This section discusses: (1) the functional units (based on Diagram 3-2, FEMDM); and (2) instruction fetching and execution, the Universal instruction set by instruction class, and power considerations.

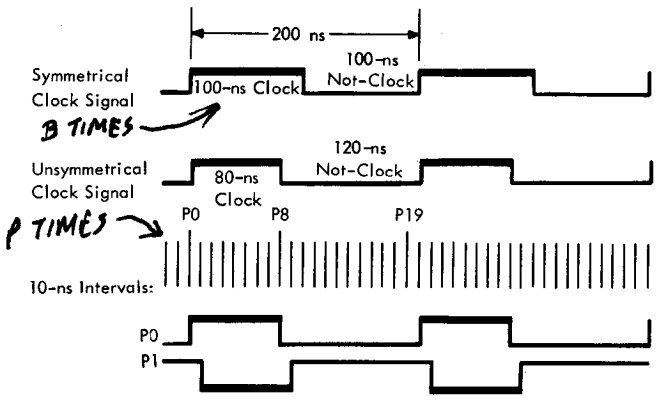
CONTROL

The 2065 CPU operates with a basic clock cycle period of 200 ns under control of ROS. The following paragraphs discuss: (1) CPU timing and data transfer; (2) ROS – what it is, how it controls the CPU, and its data flow; (3) the additional control provided by the PSW register.

CPU Timing

- Basic clock cycle period is 200 ns.
- Symmetrical clock signal consists of 100-ns clock portion and 100-ns not-clock portion.
- Unsymmetrical clock signal consists of 80-ns clock portion and 120-ns not-clock portion.

The basic CPU clock cycle period is 200 ns, divided into clock and not-clock portions. A clock signal generator provides a 5-megaHertz (5-mHz) symmetrical (100-ns/100-ns) clock signal. Two types of clock signal generators are used: a continuously running crystal-controlled oscillator in the Model G65, H65, and I65 CPU or a gated delay-line oscillator in the Model IH65 and J65 CPU. To provide additional time for CPU logic functions, the symmetrical clock signal is modified to give a 5-mHz unsymmetrical (80-ns/120-ns) clock signal. Finer intracycle control is obtained by dividing each of the two clock signals into 20 intervals of approximately 10 ns each. These intervals, named P0, P1, P2, . . . P19, are created by inverters which delay the signal by about 10 ns. Thus the notations P0, P1, P2, . . . P19 refer to signals which are inverted and are delayed 10 ns with respect to the previous signal:



The clock signals are distributed to logic gates A through E. Adjustable time delays within the logic gates synchronize the clock signals with a reference signal, thereby eliminating the various amounts of delay introduced by the distribution cables. The distribution of the clock signals to the CPU processing logic is stopped upon detection of an error or during scan, logout, single-cycle, and certain ROS operations. When the CPU clock signals are stopped, the BCU must continue to service the I/O channels and the scan logic must be operable. Therefore, the clock signals to these areas of the logic are not stopped, except under certain circumstances in the Model IH65 and J65 CPU.

Data Transfer

Data is transferred into a register, into an adder, and into and out of LS by gating signals controlled by ROS (Figure 1-16). Referring to Figure 1-16, note that data from PAL is always available at the A-register, B-register, and T-register, but is transferred only into the selected register by means of the corresponding gating signal from ROS. When gating data into an adder, timing considerations require the use of 'gate control' triggers; these triggers, which are set by the ROS decode logic, generate the required gating signal.

When transferring data into LS, the gating signal from ROS is combined with a signal from the LS addressing logic to develop a 'write LS' signal, which gates the data into LS. When transferring data from the LS, an address signal selects 1 of 24 LS registers, the contents of which are transferred to the LS bus. A second gating signal transfers the data from the LS bus to the S- or T-register.

Read-Only Storage

The CPU is controlled by ROS, a permanently recorded microprogram, supplemented by conventional control logic. A read-only storage is a storage device which contains information (1's and 0's) of a nondestructive nature. The 2065 CPU utilizes a capacitive read-only storage, in which bits are stored in the form of a capacitance between a fixed drive-plate pattern etched at right angles to a sense-plate pattern. Sense and drive plates are separated by a Mylar† film (approximately 1 mil thick), and the resulting sandwich is held together under pressure. A 1-signal is coupled from a drive line to one of a pair of sense lines, and a 0-signal is coupled to the other. Sense line outputs are detected in a differential amplifier which in turn feeds a latch. When decoded, the information in ROS controls gates to route data in the CPU. Access time is approximately 95 ns.

†Trademark of E. I. duPont de Nemours & Co. (Inc.)

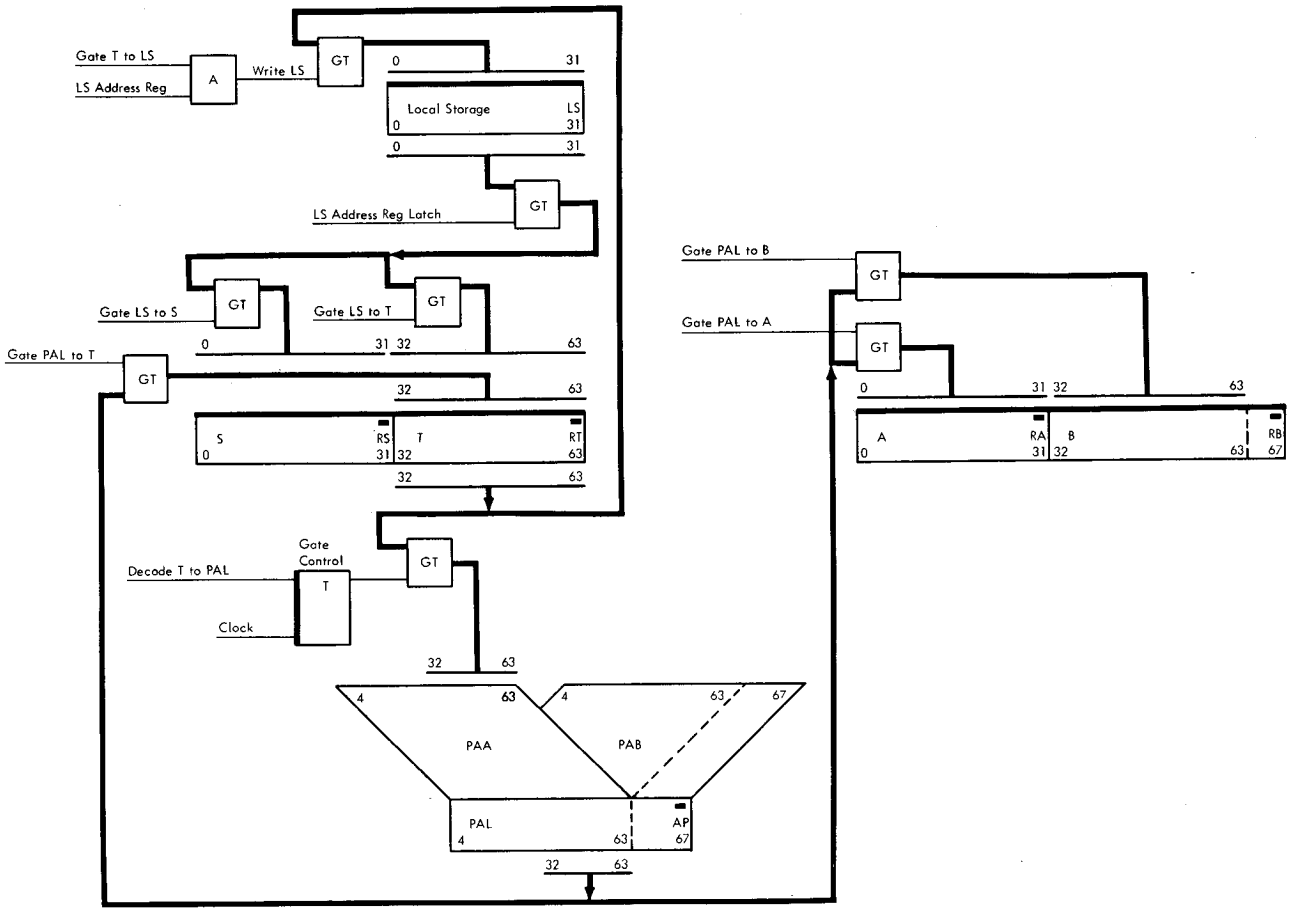


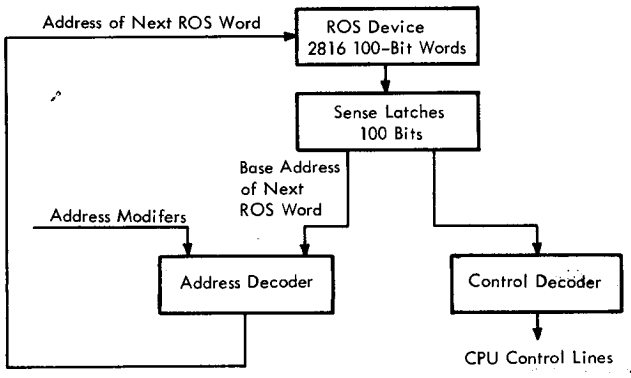
Figure 1-16. Data Transfer Scheme

Relationship of ROS to Conventional Controls

- ROS words replace most conventional sequence triggers and control lines.
- ROS word is a unique bit configuration and controls CPU during machine cycle it is in use.
- In addidion to control data, ROS word holds address of next ROS word and branch tests, if any.
- For branches, 1 ROS word is associated with each possible condition.

To understand ROS operation, it is helpful to note its relationship to conventional controls. Conventional controls may be characterized by sequence triggers, and by the control lines activated by the sequence triggers as a function of the operation to be performed and data conditions. Each cycle that the CPU may take represents a state of the CPU as defined by the control circuits. Each state, in turn, specifies which control lines are to be activated during that cycle and which state is to follow next. The defined state will cause the next sequence trigger to be set in the following cycle. In some cases, the next state may be contingent upon a branch condition in which one of two or more sequence triggers must be selected.

In ROS-controlled CPU's, the sequence triggers are replaced by micro-instructions or ROS words. Each ROS word consists of a predetermined bit pattern and represents a state of the CPU. A micro-instruction is read into the sense latches from the ROS device as follows:



Decoding of the bit pattern activates control lines which initiate operations or functions in the CPU under timing control of ROS decode logic. The base address of the next ROS word to be used is also included in each ROS word. Data conditions within the CPU may modify the address if

the bit pattern indicates a test for branching (e.g., branch if overflow occurs). One ROS word is associated with each possible condition; the base address is modified to address the ROS word which satisfies the data conditions. Thus ROS eliminates the need for most of the complex sequencing networks.

ROS Word

- ROS word is divided into 100 bits, grouped into 21 control fields.
- Number of bits within field determines number of unique control signals within field.
- Control signals are termed *micro-orders*.

The ROS is physically organized into 16 planes, each plane holding 88 200-bit words. Through addressing, the 200-bit word is further divided in half to yield 2816 100-bit words, hereafter referred to as *ROS words*. Each ROS word consists of a unique predetermined bit configuration grouped into 21 control fields (Table 1-6). The number of bits within a field determines the number of unique control signals (micro-orders) available within that field. (In a four-bit field, for example, 16 distinct micro-orders can be defined, only one of which can be activated at any one time.) The micro-orders are grouped functionally within the fields according to two rules:

1. Micro-orders that are functionally similar (such as micro-orders that control ingating to the AB register) are grouped in one field for ease of decoding.
2. All micro-orders grouped in a field must be mutually exclusive because only one micro-order within that field may be specified at a time.

Usually, rule 1 results in rule 2.

When decoded, each micro-order activates one or more control lines that condition gates to perform the function specified by the micro-order. Each micro-order is assigned a mnemonic code (up to 12 characters) that defines the control function performed. As an example, Table 1-7 lists the micro-orders and associated microcommands pertaining to control field V of the ROS word, referenced to the bit configurations of that field, and their function.

Each ROS word is represented by a block on a Control Automation System (CAS) Logic Diagram (CLD). The CLD is to the ROS microprogram what an ALD (Automated Logic Diagram) is to logic. The blocks are connected to show the logical sequence of ROS words to perform the specific function. Refer to ALD M7061 for a definition of the CLD format and content, and of the ROS block language and information contained within the block.

Table 1-6. ROS Word Breakdown

Bits	Control Field	Function Controlled
0–5	–	Spare
6–9	A	A-, B-, and IC-register ingating
10, 11	B	LS to S- and T-register ingating
12–16	C	PSW and S-, T-, D-, G-, and Q-register ingating
17–19	D†	F-register ingating, and end-op signals
20	–	Parity
21–24	E	E- and R-register ingating
25–30	F†	Status triggers and miscellaneous control lines
31–35	G†	Status triggers, IC, and miscellaneous control lines
36, 37	–	Spare
38–42	H	LS
43–46	L	Storage requests and setting of mark triggers
47–56	NA	Base address of next ROS word
57–61	K	Y-conditional branches
62–68	J	Z- (and X-) conditional branches
69–73	M	Serial adder A bus
74–77	N	Serial adder B bus
78–80	P	Parallel adder latches
81	–	Spare
82–84	Q	Hot 1's to parallel adder A-side
85	–	Parity
86	R	F- and AB-register outgating to serial adder A-bus
87–90	T	A-, B-, IC-, and F-register outgating to parallel adder B-bus
91	–	Parity
92–95	U	S-, T-, and D-register outgating to parallel adder A-bus
96	–	Spare
97–99	V	E- and Q-register outgating to parallel adder B-bus

† Control fields D, F, and G serve two functions. In the normal processing mode, they are decoded to yield standard CPU micro-orders; in the scan mode, they are identified as field S, and they yield special scan micro-orders (using common micro-order codes). The choice of modes is controlled by a 'scan mode' trigger.

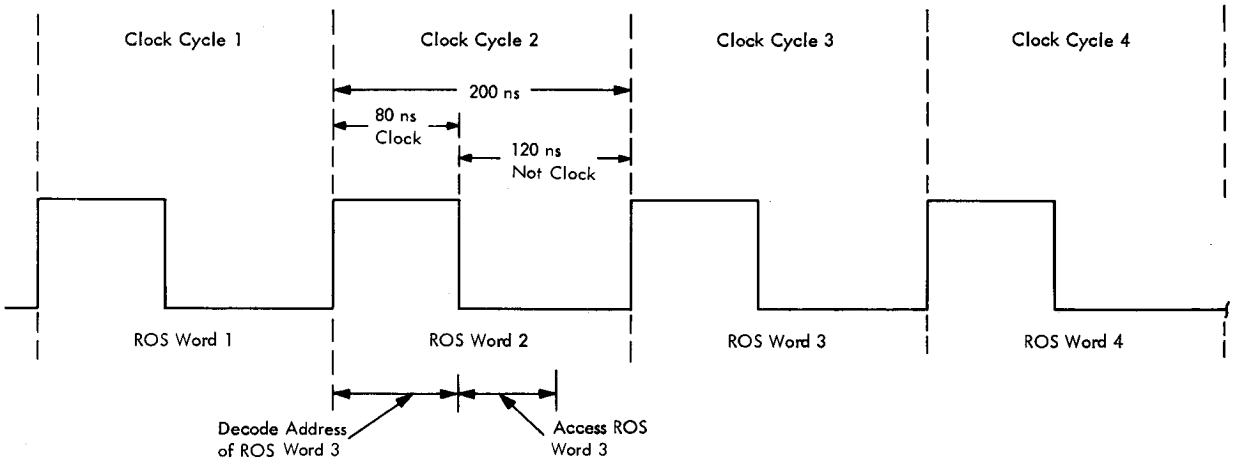
Table 1-7. Control Field V (Bits 97–99); E, Q outgates to Parallel Adder B-Bus

Bit Configuration			Micro-Order Mnemonic	Function
97	98	99		
0	0	0	0	Zero gated with parity
0	0	1	E3	E(12–15) to PB(60–63)
0	1	0	E2	E(8–11) to PB(60–63)
0	1	1	E23	E(8–15) to PB(56–63)
1	0	0	Q7	Q(52–63) to PB(52–63)
1	0	1	Q5	Q(36–47) to PB(52–63)
1	1	0	Q3	Q(20–31) to PB(52–63)
1	1	1	Q1	Q(4–15) to PB(52–63)

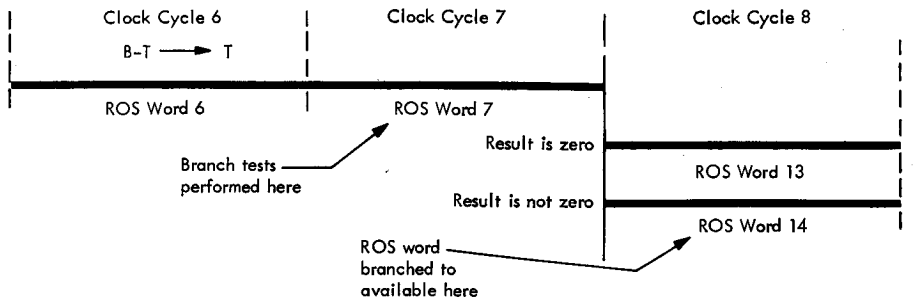
ROS Addressing and Branching

- Conditional branches are dependent on internal conditions of previous cycle.
- Word addressed as result of branch test is not available until 1 cycle later.
- ROS word is addressed by 12-bit binary address:
  0–9 is 10-bit base address.
  10 is Y-branch bit.
  11 is Z-branch bit.
- X-branch (functional branch) affects more than 1 bit.
- Overriding branch blocks 12-bit address and forces new 12-bit address into ROSAR.

As described earlier, the CPU cycle presently being executed is controlled by the ROS word addressed during the previous cycle. Referring to Figure 1-17, A, the normal sequence of ROS words is achieved by placing the address of ROS word 2 into ROS word 1, the address of ROS word 3 into ROS word 2, and so on. The address of the next ROS word is decoded during clock time of the cycle controlled by the present ROS word; the next ROS word is accessed during not-clock time of the cycle.



A. Sequential ROS Word Addressing



B. Conditional Branch ROS Word Addressing

Figure 1-17. ROS Addressing and Branching

Conditional branches are dependent on the internal conditions of the previous cycle. It is important to note that the ROS word addressed as a result of the branch test is not available until one cycle later. To explain this 1-cycle delay in addressing, assume that ROS word 6 contains the micro-orders necessary to subtract the contents of the T-register from the contents of the B-register, and to place the result into the T-register (Figure 1-17, B). Assume further that if the result is zero, a branch will be made to ROS word 13; if not zero, the next ROS word addressed will be 14. Contained in ROS word 6 is the address of ROS word 7, which defines the branch test and contains the associated branch addresses.

The results of the arithmetic operation performed in cycle 6 are tested during clock time of cycle 7. It is during this time that the address of ROS word 13 or 14 (depending on the results of the branch test) is decoded: the selected ROS word is accessed during not-clock time of cycle 7. Hence, the ROS word branched to as a result of the arithmetic operation performed during cycle 6 is not available until cycle 8.

ROS words are selected by means of a 12-bit binary address. The address is held in the ROS address register (ROSAR), whose bit positions are numbered 0 through 11, high order to low order. Bits 0 through 10 specify a 200-bit doubleword in ROS; bit 11 gates the proper 100-bit half to the ROS sense latches (Figure 1-18).

The 12-bit address is made up of three components:

1. A 10-bit base address, bits 0–9.
2. A conditional branch test, or an unconditional value of 0 or 1 applying to bit 10, designated Y-branch.
3. A conditional branch test, or an unconditional value of 0 or 1 applying to bit 11, designated Z-branch.

Included in the Z-branch field of micro-orders is a subset of branch micro-orders called X-branch or functional-branch micro-orders. The X-branch micro-orders affect more than one bit of the ROS address.

Included in the Y-branch field of micro-orders is a subset of overriding branch micro-orders. When the conditions tested by these micro-orders are satisfied, the full 12-bit address is blocked and a new 12-bit address is forced into ROSAR.

If branching conditions are to be tested, the address bits that may be affected by the branch must be set to 0's, except in the case of an overriding branch. If the branch is satisfied, 1's are forced into the ROSAR bit positions associated with that branch test; if the branch condition is not satisfied, the bits remain 0's. Thus the address is modified only if the branch is to be taken.

Addresses can be grouped into four categories: (1) no branch specified, (2) Y- and/or Z-branches, (3) X-branches, and (4) overriding branches. The following paragraphs discuss the addressing for each category. Refer to ALD M7061 for a definition of ROS addressing and branching terms used in the following paragraphs.

No Branch Specified. If no branch tests are to be made, there is only one possible ROS word that can follow, and hence only one possible next address. Accordingly, the 10-bit base address (bits 0–9) and absolute values of 1 or 0 for bits 10 and 11 are specified. The micro-orders that unconditionally set an absolute value into bits 10 and 11 are:

1. 0 in left of R-line (K0), which sets bit 10 to a 0.
2. 1 in left of R-line (K1), which sets bit 10 to a 1.
3. 0 in right of R-line (J0), which sets bit 11 to a 0.
4. 1 in right of R-line (J1), which sets bit 11 to a 1.

The appropriate Y- and Z-branch micro-orders are selected, and bits 10 and 11 are set correspondingly.

Y- and/or Z-Branches. Conditional branch addresses may be specified in which bits 10 and/or 11 are affected.

Only a Y-branch can be executed as follows. A 10-bit base address and an absolute value or X for bit 11 are specified. A branch test is defined in the Y-branch micro-order control field. If the branch condition is satisfied, bit 10 is set to a 1; if not, bit 10 remains a 0. For example, micro-order 'WCRY' sets bit 10 to a 1 if a carry is detected in the serial adder. If there is no carry, bit 10 remains a 0.

Conversely, only a Z-branch can be executed as follows. Here, a 10-bit base address and an absolute value for bit 10 are specified. If the branch test defined in the Z-branch micro-order control field is satisfied, bit 11 is set to a 1; if not, bit 11 remains a 0.

Certain situations require the use of both Y- and Z-conditional branches simultaneously. The 10-bit base address is specified, and bits 10 and 11 may assume one of the following values:

Bits		Branch Results
10	11	
0	0	Y- and Z-branch conditions both unsatisfied.
0	1	Z-branch condition only satisfied.
1	0	Y-branch condition only satisfied.
1	1	Y- and Z-branch conditions both satisfied.

X-Branches. Where a branch to one of four or more possible addresses is required (as well as some special 64-way branch tests), an X-branch is used. The X-branch may affect bits 10 and 11 (four-way branch), bits 9–11 (eight-way branch), bits 8–11 (16-way branch), or bits 6–11 (64-way branch). An example of an X-branch is the 64-way branch, 'E(2–7)→ROA', made at the end of the I-Fetch sequence per the op code to enter the execution phase of the specific instruction.

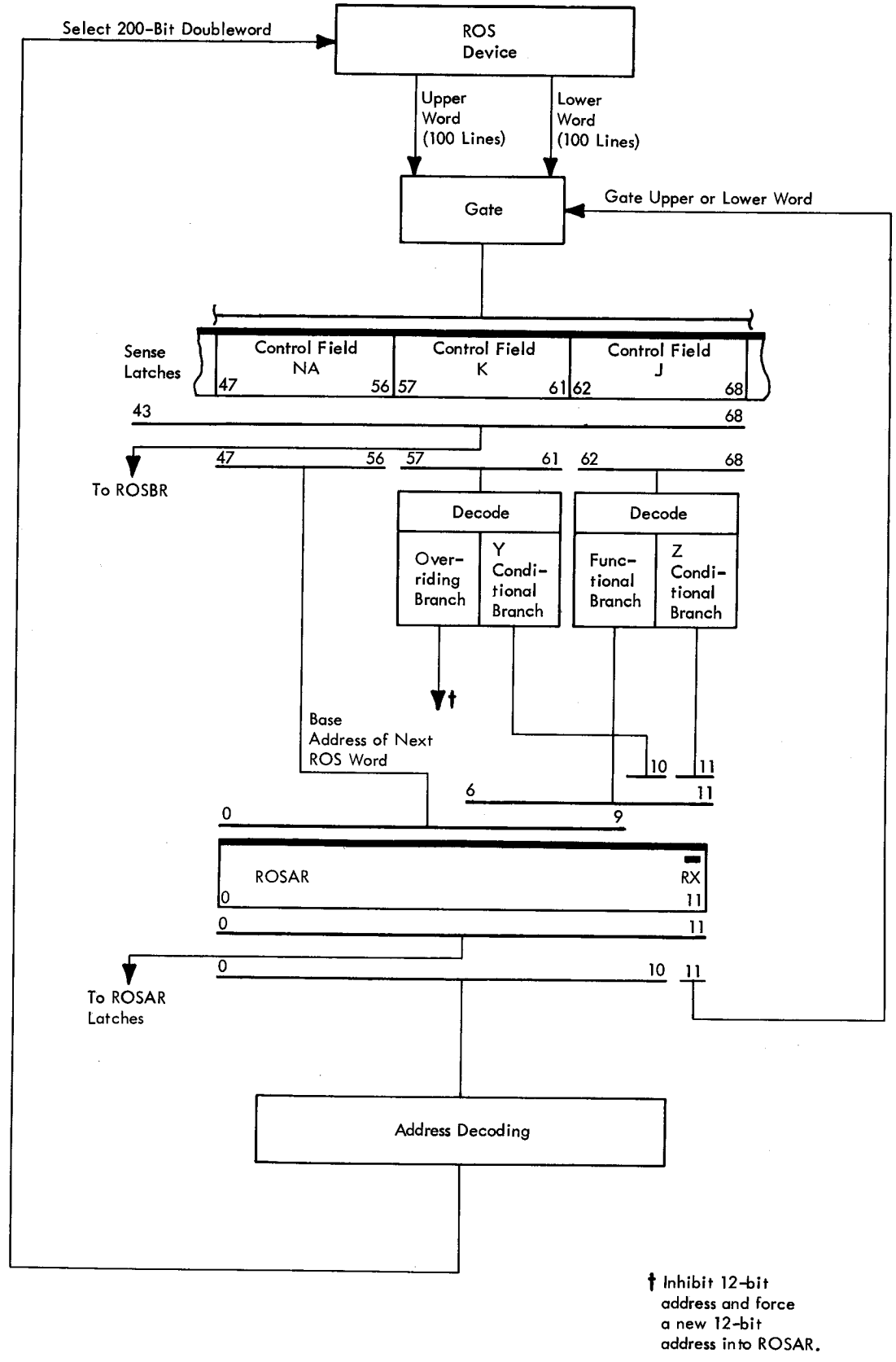


Figure 1-18. ROS Addressing Block Diagram

For these multiway branches, one condition sets the associated address bit to a 1. To illustrate, assume conditions A, B, and C sets bits 9, 10, and 11, respectively, to a 1. The possible results are:

Bits 9	Bits 10	Bits 11	Branch Results
0	0	0	None of the conditions is satisfied.
0	0	1	Condition C is satisfied.
0	1	0	Condition B is satisfied.
0	1	1	Conditions B and C are satisfied.
1	0	0	Condition A is satisfied.
1	0	1	Conditions A and C are satisfied.
1	1	0	Conditions A and B are satisfied.
1	1	1	All three conditions are satisfied.

The addressing is similar to that previously discussed. A 10-bit base address is specified, with those bits that may be affected by the X-branch set to 0. Thus, for the example given above, bit 9 of the base address is set to 0, the Y-branch micro-order control field contains micro-order 0 in left of R-line to set bit 10 to 0, and bit 11 is automatically set to 0 when the X-branch is specified. Subsequently, the bit(s) associated with successful condition(s) is set to 1. The ROS word addressed will be that ROS word whose address satisfies the branch conditions.

Overriding Branches. There are exceptional machine conditions (such as interruptions) for which the normal ROS word sequence must be stopped and a new sequence started. This change is accomplished by an overriding branch specified in the Y-branch micro-order control field.

The normal sequencing address is made up of (1) the 10-bit base address and (2) bit 11 set to 0 automatically because the overriding branch is a function of the Y-field. Because the overriding branch is specified in the Y-control field, no Y-branch can be specified.

If the overriding branch condition is satisfied, the normal full 12-bit address is blocked and a new address, as determined by the overriding branch condition, is forced into ROSAR.

ROS Data Flow

- ROS data flow units are:
  100 sense latches, 1 per ROS word bit
  ROSAR
  ROSAR latches
  ROSPARA and ROSPARB
  ROSDR
  ROSDR latches
  ROSBR
  Decode logic

- Control fields may be:
  Decoded directly from sense latches.
  Placed into ROSDR and subsequently decoded.
  Placed into ROSDR, sent to ROSDR latches, and then decoded.

The 100-bit ROS word is divided into 21 control fields. When read out from ROS, the ROS word is placed into 100 sense latches, one latch for each bit position. The control fields are handled according to the functions they control. They may be (Figure 1-19):

1. Decoded directly from the sense latches (control fields L, K, J, R, T, U, and V) or transferred directly to ROSAR (control field NA).
2. Placed into the ROS data register (ROSDR) and decoded (control fields H, M, N, P, and Q).
3. Transferred to ROSDR latches from ROSDR (control fields A–G).

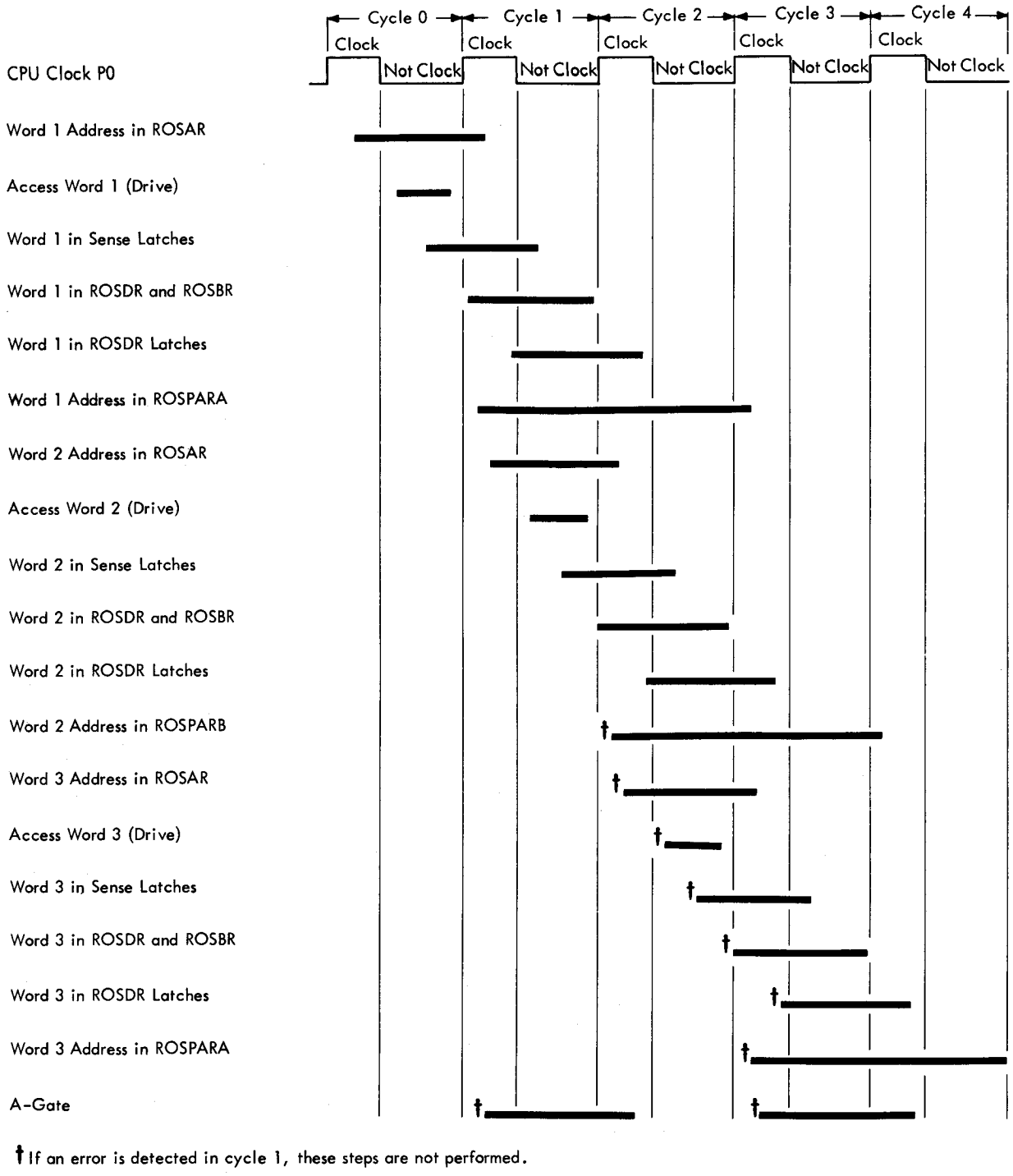
Assuming ROS words and the cycles they control are designated 1, 2, and 3, ROS word 1 is set into the sense latches during not-clock time of cycle 0 (Figure 1-20). The control fields used during clock time of cycle 1 are decoded directly from the sense latches (case 1 above). These control fields, which may be considered critical timing fields, control inputs to the adders, define the base address and branch tests for the next ROS word, and control the storage-request and mark triggers.

Although the sense latches are cleared at not-clock time of cycle 1, the control fields of ROS word 1 that are required during that time (case 2 above) are placed into ROSDR at clock time of cycle 1. These signals control the adders and LS. Note that both portions of the ROSDR associated with the adders are packaged physically with the adders they control. The balance of the ROSDR serves control fields A–H.

Control fields A–G control register inputs and triggers that are to be set during clock time of cycle 2 (case 3 above). Although the ROSDR is reset at the end of cycle 1, the ROSDR latches keep control fields A–G available for that additional 80 ns (Figure 1-20).

Control fields L, NA, K, J, R, T, U, and V are sent to the ROS backup register (ROSBR) from the sense latches. The ROSBR does not play a part in the ROS functions; it provides an indication of the subject fields during maintenance (test) mode. When the CPU stops on an error during test mode, the ROSBR contents can be used by the CE to help isolate malfunctions.

The NA control field, in addition to being stored in the ROSBR, is stored in ROSAR and provides the base address as previously explained. During each ROS cycle, the contents of ROSAR are sent to the ROSAR latches which, in turn, are alternately gated (by means of an 'A-gate' signal) to the ROS previous address A (ROSPARA) and ROS previous address B (ROSPARB) registers. These registers serve the same purpose as the ROSBR; i.e., provide the CE with an indication for maintenance use during test



† If an error is detected in cycle 1, these steps are not performed.

Figure 1-20. ROS Timing

mode. When an error causes the CPU to stop in test mode, ROSAR, ROSPARA, and ROSPARB provide the addresses of the next ROS word, current ROS word, and previous ROS word (Figure 1-20).

ROS Control of CPU

Efficient control of CPU operations is achieved by overlapping ROS words. Clock signals (P0–P19) time ROS sense latches, ROSDR, and ROSDR latches, thus allowing the processing of parts of more than one ROS word simultaneously To illustrate, Figure 1-21 shows the ROS word timing relationship for a hypothetical example.

The address of word 1 is gated into ROSAR from the ROS sense latches at P5 of word 0. [ROSAR(11) may be set as late as P7.] Information from word 1 enters the ROS sense latches at P0 + 160 ns (P16). (In normal operation, a

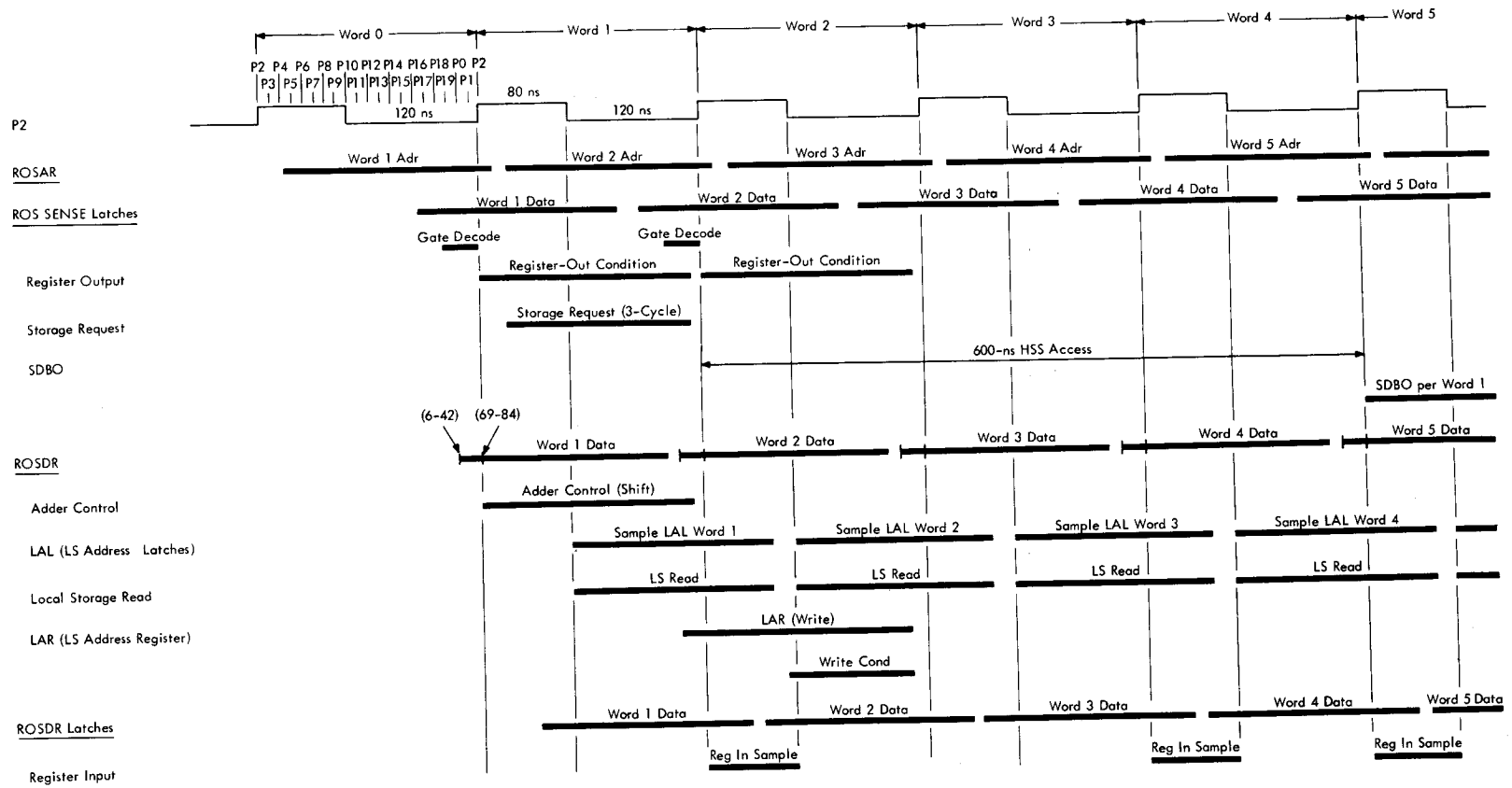


Figure 1-21. ROS Control of CPU Operations

Timing see p. 1-47

new word enters the ROS sense latches every 200 ns.) In the example, word 1 controls: (1) register output, (2) main storage request, (3) adder shift, (4) local storage write, and (5) register input. A register output micro-order gates register data into an adder at P2 by means of 'gate control' triggers. A three-cycle main storage request is initiated at P4 to fetch information which is used three cycles later. Note that register output and the main storage request are decoded directly from the ROS sense latches.

Bits 6–35 and 38–42 enter ROSDR at P0, and bits 69–80 and 82–84 enter ROSDR at P2; they are decoded for adder control and LS operations. In the example, micro-orders generate an adder shift and a local storage write operation. The shift is performed immediately, but the local storage write operation is delayed because a local storage read operation is automatically set up first. The LS address is entered into the local storage address latches (LAL); a read operation is performed, but data is not gated into a register.

The address then is gated into the local storage address register (LAR) to perform the write operation. Note that the write condition, ordered by word 1, starts after word 2 has been transferred to the ROSDR. The figure shows an LS read operation for every word because this sequence happens even if it is not ordered. When no order is given to LS, a readout of LS address 0 is performed but the data is not used. Forcing the read operation saves time if it is needed.

ROSDR(6–35) is gated to the ROSDR latches at P7 to retain word 1 information at the same time word 2 is set into ROSDR. In the example, word 1 transfers data into a register at clock time (P2) of word 2. This action, along with an LS write operation, illustrates ROS word overlap because these operations are performed at the same time the register out condition is energized from the decoding of word 2.

Word 2 has only one micro-order, register output, but gates are conditioned to transfer word 2 data from the ROS sense latches to the ROSDR and then to the ROSDR latches. Note that the LS read micro-order is active, though not ordered. Word 3 operates in a similar manner, but the only micro-order is a register input which takes place during word 4.

As the ROS words are executed, the main storage request is processed and data is returned on the Storage Data Bus Out (SDBO); word 4 contains the micro-order to gate the data into a register for further processing.