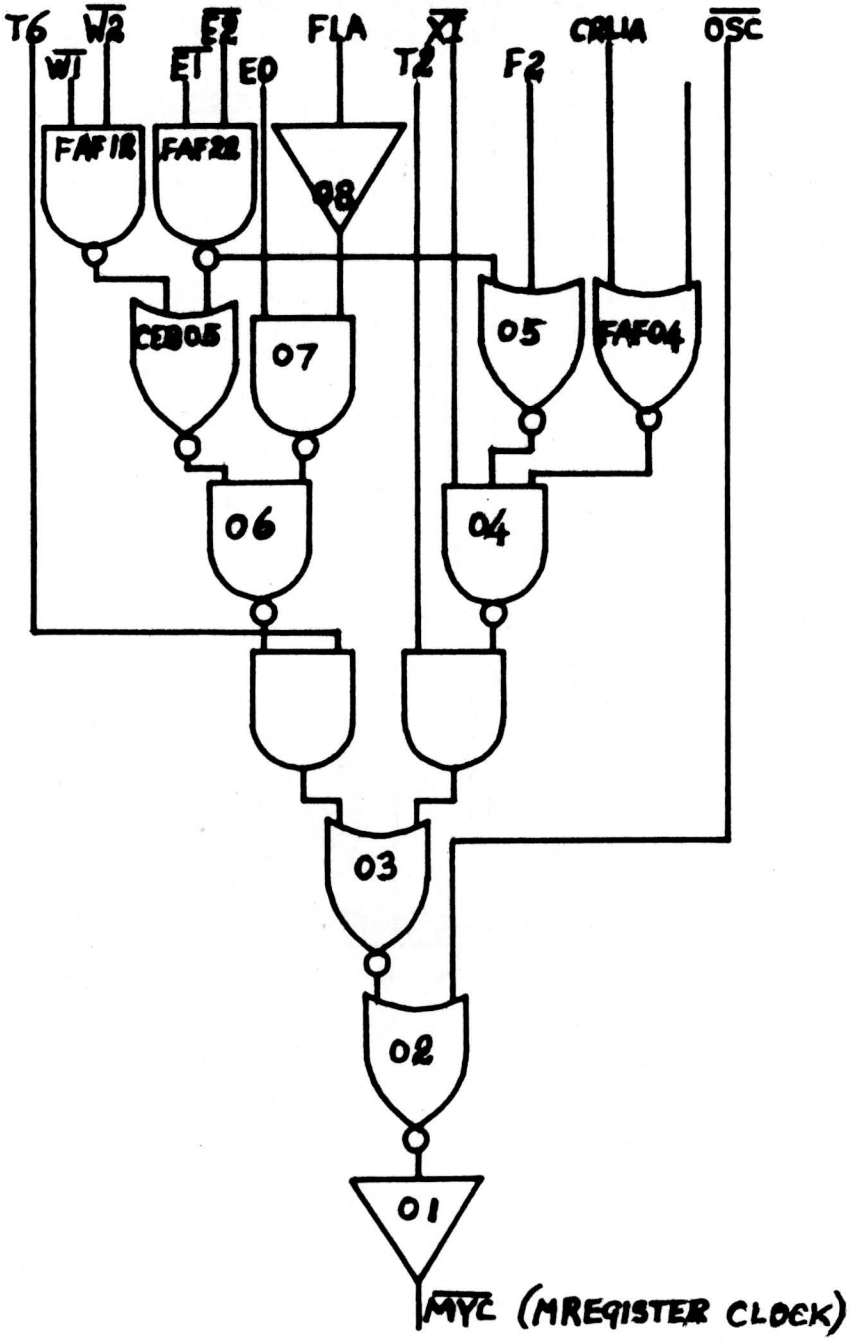


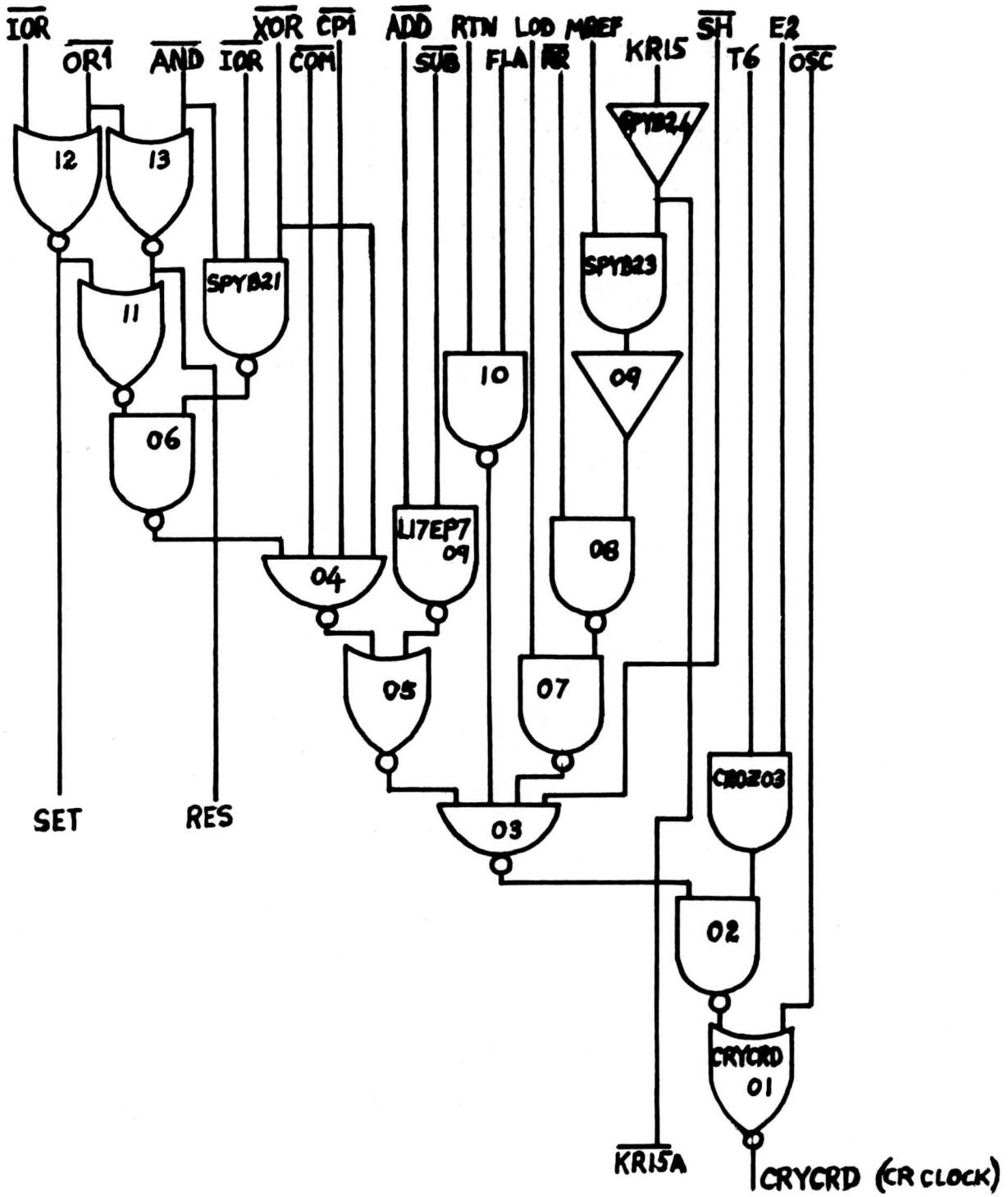
Figure C5a C GATE CONTROL

M REGISTER CLOCK PULSE



This clock pulse is produced either at T2 or T6 time depending on the type of cycle being performed. It is used on diagram M2 to trigger the M-register flip-flops.

CONDITION REGISTER CLOCK PULSE



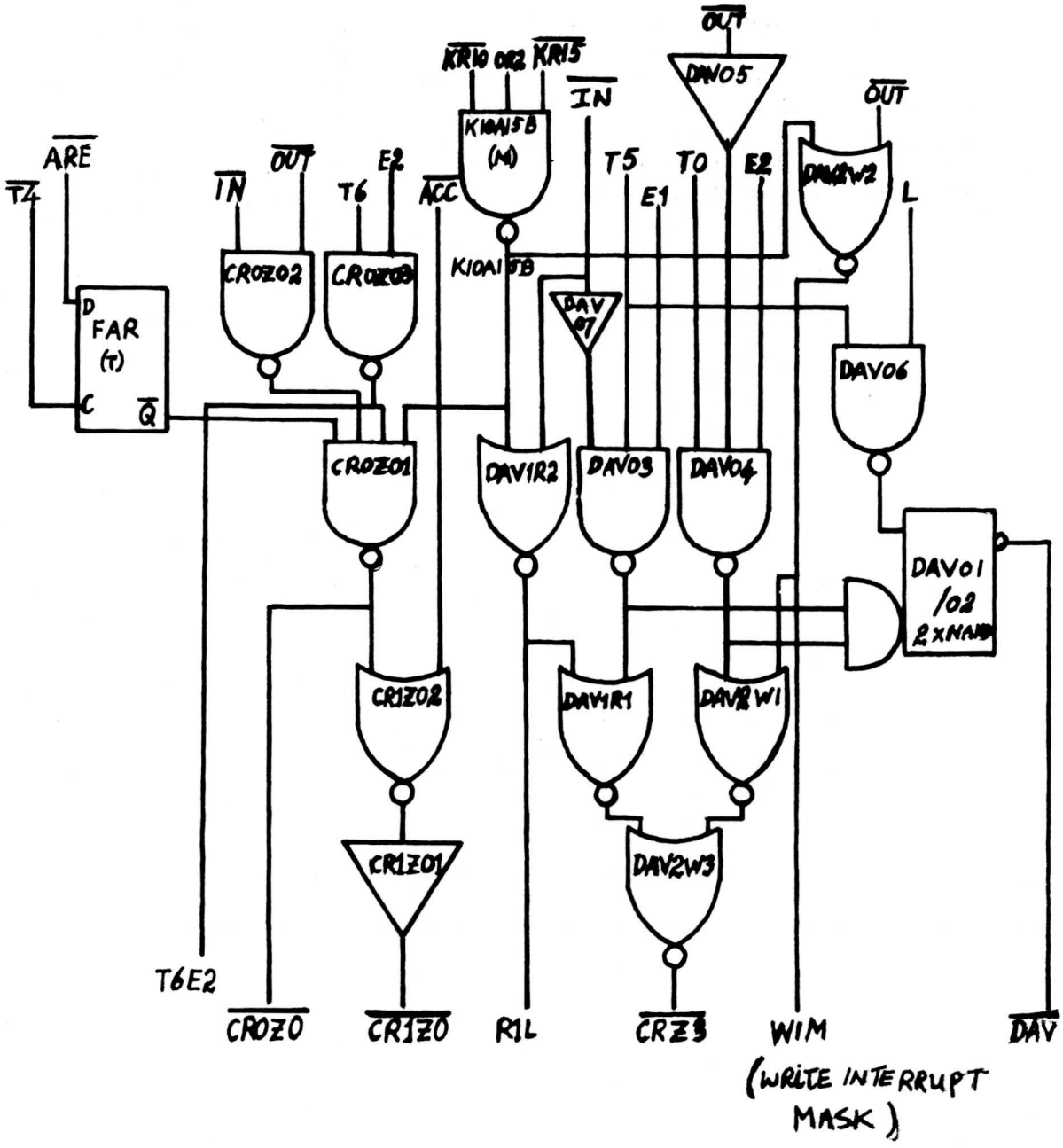
CRYCRD is used to clock the condition register (diagram A3) when various events occur. During an E1-cycle it occurs at T5 time to set the condition register to 3 if the instruction is a data input type.

During an E2-cycle, if the instruction is data output type, the signal occurs at T0 time and sets the condition register to 3. It also occurs at T6 time, either to set the condition register according to the result of performing an instruction, or to update the condition register if the instruction is an input/output type.

SET and RES are produced during logical instruction and are used on diagram C14 as part of the scratch pad enable logic, on diagram T11 as part of the interrupt logic and diagram M7 as part of the K-register decoding logic.

$\overline{\text{KR15A}}$ is produced during memory reference instructions and is used on diagram C2 as part of the AU function coding logic.

CONDITION REGISTER SET



T6E2 is used on diagram C14 as one input to AND gate 28.

$\overline{\text{CR0Z0}}$ is used on diagram A3 on the reset side of flip-flop CR24 (condition register).

$\overline{\text{CR1Z0}}$ is also used on diagram A3 on the reset side of flip-flop CR22 (condition register).

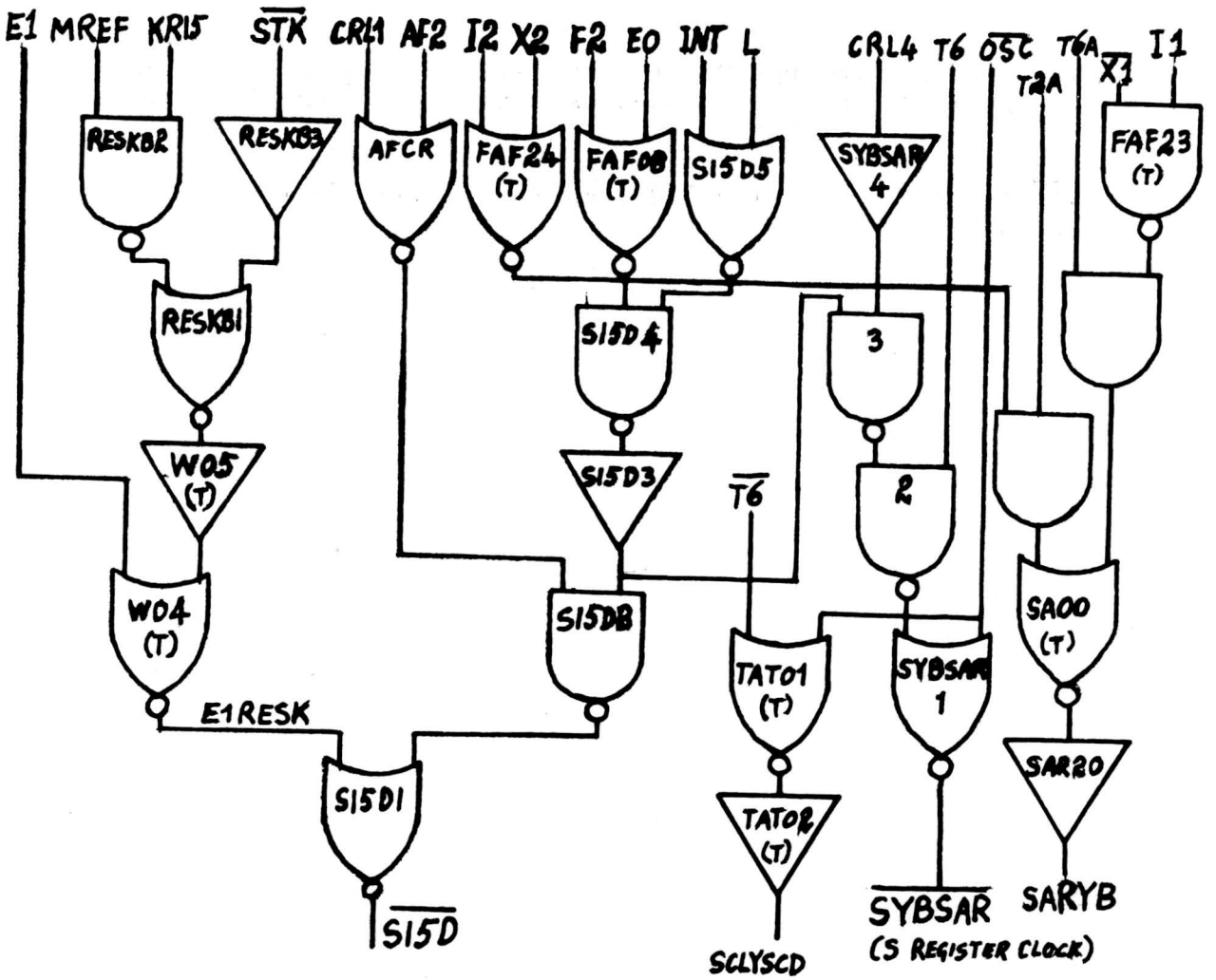
$\overline{\text{CRZ3}}$ is used on diagram A3 on the set side of flip-flops CR24 and CR22 (condition register).

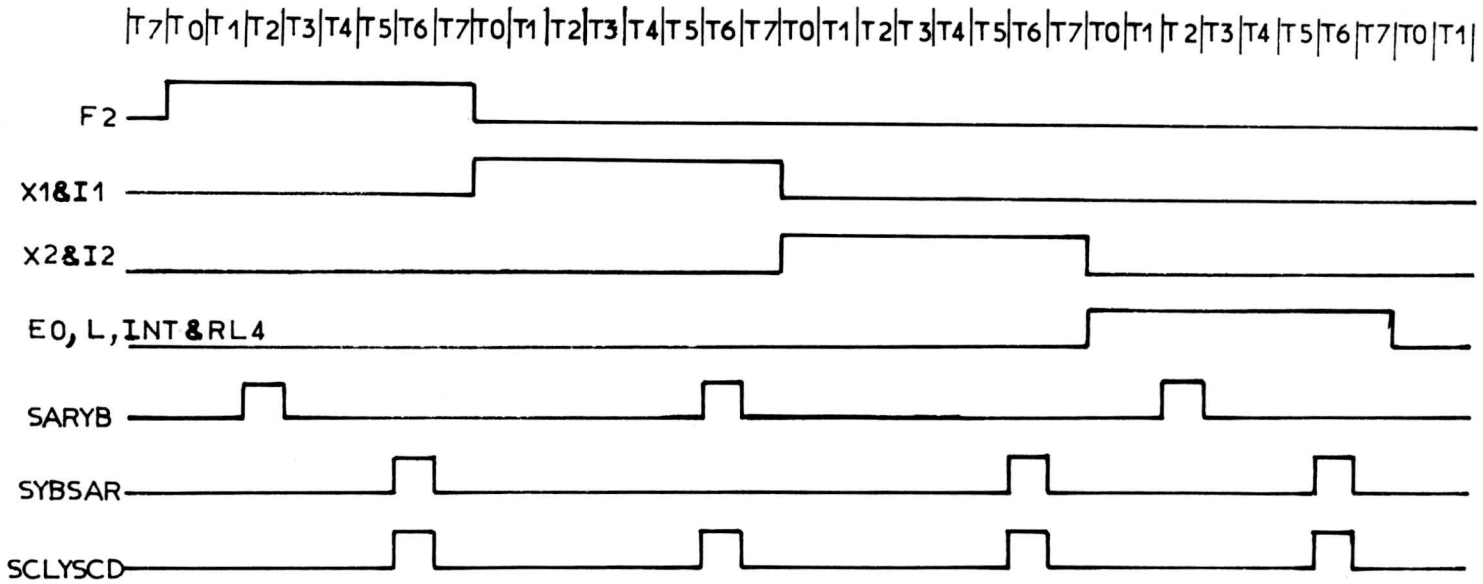
R1L is produced during a Read Interrupt Lines instruction.

WIM (Write Interrupt Mask) is produced when this instruction is performed. It is used as one input to the AND gate IL10 in the mask logic.

$\overline{\text{DAV}}$ is produced during I/Ø transfers if the device address is recognized.

S REGISTER BIT 15 AND S REGISTER CLOCK PULSES





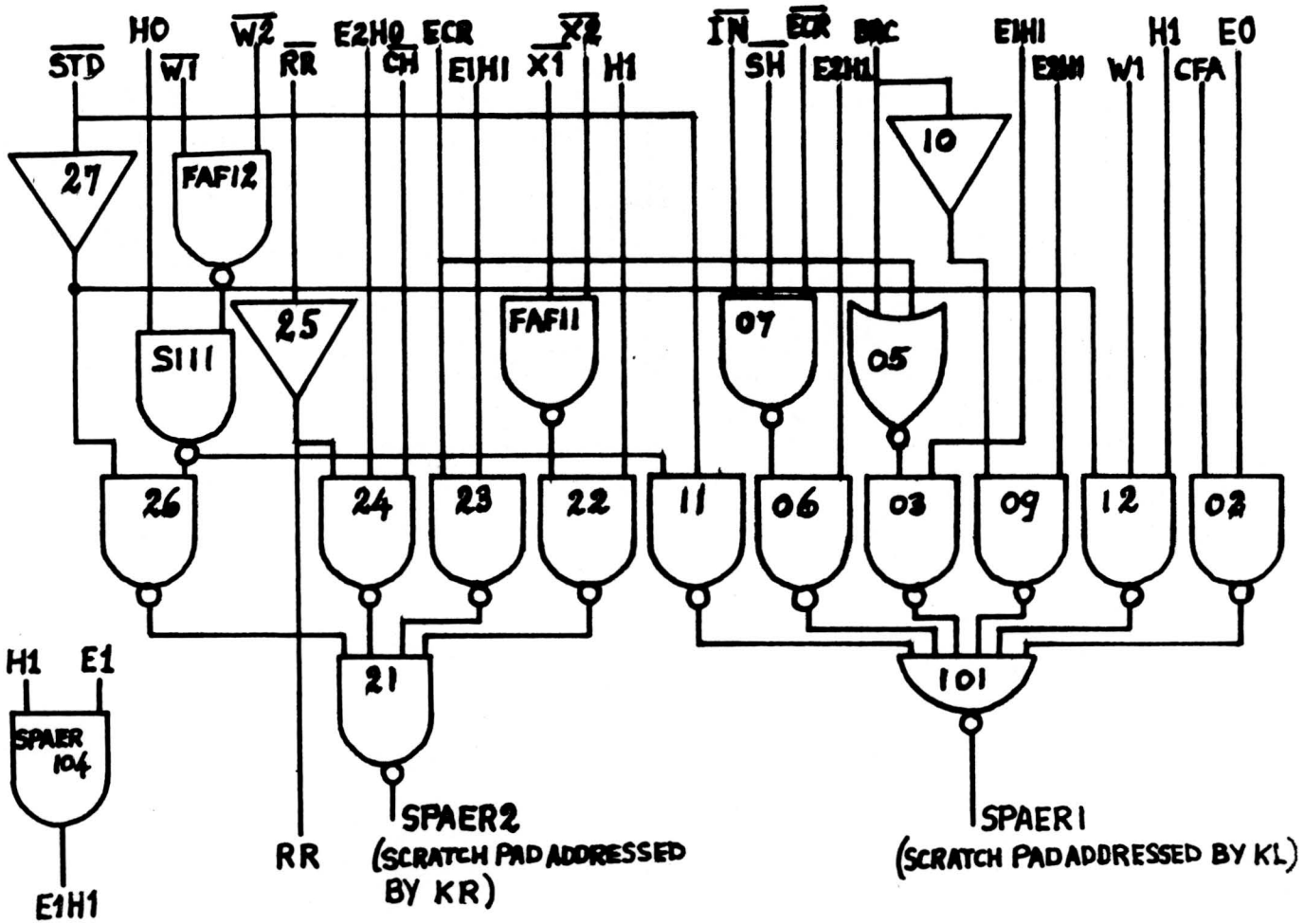
$\overline{\text{S15D}}$ is used to select the MSB (bits 00 to 08) of a 16-bit word stored in the memory. It is input to terminal D of flip-flop S15 on diagram M9. When the MSBs are required, the flip-flop is clocked by SCLYSCD.

SARYB is used to clock the contents of the B-lines into the SAR-register (diagram M9).

BYBSAR is used to clock the contents of the B-lines and the SAR-register into the S-register (diagram M9).

SCLYSCD is used to clock flip-flop S15 on diagram M9.

SCRATCH PAD ADDRESS GATING



SPAER1 is used to enable the scratch pad address lines (diagram M10) for the first operand (R1); the address is contained in bits 05 to 08 of the K-register.

SPAER2 is used to enable the scratch pad address lines (diagram M10) for the second operand (R2); the address is contained in bits 11 to 14 of the K-register.

E1H1 is used as an input to the SPAER1 logic on this page.

RR is used on diagram C5 as an input to the C-gate control logic.

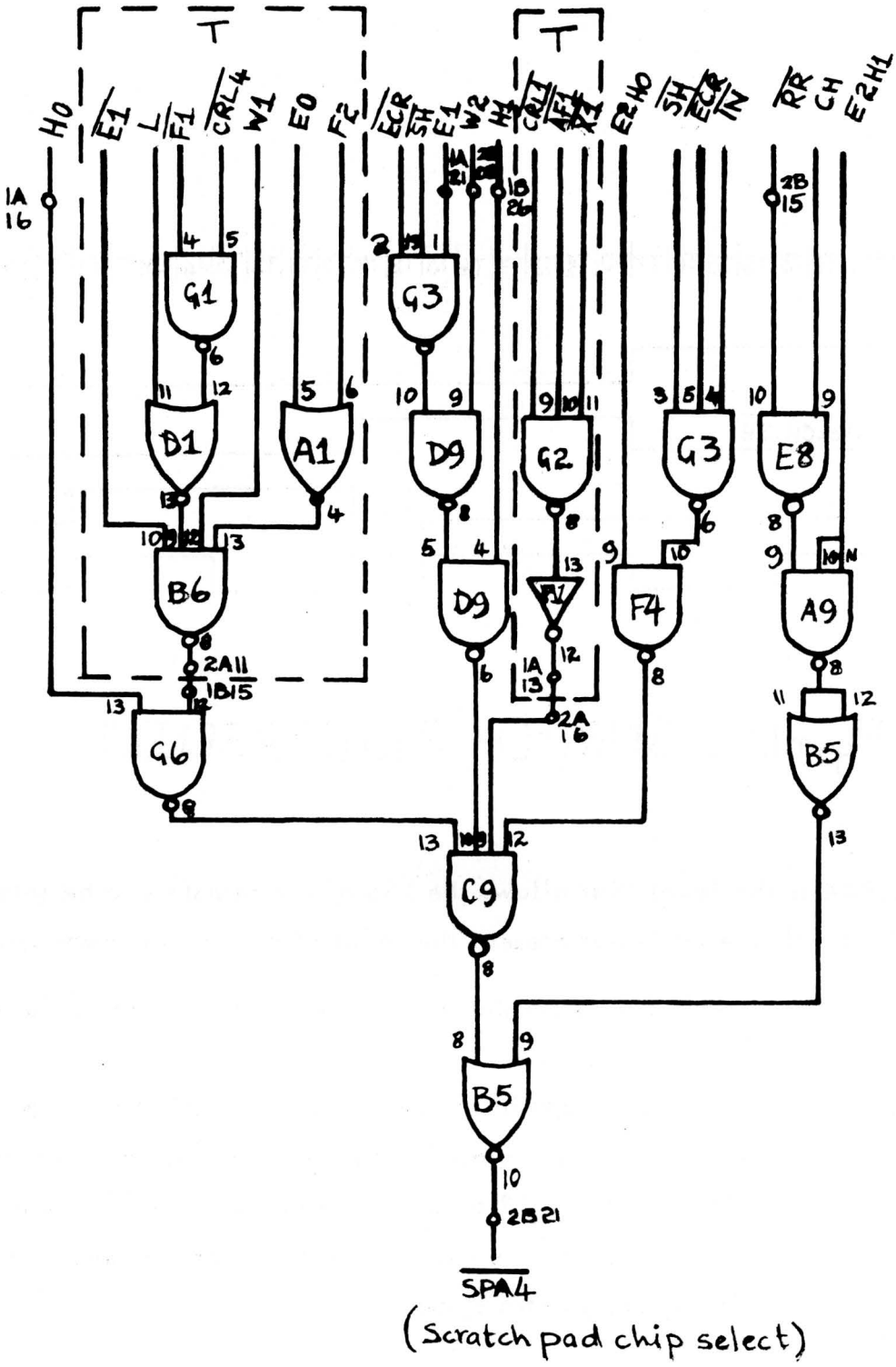


Figure C11b SCRATCH PAD CHIP SELECT

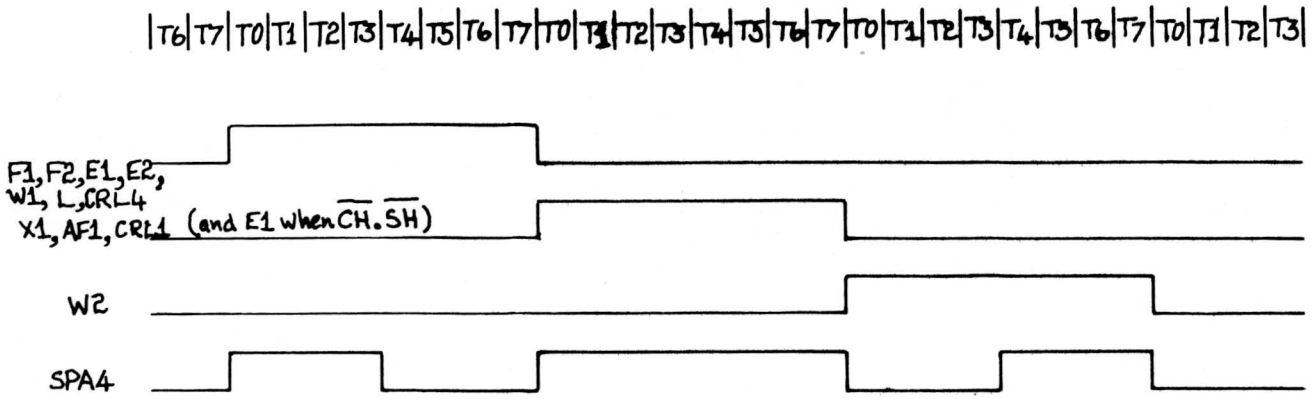


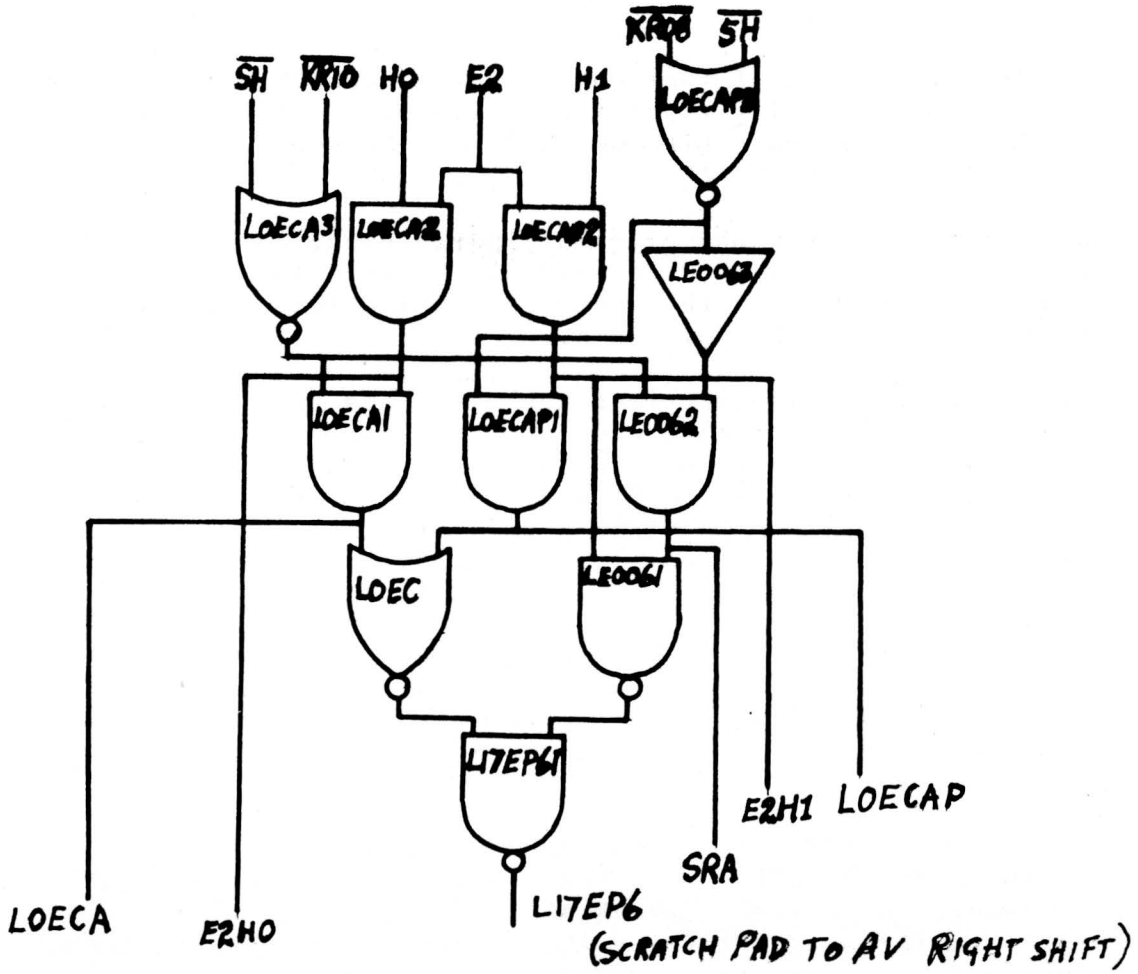
Figure C11a SCRATCH PAD CHIP SELECT

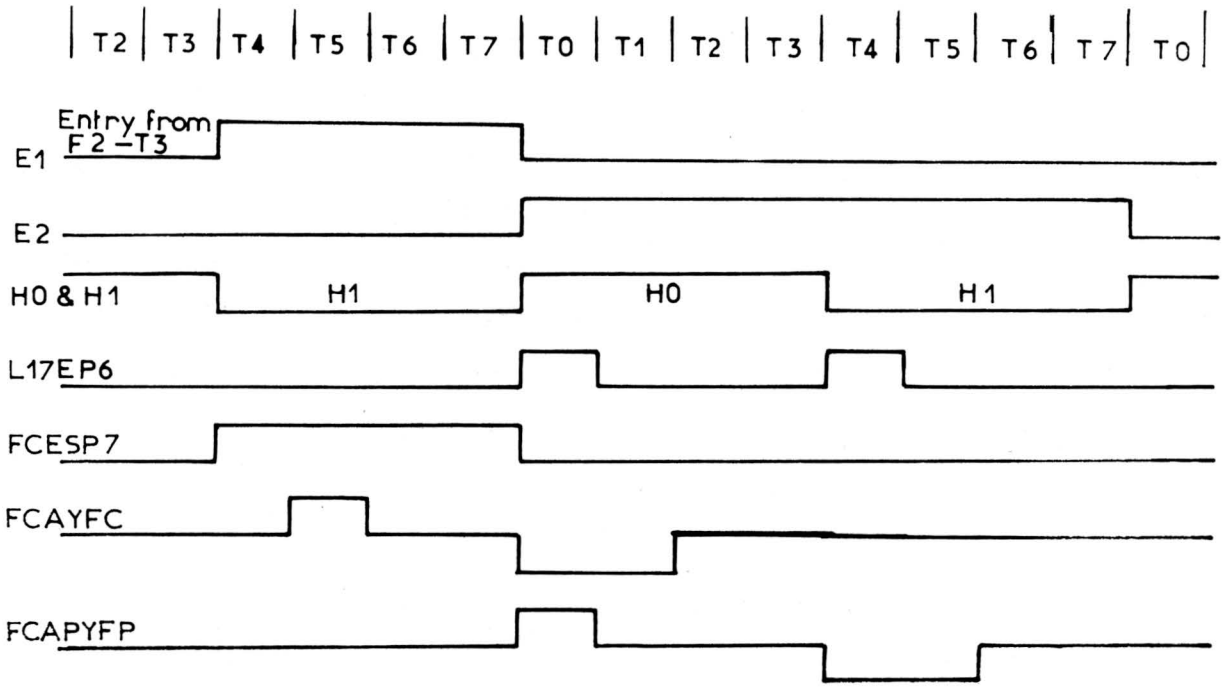
SPA4 is the level that allows the LSB of the registers to be selected. When this level is not present the MSBs of the registers are selected.

The above timing diagram shows cycles and times at which it is used.

Note: During shift right instructions the chip selection order is reserved. The MSBs are selected during the E1 cycle and then the LSBs and MSBs during the E2 cycle. This allows the carry logic to be used for shifting the rightmost bit into the leftmost bit position without using extra cycles.

SCRATCH PAD TO AU RIGHT SHIFT (AL GATE) PULSE





NOTE: Signals FCAYFC and FCAPYFP will never be negative as shown on this diagram, they are only shown this way to indicate the time at which the contents of the flip-flops are gated into the Most Significant bit of the AU.

L17EP6 is the signal that controls the scratch pad input to the AU during shift right instructions. It is used (diagram A8) to gate bits 00 to 06 of the scratch pad into bits 00 to 07 of the AU.

During the shift right circular instruction, both the FCA and FCP flip-flops are used to hold the least significant bit of each half-word until the appropriate time for gating into the most significant bit of the AU.

The shift right instructions are performed in the following way. During the E1 cycle, the state of bit 07 of the most significant half-word of the register is forced into the FCA flip-flop by signals FCESP7 and FCAYFC. At T0 time in the E2 cycle, the contents of this flip-flop are gated into the most significant bit position of the AU at the same time that bits 00 to 06 of the least significant half-word of the register are gated into bits 01 to 07 of the AU.

At T1 time, the state of bit 07 of the least significant half-word of the register is gated into the FCAP flip-flop by signal FCAPYFC.

At T4 time of the E2 cycle, bits 00 to 06 of the most significant half-word of the register are gated into bits 01 to 07 of the AU. If the instruction is shift right circular, the state of the FCAP flip-flop is gated into the most significant bit of the AU but, for shift right arithmetic instructions, the state of FCAP is ignored by the shift logic.

L17EP7 controls the gating of bits 01 to 07 of the scratch pad register to bits 01 to 07 of the AU.

LOESPO controls the gating of bit 00 of the scratch pad register to bit 00 of the AU. In shift right instructions, the signal inhibits bit 00 from being being gated into bit 00 of the AU.

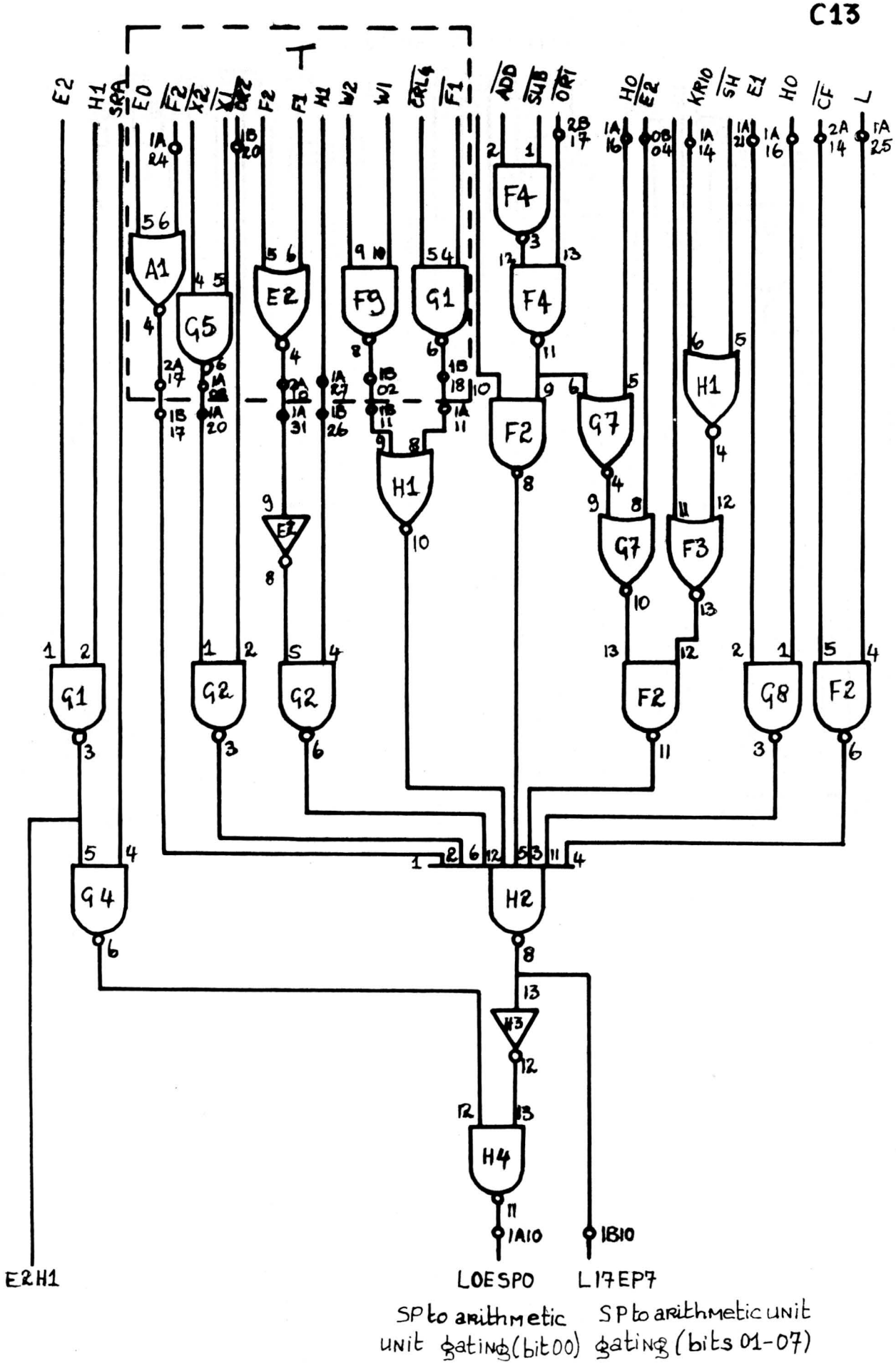
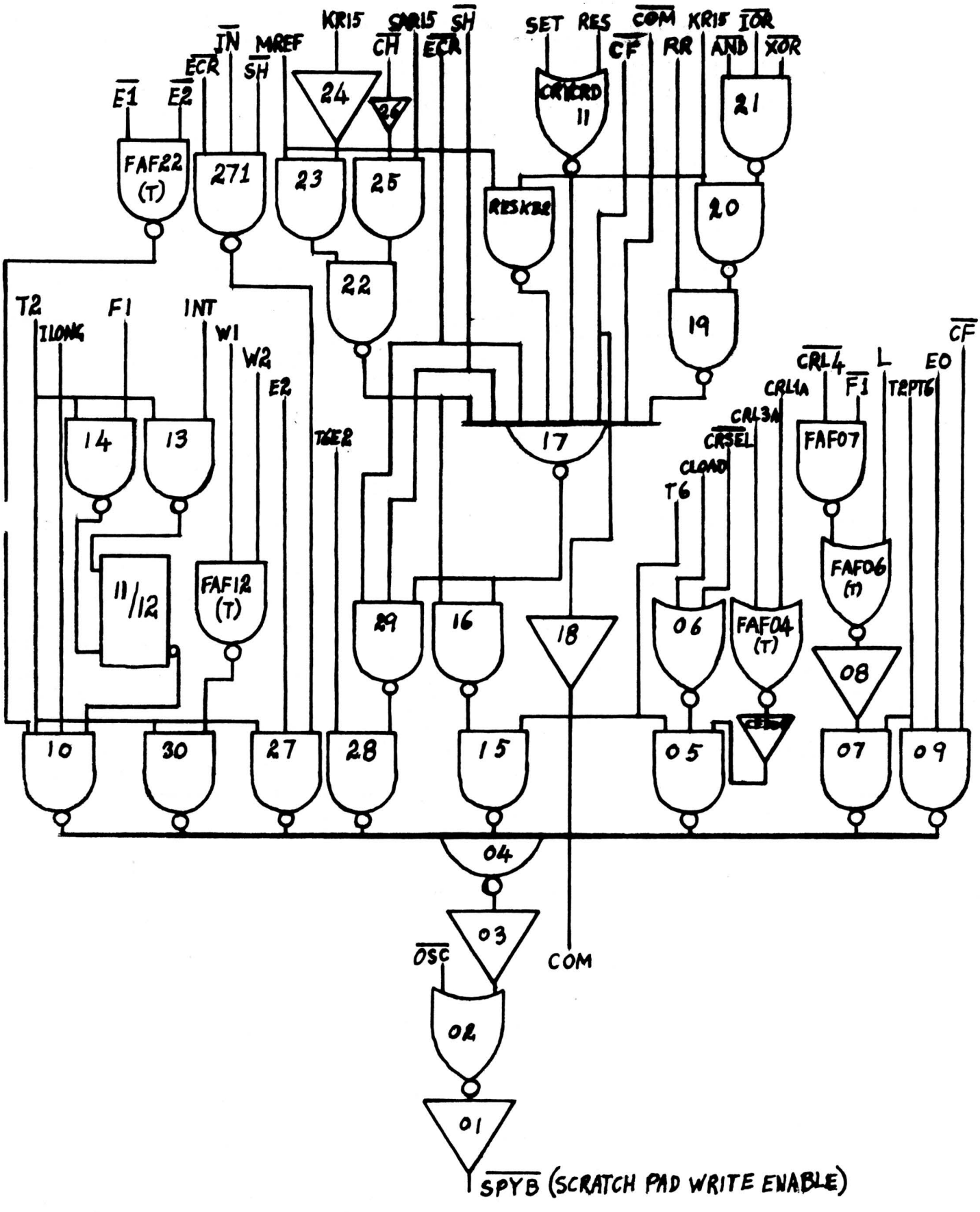


Figure C13

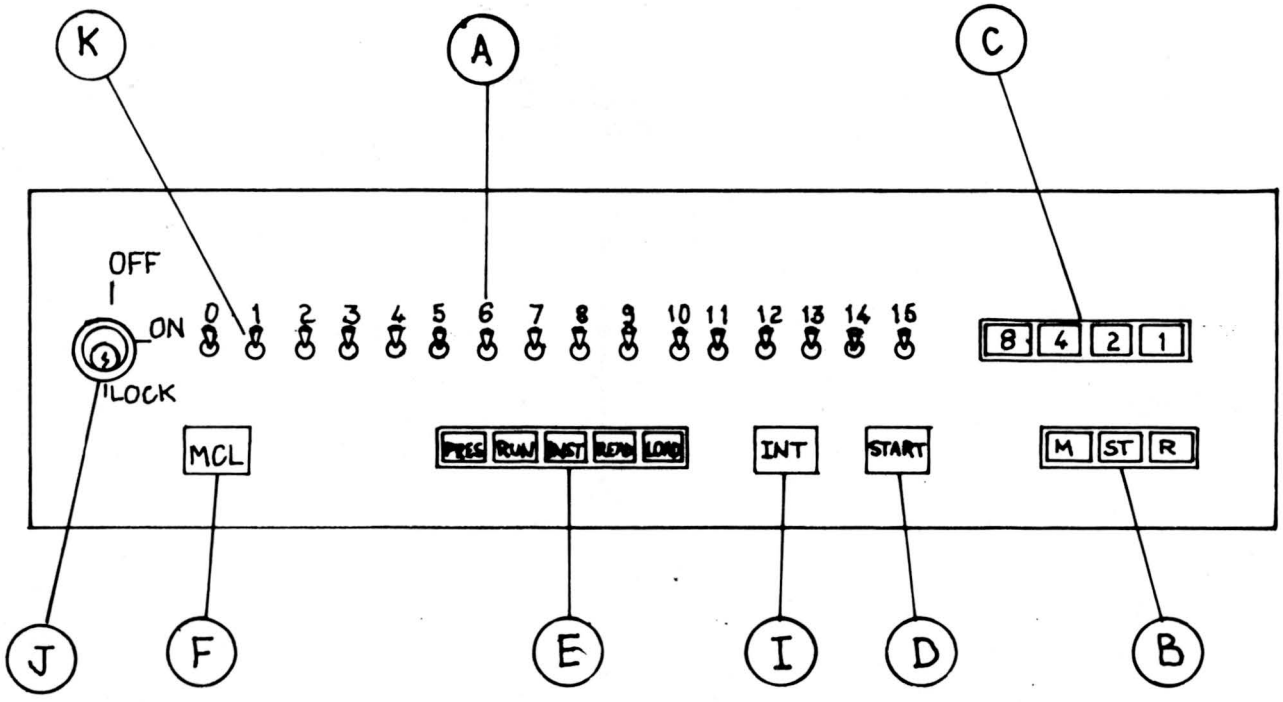
SCRATCH PAD TO AU (AL GATE) PULSE

SCRATCH PAD WRITE ENABLE



SPYB is the scratch pad write enable signal. It is produced at T2 and T6 time of the appropriate cycles and may or may not be used, depending upon the type of instruction being performed.

CONTROL PANEL



A - 16 indicators: used to display any functional register.

B - Register selector: selects the register to be displayed or loaded.

C - File register selector: when the B selector is in 'R' position these switches may be used to select one of the 16 multipurpose registers.

D - Start button/Run indicator: used to start any panel operations. An indicator in the button is light when the computer is running.

E - Mode selector: used to select one of the following modes:

RUN : Normal operating mode
INST : Instruction by instruction mode
READ : Read data from memory
LOAD: Write data into memory
PRES : Stop on preset address (not available on ALPHA model).

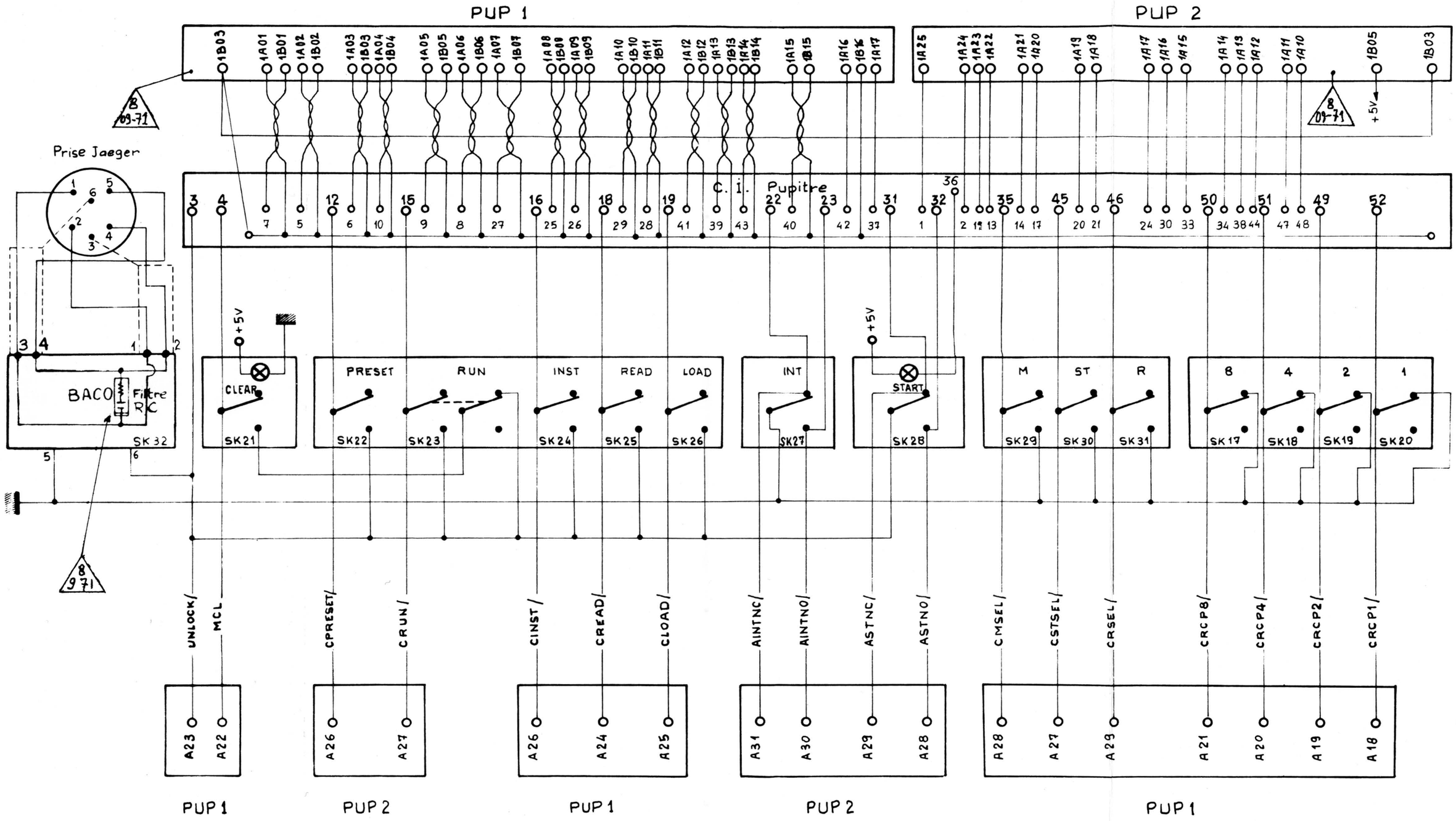
F - Master Clear push button.

I - Program interrupt push button

J - Control panel key lock used to switch the Power ON or OFF and to disable the other switches (lock position)

K - 16 data switches.

CONTROL PANEL LOGIC CONNECTIONS



CONTROL PANEL LOGIC SWITCHES

CRCP8 50 R17
CRCP4 51 R18
CRCP2 49 R19
CRCP1 52 R20
UNLOCK/ 3 R21
MCL 4 R22
C PRESET 12 R23
C RUN/ 15 R24
C INST/ 16 R25
C READ/ 18 R26
CLOAD/ 19 R27
AINTNC/ 22 R28
AINTNO/ 23 R29
ASTNC/ 31 R30
ASTNO/ 32 R31
CMSEL/ 34 R33
CSTSEL/ 45 R34
CRSEL/ 46 R35
R32

ABOU01 5 ABOU02 6 ABOU03 10 ABOU04 9 ABOU05 8 ABOU00 7
ABOU07 25 ABOU08 26 ABOU09 29 ABOU10 28 ABOU06 27
ABOU12 39 ABOU13 43 ABOU14 40 ABOU15 42 ABOU11 41

U1 U2 U3

R36 R37 R38 R39 R40 R41 C1
R42 R43 R44 R45 R46 C2
R47 R48 R49 R50 R51 C3

LR2 LR3 LR4 LR6 LR5 LR1
LR8 LR9 LR10 LR11 LR7
LR13 LR14 LR15 LR16 LR12

R2 R3 R4 R5 R6 R1
R8 R9 R10 R11 R7
R13 R14 R15 R16 R12

SK2 SK3 SK4 SK5 SK6 SK1
SK8 SK9 SK10 SK11 SK7
SK13 SK14 SK15 SK16 SK12

AFRUN 37 TS1
LR17 36 TS2

13 09.71

C KEY 01 2 C KEY 02 11 C KEY 03 13 C KEY 04 14 C KEY 05 17 C KEY 00 1
C KEY 07 21 C KEY 08 24 C KEY 09 30 C KEY 10 33 C KEY 06 20
C KEY 12 38 C KEY 13 44 C KEY 14 47 C KEY 15 48 C KEY 11 35

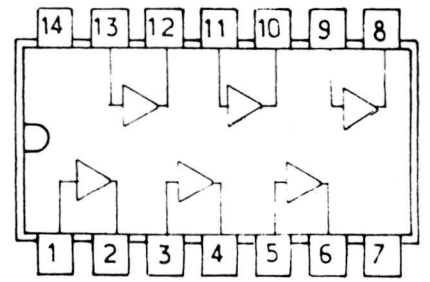
R34 33 +5V
C4
0V

13 09.71

+12V

CP1

SN 7416
U1, U2, U3



The three hardware options available for the P850 are all contained on one card called OPTI. These options are:

additional interrupt lines
power fail/automatic restart
real-time clock

and one, two or all three options may be included on the card.

The logic for each option is described and illustrated in the following sections.

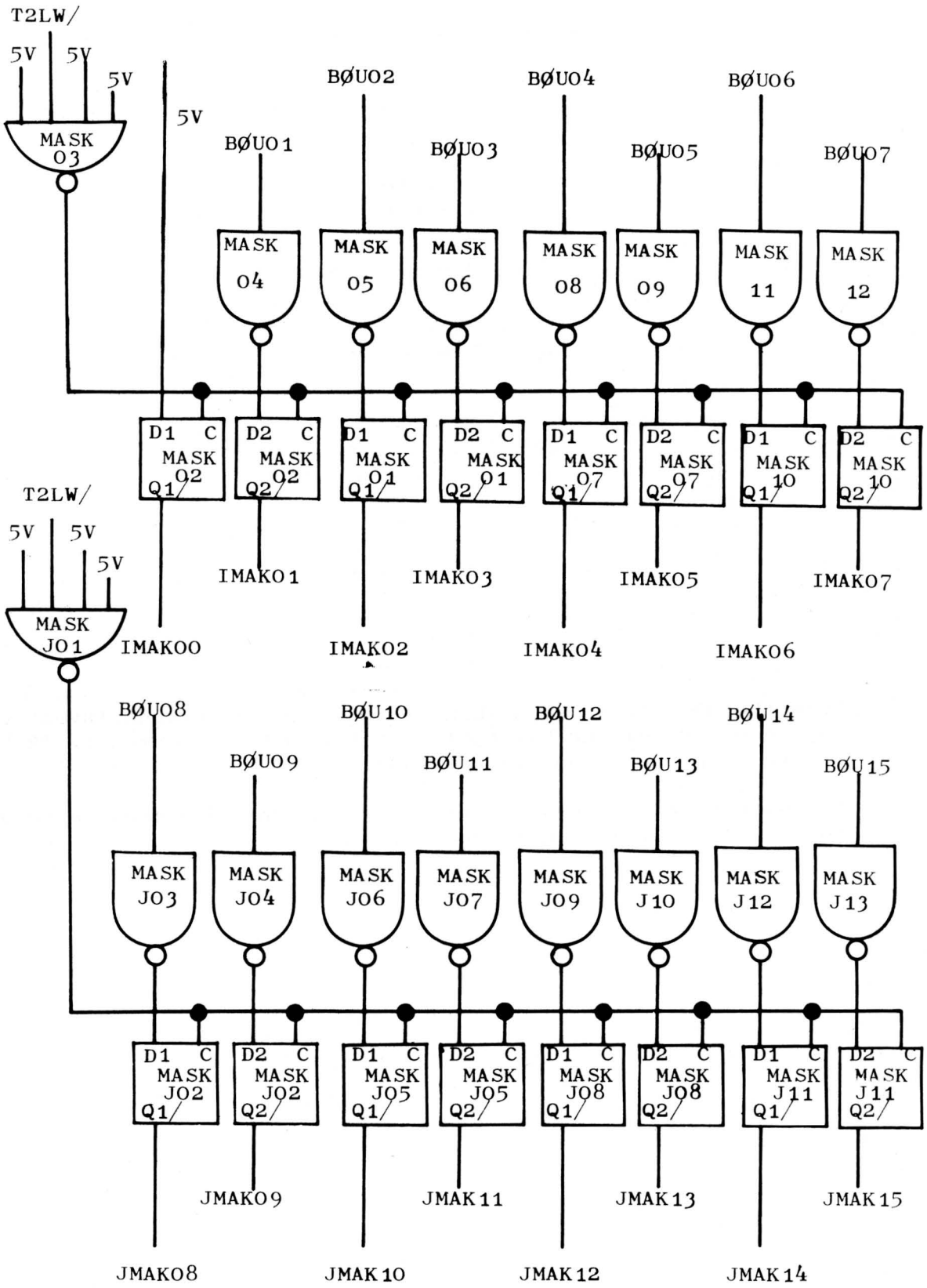


Figure 3.1 Mask register logic

3.1 INTERRUPT LOGIC

A single interrupt line is fitted as standard to all P850 processors and the logic for this line will be found on diagram T11 in Part 2 of this book.

The logic for the additional interrupt lines can be added as two separte groups. The first contains logic for 7 additional lines and the logic elements can be identified by the letter I. If this group is added to the standard interrupt line, a total of 8 interrupt lines can be serviced. When more than 8 interrupt lines are needed, logic for a group of 8 more interrupt lines can be added giving a maximum of 16 interrupt lines. The logic elements of this second group can be identified by the letter J.

The standard interrupt line cannot be masked, but all the addtional lines can be masked by bi-stable latches shown on Figure 3.1 using the WIM instruction.

This instruction loads the contents of a register (which has previously had the required mask pattern loaded into it) on to the BØU lines. The register into which the mask pattern has been loaded will have a 1 in the bit positions of the lines which are to be masked and a 0 in the bit positions of the lines which are to be enabled.

From the BØU lines the mask pattern is gated through inventors to the D side of the bi-stable latches and is clocked by T2LW/ (which is produced on diagram T11).

The output levels from the bi-stable latches are used as input to the NØR elements on Figures 3.2 and 3.3.

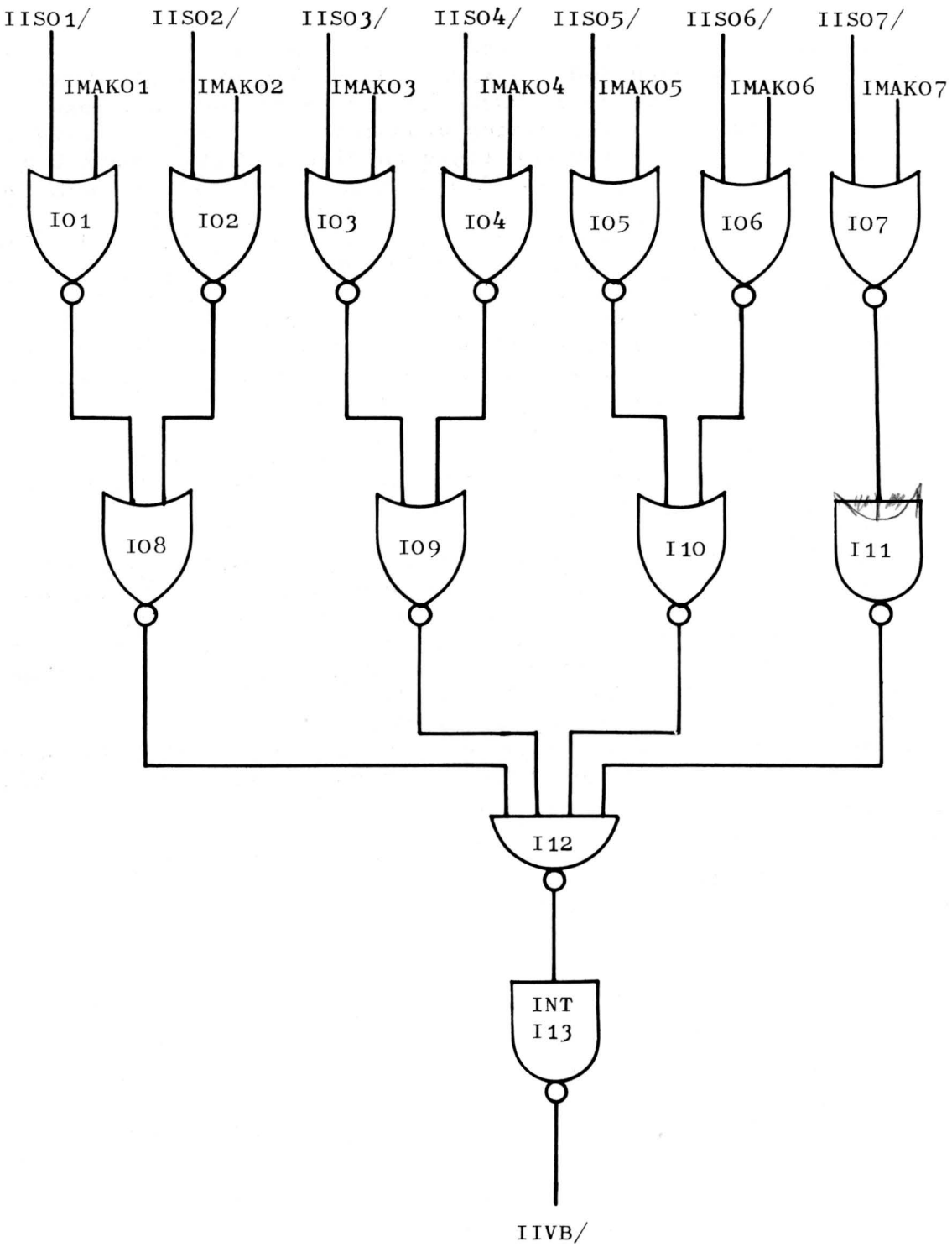


Figure 3.2 Optional interrupt logic (first 7 additional lines)

The logic on both Figures 3.2 and 3.3 functions in the same way.

Levels from the mask bi-stables and from the interrupt lines are gated through NØR elements to two 4 input NAND gates, I12 and J13. Any one of these inputs going low will result in levels IIUB/ and JIVC/ being produced. These two levels are used by the interrupt logic on diagram T11 in part 2 of this book.

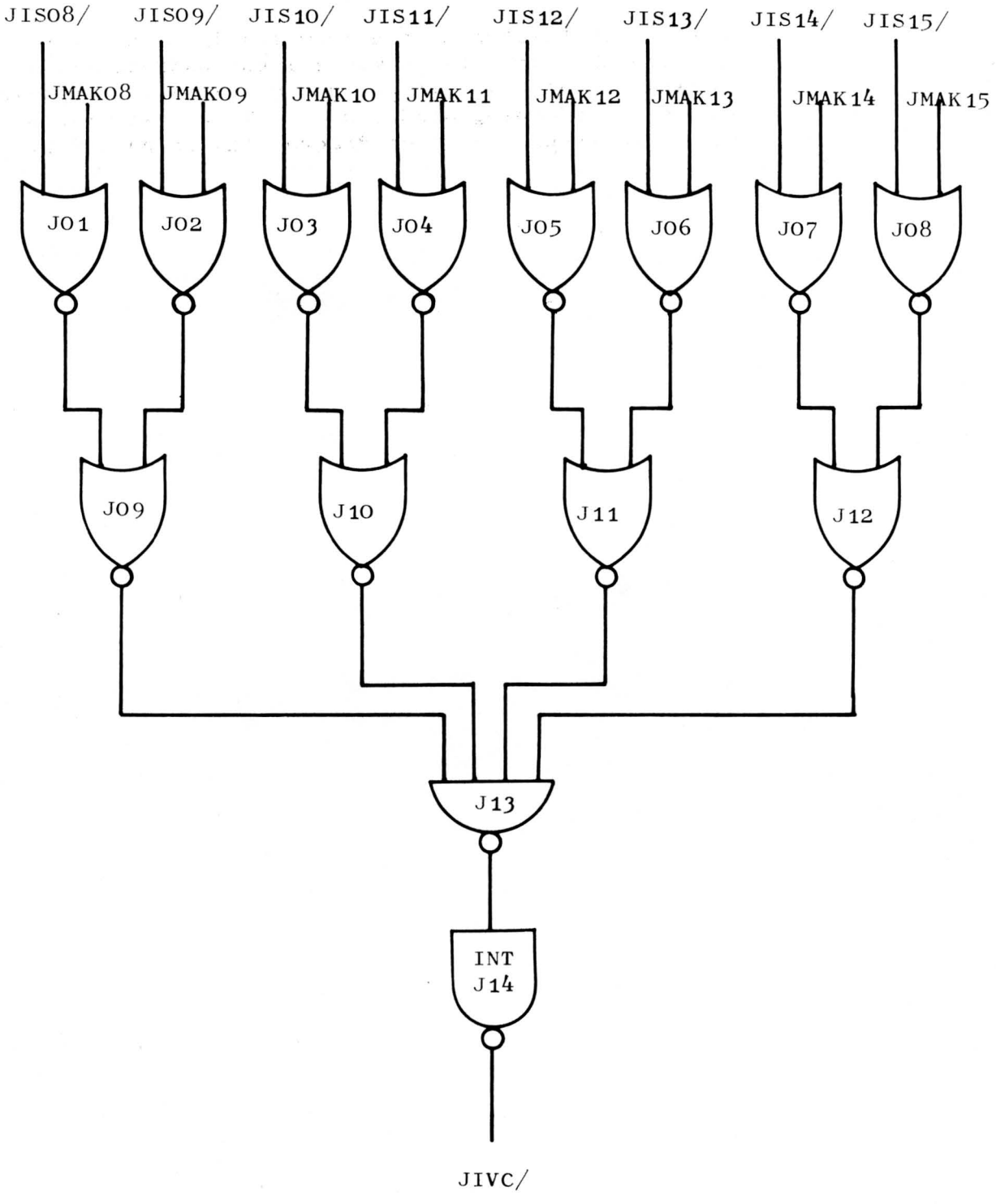


Figure 3.3 Optional interrupt logic (second 8 additional lines)

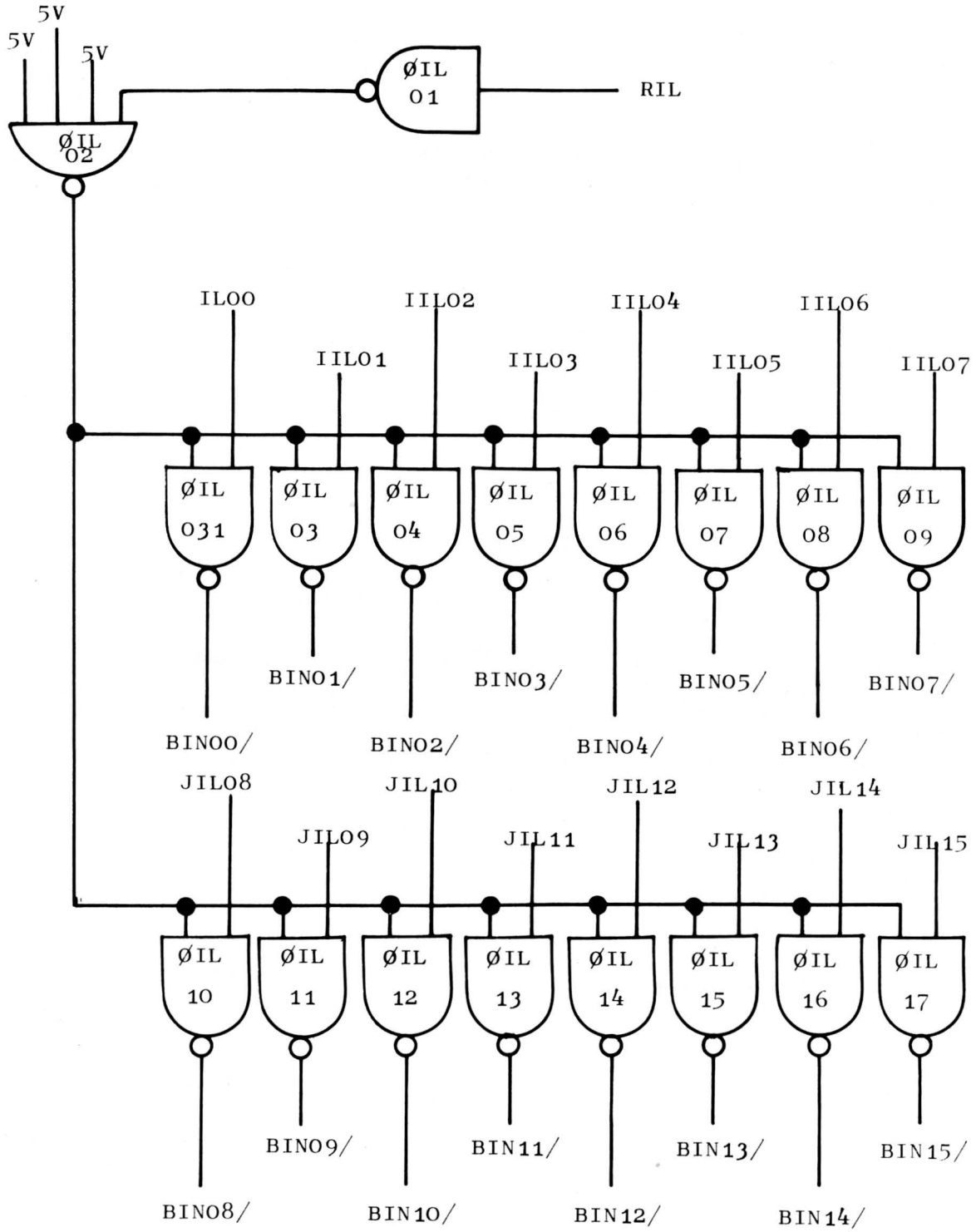


Figure 3.4 Read interrupt lines logic

The logic on this diagram is used by the RIL instruction. This instruction allows the state of the interrupt lines to be loaded into a register.

The interrupt lines are connected to one side of 2 input NAND gates. These gates are enabled by the level RIL which is produced on diagram C8 in part 2 of this book.

The outputs from these gates are connected to the BIN lines and are gated into the selected register during the E1 and E2 cycles of the instruction.

Active and not masked interrupt lines are indicated by a 1 in the corresponding bit positions.

Inactive or masked lines are indicated by a 0.

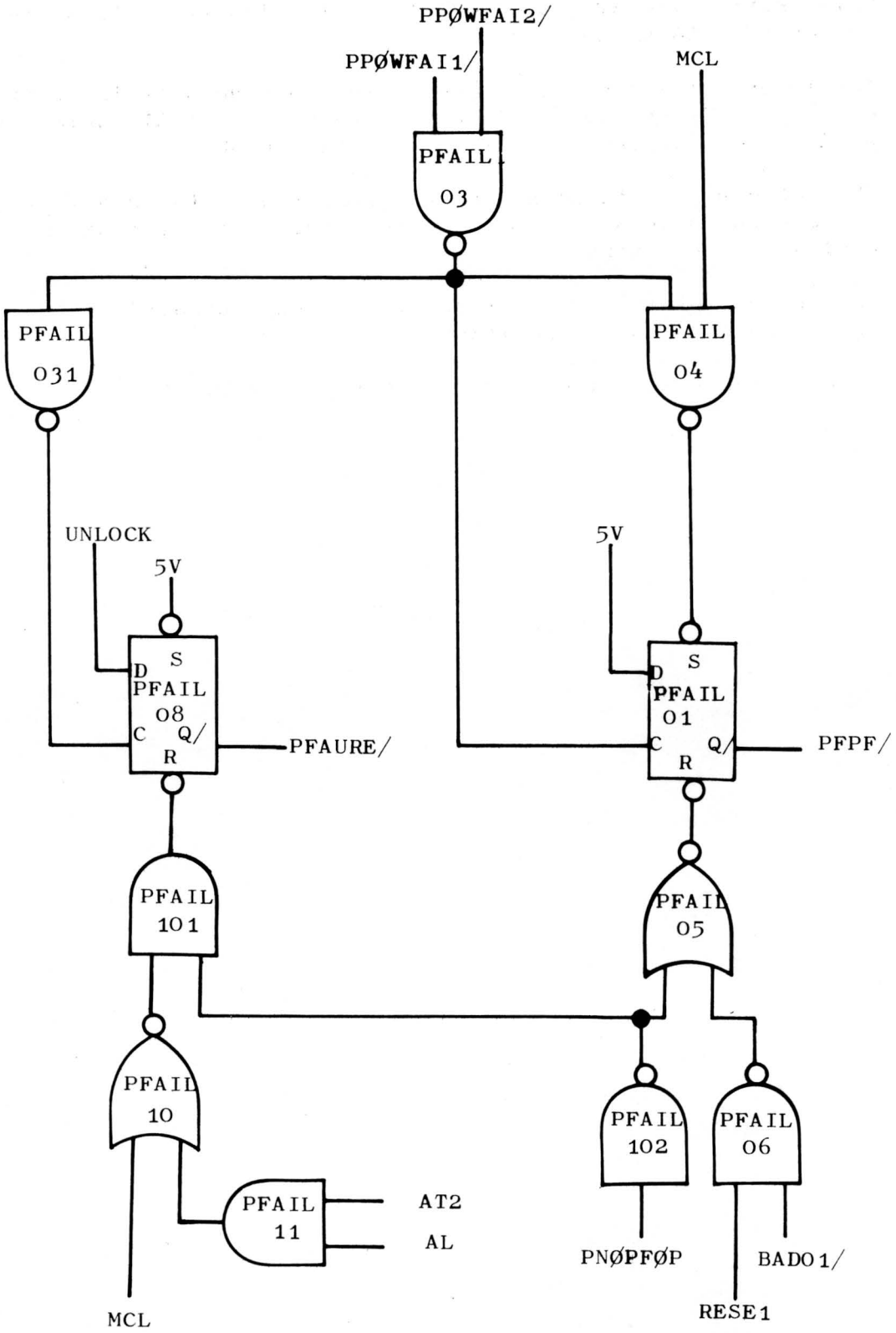


Figure 3.5 Power fail/automatic restart logic

3.2 POWER FAIL

When this option is included on the card, the output from the PFAILO1 flip-flop must be connected to the standard interrupt line INOO which is not maskable.

When the power supply(ies) is (are) operating, the inputs to PFAILO3 are high (1). A power failure sends the output of this gate high and this high is used to trigger flip-flops PFAILO8 and PFAILO1.

PFPF/ is used to generate an interrupt and this interrupt carryies out a routine to store information, relevant to the current instruction, in the memory. The interrupt will occur 1 millisecond before the d.c; supplies fall below the normal operating level.

PFAURE/ is used (on diagram T5) for the automatic restart routine to enable the FRUN flip-flop.

This routine will operate only if the key-lock is in the LOCK position. One of the things this routine does is to generate MCL to clear all the control flip-flops (including the power/fail flip-flops).

NOTE: A break in mains voltage of less than 20 milliseconds will not be detected.

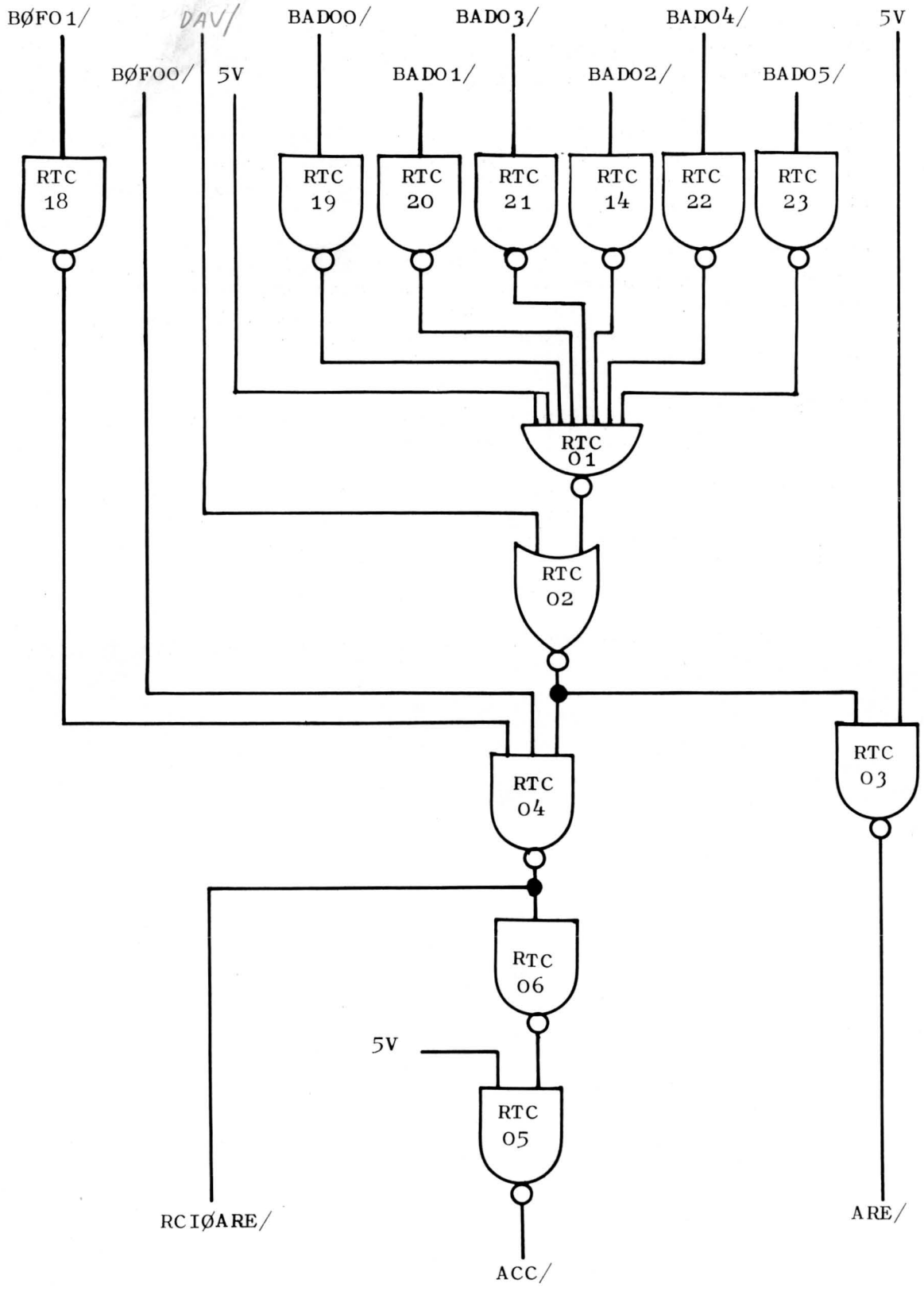


Figure 3.6 Real-time clock logic

3.3 REAL TIME CLOCK

The real-time clock can be turned ON or OFF by a CIØ instruction. It is programmed in the same way as a peripheral device and it has a fixed address of 63.

When this option is addressed by program, the BAD and BØF lines activate the logic on Figure 3.6 producing levels RCIØARE/, ACC/ and ARE/ which are used on Figure 3.7.

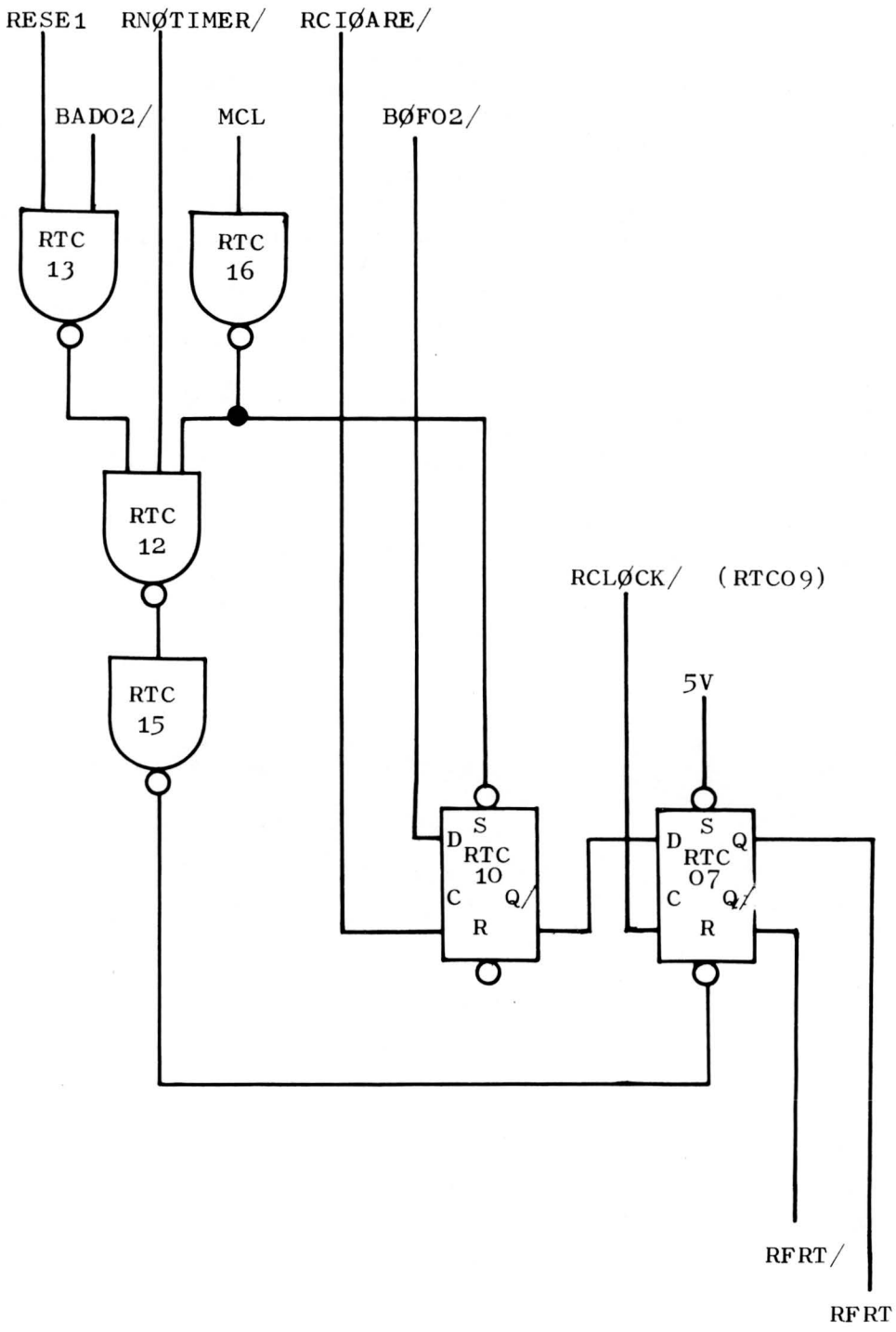


Figure 3.7 Real-time clock logic

The internal timer is synchronised by the frequency of the mains voltage supply to which the CPU is connected. It produces 20 millisecond pulses (RCLØCK) and these are used to trigger flip-flop RTC07. The output from this flip-flop (RFRT) generates an interrupt which is used on the ØAUR card.

The clock is turned ON or OFF by flip-flop RTC10.

This part is divided into 3 sections. The first section contains the ICM diagrams that show the pin connections of each individual integrated circuit module.

The second section contains the diagrams of each individual card for the basic processor. These show the types of module and their relative position on the card, and the pin connections for each card.

The last section contains diagrams that show all other pin and cable connections.

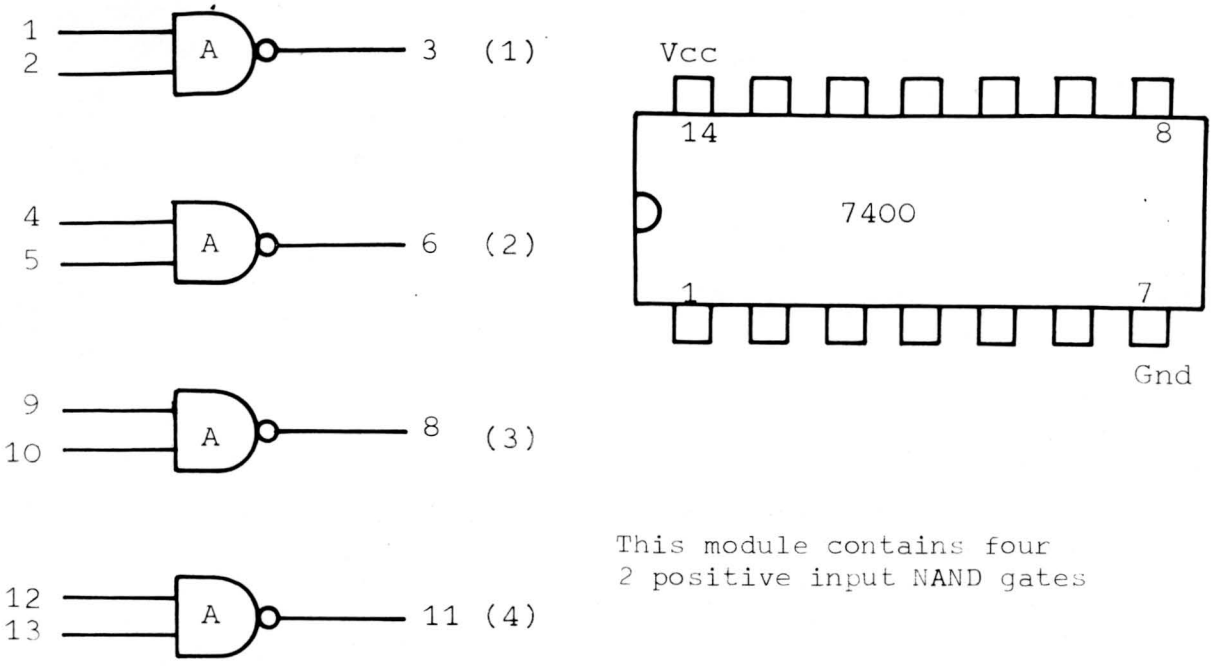


Figure 4.1 Module 7400

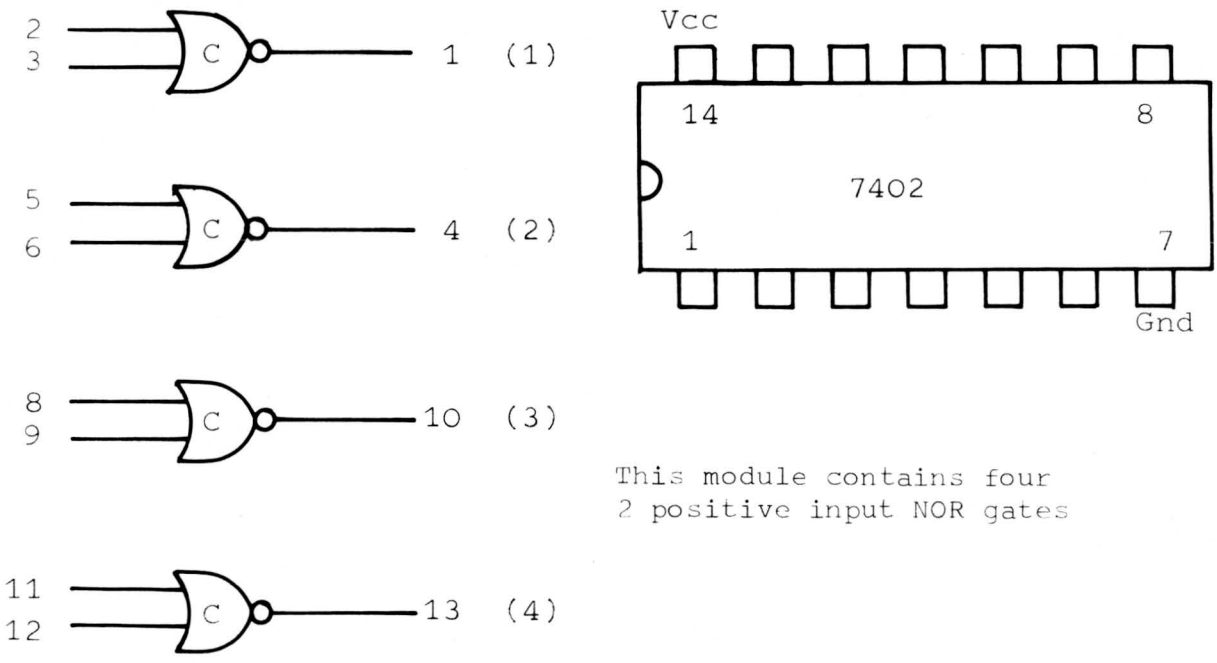


Figure 4.2 Module 7402

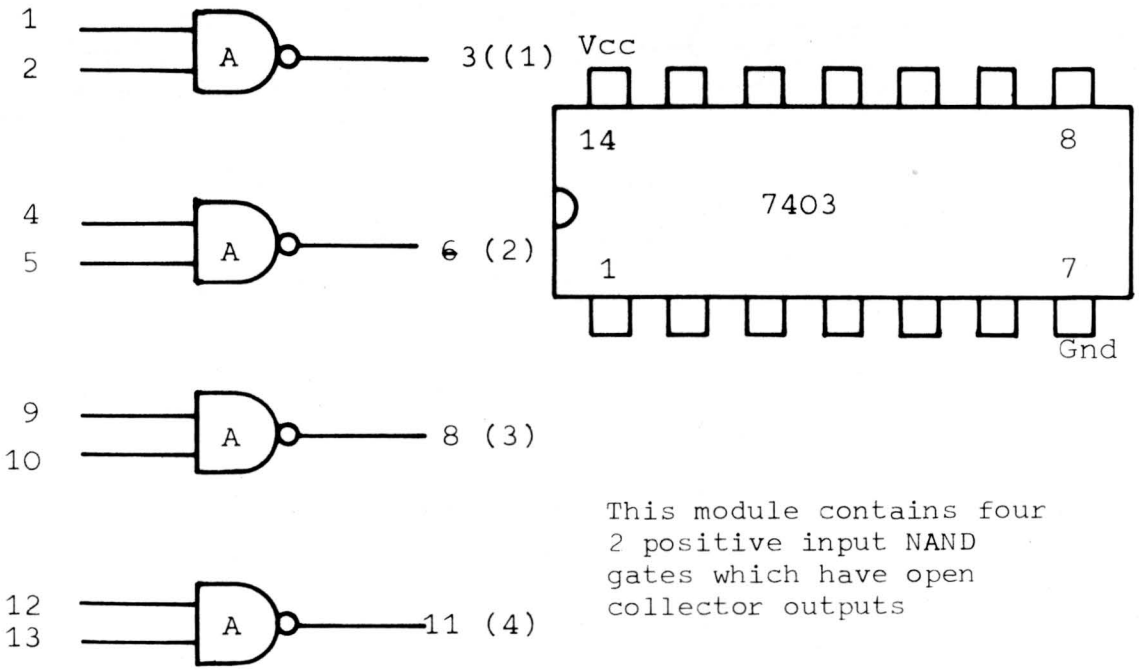


Figure 4.3 Module 7403

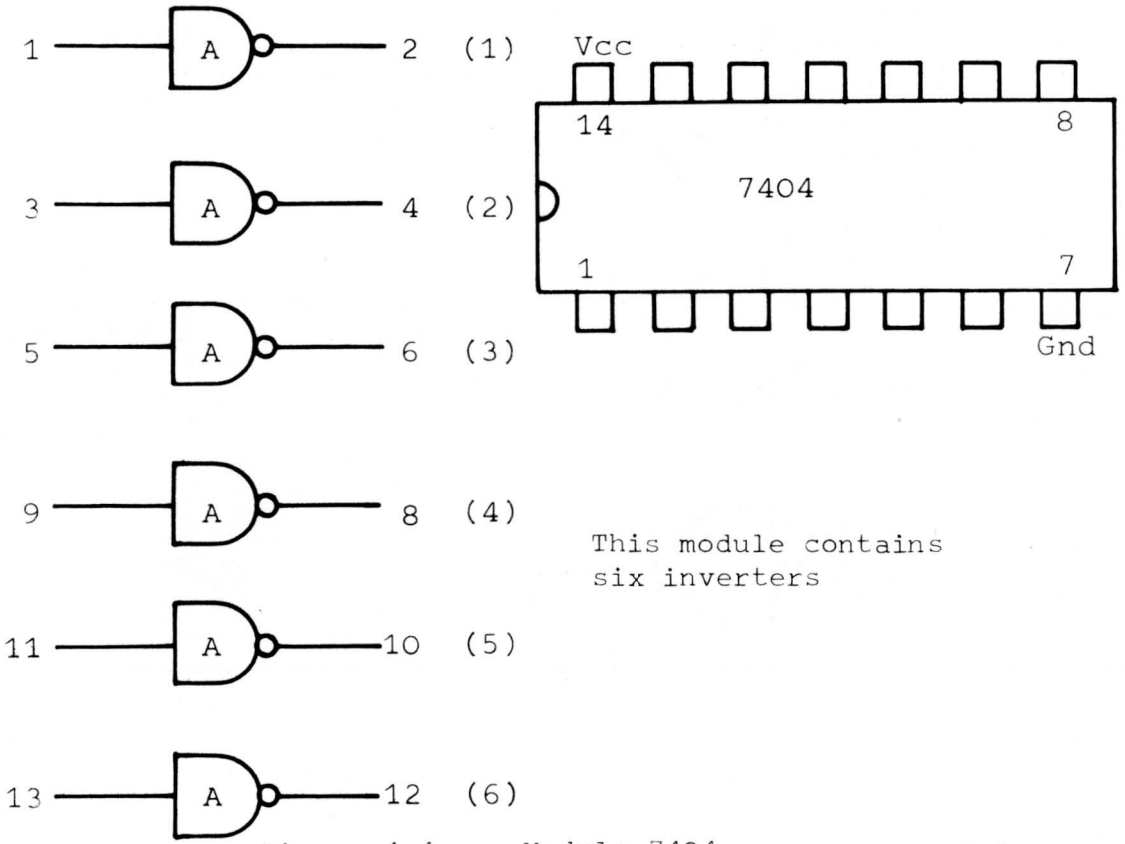


Figure 4.4 Module 7404

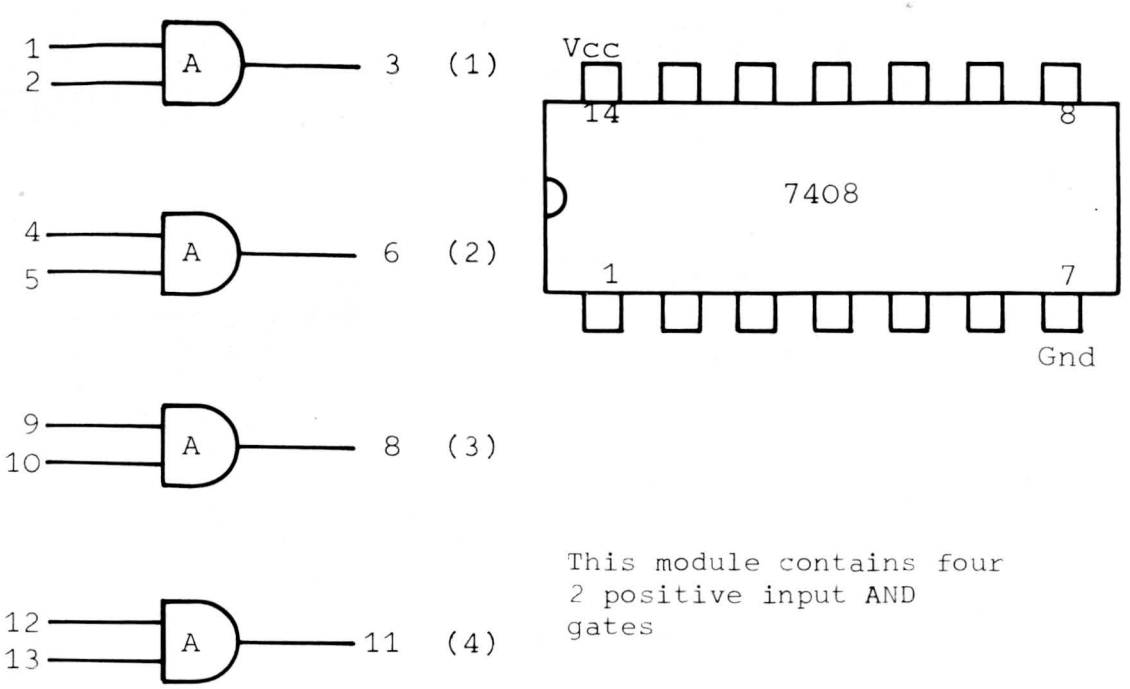


Figure 4.5 Module 7408

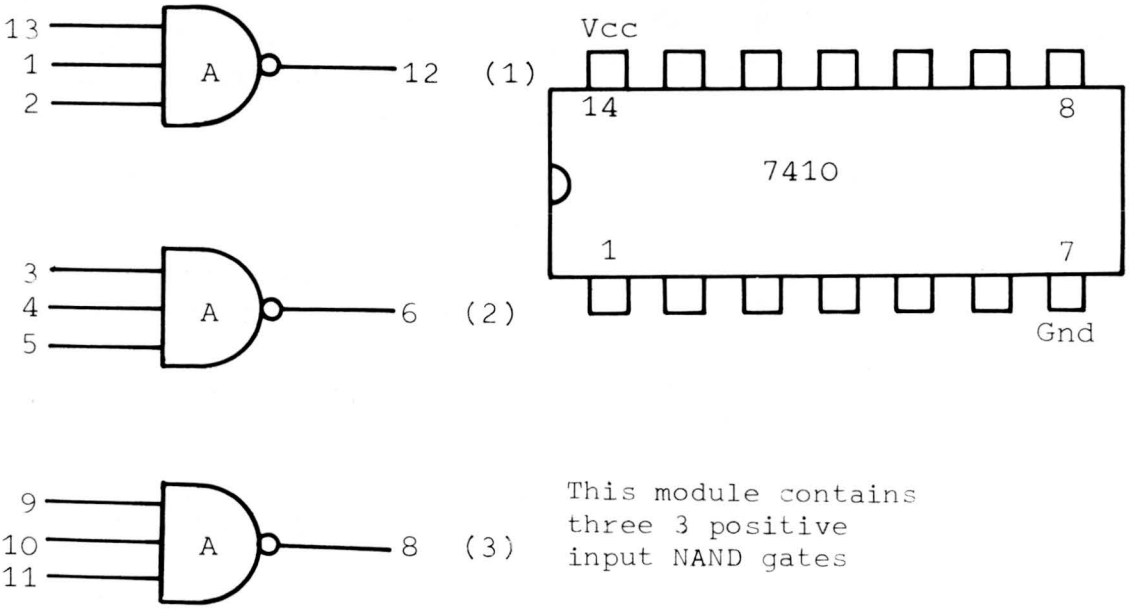
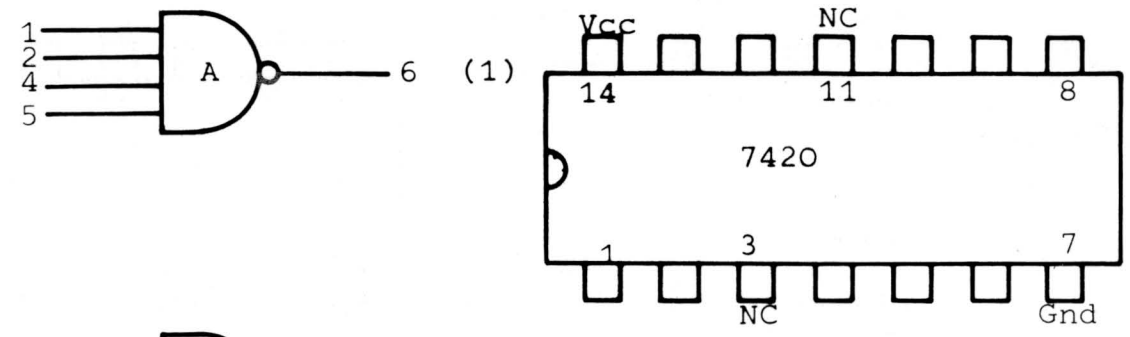
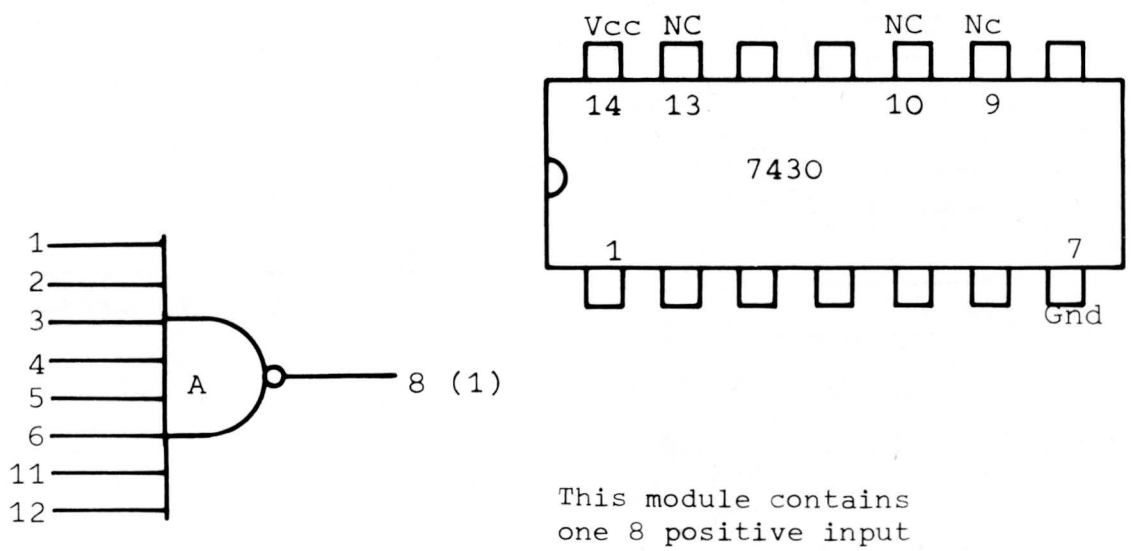


Figure 4.6 Module 7410



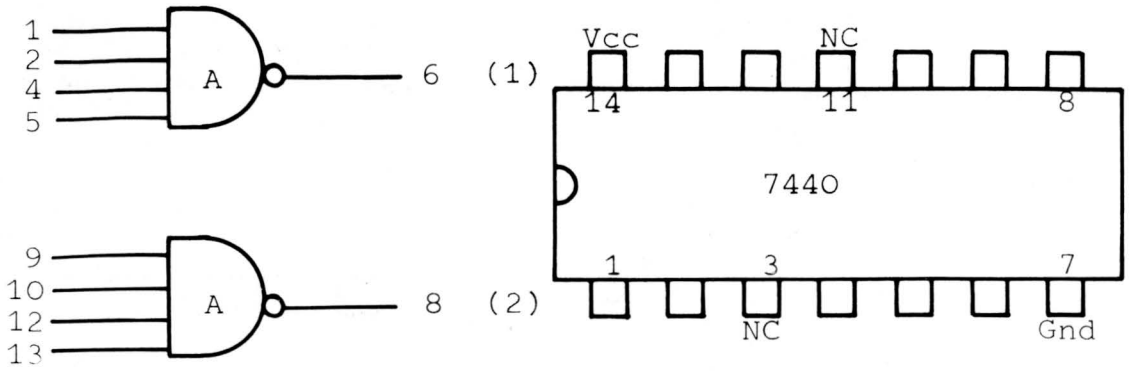
This module contains two 4 positive input NAND gates

Figure 4.7 Module 7420



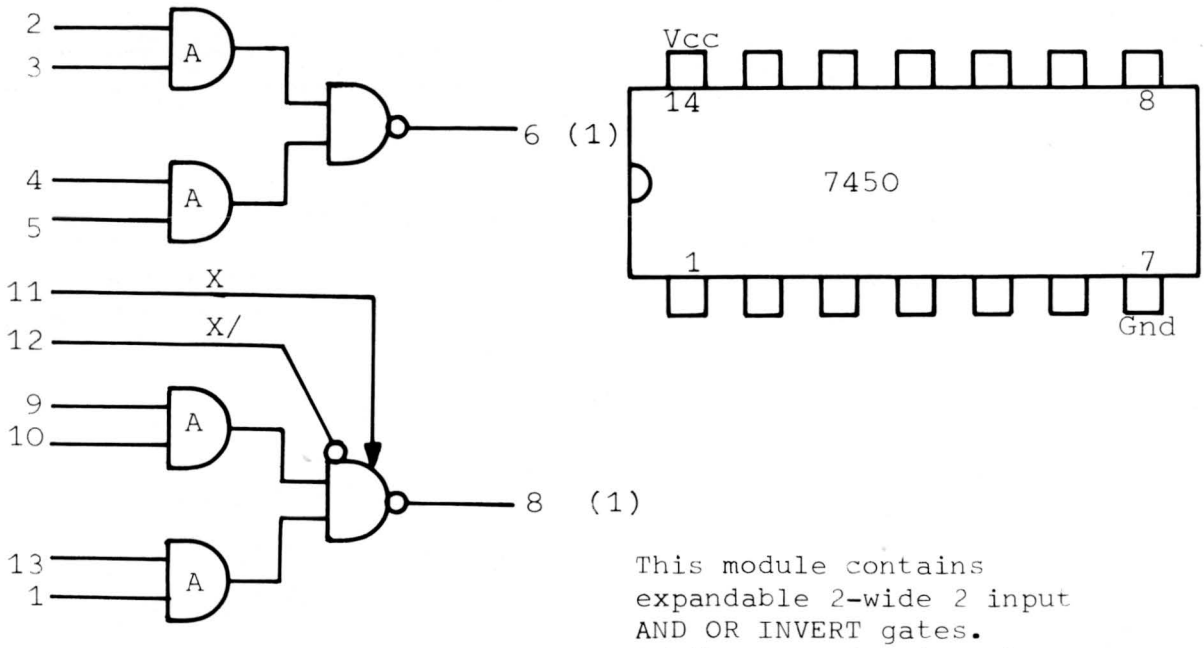
This module contains one 8 positive input NAND gate

Figure 4.8 Module 7430



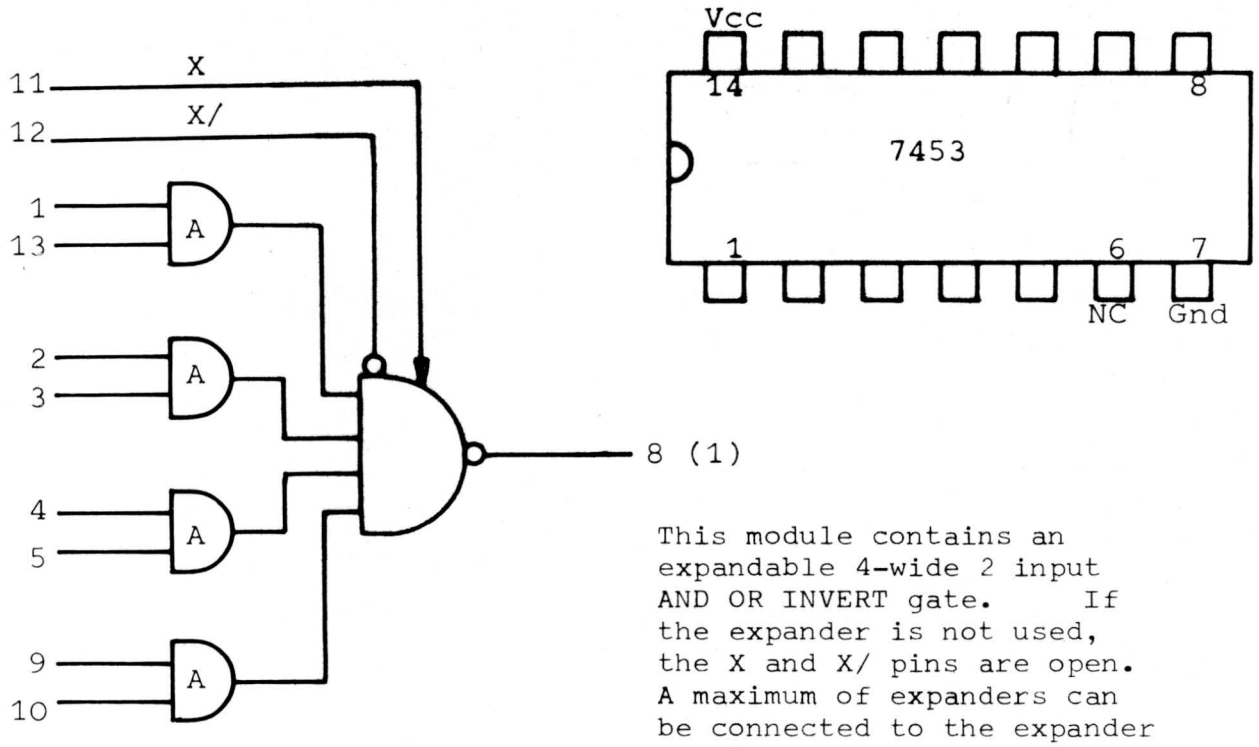
This module contains two
4 positive input NAND gates

Figure 4.9 Module 7440



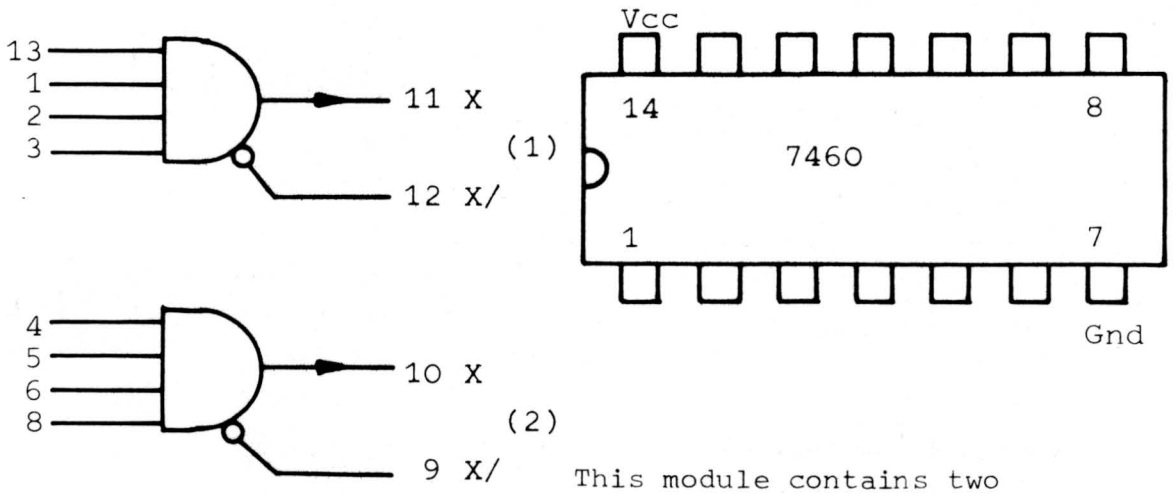
This module contains
expandable 2-wide 2 input
AND OR INVERT gates.
If the expander is not
used, the X and X/ pins
are open. A maximum
of four expanders can be
connected to the expander
inputs

Figure 4.10 Module 7450



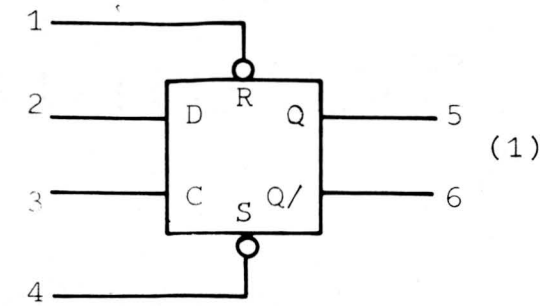
This module contains an expandable 4-wide 2 input AND OR INVERT gate. If the expander is not used, the X and X/ pins are open. A maximum of expanders can be connected to the expander inputs

Figure 4.11 Module 7453

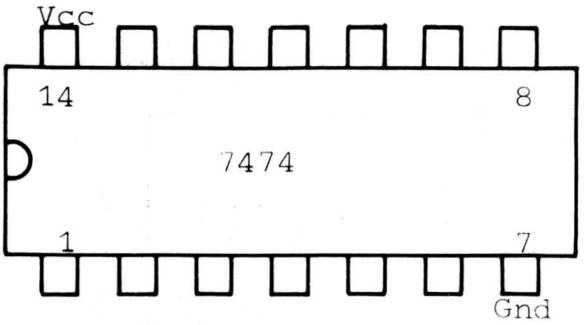


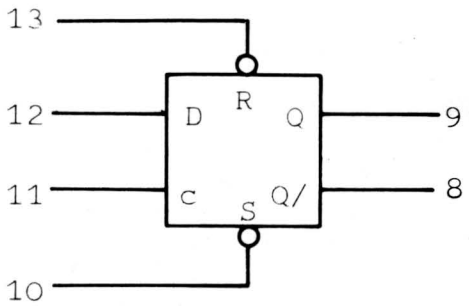
This module contains two 4 input expander gates. A maximum of four expanders can be input into one 7450 or 7453 module

Figure 4.12 Module 7460



(1)

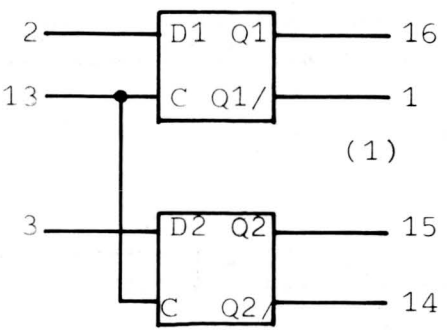


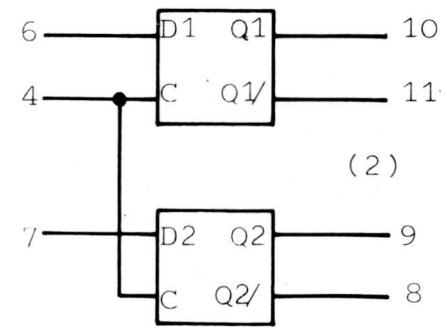


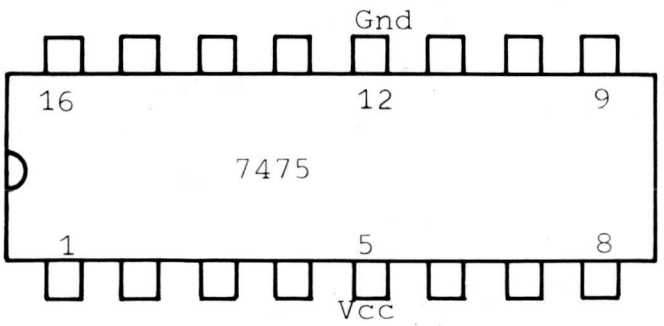
(2)

This module contains two D-type edge triggered flip-flops. If R and S are not used, they must be connected to a fixed voltage

Figure 4.13 Module 7474

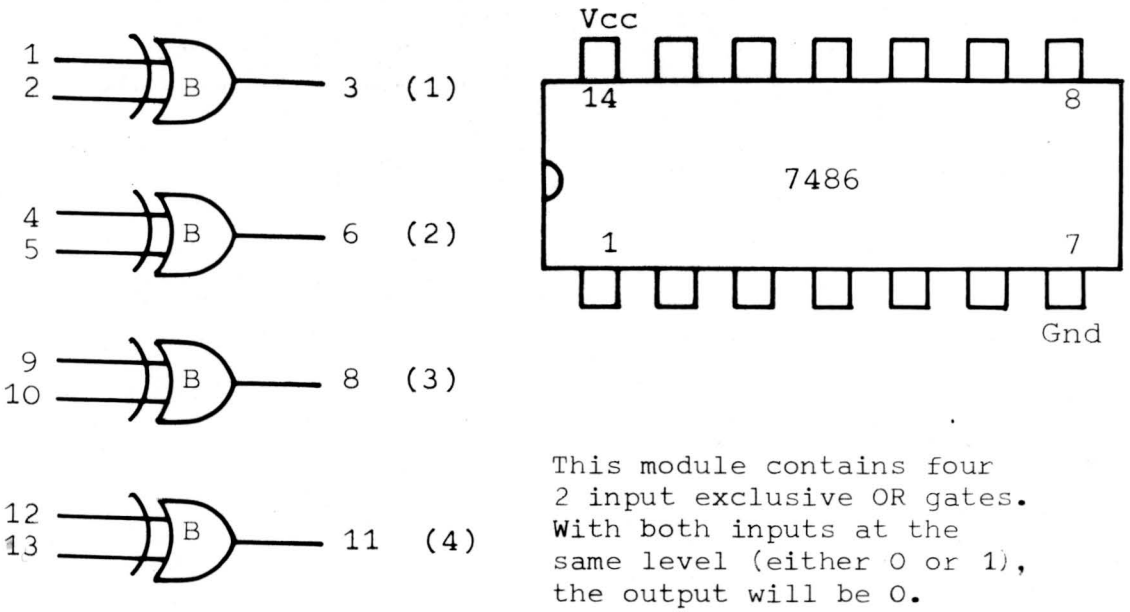






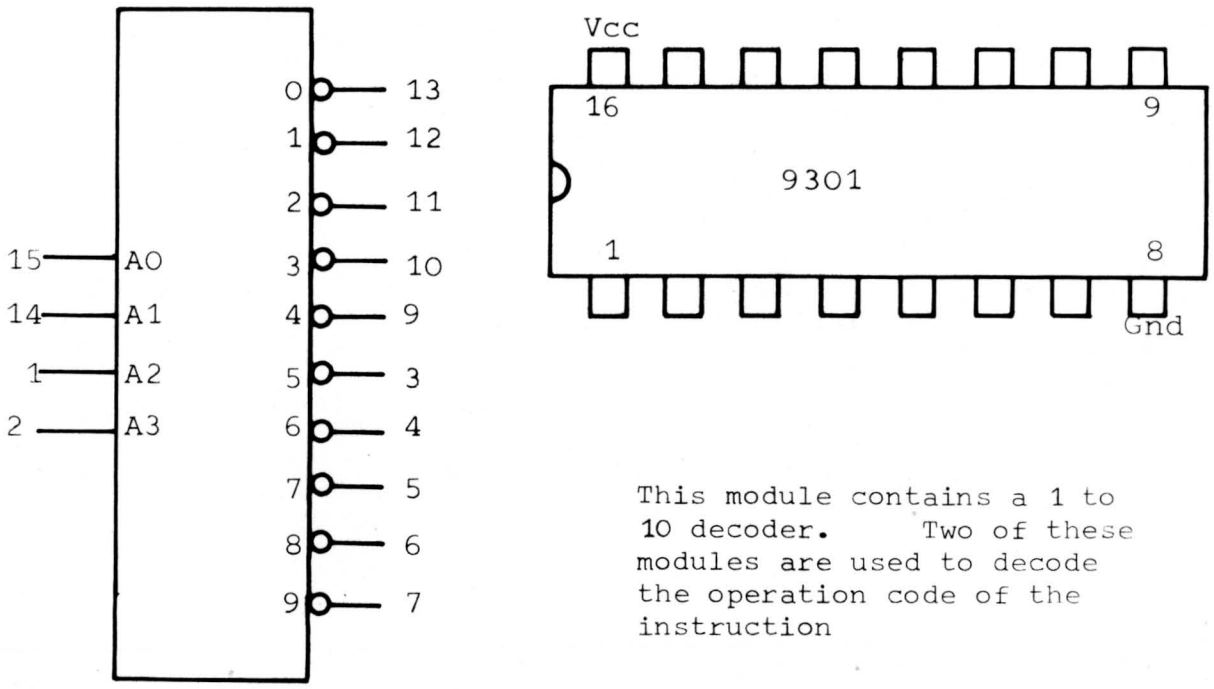
This module contains two 2-bit bi-stable latches with a clock pulse common to two bits

Figure 4.14 Module 7475



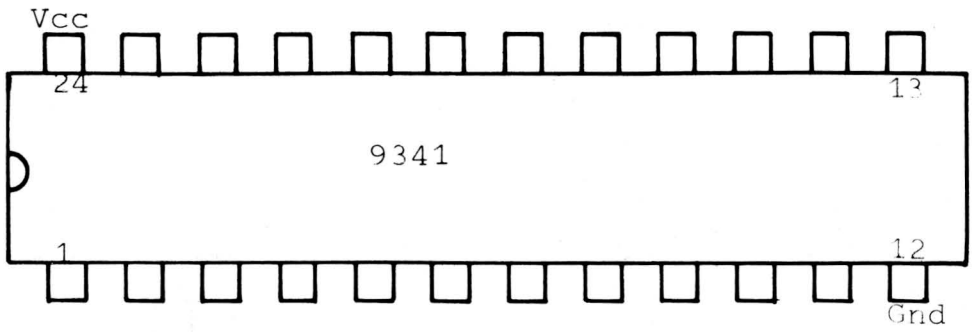
This module contains four 2 input exclusive OR gates. With both inputs at the same level (either 0 or 1), the output will be 0. With both inputs different, the output will be 1

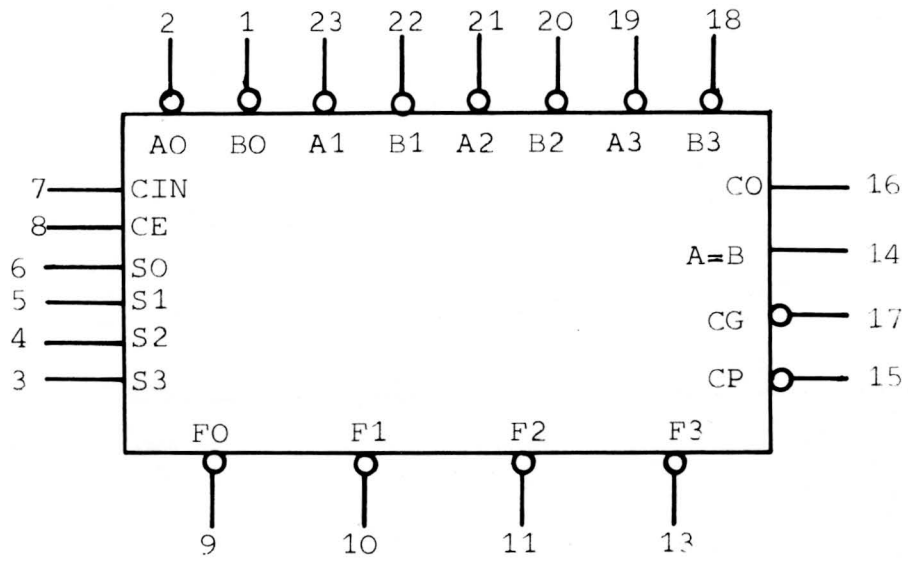
Figure 4.15 Module 7486



This module contains a 1 to 10 decoder. Two of these modules are used to decode the operation code of the instruction

Figure 4.16 Module 9301

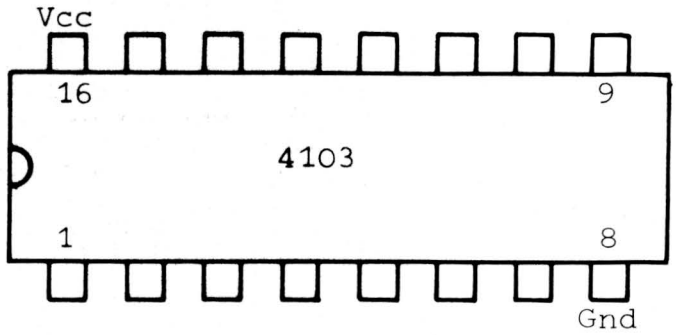


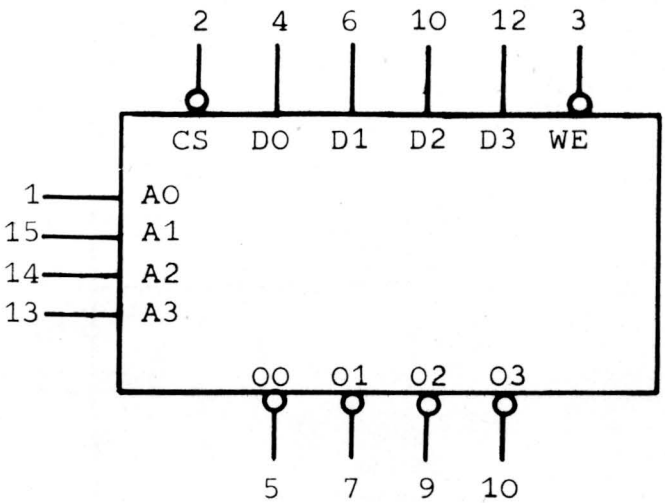


This module contains a 4-bit arithmetic/logic unit. Two of these modules are used to perform the arithmetic and logic functions of the processor

zelfde pinkonfiguratie als SN74181

Figure 4.17 Module 9341





This module contains a 64-bit random access memory unit. It is organized as sixteen 4-bit words with inputs AO to A3 used for addressing purposes, inputs DO to D3 used for data input; the output lines are O0 to O3. CS is the chip select input pin and pin WE is for the write enable signal. Four of these modules are used for the scratch pad registers in the processor

Figure 4.18 Module 4103

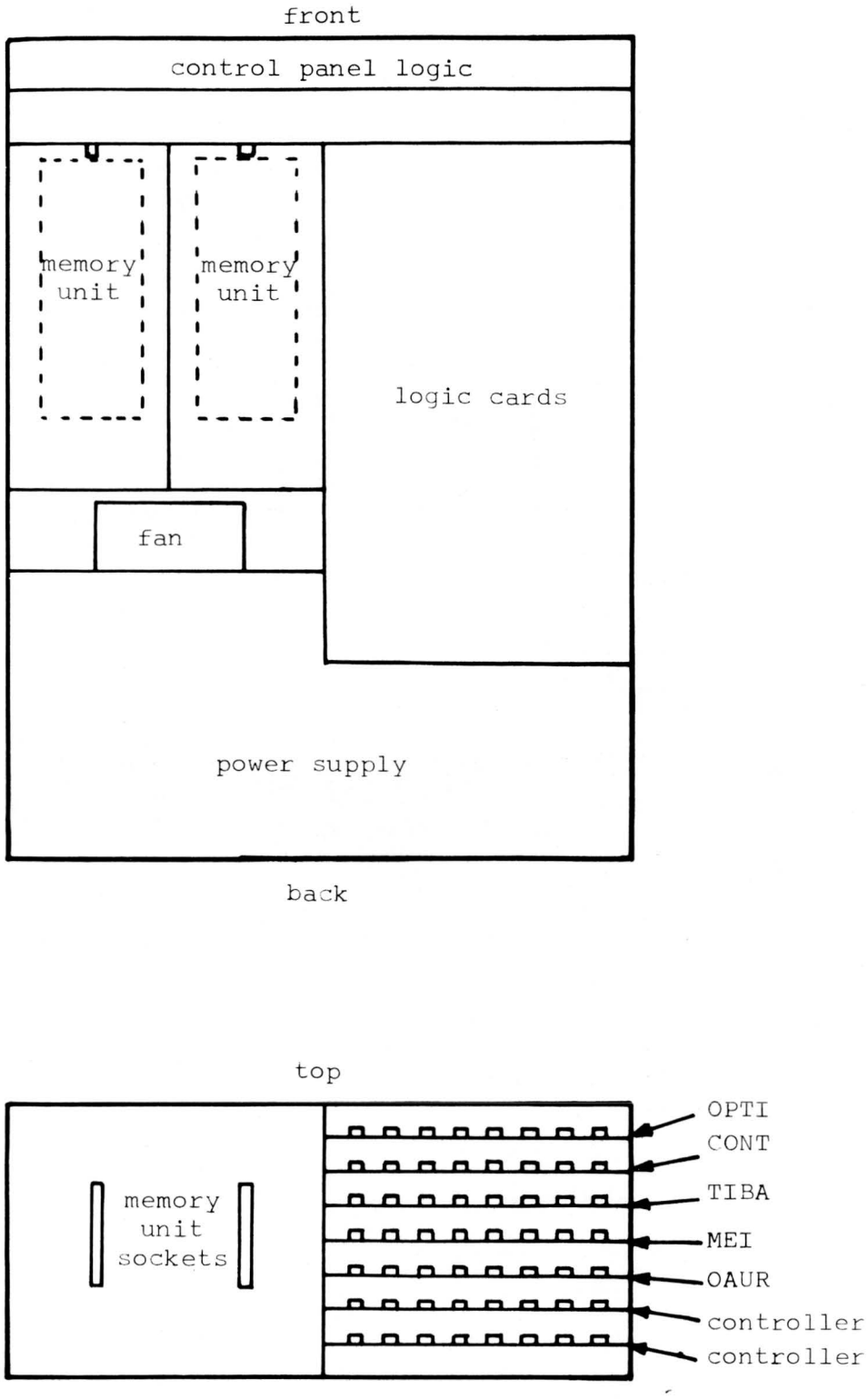


Figure 4.19 CPU layout and logic card positions

The signal, control or instruction names used in the logic and timing diagrams are made up of one or a combination of abbreviations or mnemonics. To assist in recognizing the meaning of these names, the following list of abbreviations has been compiled and placed in alphabetical order. The following example shows how to use the list:

BULYB: this breaks down into three components, i.e.

BUL, Y and B

BUL means BU register left
Y means read the contents of
B means the internal bus lines (B-lines)

therefore the meaning of BULYB is: the contents of the B-lines are read into the BUL register.

At the end of the list of abbreviations there is a list of logic drawing interconnections which will assist in finding the various signals used in the logic diagrams in Part 2.

Definition of abbreviations

name	meaning
ACB	absolute conditional branch (instruction mnemonic)
ACC	accept command (used during I/Ø transfers)
ADD	add function
AL	arithmetic left
AND	logical AND (instruction mnemonic)
ARETWØ	signal used to update the P- and S-registers
AR	arithmetic right
AU	arithmetic unit
B	bus or B-lines
BAD	bus address lines (used during I/Ø transfers)
BIN	bus input lines (I/Ø bus)
BØ	bootstrap
BØF	bus operation function (used in I/Ø transfers)
BRC	branch on condition
BSAR	contents of B-lines and SAR-register
BUL	most significant 8 bits of the BU-register
BUR	least significant 8 bits of the BU-register
C	clock
C2	two's complement (mnemonic)
CA	carry
CF	call function (mnemonic)
C-gates	gates that enable the input lines to the M-register
CH	character
CIN	carry into AU
CØ	carry out from AU
CØM	complement
CR	condition register
CV	condition verified
DAV	device address valid (used during I/Ø transfers)
DEF7	logic signal that forces hexadecimal value F7 into the K-register during a call function
D-gates	gates that enable the MØL contents to be read into either the K-register or the C-gates
E, EX or Exec	execute

name	meaning
F	flip-flop or fetch
FAR	address recognized flip-flop (used during I/Ø transfers)
FAURE	automatic restart flip-flop
FCE	force
FENB	enable interrupt flip-flop
FLA	signal used to perform some cycles twice during call, return and load increment instructions
FLK	link to monitor (instruction mnemonic)
FPF	power fail flip-flop
FRT	real-time clock flip-flop
HL	halt
IG	interrupt generated
ILØNG	an instruction with the address or constant contained in the second word
IM	interrupt mask register
IN	input
INT	interrupt
IØ or I/Ø	input/output
KEY	key lock on control panel
KL	K-register left bits 00 to 07
KR	K-register right bits 08 to 15
L	last or left or load
LCL	clear-write (LCL/) or read-restore (LCL)
LØAD	load from control panel
LØD	load
LØG	logical
M	memory or M-register
MREF	memory reference
NU	null
Ø	output
ØSC	oscillator
ØUT	output
ØVF	overflow

name	meaning
P	P-register or program counter
PFAURE	power fail/automatic restart
PST	start pulse
RCP	read control panel switches
RES	result
RESET	reset flip-flop
RESM	result in memory
RIL	read interrupt lines (instruction mnemonic)
RR	register-to-register instruction
RSEL	register selected from control panel
RTN	return from function (instruction mnemonic)
RUN	run mode
S	selection or select
SAR	buffer register used during the updating of the S-register
SET	set flip-flop
SH	shift
SLA	shift left arithmetic
SLL	shift left logical
SP	scratch pad registers
SPA	scratch pad address
SPA4	signal that selects the appropriate SP chip
SRA	shift right arithmetic
SRC	shift right circular
STA	start
STD	standard
STK	stack
STNC	start button normally closed contact
STNØ	start button normally open contact
STR	start routine
STSEL	status selected from control panel
SUB	subtract
T	timing
W	write
WIM	write interrupt mask (instruction mnemonic)
X	indexing
Y	reads

Logic drawing interconnection list

signal	from	to
ACB/	C1	C2
ACC/		C8
ADD/	C1	A3,C2,C7,C13
ADSLA	A3	C1
AF1	T5	T8,T9,T10
AF1/	T5	C11
AF2	T5	T9,T10,C9,T12
AF2/	T5	T8,T10
ALO/	A2	A3
AND/	C1	C2,C7,C14
ARE/		C8
AREM	C3	M3
ARETWØ	C3	M3
ARO	M3	A3
AUO/	A4	A3
BADOO/	M4	T6,C1
BADO1/	M4	
BADO2/	M4	
BADO3/	M4	
BADO4/	M4	
BADO5/	M4	
BEBINR	T9	A1
BEBINL	T9	A1
BEE	T9	A5
BESTL	T9	A5
BESTR	T9	A5
BINOO to 15/		A1
BØ/		A2
BØFO1/	M4	
BØFO2/	M4	
BØFOO/	M4	
BØUOO		T11
BRC	C1	C10
BULLYBL	T8	A2
BULRYBR	T8	A2

signal	from	to
BURLYBL	T8	A2
BURRYBR	T8	A2
BUOO to 15		A2
BØT	M6	
BO/	A1,A5	M9
B1/	A1,A5	M9
B2/	A1,A5	M9
B3/	A1,A5	M9
B4/	A1,A5	M8,M9
B4A	M8	M6
B5/	A1,A5	M8,M9
B5A	M8	M6
B6/	A1,A5	M8,M9
B6A	M8	M6
B7/	A1,A5	M8,M9
B7A	M8	M6
CE/	C2	C4
CEB	C5	M1
CFA	C4	C10
CF	M7	T6
CF/	M6	M7,T4,C13,C14
CFBEO/	T6	
CFEO/	C4	C2
CFHL	T10	T12
CH/	C1	C2,C4,C10,C11,C14
CINEFC	C4	A7
CINEFP	T8	A7
CINE1/	C4	A7
CINST/		T10
CMSEL/		T8
CØM	C14	A3
CFRUNZØ	T10	
CLØAD		T10,C5,C14
CØM/	C1	C2,C7,C4,C14
CPI/	C1	C2,C7
CRCP1		M10

signal	from	to
CRCP2		M10
CRCP4		M10
CRCP8		M10
CREAD		T10
CRELO/	T10	T12
CRL1	T10	A1,T9,C9
CRL1/	T10	T5,C11
CRL1A	T10	T8,C3,C5,C6
CRL1RL3	T10	
CRL3	T10	A1,T9
CRL3A	T10	T8,C3,C5,C6,C14
CRL4	T10	C9
CRL4/	T10	T8,T7,C11,C13,C14
CRO	A3	A5
CRO/	A3	M6
CR1	A3	A5
CR1/	A3	M6
CROE1A	A3	A3
CROZØ/	C8	A3
CR1ZØ/	C8	A3
CRZ3	C8	A3
CRYCRD	C7	A3
CRSEL/		C14
CSTSEL/		T9
CV/	M6	C1
DAV/	C8	
DEF7	T11	M1
DEMOL	T11	M1
ECR	C4	C10,C14,C11
ECR/	C4	C10,C13,C14
EXINST	T12	T11
EO	T6	T7,T11,C3,C4,C6,C9,C10,C11,C13,C14
EO/	T6	T8,T4
E1	T6	M7,T4,T6,T8,C3,C4,C8,C9,C11,C13,T2
E1/	T6	T7,T11,T12,C2,C3,C5,C6,C11,C14

signal	from	to
E1H1	C10	C10
E1RESK	T4	C9
E2	T6	A7,T7,T8,T9,C4,C5,C7,C8,C12,C13,C14, T12
E2/	T6	M7,T8,T9,C2,C3,C5,C14
E2HO	C12	C2,C10,C11
E2H1	C12,C13	C10
EOO	A4	A5
EO1	A4	A5
EO2	A4	A5
EO3	A4	A5
EO4	A4	A5
EO5	A4	A5
EO6	A4	A3,A5
EO7	A4	A3,A5
FCAYFC	T7	A7
FCESP7	T7	A7
FCAPYFP	T7	A7
FCP	T7	A5
FCP/	T7	
FENB	T11	A5
FLA	M7	T6,C2,C4,C6,C7
FLAB/	T6	T9,T12
FLACR	M7	T6,T12
FLK	M7	A5
FNUYNU	T7	A3
FRUN	T5	T5,T10,T12
F1	T12	T5,T7,T8,C3,C13,C14
F1/	T12	C11
F2	T12	T4,T6,T8,C6,C9,C11,C13
HO	T1,T2	T8,T9,C2,C3,C4,C5,C10,C11,C12
H1	T1,T2	T8,T9,C2,C3,C4,C5,C10,C11,C12,C13
H1E1	T2	M3
IIVB		T11
ILØNG	M7	C3,C14
IM/	C2	C4
IN/	C1	T9,C8,C10,C11,C14

signal	from	to
INT	T11	A5,T9,C9,C14
INT/	T11	T5,T6,T12
INTNC/		T7
INTNØ/		T7
IØR/	C1	C2,C7,C14
IR/	T11	T12
ISOO/		T11
I2	T11	T6,C9
I1/	T11	C9
JIVC		T11
KEYOO to 15		A1
KLYD/	T12	M5
KLOO	M5	M3,M6,M10,C3
KLO1	M5	C1
KLO2	M5	C1
KLO3	M5	C1
KLO4	M5	C1
KLO5	M5	M6,M7,M10
KLO6	M5	M6,M7,M10
KLO7	M5	M6,M7,M10
KRYD/	T12	M4
KRO8	M4	M3,M5,M7,M10,T10
KRO8/	M4	C4,C12
KRO9	M4	M3,M5,M6,T4,T11
KRO9A/	T11	T6,C1
KR10	M4	M3,M5,M6
KR10/	M4	T7,C8,C12,C13
KR10A	T6	T7,T11
KR11	M4	M7,M6,M10
KR12	M4	M3,M6,M7,M10
KR13	M4	M3,M6,M7,M10
KR14	M4	M3,M6,M7,M10
KR15	M4	M3,M6,T4,C2,C9,C13,C14
KR15/	M4	C7,C8
KR15A/	C7	C2
K10A15B	M4	C8

signal	from	to
L	T12	T7,T8,T10,T11,T12,C3,C4,C8,C9,C11, C13,C14
LEBUL	C4	A2
LEBUR	C4	A2
LØD	C1	M6,C2,C7
LØECA	C12	A7
LØECAP	C12	A7
LØESPO	C13	A7
L17EP6	C12	A8
L17EP7	C13	A8
MCL	M8	
MCL/		M8
MCLM/	M8	M9
MCLT1/	M8	T5,T10,T12
MCLT2	M8	T4,T6
MREF	M6	T4,T6,C2,C7,C9,C13,C14
MYC/	C6	M2
MO-7	M2	M3
NU	A4	A3
ØR1/	M5	M6,C2,C7,C13
ØR2	M4	C8
ØR2/	M4	M7,C13
ØSC/		A4,T1,T6,C6,C7,C9,C13,T3
ØUT/	C1	C2,C8
PFAURE/		T5
RBC/	C1	C2,C4
RES	C7	T7,T11,C14,R10
RESET1		M8
RESET2		M8
RESKB/	T4	T6
RESM	T4	C2
RESM/	T4	C2
RFRT		A5
RIL	C8	
RLI	M6	T6

signal	from	to
RR	C10	C4,C5
RR/	M5	M5,C4,C7,C10
RTN/	M6	M7
RTN	M7	T11,C7
RTO	T1	T2
RT1	T1	T2
RT2	T1	T2
RT3	T1	T2
SO	C2	A4
S1	C2	A4
S2	C2	A4
S3	C2	A4
SO4/	M8	M7
SO5/	M8	M7
SO6/	M8	M7
SO7/	M8	M7
SO8/	M8	M7
SO9/	M8	M7
S10/	M8	M7
S11/	M8	M7
S12/	M8	M7
S13/	M8	M7
S14/	M8	M7
S15D/	C9	M9
SARYB	C9	M9
SARO8	M9	M6
SARO9	M9	M6
SAR15	M9	C14
SCLYSCD	C9	M9
SET	C7	M7,T11,C14
SH	C1	T7,T8,C12
SH/	C1	C4,C7,C10,C11,C12,C13,C14
SHK10B	C1	C2
SKSD	T4	T6
SPAERCP	T10	M10
SPAER1	C10	M10

signal	from	to
SPAER2	C10	M10
SPA0/	M10	M10
SPA1/	M10	M10
SPA2/	M10	M10
SPA3/	M10	M10
SPYB/	C14	A6
SPA4	C11	A6
SRA	C12	C13
STD/	M6	M7,T4,T9,C10
STK/	C1	M7,A3,T4,C2,C9
SUB/	C1	A3,C2,C4,C7,C13
STNØ/		T5
STNC/		T5
SUSLA	C1	A3
SYBSAR/	C9	M8,M9
TF/	T3	T10
TFP/	T3	T4,T5,T6,T11,T12
T0	T1/T2	T7,C8
T1	T1/T2	T7
T2	T1/T2	C6,C14
T3	T1/T2	T6
T4/	T1/T2	M7,C8
T5	T1/T2	T7,C8
T6	T1/T2	T5,T10,C6,C7,C8,C9,C14
T6/	T2	C9
T7	T1/T2	T3
T2A	T2	M7,T5,T8,T11,T12,C9
T2PT6	T1/T2	C14
T6A	T2	T8,T10,T11,C9
T6E2	C8	C14
T105/	T1/T2	A4
T2LW/	T11	
WIM	C8	T11
W1	T4	T6,T7,T9,C4,C10,C11,C14
W1/	T4	C5,C6,C10
W2	T4	M7,T6,T8,T12,C11,C14

signal	from	to
W2/	T4	C5,C6,C10
W2D	T4	C3
W1PW2	T4	C2
XEC	T4	T6,C3
XEC/	C1	T4,T12
XI/	T11	C3,C6
XØR/	C1	C2,C7,C14
X1	T4	T7,C11
X1/	T4	T11,C10,C13,C9
X2	T4	T6,T11,C9
X2/	T4	T8,T11,C10,C13

SECTION VI

POWER SUPPLY

The power supply unit provides + and - 5 volts d. c. (for up to 1K of memory the logic cards of the CPU and the control units cards), the 12 volts a. c. supply for the indicator lamps, and plus and minus 12 volts d. c. used by the logic of the power supply unit. In addition it provides power fail and reset signals used by logic in the CPU.

When more than one power supply is used, two additional signals are used to ensure that the correct switching sequence is maintained during power ON/OFF operations.

6. 1 LAYOUT

Figure 6. 1 shows the layout of the unit. The power transistors and thyristors for the + and - 5 volts d. c. supplies are mounted on heat sinks RD1 and RD2. The HI1 card contains the power fail and reset line logic and the + and - 12 volts d. c. supplies used by this unit. Card HI2 contains the control logic for both voltage and current and the power ON/OFF sequencing logic for the + and - 5 volts d. c. supplies.

The mains input fuse (F1) is mounted on the backpanel of the unit, the two d. c. fuses (F2 and F3) are mounted on the heat sink inside the unit.

6. 2 OPERATION

The power supply is of the series regulated type and has outputs sequenced such that data stored in the memory is not disturbed when the mains is switched ON or OFF. The timing of the switching sequence is shown in Figure 6. 2, together with the timing of the power fail and reset signals.

Figure 6. 3 shows a general electrical scheme of the power supply. When the mains supply is switched on, via the key-lock switch, the two rectified and smoothed 5 volts d. c. outputs will appear on the collectors of the series power

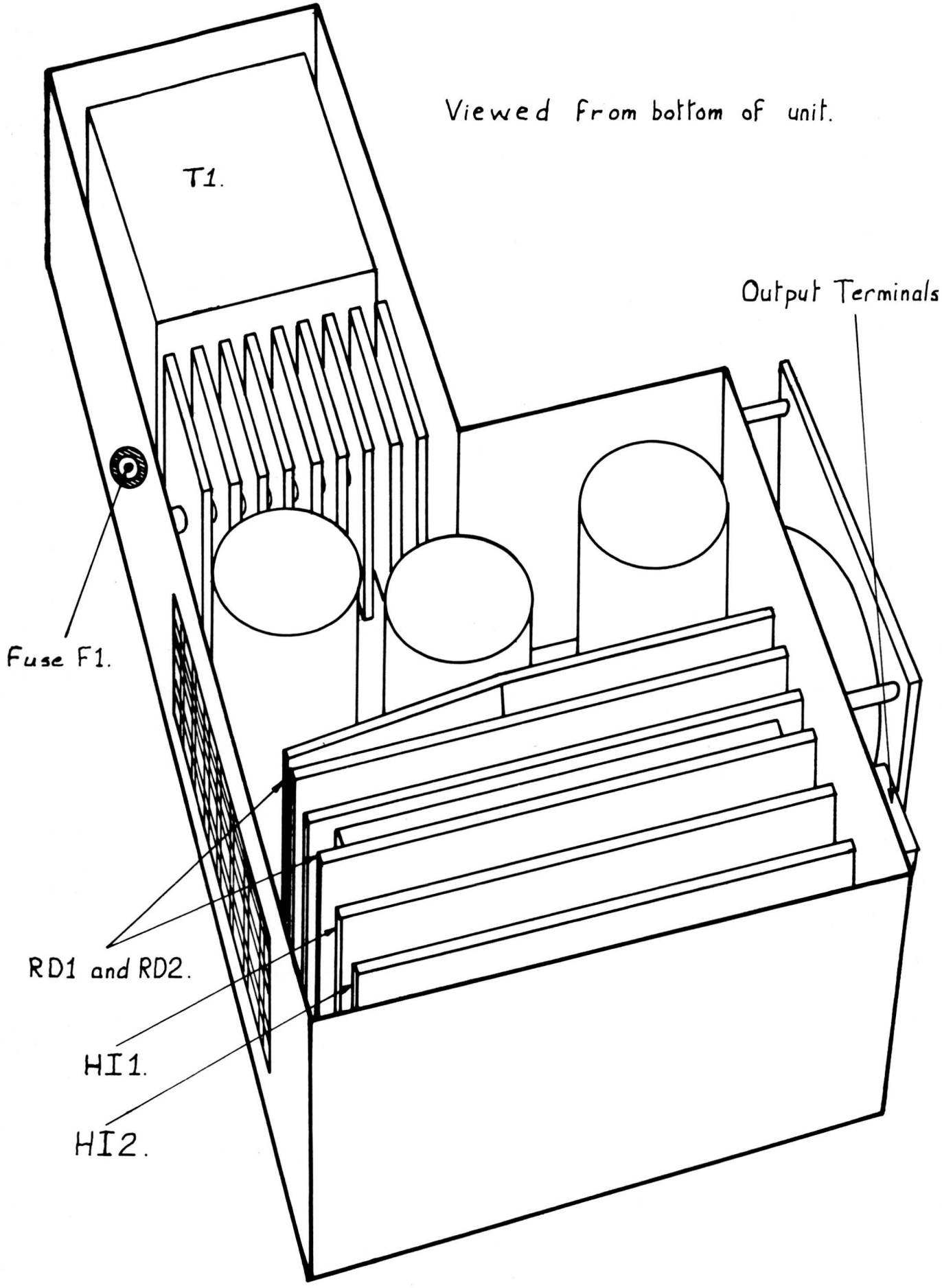


Figure 6.1 Layout

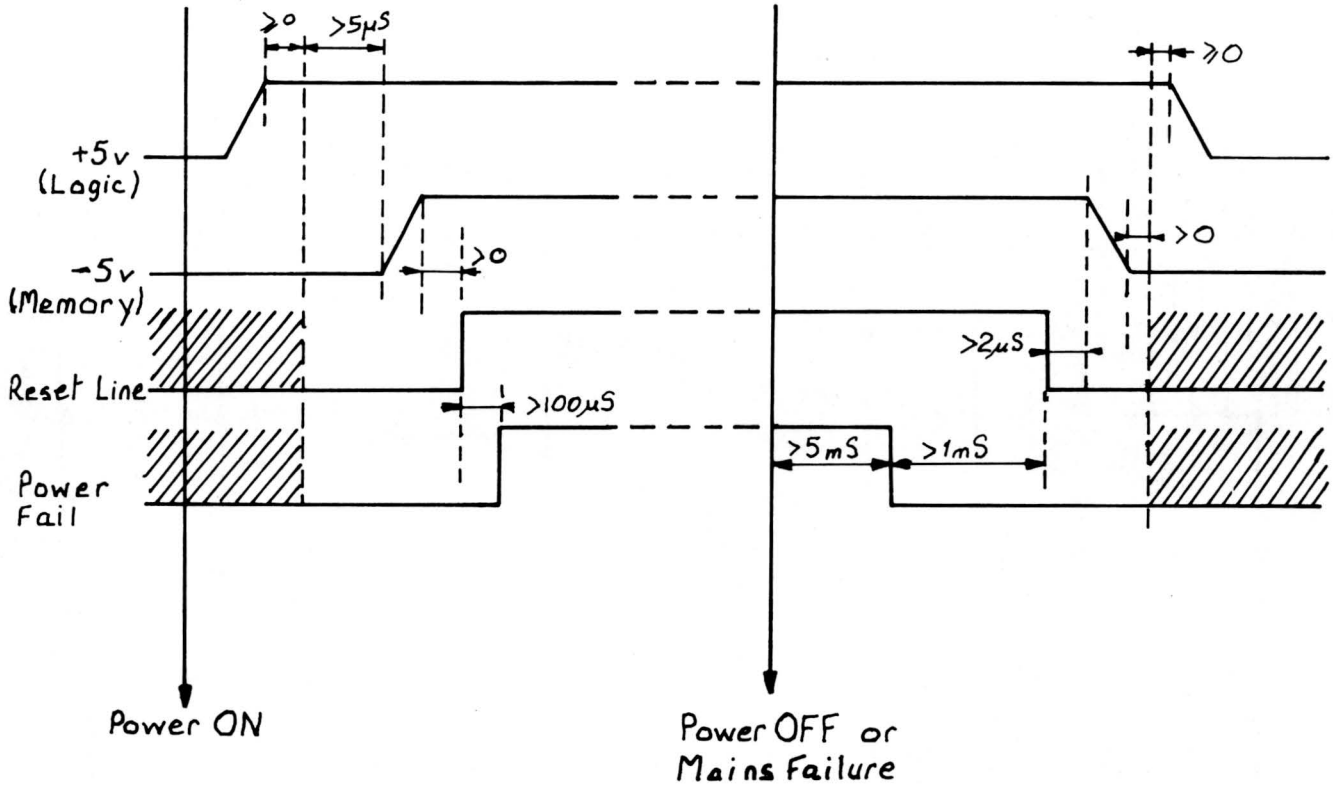


Figure 6.2 Switching Sequence

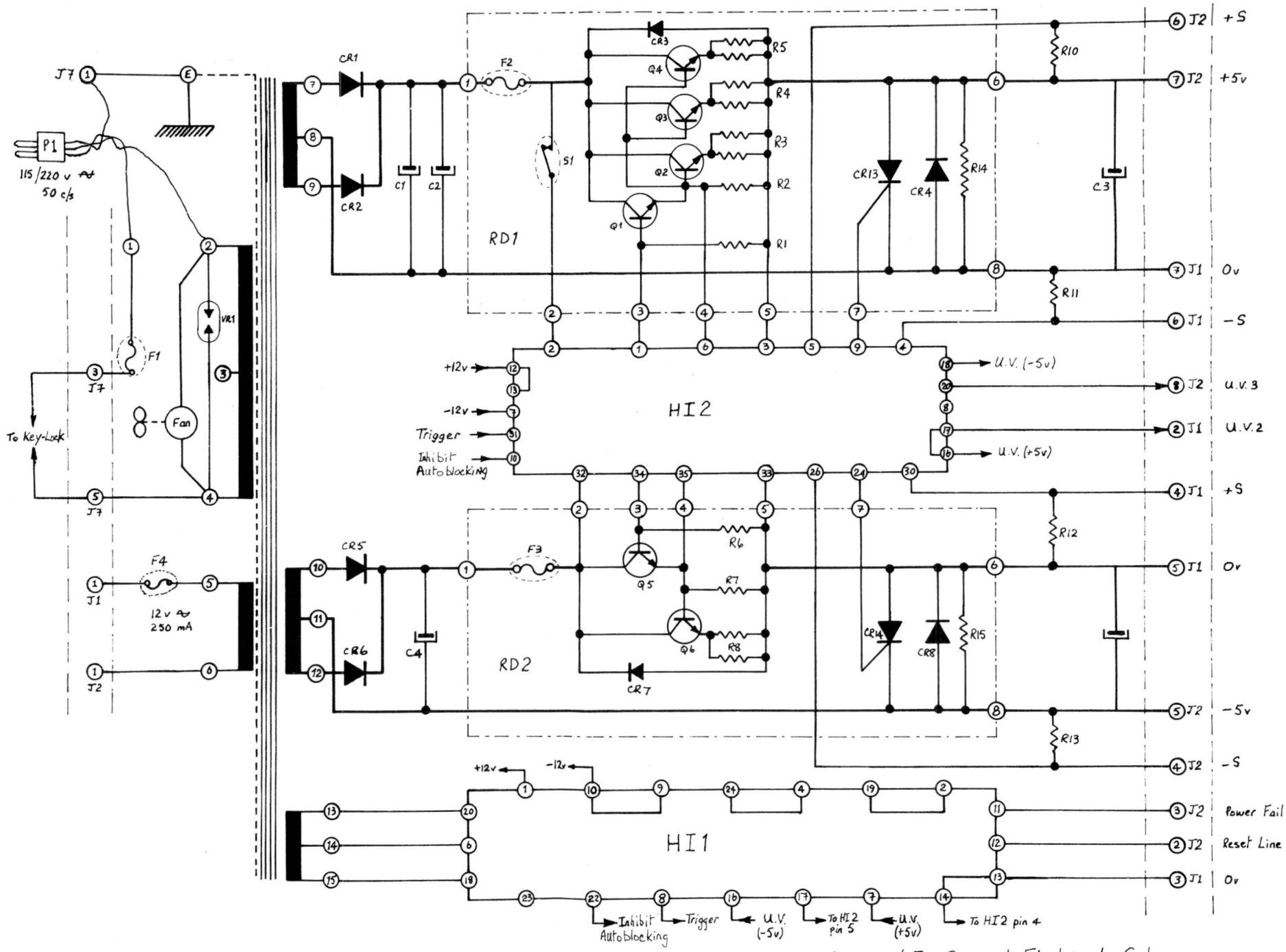


Figure 6.3 General Electrical Scheme

transistors. At this time the transistors are cut off so the output is prevented from appearing on the output terminals. When the outputs have stabilised, the delay logic (on the HI2 card) switches the transistors on and the output becomes available on the output terminals.

When the mains supply is switched off, or there is a power failure, a trigger pulse is sent from the security card (HI1) to the thyristor of the - 5 volts d. c. supply. This cuts off the - 5 volts from the output terminals, and the loss of the - 5 volts activates the undervoltage logic which in turn cuts off the + 5 volts from the output terminals.

6.3 SECURITY CARD (HI1)

(see figures 6.4 and 6.5)

The stabilised + and - 12 volts d. c. is used by logic both on this card and on the HI2 card. The mains input to the line detector circuit is also taken from the same transformer winding.

POWER FAIL LOGIC The power fail detection is done by the Q1 networks, R1 having been set to detect breaks greater than 5 milliseconds (all adjustment procedures have been carried out by the manufacturer and the variable resistors, shown on Figures 6.5 and 6.7, replaced by fixed resistors). During normal operation the output from Q7 is high. When the mains is switched off a power failure occurs, a signal from Q1 drives the output of Q7 low and this low signal activates the power fail logic of the CPU. The output from Q1 also drives Q10 and Q2 producing FL1 and FL2.

DELAY LOGIC The output from Q2 is also used to drive the delay network of H1, Q5 and Q6. This network can also be activated by a failure of the - 5 volts during normal operation. When the delay time has expired, the output from Q6 is used to drive the thyristor trigger circuit.

TRIGGER CIRCUIT The output from Q6 causes Q3 to trigger the thyristor half of the two cross-coupled nand gates in H2. The output from H2 drives Q9 and the output from this transistor is used as the crowbar trigger on the HI2 card. This circuit will also be activated if the + 5 volts fails during normal operation.

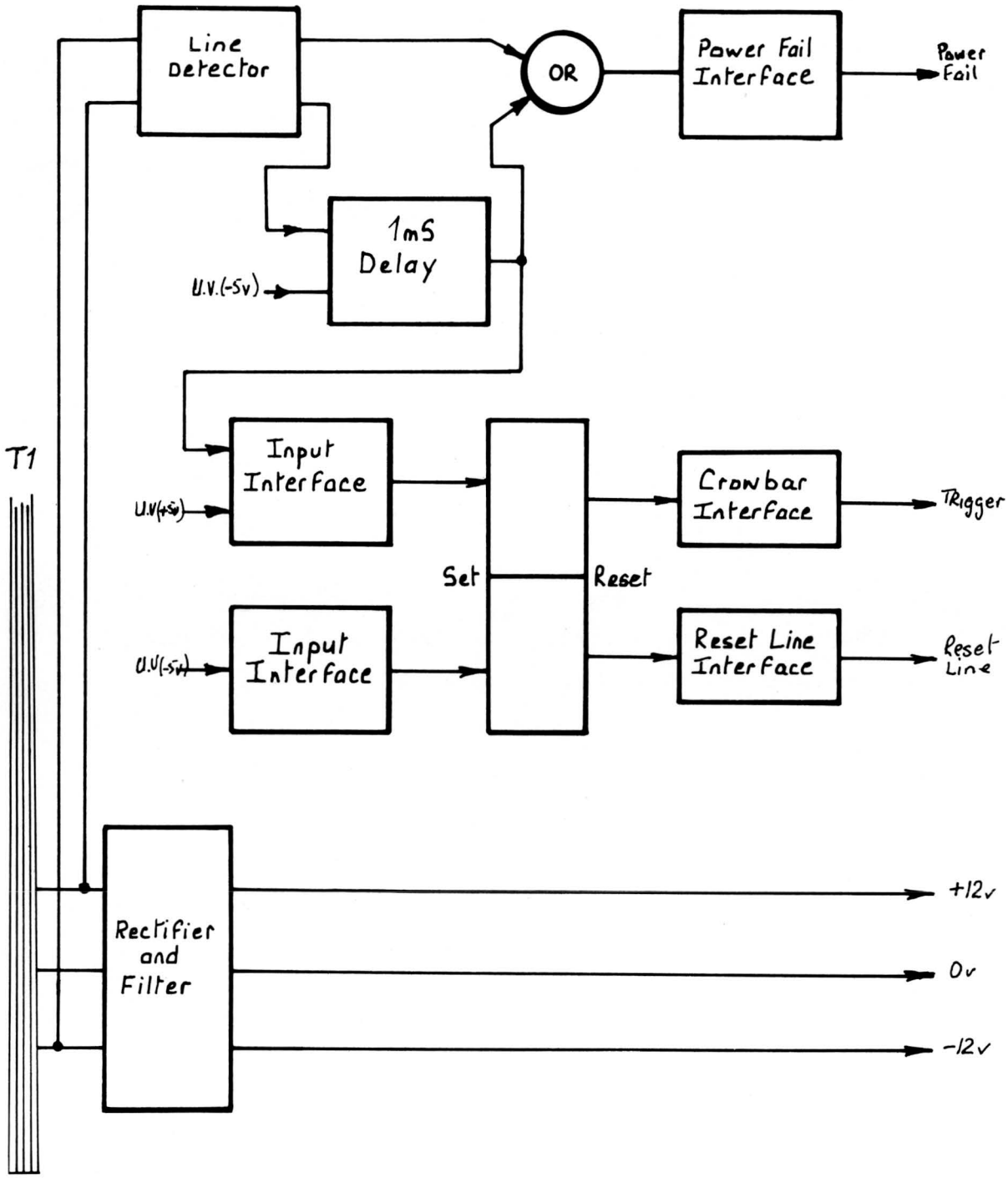


Figure 6.4 Block Diagram of HI1 Card

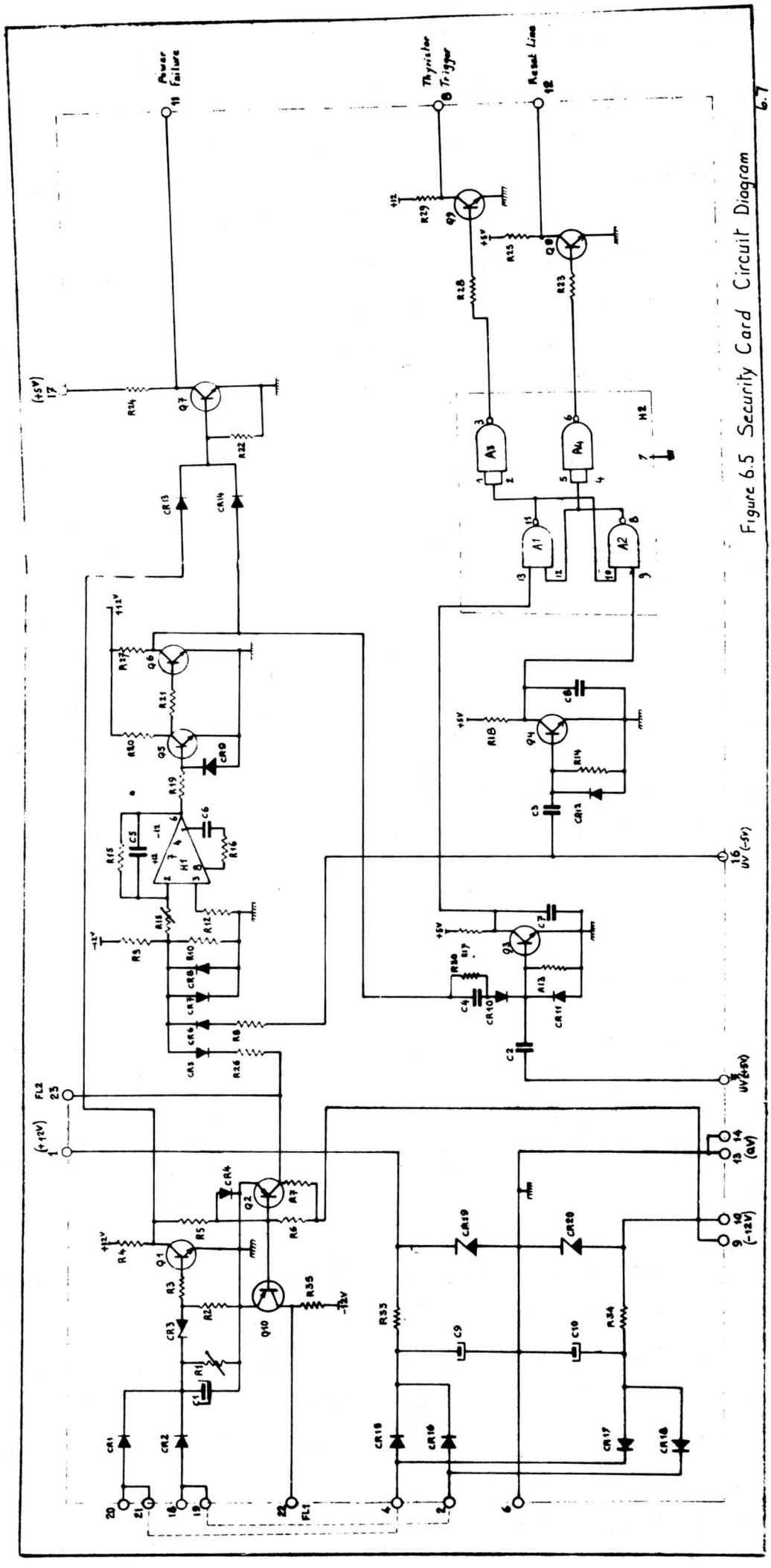


Figure 6.5 Security Card Circuit Diagram 6.7

RESET LINE LOGIC This circuit is activated by a signal from the - 5 volts undervoltage circuit, on the HI2 card, shortly after the - 5 volts is available on the output terminals. The signal drives Q4 whose output triggers the reset half of the cross-coupled nand gates in H2. The output from H2 drives Q8 and the output from Q8 will be high during normal operation of the power supply. When the mains is switched off or there is a power failure , the output of Q8 will be driven low due to the action of the cross coupled nand gates, in H2, when the trigger circuit operates.

6.4 AMPLIFIER CARD (HI2)

(see figures 6.6 and 6.7)

The circuit can be divided into two and both halves operate in a similar way. The - 5 volts half has two extra input to it, (one in the gating circuit and the other from the crowbar trigger) and the delay times are slightly different; to therefore to avoid repetative descriptions, both halves of the circuit will be explained together. Where differences occur, both circuits will be described.

OPERATIONAL AMPLIFIERS H1 and H3 These modules perform the function of the current amplifier, current limiting and voltage amplifier.
The outputs from the modules controls the operation of the series power transistors and, when the mains power has just been switched on, the power-on sequence of the + and - 5 volts d. c. via the undervoltage, delay and gating circuits.

OPERATIONAL AMPLIFIERS H2 and H4 These modules function as undervoltage circuits. The outputs from them are used as inputs to the delay circuits, the security card and to the other power supply when two units are used in the CPU.

OVERVOLTAGE CIRCUITS These are the Q4 and Q5 and the Q9 and Q10 networks. The voltage to the bases of Q5 and Q10 is set so that until the output voltage (of the power supply) rises above 5. 5 volts, Q5 and Q10 will conduct and their outputs will drive Q4 and Q9. The output from these transistors will trigger the appropriate thyristor cutting off the supply to the output terminals. The base of Q10 can also be driven by the crowbar trigger output from the security card.

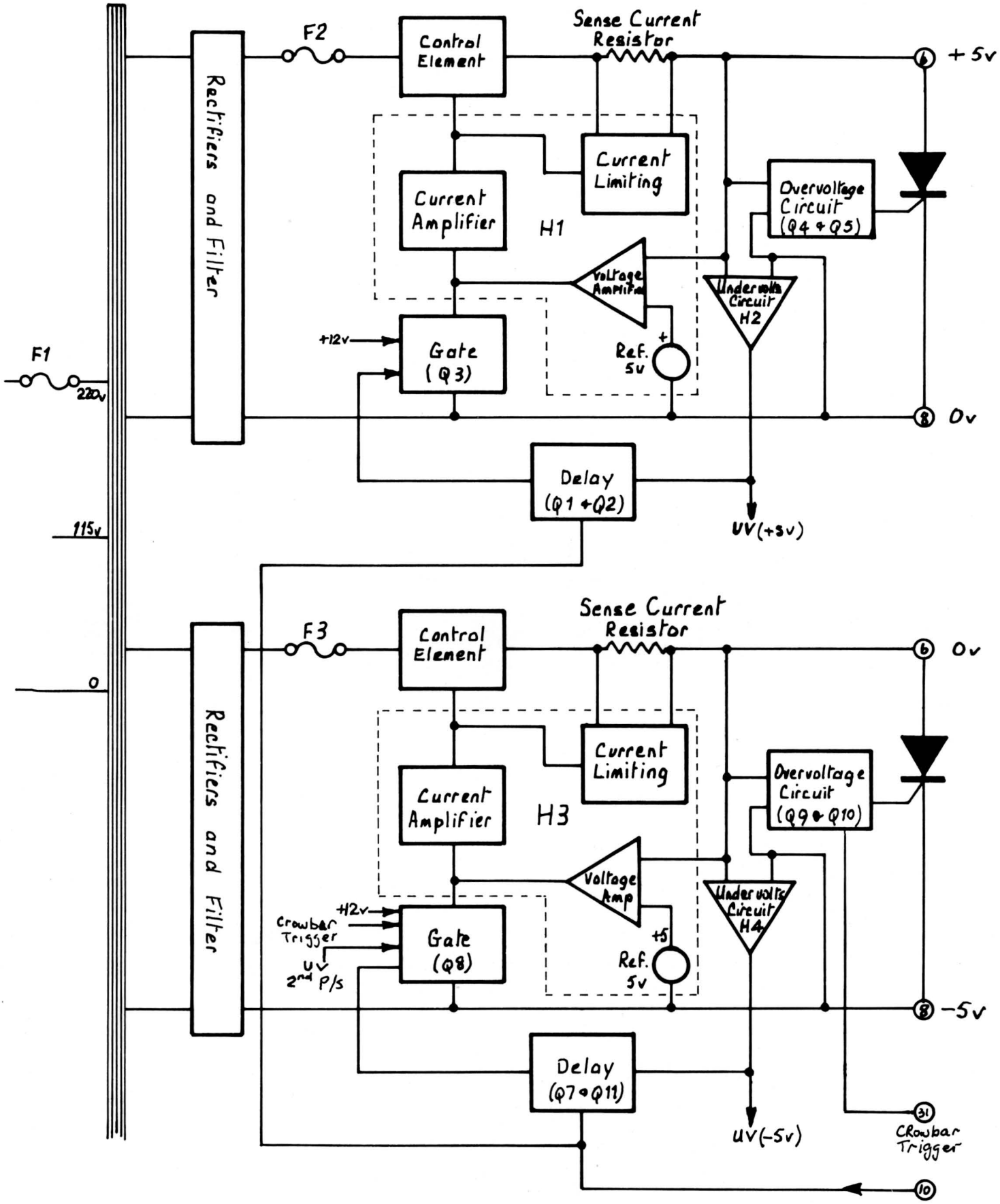


Figure 6.6 Block Diagram of HI2 Card

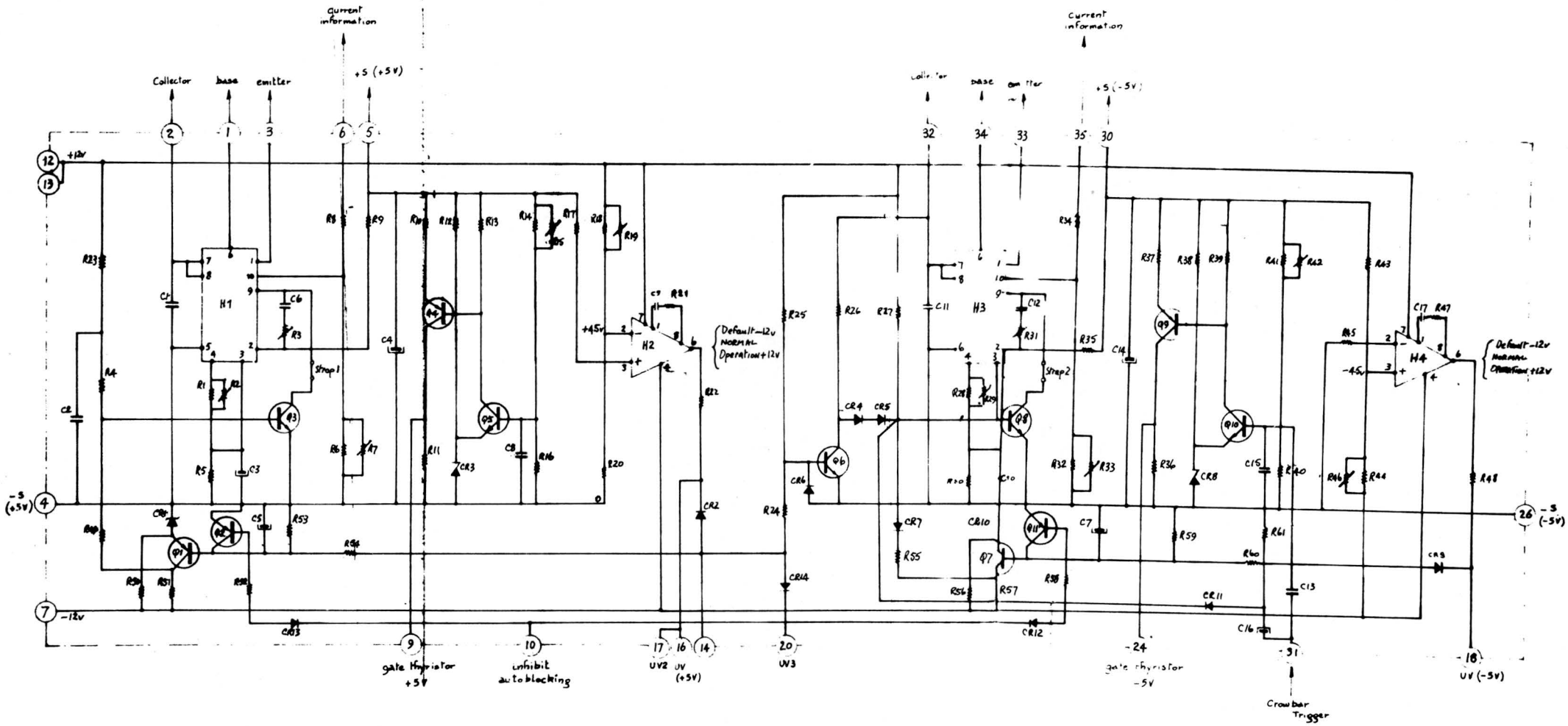


Figure 6.7 Amplifier Card Circuit Diagram

DELAY CIRCUITS These are the Q1 and Q2 and the Q7 and Q11 networks. The delay is achieved by the CR networks C5 and R53 and C7 and R59. During the switch on period, these circuits delay the application of the supply outputs to the output terminals until all the circuits have stabilised. The outputs from Q1 and Q7 then activates the gating circuits which in turn permit the H1 and H3 modules to switch the supplies to the output terminals. When a power failure is detected, the delay circuits are inhibited by a signal from the security card.

GATING CIRCUITS These are the Q3 and the Q6 and Q8 networks. They are driven by the output signal from the delay circuits and the outputs from Q3 and Q8 will cause the H1 and H3 modules to cut off the supply from the output terminals. They will be activated by an undervoltage condition either during normal operation or during a switch-off sequence. The -5 volt circuit has an extra input to the gate from Q6. This input will be activate when an undervoltage condition exists in the second power supply (when two supplies are used by the CPU).

6.5 OTHER COMPONENTS

These are the components not already described in previous sections (see figure 6.3.)

VR1: This is a spark gap device which will suppress spikes on the incoming a.c. mains.

S1: This a thermal operated switch which will operate if the internal temperature becomes excessive.

VOLTAGE SENSE RESISTORS These are resistors R10, R11, R12 and R14. These resistors prevent damage to power supply if it is switched on without the external load being connected.

6.6 OUTPUT TERMINALS

Figure 6.8 shows the terminals to which the various outputs are connected. The configuration will be the same when more than one power supply is used,

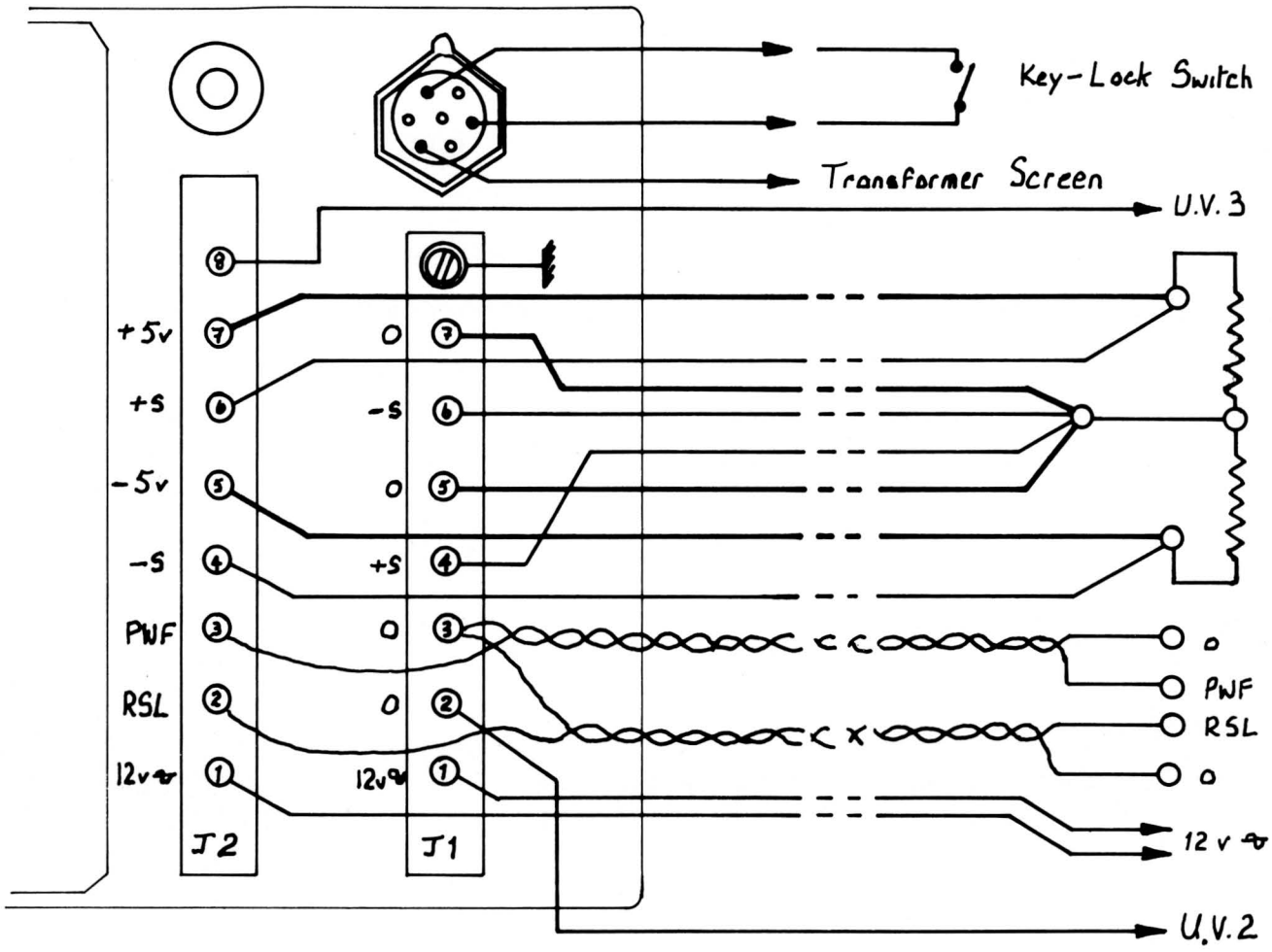


Figure 6.8 Output Terminals

but in this case J2/8 (U.V.3) will be connected to J1/2 (U.V.2) of the second power supply and J2/8 (U.V.3) of the second power supply will be connected to J1/2 (U.V.2) of the first power supply.

6.7 CONTROL OF TWO POWER SUPPLIES

The interconnection of two power supplies has already been detailed in the previous section. Figure 6.9 shows the power on/off sequence and power fail and reset line timing for two power supplies. The reset logic in the CPU will only be activated when both reset line outputs are high. On the other hand either of the power fail outputs going low will activate the power fail logic in the CPU.

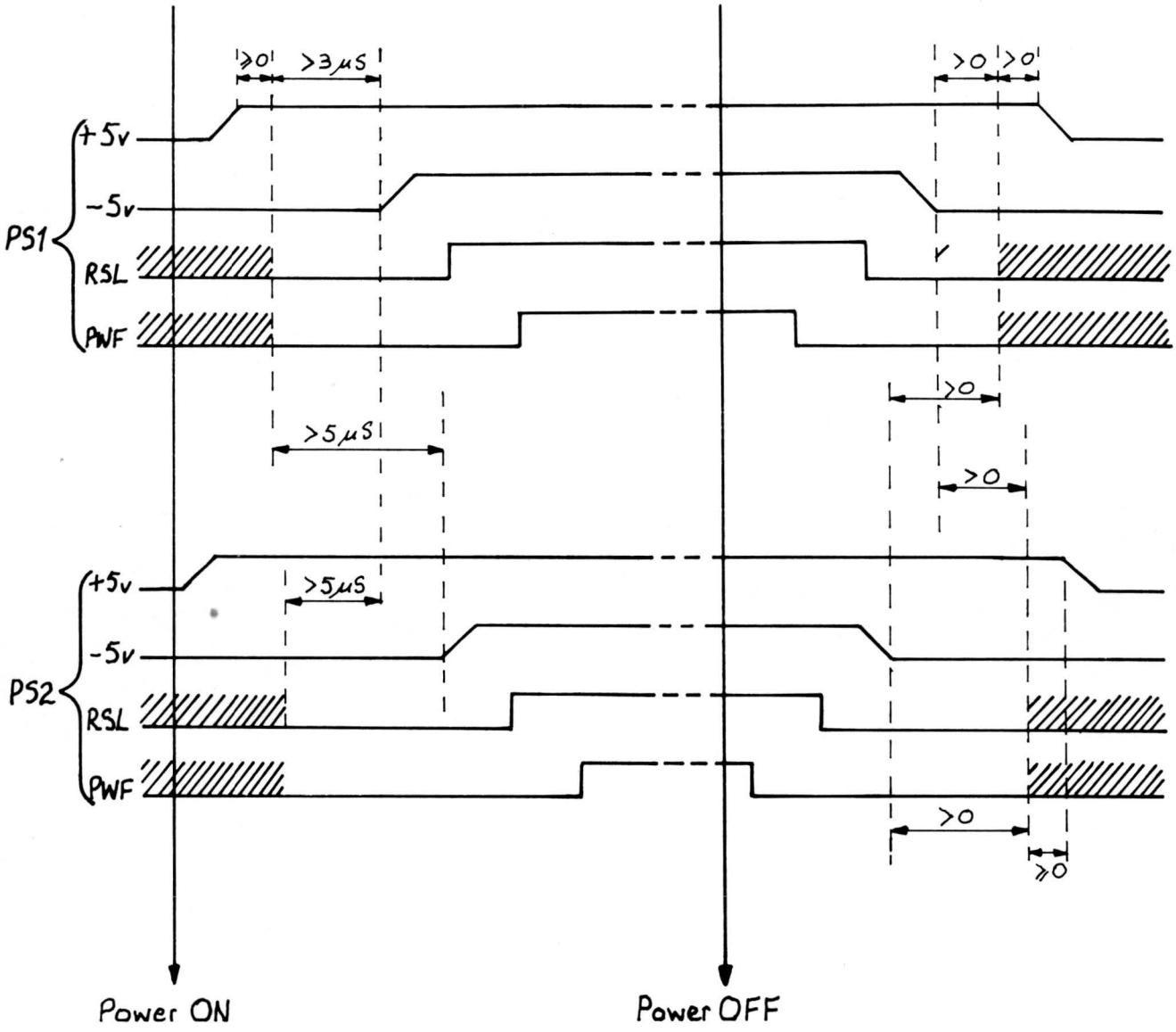


Figure 6.9 Timing Sequence for Two Power Supplies

SECTION VII

INSTRUCTIONS FOR REMOVAL AND REPLACEMENT OF SUBASSEMBLIES

This section contains the instructions for the removal and replacement of the subassemblies of the CPU. A separate section is devoted to each type of CPU and the engineer has only to refer to the section relevant to the CPU he is servicing. The other sections may be needed later when the customer either extends his existing system or changes from a Table-Top to rack mounting version or visa-versa.

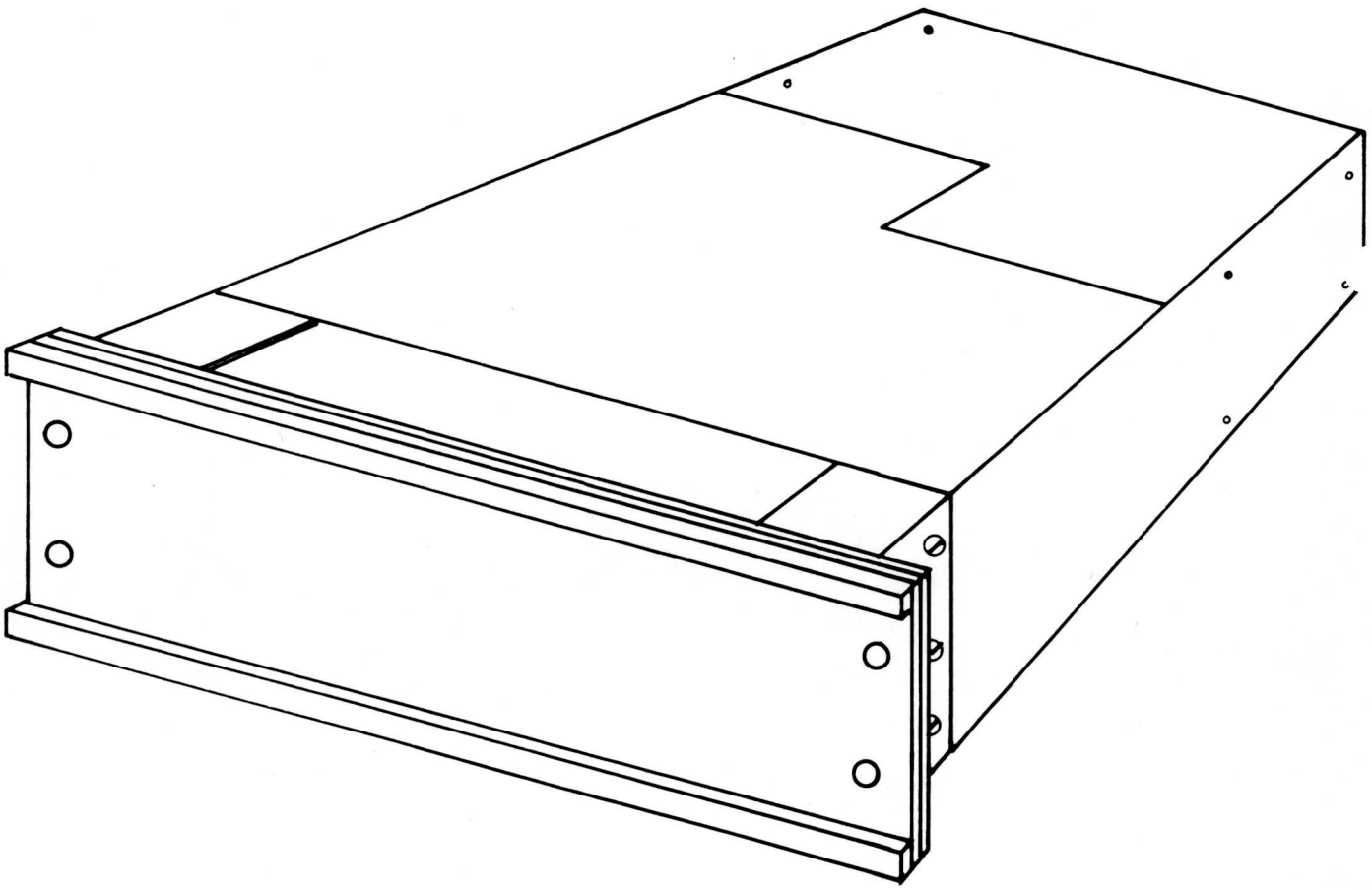


Figure 7.2 2K Rack Version

7.2 INSTRUCTIONS FOR REMOVING AND REPLACING SUBASSEMBLIES

The subassemblies of the P850 rack-mounting CPU can be removed and replacing using the following instructions.

7.3 REMOVAL FROM THE RACK

(1) Switch OFF the CPU at the control panel and remove the mains plug from the mains socket.

(2) Remove the screws holding the CPU to the rack.

(3) Remove the CPU from the rack by pulling the unit towards you, making sure that the cables at the rear of the unit are not damaged.

The subassemblies can now be accessed.

7.4 CONTROL PANEL

The top of the control panel is attached to the chassis by two hinges. Access to the switches, indicator etc, can be made using the following routine:

(1) Remove the two metal fixing strips (at the top and bottom of the panel) by pulling them towards you. These strips are held in place by spring clips (two to each strip) and are quite easily detached.

(2) The engraved plexiglass panel can now be removed.

(3) Remove the six countersunk screws that hold the metal panel to the mounting.

CAUTION: take care not to damage the connectors or wiring when carrying out the next two steps.

(4) The panel will now hinge upwards.

(5) Carefully unplug the two connectors (linking the panel to the chassis) by pulling the connectors straight out without exerting any side pull.

Replace the control panel by carrying out the above instructions in the reverse order.

7.5 REMOVAL OF THE CONTROL PANEL CIRCUIT BOARD

(1) Ensure that the machine is switched OFF at the control panel

(2) Remove the four fixing screws and withdraw the circuit board.

This allows access to switches and indicator lamps.

Care should be taken that the board does not come into contact with the mains switch contacts whilst taking measurements with the power switched ON.

The board is replaced by carrying out the above in the reverse order.

7.6 REMOVAL AND REPLACEMENT OF THE POWER SUPPLY

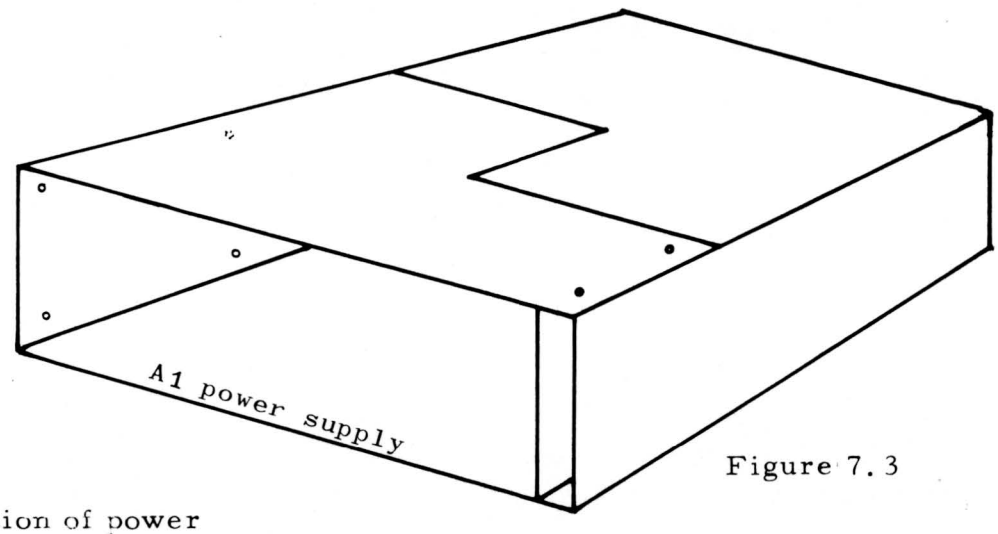


Figure 7.3

position of power supply retaining screws

Ensure that the mains plug has been disconnected.

7.7 A1 POWER SUPPLY REMOVAL

(1) Remove the mains cable clamp

(2) Remove the eight retaining screws (see Figure 7.3) four on each side.

(3) Very carefully withdraw the power supply from the back of chassis, taking care not to damage the cable.

This operation will allow access to the logic and CU cards of the CPU and memory unit(s) n° 1 and 2.

7.8 REPLACING THE POWER SUPPLY

The power supply should be replaced by reversing the order of the steps used for its removal. Take care when replacing the supply not to trap or damage the cable.

7.9 REMOVAL AND REPLACEMENT OF CIRCUIT CARDS

The machine must be switched off at the control panel before either removing or replacing a card. Before a control unit card can be removed, the input/output socket that connects the peripheral to the card must be carefully removed.

The circuit cards should be removed by pulling straight out without exerting any lateral pressure on the card. For this reason it is advisable to use both hands when either removing or replacing a card.

When replacing a card, care should be taken to ensure that it engages correctly in location slots on each side of the chassis. No excessive pressure must be used when replacing a card otherwise the card or circuits may be damaged.

7.10 REMOVAL AND REPLACEMENT OF THE MEMORY UNIT(S)

Switch off the machine at the control panel before removing or replacing a memory unit.

To remove the memory unit, pull down on the RED lever (top center of the unit) to release the retaining latch and withdraw the unit.

When replacing a memory unit, make sure that the RED retaining latch is properly engaged in its socket.

7.11 TAKING MEASUREMENTS

CONTROL PANEL CIRCUIT BOARD

When taking measurements on this board, the probe must not be connected directly to the pins. The probe should be attached firmly to some fixed point and a single flexible wire should be connected to the probe. At the other end of the wire should be a single pin socket which is then inserted in the appropriate circuit board pin. This method prevents any undue tension being exerted on the pins causing pin flexion or short-circuit with neighbouring pins.

LOGIC CARD MEASUREMENTS

Before measurements can take place, the card must first be removed and replaced by an extension card. The card to be tested is then inserted in the extension card sockets. A probe with a fine point should be used because of the high density of printed circuit tracks.

Measurements to individual integrated circuit modules must be made using the clip specially designed for this purpose.

NOTES:

(1) The removal of one or more of the logic cards for the purpose of making special measurements will not adversely affect the operation of the remaining cards.

(2) Whilst making tests on a disassembled machine, care should be taken that the mains leads of the switch do not come into contact with any metal object.

A check should also be made that the blower fans operate normally and that any mains connections are properly isolated.

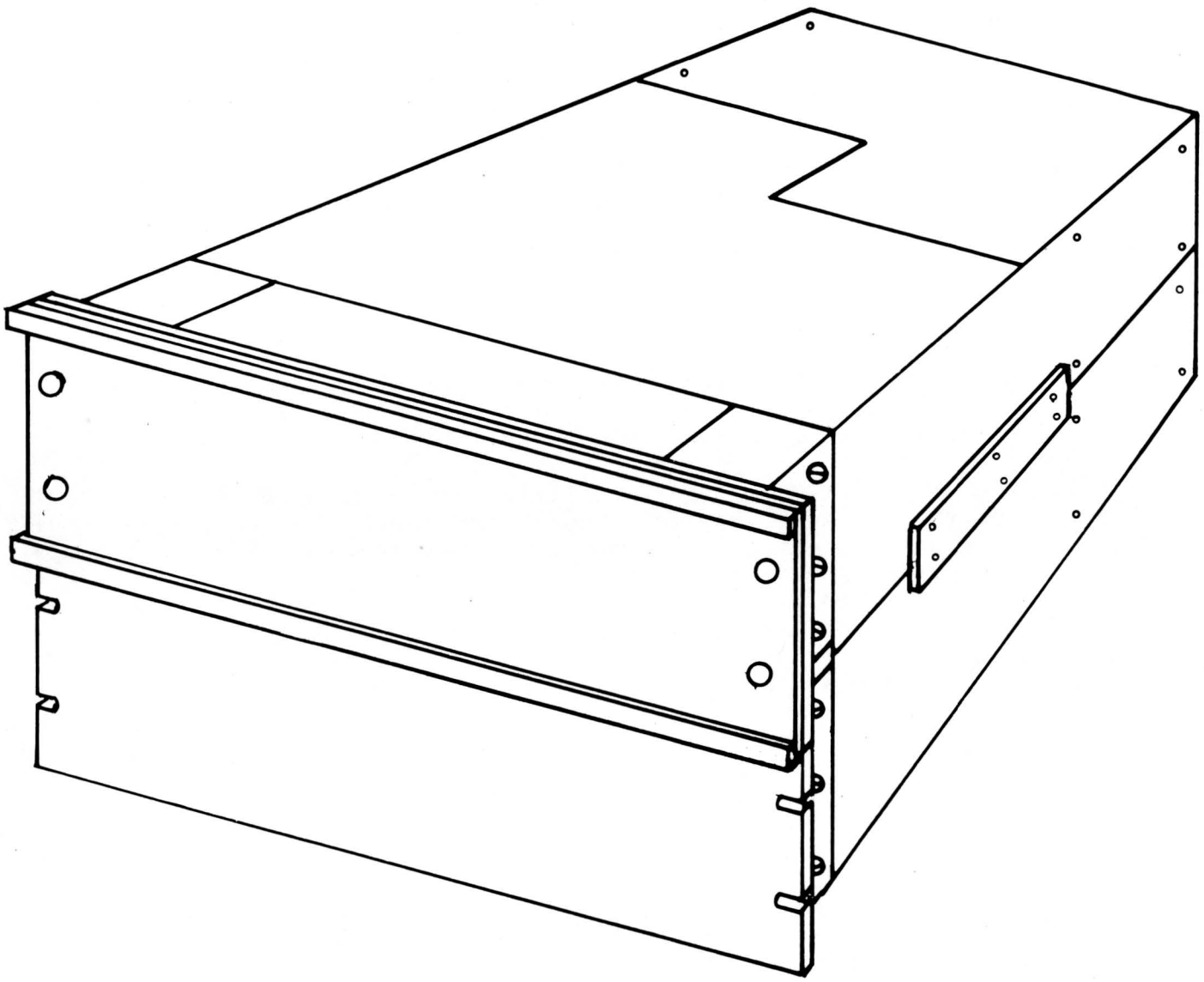


Figure 7.3 4K Rack Version

7.12 INSTRUCTIONS FOR REMOVING AND REPLACING SUBASSEMBLIES

The subassemblies of the P850 rack-mounting CPU can be removed and replaced using the following instructions.

7.13 REMOVAL FROM THE RACK

(1) Switch OFF the CPU at the control panel and remove the mains plug from the mains socket.

(2) Remove the screws holding the CPU to the rack.

(3) Remove the CPU from the rack by pulling the unit towards you, making sure that the cables at the rear of the unit are not damaged.

The subassemblies can now be accessed.

7.14 CONTROL PANEL

The top of the control panel is attached to the chassis by two hinges. Access to the switches, indicator, etc, can be made using the following routine :

(1) Remove the two metal fixing strips (at the top and bottom of the panel) by pulling them towards you. These strips are held in place by spring clips (two to each strip) and are quite easily detached.

(2) The engraved plexiglass panel can now be removed.

(3) Remove the six countersunk screws that hold the metal panel to the mounting.

CAUTION : take care not to damage the connectors or wiring when carrying out the next two steps.

(4) The panel will now hinge upwards

(5) Carefully unplug the two connectors (linking the panel to the chassis) by pulling the connectors straight out without exerting any side pull.

Replace the control panel by carrying out the above instructions in the reverse order.

7.15 REMOVAL OF THE CONTROL PANEL CIRCUIT BOARD

(1) Ensure that the machine is switched OFF at the control panel.

(2) Remove the four fixing screws and withdraw the circuit board.

This allows access to the switches and indicator lamps.

Care should be taken that the board does not come into contact with the mains switch contacts whilst taking measurements with the power switched ON.

The board is replaced by carrying out the above in the reverse order.

7.16 REMOVING THE BOTTOM FRONT PANEL

This panel is attached to the chassis by two hinges on the bottom edge.

Remove the six fixing screws and carefully pull the panel downwards.

Replace using the reverse order.

7.17 REMOVAL AND REPLACEMENT OF THE POWER SUPPLIES

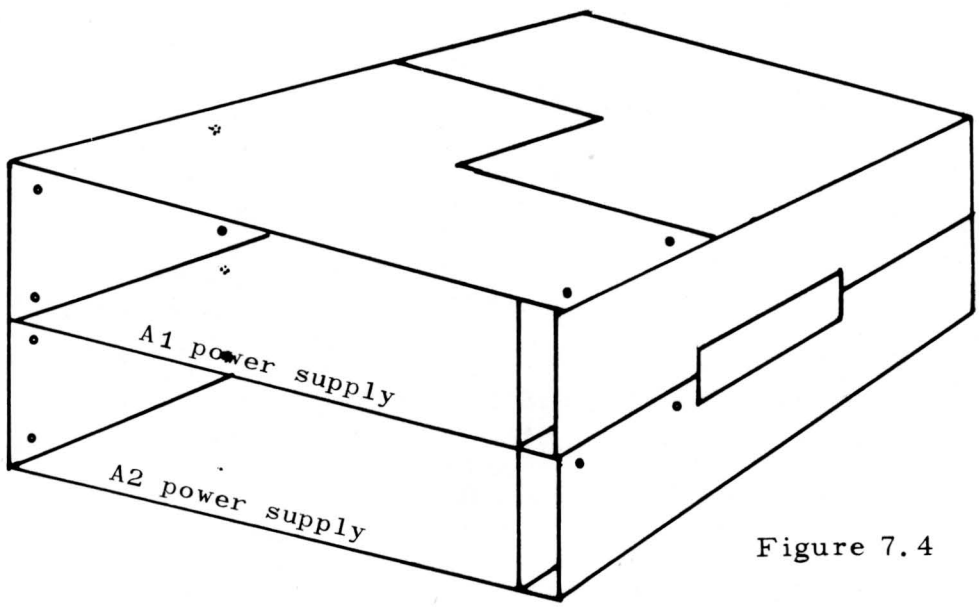


Figure 7.4

position of power
supply retaining screws

Ensure that the mains plug has been disconnected.

7.18 A1 POWER SUPPLY REMOVAL

(1) Remove the mains cable clamp.

(2) Remove the eight retaining screws (see Figure 7.4) four on each side.

(3) Very carefully withdraw the power supply from the back of the chassis, taking care not to damage the cable.

This operation will allow access to the logic cards of the CPU and memory unit(s) n°1 and 2 in the top chassis.

7.19 A2 POWER SUPPLY REMOVAL

(1) Remove the mains cable clamp.

(2) Remove the power supply retaining screws (see figure 7.4) and withdraw the power supply in the same way as for power supply A1.

This operation will allow access to the CU cards and memory unit(s) n° 3 and 4.

7.20 REPLACING THE POWER SUPPLIES

The power supplies should be replaced by reversing the order of the steps used for their removal. Take care when replacing the supplies not to trap or damage the cable.

7.21 REMOVAL AND REPLACEMENT OF CIRCUIT CARDS

The machine must be switched off at the control panel before either removing or replacing a card. Before a control unit card can be removed, the input/output socket that connects the peripheral to the card must be carefully removed.

The circuit cards should be removed by pulling straight out without exerting any lateral pressure on the card. For this reason it is advisable to use both hands when either removing or replacing a card.

When replacing a card, care should be taken to ensure that it engages correctly in location slots on each side of the chassis. No excessive pressure must be used when replacing a card otherwise the card or circuits may be damaged.

7.22 REMOVAL AND REPLACEMENT OF THE MEMORY UNIT(S)

Switch off the machine at the control panel before removing or replacing a memory unit.

To remove the memory unit, pull down on the **RED** lever (top centre of the unit) to release the retaining latch and withdraw the unit.

When replacing a memory unit, make sure that the **RED** retaining latch is properly engaged in its socket.

7.23 TAKING MEASUREMENTS

CONTROL PANEL CIRCUIT BOARD

When taking measurements on this board, the probe must not be connected directly to the pins. The probe should be attached firmly to some fixed point and a single flexible wire should be connected to the probe. At the other end of the wire should be a single pin socket which is then inserted in the appropriate circuit board pin. This method prevents any undue tension being exerted on the pins causing pin flexion or short-circuits with neighbouring pins.

LOGIC CARD MEASUREMENTS

Before measurements can take place, the card must first be removed and replaced by an extension card. The card to be tested is then inserted in the extension card sockets. A probe with a fine point should be used because of the high density of printed circuit tracks.

Measurements to individual integrated circuit modules must be made using the clip specially designed for this purpose.

NOTES :

(1) The removal of one or more of the logic cards for the purpose of making special measurements will not adversely affect the operation of the remaining cards.

(2) Whilst making tests on a disassembled machine, care should be taken that the mains leads of the switch do not come into contact with any metal object.

A check should also be made that the blower fans operate normally and that any mains connections are properly isolated.

P850 Table-Top 4K

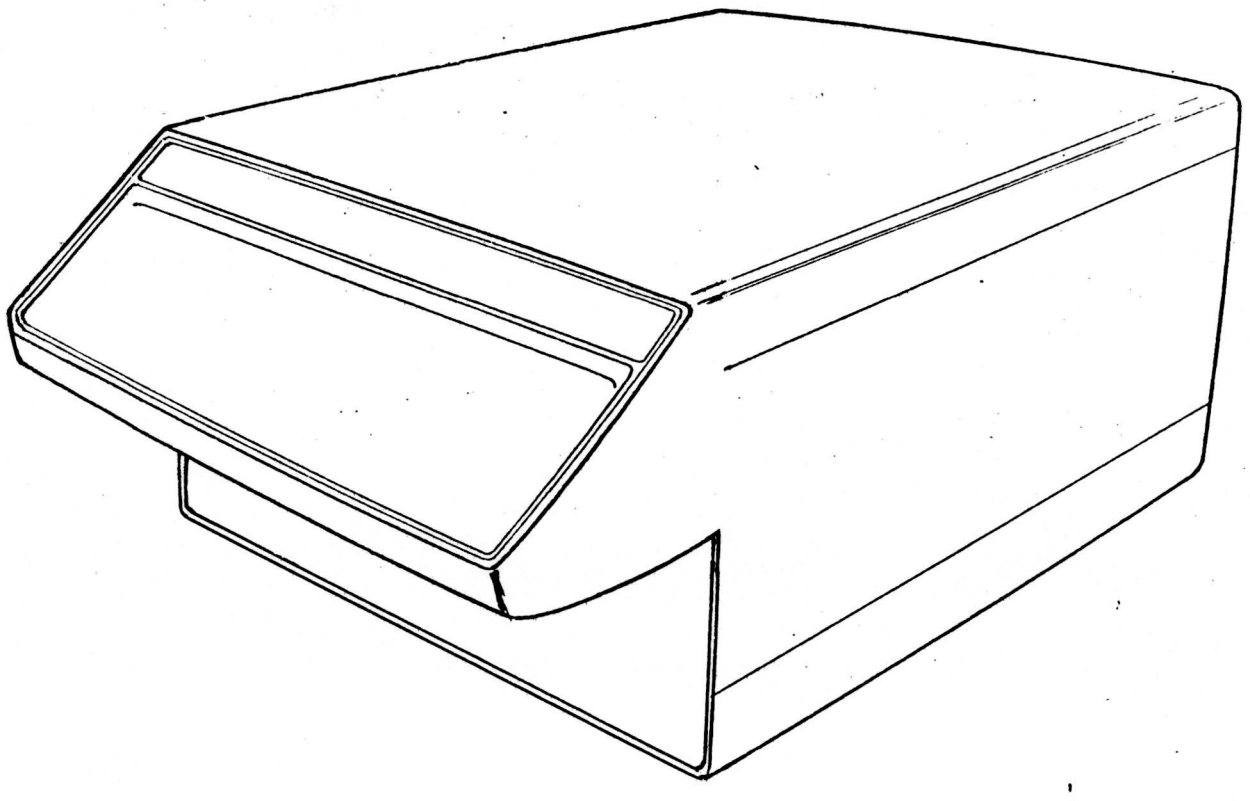


Figure 7.5 4K Table-Top

7.24 INSTRUCTIONS FOR REMOVING AND REPLACING SUBASSEMBLIES

The P850 computer is produced as a table-top model complete with cover. This cover is removable to allow easy access to the various components of the computer.

7.25 REMOVAL OF THE COVER

The following routine must be used for removing the cover:

(1) Switch off the computer at the control panel and remove the mains plug.

(2) Remove the back grid.

(3) Remove the front grid.

(4) Remove the two side panels. These items simply lift off and are replaced by pressing back into position.

(5) Remove the top of the cabinet. This is done by removing the four retaining screws holding the cabinet to the mounting.

These steps will allow access to the subassemblies and cards in the top chassis. To gain access to the subassemblies and cards in the bottom chassis the control panel and the bottom half of the cabinet must first be removed. This can be done using the following routine:

(6) Using the correct tool, remove the four screws that secure the control panel to the mountings.

CAUTION: Take care not to damage the connectors or wiring when carrying out the next step.

(7) Carefully unplug the two connectors (linking the panel to the panel to the chassis) by pulling the connectors straight out without exerting any side pull.

(8) Remove the cable clamp that secures the main cable to the chassis.

(9) Remove the eight retaining screws (four on each side) and withdraw the chassis from the bottom half of the cabinet.

7.26 REMOVAL AND REPLACEMENT OF THE POWER SUPPLIES

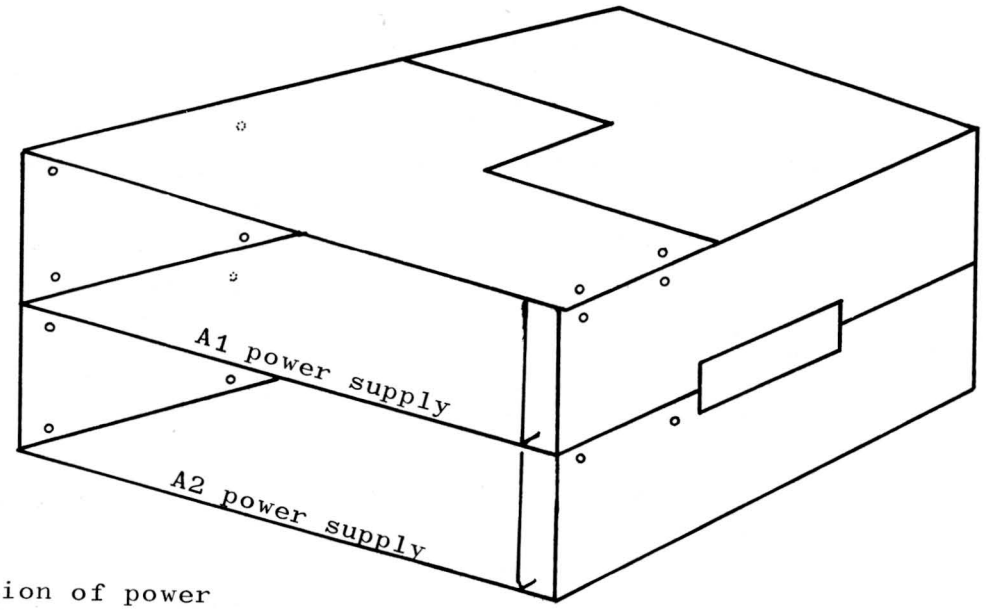


Figure 7.6 Retaining Screws

Ensure that the mains plug has been disconnected.

7.27 A1 POWER SUPPLY REMOVAL

(1) Remove the top member of the mounting frame.

(2) Remove the mains cable clamp.

(3) Remove the eight retaining screws (see Figure 7.6) four on each side.

(4) Very carefully withdraw the power supply from the back of the chassis, taking care not to damage the cable.

This operation will allow access to the logic cards of the CPU and memory unit(s) n° 1 and 2 in the top chassis

7.28 A2 POWER SUPPLY REMOVAL

Before this supply can be removed, the control panel and the bottom half of the cabinet must be removed as described above.

(1) Remove the mains cable clamp.

(2) Remove the power supply retaining screws (see Figure 7.6) withdraw the power supply in the same way as for power supply A1.

This operation will allow access to the CU cards and memory unit(s) n° 3 And 4.

7.29 REPLACING THE POWER SUPPLIES

The power supplies should be replaced by reversing the order of the steps used for their removal. Take care when replacing the supplies not to trap or damage the cable.

7.30 REMOVAL AND REPLACEMENT OF CIRCUIT CARDS

The machine must be switched off at the control panel before either removing or replacing a card. Before a control unit card can be removed, the input/output socket that connects the peripheral to the card must be carefully removed.

The circuit cards should be removed by pulling straight out without exerting any lateral pressure on the card. For this reason it is advisable to use both hands when either removing or replacing a card.

When replacing a card, care should be taken to ensure that it engages correctly in location slots on each side of the chassis. No excessive pressure must be used when replacing a card otherwise the card or circuits may be damaged.

7.31 REMOVAL AND REPLACEMENT OF THE MEMORY UNIT(S)

Switch off the machine at the control panel before removing or replacing a memory unit.

To remove the memory unit, pull down on the RED lever (top centre of the unit) to release the retaining latch and withdraw the unit.

When replacing a memory unit, make sure that the RED retaining latch is properly engaged in its socket.

7.32 REMOVAL OF THE CONTROL PANEL CIRCUIT BOARD.

Switch off the machine at the control panel before removing or replacing the circuit board.

Remove the four fixing screws and withdraw the circuit board. This will allow access to the switches and indicator lamps. Care should be taken that the board does not come into contact with the mains switch contacts whilst taking measurements with the power switched on.

7.33 TAKING MEASUREMENTS

CONTROL PANEL CIRCUIT BOARD

When taking measurements on this board, the probe must not be connected directly to the pins. The probe should be attached firmly to some fixed point and a single flexible wire should be connected to the probe. At the other end of the wire should be a single pin socket which is then inserted in the appropriate circuit board pin. This method prevents any undue tension being exerted on the pins causing pin flexion or short-circuits with neighbouring pins.

LOGIC CARD MEASUREMENTS

Before measurements can take place, the card must first be removed and replaced by an extension card. The card to be tested is then inserted in the extension card sockets. A probe with a fine point should be used because of the high density of printed circuit tracks.

Measurements to individual integrated circuit modules must be made using the clip specially designed for this purpose.

NOTES:

(1) The removal of one or more of the logic cards for the purpose of making special measurements will not adversely afffect the operation of the remaining cards.

(2) Whilst making tests on a disassembled machine, care should be taken that the mains leads of the switch do not come into contact with any metal object.

A check should also be made that the blower fans operate normally and that any mains connections are properly isolated.