Contents

Introduction

This manual was written to provide an understanding of the microprogramming language for the people involved with creating or interpreting microprograms on the IBM System 360-Model 30.

Microprogramming provides the control for most of the functions performed in the IBM System 360-Model 30. These functions consist of memory control, Arithmetic and Logic Unit Controls (ALU), hardware register input and output controls, machine status controls, Read Only Storage (ROS), sequencing controls and some I/O controls.

The microprogramming instructions are explained individually to simplify the complexity of the microprogramming language. Each instruction has a description of the control field bit structure, branching conditions, hardware registers and latches, and buss and tag lines involved. Charts are also provided for timings, microword formats, data-flow and local storage layout to clarify the microprogramming functions.

Model 30 Machine Control Specifications

The Read Only Storage (ROS) must provide control for most of the functions performed in the Model 30. These functions, to repeat, consist of memory control, ALU (Arithmetic and Logic Unit) controls, hardware register input and output controls, machine status controls, ROS sequencing control and some I/O control.

The Data Flow Chart for the Model 30 is shown in the Appendix. The breakdown of the ROS output, as punched in a bit card, can also be found in the Appendix. A GEOG address is the location of the ROS word on the microprogramming flow diagram. The HEX address is the location of the word in the ROS. The output is described in columns 11 through 72 (62 bits) of the CCROS Document Card.

The ROS output is divided into subfields called control fields. Some control fields may be decoded and used directly from the Sense Amplifier Latches (SAL). Some control fields require a control register to hold the ROS output information until it is used.

Control Field Description

CN—6 Bits	—	Next ROS address
PN—1 Bit	—	Odd parity on CN
PA—1 Bit	—	Odd parity on address of ROS word
CH—4 Bits	—	ROS address branching
CL—4 Bits	—	ROS address branching
CM—3 Bits	—	Address register/read write select
#CU—2 Bits	—	Main storage or local storage data destination
#CA—4 Bits	—	Input source for A buss and A register
AA—1 Bit	—	Alternate CA decoder bit
CB—2 Bits	—	Input source for B buss and B register
#CK—4 Btis	—	Constant generator
AK—1 Bit	—	Alternate CK decoder bit
PK—1 Bit	—	Parity for CK field
PS—1 Bit	—	Odd parity for SAL's

*CD—4 Bits — Destination from Z buss

*CF—3 Bits — Controls the Hi/Lo, crossed/straight functions of the A register entry into the ALU

*CG—2 Bits — Controls the Hi/Lo functions of the B register entry into the ALU

*CV—2 Bits — True/complement and Binary/decimal controls

*CC—3 Bits — Carry control and logic control

*CS—4 Bits — Status control

*AS—1 Bit — Alternate CS decoder bit

*PC—1 Bit — Odd parity on control registers

#—Fields with alternate decoders
*—Fields with control register

Control Field (Detailed Explanations)

CN Field — The high order 6 bits of the ROS address register X are loaded from the CN field.

PN Field — Provides odd parity on CN for generating parity on X when the other two bits of X are known.

PA Field — Provides a check on the word read from ROS. It is odd parity on the 12 bit address WX.

CH and CL Fields — These fields handle the branching of the ROS address. The CH field controls the X register 6 bit in the ROS address register, and the CL field controls the X register 7 bit. All conditions must be set before the cycle in which the branch is interrogated.

CH = 0000 or 0001
In this condition the constants are used for forced branching. CH = 0000 forces the X register 6 bit off (0), and CH = 0001 forces the X register 6 bit on (1). (See example 1A, page 17.)

CH = 0010 (RO)
This is a branch on a latch in the R register. If the R register 0 bit is a 1, the X register 6 bit is forced on (1), satisfying the branch condition.

CH = 0011 (V67 = 0) (GMWM—If 1401 feature)
This is a conditional branch. If the V register 6 and 7 bits are 0, the X register 6 bit is forced on (1), satisfying the branch condition. The GMWM, if a 1401 feature, is a conditional branch. (See example 1B, page 17.)

CH = 0100 (STI)
This is a conditional branch. If the status-in-line (Multiplex Channel) for the interface is up, the X register 6 bit is forced on (1) satisfying the branch condition. (See example IB, page 17.)

CH = 0101 (OPI)
This is a conditional branch. If the operational-in-line (Multiplex Channel) for the interface is up, the X register 6 bit is forced on (1), satisfying the branch condition. (See example IB, page 17.)

CH = 0110 (AC)
This is a conditional branch. If an adder carry resulted in the previous cycle, the X register 6 bit if forced on (1), satisfying the branch condition. (See example 1B, page 17.)

CH = 0111 (SO)
This is a branch on a latch in the S register. If the S register 0 bit is a 1, the X register 6 bit is forced on (1), satisfying the branch condition. The SO bit is the true/complement latch. (See example 1C, page 18.)

CH = 1000 (S1) (R2—If 1401 feature)
This is a branch on a latch in the S register. If the S register 1 bit is a 1, the X register 6 bit is forced on (1), satisfying the branch condition. The 1401 feature R2 works the same except that it deals with the R register. (See example 1C, page 18.)

CH = 1001 (S2)
This is a branch on a latch in the S register. If the S register 2 bit is a 1, the X register 6 bit is forced on (1), satisfying the branch condition. (See example 1C, page 18.)

CH = 1010 (S4)
This is a branch on a latch in the S register. If the S register 4 bit is a 1, the X register 6 bit is forced on (1), satisfying the branch condition. (See example 1C, page 18.)

CH = 1011 (S6)
This is a branch on a latch in the S register. If the S register 6 bit is a 1, the X register 6 bit is forced on (1), satisfying the branch condition. (See example 1C, page 18.)

CH = 1100 (GO)
This is a branch on a latch in the G register. If the G register 0 bit is a 1, the X register 6 bit is forced on (1), satisfying the branch condition. (See example 1C, page 18.)

CH = 1101 (G2)
This is a branch on a latch in the G register. If the G register 2 bit is a 1, the X register 6 bit is forced on (1), satisfying the branch condition. (See example 1C, page 18.)

CH = 1110 (G4)
This is a branch on a latch in the G register. If the G register 4 bit is a 1, the X register 6 bit is forced on (1), satisfying the branch condition. (See example 1C, page 18.)

CH = 1111 (G6)
This is a branch on a latch in the G register. If the G register 6 bit is a 1, the X register 6 bit is forced on (1), satisfying the branch condition. (See example 1C, page 18.)

CL Field

— CL = 0000 or 0001
In this condition the constants are used for forced branching. CL = 0000 forces the S register 7 bit off (0), and CL = 0001 forces the X register 7 bit on (1). (See example 1A, page 17.)

CL = 0010 (CAnn → W)
CAnn → W is used for module changing. The CCROS modules contain 256 bytes, therefore, the second "n" (in CAnn) would be used for addressing the basic CCROS. If the machine has a 1401 feature, an additional CCROS is added. Therefore, the first "n" would be used for addressing the 1401 feature CCROS. CA 0 8 → W and AA = 0 resets the W register 3 bit and forces the X register 7 bit on. The address of the next microword is in the 800 block. (See example 1D, page 18.)

CA 18 → W and AA = 1 forces the W register 3 bit on and puts the machine into the 1401 mode. The address of the next microword is in the 1800 block.

CL = 0011 (AI)
This is a conditional branch. If the address-in-line (Multiplex Channel) for the interface is up, the X register 7 bit is forced on (1), satisfying the branch condition. (See example 1B, page 17.)

CL = 0100 (SVI)
This is a conditional branch. If the service-in-line (Multiplex Channel) for the interface is up, the X register 7 bit is forced on (1), satisfying the branch condition. (See example 1B, page 17.)

CL = 0101 (R = VDD)
This is a conditional branch. If the R register contains a valid decimal digit, the X register 7 bit is forced on (1), satisfying the branch condition. (See example 1B, page 17.)

CL = 0110 (IBC) (R1—If 1401 feature)
This is a conditional branch. If a carry results out of the 1 bit position on the output of the ALU, the X register 7 bit is forced on (1), satisfying the branch condition. (See example 1B, page 17.)

Bit positions 0 1 2 3 4 5 6 7

CL = 0111 (Z = 0)
This is a conditional branch. If the Z buss contains all zeros, the X register 7 bit is forced on (1), satisfying the branch condition. (See example 1B, page 17.)

CL = 1000 (G7)
This is a branch on a latch in the G register. If the G register 7 bit is a 1, the X register 7 bit is forced on (1), satisfying the branch condition. (See example 1C, page 18.)

CL = 1001 (S3)
This is a branch on a latch in the S register. If the S register 3 bit is a 1, the X register 7 bit is forced on (1), satisfying the branch condition. The S3 bit is the carry latch. (See example 1C, page 18.)

CL = 1010 (S5)
This is a branch on a latch in the S register. If the S register 5 bit is a 1, the X register 7 bit is forced on (1), satisfying the branch condition. (See example 1C, page 18.)

CL = 1011 (S7)
This is a branch on a latch in the S register. If the S register 7 bit is a 1, the X register 7 bit is forced on (1), satisfying the branch condition. (See example 1C, page 18.)

CL = 1100(G1) (R3—If the 1401 feature)
This is a branch on a latch in the G register. If the G register 1 bit is a 1, the X register 7 bit is forced on (1), satisfying the branch condition. The 1401 feature R3 works the same except it deals with the R register. (See example 1C, page 18.)

CL = 1101 (G3)
This is a branch on a latch in the G register. If the G register 3 bit is a 1, the X register 7 bit is forced on (1), satisfying the branch condition. (See example 1C, page 18.)

CL = 1110 (G5)
This is a branch on a latch in the G register. If the G register 5 bit is a 1, the X register 7 bit is forced on (1), satisfying the branch condition. (See example 1C, page 18.)

CL = 1111 (INTR)
This is a conditional branch. If any of the following interrupt lines MPX,

SEL 1, SEL 2, EXT., TIMER are up, the X register 7 bit is forced on (1), satisfying the branch condition. (See example 1B, page 17.)

CM Field — This field controls the reading and the writing of memory.

CM = 0000 (Write)
This decode sends a write call to memory. It resets the allow write latch. If one write follows another write, the second write is ignored because the allow write latch was reset by the first write operation. If the write follows a read, the data read will set into R register and be regenerated. (See example 2A, page 19.)

CM = 0001 (Compute)
This decode asks for neither a read nor write call. This decode can be used for generating new R register data which will be written back, with either a write or store, during the next cycle. (See example 2B, page 19.)

CM = 0010 (Store)
This decode sends a write call to memory. It also resets the allow write latch. If the store follows a read, the *DATA READ WILL NOT* be set into the R register, but what is already in the R register will be regenerated. (See example 2A, page 19.)

CM = 0011 (IJ → MN)
This decode sends a read call to memory and specifies an address for the memory address MN register. It turns on the allow write latch and gates the IJ register address to the MN register. It also addresses that location of either the main storage or local storage. The data from the addressed location is read out and into the R register. If one read follows another read, the second read is ignored (no read call), but the IJ address is gated to the MN register. (See example 2C, page 19.)

CM = 0100 (UV → MN)
This decode sends a read call to memory and specifies an address for the MN register. This decode turns on the allow write latch and gates the UV register address to the MN register and addresses that location of either the main storage or the local storage. The data from the addressed location is read out and into the R register. If one read follows another read, the second read is ignored (no read call), but the UV address is gated to the MN register. (See example 2C, page 19.)

CM = 0101 (T → N)
This decode sends a read call to memory and specifies an address for the Lo order 8 bits of the MN register. This decode turns on the allow write latch and gates the T register address to the N register (the M register is reset). It addresses that location of either the main storage or the local storage. The data from the addressed location is read out and into the R register. If in the 1401 mode, it gates the LT register address to the MN register. If one read follows another read, the second read is ignored (no read call), but the T or LT address is gated to the N or MN register. (See example 2C, page 19.)

CM = 0110 (hh[1])
This decode sends a read call to memory and specifies an address for the MN register. It turns on the allow write latch and gates the HEX address of one of the local storage coordinates (determined by the CK field and CN field next address) to the MN register. The decode addresses that location of the local storage. The data from the addressed location is read out and into the R register. If one read follows another read, the second read is ignored (no read call), but the HEX address is gated to the MN register. (See example 2D, page 20.)

CM = 0111 (GUV → MN)
This decode sends a read call to memory and specifies an address for the MN register. This decode turns on the allow write latch and gates the GUV (selector channel) register address to the MN register. It also addresses that location of either the main storage or the local storage. The data from the addressed location is read out and into the R register. If one read follows another read, the second read is ignored (no read call), but the GUV address is gated to the MN register. (See example 2C, page 19.)

CU Field — This field specifies the area of memory to be addressed (MS, LS and MPX), depending on the value of the CM field.

CU = 0000 (MS)
If the CM decode is any combination of conditions 3 through 7, it specifies that the machine is addressing the main storage. The R register is assumed to be the destination for the data from memory. (See example 3A, page 21.)

CU = 0001 (LS)
If the CM decode is any combination of conditions 3 through 7, it specifies that the machine is addressing local storage (CPU Bump). The R register is assumed to be the destination for the data from memory. (See example 3B, page 21.)

CU = 0010 (MPX)
If the CM decode is any combination of conditions 3 through 7, it specifies that the machine is addressing MPX (UCW Bump). The R register is assumed to be the destination for the data from memory. (See example 3C, page 22.)

CU = 0011 (M/LS)
This decode is a function of the macroinstruction format and specifies the LS (CPU Bump) if the G register 0 and 1 bits are off (0). If either the G register 0 and/or 1 bits are on (1), main storage is selected. The R register is assumed to be the destination for data from memory. (See example 3D, page 22.)

Alternate CU Field — Activated by CM = 0000, 0001 or 0010.

CU = 0001 (Use GR)
This decode specifies that the selector channel data register (GR) will be the destination for data from memory. (See example 4A, page 22.)

CU = 0010 (K → W)
This decode is used for changing modules. It uses the CK field to specify the value. It does not change the W3 bit. (See example 4B, page 23.)

CU = 0011 (FWX → WX)
This decode gates the FWX register (backup ROSAR) into the WX register (ROSAR). It restores the link address of a microprogram routine disrupted by an I/O trap. (ROSAR—Read Only Storage Address Register.) (See example 4C, page 23.)

CA Field — This field names the desired input to the A register.

CA = 0000 (FT)
This decode specifies the MPX tags-in-buss (FT) as the input to the A register. The A register parity check is blocked. (See example 5, page 24.)

CA = 0001 (TT)
This decode specifies the 1050 tags-in-buss (TT) as the input to the A register. The A register parity check is blocked. (See example 5, page 24.)

CA = 0100 (S)
This decode specifies the S register as the input to the A register. The A register parity check is blocked. (See example 5, page 24.)

CA = 0101 (H)
This decode specifies the H register as the input to the A register. The A register parity check is blocked. (See example 5, page 24.)

CA = 0110 (FI)
This decode specifies the MPX buss-in-line (FI) as the input to the A register. (See example 5, page 24.)

CA = 0111 (R)
This decode specifies the R register as the input to the A register. (See example 5, page 24.)

CA = 1000 (D)
This decode specifies the D register as the input to the A register. (See example 5, page 24.)

CA = 1001 (L)
This decode specifies the L register as the input to the A register. (See example 5, page 24.)

CA = 1010 (G)
This decode specifies the G register as the input to the A register. (See example 5, page 24.)

CA = 1011 (T)
This decode specifies the T register as the input to the A register. (See example 5, page 24.)

CA = 1100 (V)
This decode specifies the V register as the input to the A register. (See example 5, page 24.)

CA = 1101 (U)
This decode specifies the U register as the input to the A register. (See example 5, page 24.)

CA = 1110 (J)
This decode specifies the J register as the input to the A register. (See example 5, page 24.)

CA = 1111 (I)
This decode specifies the I register as the input to the A register. (See example 5, page 24.)

Alternate CA Field — Activated by AA = 1

CA = 0000 (F)
This decode specifies the external interrupt register (F), in the complement form, as the input to the A register. (See example 5, page 24.)

CA = 0001 (SFG)
This decode specifies the console switches F and G as the input to the A register. (See example 5, page 24.)

CA = 0010 (MC)
This decode specifies the machine check register (MC) as the input to the A register. The A register parity check is blocked. (See example 5, page 24.)

CA = 0100 (C)
This decode specifies the interval timer register (C) as the input to the A register. The A register parity check is blocked. (See example 5, page 24.)

CA = 0101 (Q)
This decode specifies the memory protect register (Q) as the input to the A register. (See example 5, page 24.)

CA = 0110 (JI)
This decode specifies the direct data channel buss-in-line (JI), in the complement form, as the input to the A register. (See example 5, page 24.)

CA = 0111 (TI)
This decode specifies the 1505 buss-in-line (TI) as the input to the A register. (See example 5, page 24.)

CA = 1100 (GR)
This decode specifies the selector channel data register (GR) as the input to the A register. (See example 5, page 24.)

CA = 1101 (GS)
This decode specifies a selector channel buss by which internal values can be gated to the A register. (See example 5, page 24.)

CA = 1110 (GT)
This decode specifies a selector channel buss by which internal latches and tags-in-line values can be gated to the A register. (See example 5, page 24.)

CA = 1111 (GJ)
This decode specifies that internal selector channel buss (GJ) be used as the source for input to the A register. The GJ buss, depending on CK field values, may be made up of several sources from within the selector channel. (See example 5, page 24.)

CB Field — This field names the desired input to the B register.

CB = 0000 (R)
This decode specifies the R register as the input to the B register. (See example 6, page 24.)

CB = 0001 (L)
This decode specifies the L register as the input to the B register. (See example 6, page 24.)

CB = 0010 (D)
This decode specifies the D register as the input to the B register. (See example 6, page 24.)

CB = 0011 (K)
This specifies the CK field as the input to the B register. The CK value is gated into both the high and low halves of B with the parity being forced to 1. (See example 6, page 24.)

CK Field — This field allows the microprogrammer to use constants from the ROS. When used in a microprogram, this field is called K and its value is specified. K may be gated to B for entry into the adder, to MN for addressing a special area of any bump, to the I/O channels for control, and to the ROSAR for changing the high-order 4 bits (W) of the ROS address. (See example 7, page 24.)

Alternate CK Field — Activated by AK = 1

CK = 0001 (UV → WX)
This decode specifies that 12 bits from the UV register will be gated into the WX register.

CK = 0010 (Restore Wrap)
This is a 64K option. This condition allows the machine to store the value of the buffer wrap latch into the wrap latch. (See example 8, page 25.)

CK = 0011 (Wrap)
This is a 64K option. This condition allows the machine to test the wrap latch and determine whether or not the I or U registers were wrapped. If I was wrapped, it blocks the X6 branch. If U was wrapped, it blocks the X7 branch. (See example 8, page 25.)

CK = 0100 (SHI)
This decode specifies the console switch values of H and I as the sources for the K buss.

CK = 0101 (AC Force)
This decode specifies that if an address carry occurred in the previous cycle, it forces the X register to zeros.

CK = 0110 (0 → line)
This resets the 1050 to home loop.

CK = 0111 (1 → line)
This condition sets the 1050 to line loop.

CK = 1000 (1 → OE)
When this condition first occurs, it sets the odd/even latch to even. The second time, it introduces an ALU check and resets the odd/even latch.

CK = 1001 (ASCII)
This condition tests the ASCII latch, and if on, it forces the X6 bit to 0.

CK = 1010 (INST)
This condition tests for interrupts; meanwhile, it creates a four-way branch. If a MPX interrupt is indicated, the X6 and X7 bits remain the same.
If the interrupt is a SEL. 1, the X7 bit is forced to 0. If the interrupt is a SEL. 2, the X6 bit is forced to 0. If there is a TIMER or EXT. interrupt, the X6 and X7 bits are both forced to 0.

CK = 1011 (0 → MC)
This condition resets the machine check register.

CK = 1100 (Store Wrap)
This is a 64K option. It stores the value of the wrap latch into the buffer wrap latch. (See example 8, page 25.)

CK = 1101 (0 → IPL)
This resets the load request, ALU check, odd/even, and SX diagnostic latches.

CK = 1110 (0 → F)
This resets the external interrupt (F) 0 bit. And, if the L register 1 bit is on, the F register 1 bit is reset. The same relationship exists for L2 through 7, and F2 through 7.

CK = 1111 (1 → FO)
This sets the external interrupt register (F) 0 to bit to a 1.

PK Field — When CK is used at a position other than B, it must carry a parity bit. When K goes to W, PK is odd parity. When K goes to the MN register, PK is odd or even parity according to whether CNO is 0 or 1. When K goes to I/O, PK indicates control and not parity. When CA is gated to W, a parity bit is needed and PK is odd parity on the 5 bits consisting of 4 CA bits and the AA bit.

PS Field — This bit is called SAL parity and is odd parity on PA, CH, CL, CM, CU, CA, CB, CK, and PK.

CD Field — Specifies the destination for the ALU output.

CD = 0000 (Z)
This specifies no destination, and allows an ALU function where an output zero test is desired without changing the data held in the registers. (See example 9A, page 25.)

CD = 0001 (TE)
This specifies the 1050 buss out (TE) as the destination for the ALU output.

CD = 0010 (JE)
This specifies the direct data channel buss out (JE) as the destination for the D register output (complement), and not the normal ALU output.

CD = 0011 (Q)
This specifies the memory protect register (Q) as the destination for the ALU output.

CD = 0100 (TA)
This specifies the 1050 tags out (TA) as the destination for the ALU output.

CD = 0101 (H)
This specifies the hold register (H) as the destination for the ALU output.

CD = 0110 (S)
This specifies the S register as the destination for the ALU output.

CD = 0111 (R)
This specifies the R register as the destination for the ALU output. The * denotes the time in a read-compute-write sequence that R may be designated as a destination. (See example 9B, pages 25 and 26.)

Read — *Compute — Write

*Read — Store
Read — Write

Read — *Compute — *Compute — Write

Read — *Compute — *Compute — Read (Dummy)
*Compute — Write

CD = 1000 (D)
This specifies the D register as the destination for the ALU output.

CD = 1001 (L)
This specifies the L register as the destination for the ALU output.

CD = 1010 (G)
This specifies the OP register (G) as the destination for the ALU output.

CD = 1011 (T)
This specifies the T register as the destination for the ALU output.

CD = 1100 (V)
This specifies the V register as the destination for the ALU output.

CD = 1101 (U)
This specifies the U register as the destination for the ALU output.

CD = 1110 (J)
This specifies the J register as the destination for the ALU output.

CD = 1111 (I)
This specifies the I register as the destination for the ALU output.

CF Field — This field controls the ALU—A register entry for High-Low 4 bit gating and straight crossed switching.

CF = 0000 (0)
This specifies the ALU—A register entry will be blocked and forced to zeros. (See example 10A, page 26.)

CF = 0001 (L)
This specifies the ALU—A register entry Low 4 bits will be gated and the High 4 bits will be blocked and forced to zeros. (See example 10A, page 26.)

CF = 0010 (H)
This specifies the ALU—A register entry High 4 bits will be gated and the Low 4 bits will be blocked and forced to zeros. (See example 10A, page 26.)

CF = 0011 (Straight)
This specifies the ALU—A register entry High and Low 4 bits will be gated. (See example 10A, page 26.)

CF = 0100 (Stop)
This specifies a machine stop function. The machine stops at the end of the cycle prior to the microword containing this instruction. The WX register contains the address of this microword.

CF = 0101 (XL)
This specifies the ALU—A register entry will cross the High 4 bits into the Low 4 bits and the Low 4 bits into the High 4 bits. Then it gates the Low 4 bits to the ALU and blocks the High 4 bits, forcing them to zeros. (See example 10B, page 26.)

CF = 0110 (XH)
This specifies the ALU—A register entry will cross the High 4 bits into the Low 4 bits and the Low 4 bits into the High 4 bits. It then gates the High 4 bits to the ALU and blocks the Low 4 bits, forcing them to zeros. (See example 10B, page 26.)

CF = 0111 (X)
This specifies the ALU—A register entry will cross the High 4 bits into the Low 4 bits and the Low 4 bits into the High 4 bits. Then both the High and Low 4 bits are gated to the ALU. (See example 10B, page 26.)

CG Field — This field controls the ALU—B entry High—Low 4 bit gating.

CG = 0000 (0)
This specifies the ALU—B register entry will be blocked and forced to zeros. (See example 11, page 27.)

CG = 0001 (L)
This specifies the ALU—B register entry Low 4 bits will be gated, and the High 4 bits will be blocked and forced to zeros. (See example 11, page 27.)

CG = 0010 (H)
This specifies the ALU—B register entry High 4 bits will be gated, and the Low 4 bits will be blocked and forced to zeros. (See example 11, page 27.)

CV Field — This field controls the ALU true-complement and binary-decimal functions.

CV = 0000 (+)
This specifies a true binary add.

CV = 0001 (−)
This specifies a complement binary add. The B register data is complemented at the ALU—B register entry.

CV = 0010 (± 2)
This specifies a binary add under true-complement control. This is dependent on the SO bit which is the true-complement latch.
SO = 0—True add.
SO = 1—Complement add.

CV = 0011 (± 3)
This specifies a decimal add under true-complement control. This is dependent on the SO bit which is the true-complement latch.
SO = 0—True add.
SO = 1—Complement add.

CC Field — This field controls the carry inputs and outputs of the ALU and the logic functions (AND, OR, XOR).

CC = 0000 (0)
This decode specifies that the carry-input line is to be 0 and ignores the carry out of the ALU.

CC = 0001 (1)
This decode specifies that the carry-input line is to be 1, and ignores the carry out of the ALU. (See example 12A, page 27.)

CC = 0010 (.)
This decode specifies the AND function and ignores the carry out of the ALU. The AND function requires a coincidence of bits to obtain an output. (See example 12B, page 27.)

CC = 0011 (Ω)
This decode specifies the OR function and ignores the carry out of the ALU. The OR function requires a bit on either side to obtain an output. (See example 12C, page 28.)

CC = 0100 (O C)
This decode specifies that the carry-input line is to be 0 and sets the S3 bit to 1 if a carry results. The S3 bit is the carry latch. (See example 12D, page 28.)

CC = 0101 (I C)
This decode specifies that the carry-input line is to be 1 and sets the S3 bit to 1 if a carry results. The S3 bit is the carry latch.

CC = 0110 (C C)
This decode specifies the value of the carry latch onto the carry-input line and sets the S3 bit to 1 if a carry results. The S3 bit is the carry latch. (See example 12E, page 28.)

CC = 0111 (∀)
This decode specifies the XOR (Exclusive OR) function and ignores the carry out of the ALU. The XOR function requires no coincidence of bits to obtain an output. (See example 12F, page 28.)

CS Field — This field controls the individual sets and resets of status in the S register. It also controls some I/O lines. CS has an alternate decoder activated by the bit AS = 1.

CS = 0001 (LZ→S5)
This specifies that if the Low 4 bits of the Z buss are zeros, the S register

5 bit will be set on (1). If the Low 4 bits of the Z buss are not zeroes, the S register 5 bit will be reset off (0). (See example 13A, page 29.)

CS = 0010 (HZ → S4)
This specifies that if the High 4 bits of the Z buss are zeros, the S register 4 bit will be set on (1). If the High 4 bits of the Z buss are not zeros, the S register 4 bit will be reset off (0). (See example 13A, page 29.)

CS = 0011 (LZ → S5, HZ → S4)
This specifies that if the High-Low 4 bits of the Z buss are zeros or any combination thereof, the corresponding S register 4 and 5 bits will be set on (1). If the High-Low 4 bits of the Z buss are not zeroes or any combination thereof, the corresponding S register 4 and 5 bits will be reset off (0). (See example 13A, page 29.)

CS = 0100 (0 → S4, S5)
This specifies that the S register 4 and 5 bits will be reset off (0).

CS = 0101 (Treq → S1)
This specifies that if a 1050 request occurs, the S register 1 bit will be set on (1).

CS = 0110 (0 → SO)
This specifies that the S register 0 bit (true-complement latch) will be reset off (0).

CS = 0111 (1 → SO)
This specifies that the S register 0 bit (true-complement latch) will be set on (1).

CS = 1000 (0 → S2)
This specifies that the S register 2 bit will be reset off (0).

CS = 1001 (ANSNZ → S2)
This specifies that if the Z buss (results of an arithmetic statement) is non-zero, the S register 2 bit will be set on (1). If the Z buss is zero, the S register 2 bit would not be reset. (See example 13A, page 29.)

NOTE

A special function of the above decode takes place if the suppress-machine trap latch is on (1), the CPU check switch is in diagnostic mode, and the Z buss is not zero (0); the machine is forced to a hard stop.

CS = 1010 (0 → S6)
This specifies that the S register 6 bit will be reset off (0).

CS = 1011 (1 → S6)
This specifies that the S register 6 bit will be set on (1).

CS = 1100 (0 → S7)
This specifies that the S register 7 bit will be reset off (0).

CS = 1101 (1 → S7)
This specifies that the S register 7 bit will be set on (1).

CS = 1110 (K → FB)
This specifies the controls for numerous MPX channel conditions.

K = 1100 P1—Sets the MPX channel interrupt latch on.

K = 0110 P1—Sets the MPX operation latch on.

K = 1010 P1—Sets the suppress-out latch on.
K = 0101 P1—Sets the operational-out control latch on.
K = 0011 P1—The set or reset depends on the R register mask bits.

With the instruction K = 0011 P1:

a. If the R register 0 bit is on (1), the MPX mask latch will be set on.

b. If the R register 1 bit is on (1), the Selector Channel 1 mask will be set on.

c. If the R register 2 bit is on (1), the Selector Channel 2 mask will be set on.

d. If the R register 7 bit is on (1), external trap mask will be set on.

K = 1001 P1—The set or reset depends on the S register 0, 1 and 2 bits. (See example 13B, page 29.)

a. With the above instruction, if the S register 0 bit is on (1), it sets the XX high latch on.

b. If the S register 1 bit is on (1), it sets the X high latch on.

c. If the S register 2 bit is on (1), it sets the X low latch on. The XL, XH, and XXH latches force the M register 1, 2, and 3 bits, which in turn address a specific bump. The latches and M register bits may appear in combinations.

CS = 1111 (K → FA)
This specifies the controls for MPX channel tag lines and conditions.
K = 0000 PI—Sets the command-start latch on.
K = 1000 PO—Sets the buss-out register from the R register.
K = 0100 PO—Sets the address-out-line on.
K = 0010 PO—Sets the command-out-line on.
K = 0001 PO—Sets the service-out-line on.

NOTE

The FA register will frequently appear as a combination of these.

Example: K → FA
K = 1100 P1 which sets the buss-out CTRL, address-out, and command-start latch on.

Alternate CS Field — This field is activated by AS = 1. It controls the selector channel hardware.

CS = 0110 (GUV → GCD)
This specifies that the selector channel data address register (GUV) is gated to the selector channel count register (GCD).

CS = 0111 (GR → GK)
This specifies that the GR register is gated to the selector channel protect key register (GK).

CS = 1000 (GR → GF)
This specifies that the GR register is gated to the selector channel flag register (GF).

CS = 1001 (GR → GG)
This specifies that the GR register is gated to the selector channel command register (GG).

CS = 1010 (GR → GU)
This specifies that the GR register is gated to the selector channel data address register (GU).

CS = 1011 (GR → GV)
This specifies that the GR register is gated to the selector channel data address register (GV).

CS = 1100 (K → GH)
This specifies that the CK field values are decoded with gated K to GH to determine specific selector channel functions.

* K = 0—Selector channel 1 and 2 machine reset.

* K = 1—Diagnostic and tag controls are set.

* K = 2—Tag control is reset.

K = 7—Chain detect is set.

*—Hardware is added for the R and S diagnostic functions.

CS = 1101 (GI → GR)
This specifies that the selector channel buss-in-line (GI) is gated to the GR register.

CS = 1110 (K → GB)
This specifies that the CK field values are decoded with gated K to GB to determine specific selector channel functions.

K = 0—Program Check
K = 1 and KP = 0—Selector channel 1
and KP = 1—Selector channel 2
K = 2—Operational-out reset
K = 3—PCI flag is reset
K = 4—Selector channel interrupt is set
K = 5—Channel control check
K = 6—GR to zero is set
K = 7—CPU stored
K = 8 and KP = 0—Count ready is reset
and KP = 1—Count ready is set
K = 9 and KP = 0—Channel reset
and KP = 1—Poll control reset and channel reset
K = 10 and KP = 0—Suppress-out is reset
and KP = 1—Suppress-out is set
K = 11 and KP = 0—Poll control is reset
and KP = 1—Poll control is set
K = 12—Select-out is reset
K = 13—Channel busy is set
K = 14—Halt I-O latch is set
K = 15—Interface control check

NOTE

KP is K field parity bit.

CS = 1111 (K → GA)

This specifies that the CK field values are decoded with gated K to GA to determine specific select channel functions.

K = 0001—Sets the service-out line on.

K = 0010—Sets the command-out line on.

K = 0100—Sets the address-out line on.

K = 1000—Sets the buss-out control line on.

NOTE

The CK values will frequently appear as a combination of these.

Example: K → GA

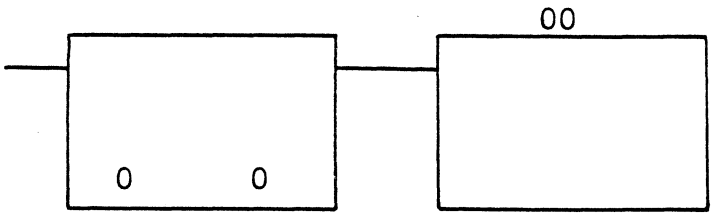
K = 1100 which sets the address-out line, the-control line, and the selection sequence latch on. This turns on the select-out line.

Examples of Microword Format

	← SYMBOLS ALLOWED →		← EXAMPLES →
EMIT	K = - - - - - (P -) BIN, DEC		K = 0111 P0 BIN
ARITHMETIC	CA, ALT. CA, CC, CD, CF, CG, SHJ, CV		RXL. DH → SC
STORAGE	CM, CU, USE GR		IJ → MN MS
MISCELLANEOUS CONTROLS	STORE WRAP, RESTORE WRAP, 0 → LINE, 1 → LINE, 1 → OE, 0 → MC, 0 → IPL, 0 → F, CS, ALT. CS		ANSNZ → S2
ROS ADDRESSING	WRAP, ASCII, INTST, UV → WX, CAnn, → W, K → W, FWX → WX		UV → WX
BRANCHING	CH	CL	R0 S3

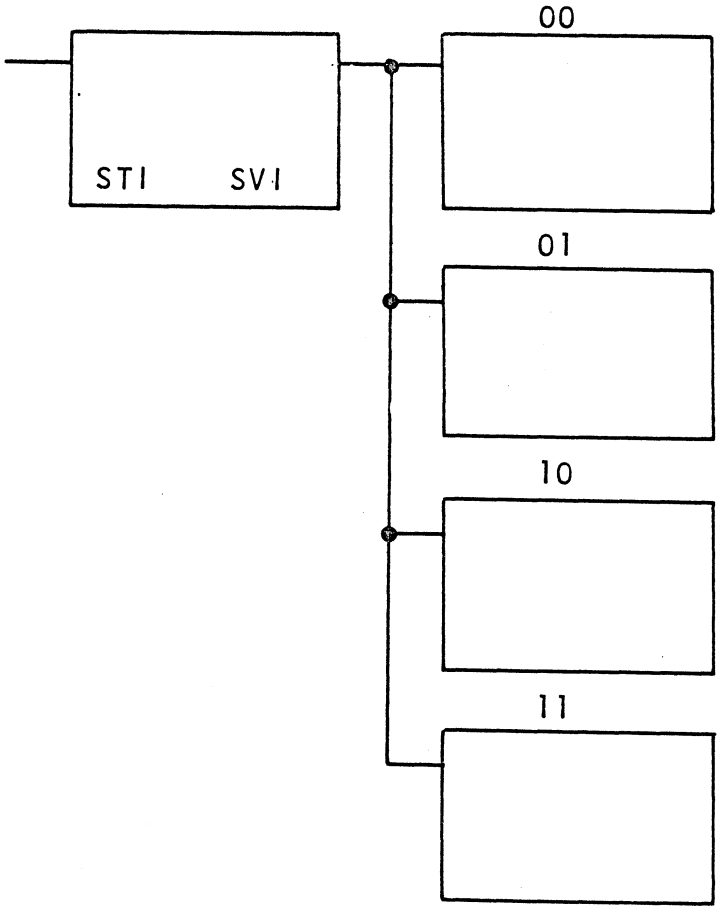
1 CH AND CL BRANCHING

A. CH and CL = 0000 or 0001



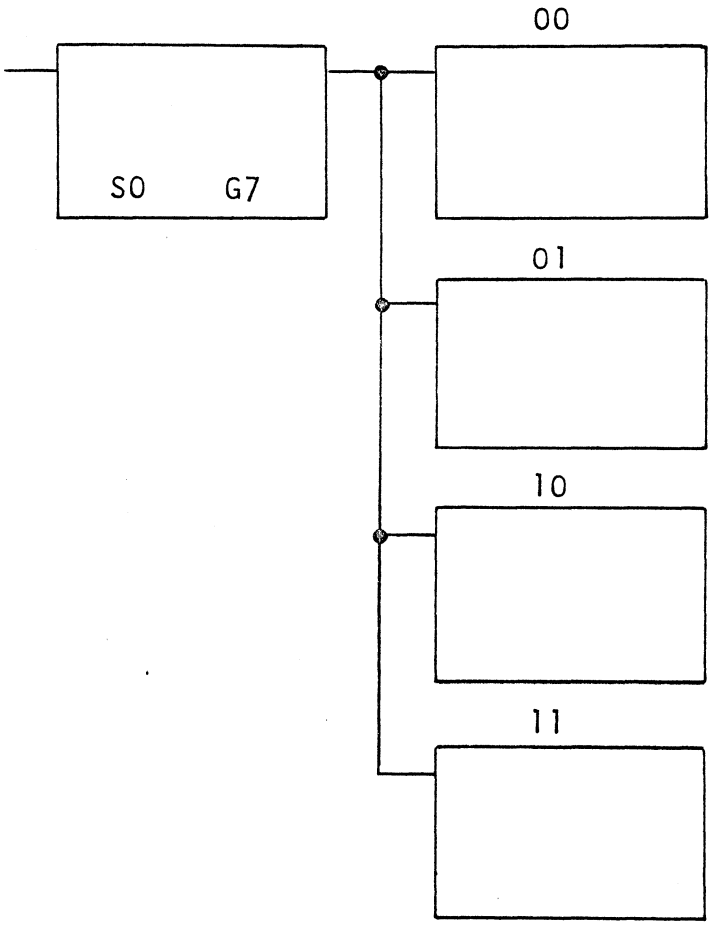
Signifies forced branching. If CH and CL are zeros, it forces a branch to the next microword identified by 00. If CH and CL are 01, it forces a branch to the next microword identified by 01.

B. CH = 0011 through 0110 and CL = 0011 through 0111 and 1111



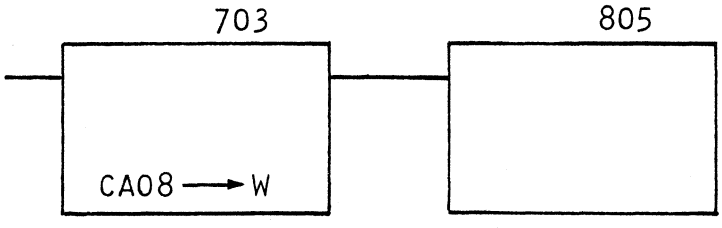
This signifies conditional branching. If the Status In Line (STI) is up and the Service In Line (SVI) is down, this condition forces a branch to the microword identified by 10.

C. CH = 0111 through 1111 and CL = 1000 through 1110



This signifies branching on latches in registers. If the S register 0 bit and the G register 7 bit are on, this condition forces a branch to the microword identified by 11.

D. CL = 0010



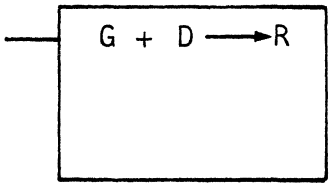
This signifies a module change. This condition indicates that the address of the next microword is in the 800 block.

2 CM FIELD—READ, COMPUTE, AND WRITE

A. CM = 0000 and 0010

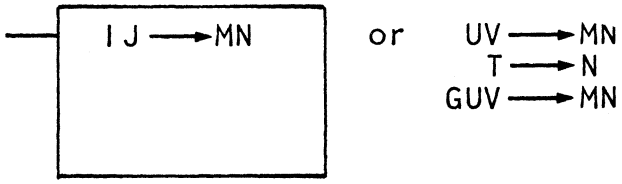
Either instruction sends a write call to memory if preceded by a read cycle.

B. CM = 0001



Signifies a compute cycle and does not ask for a read or write call.

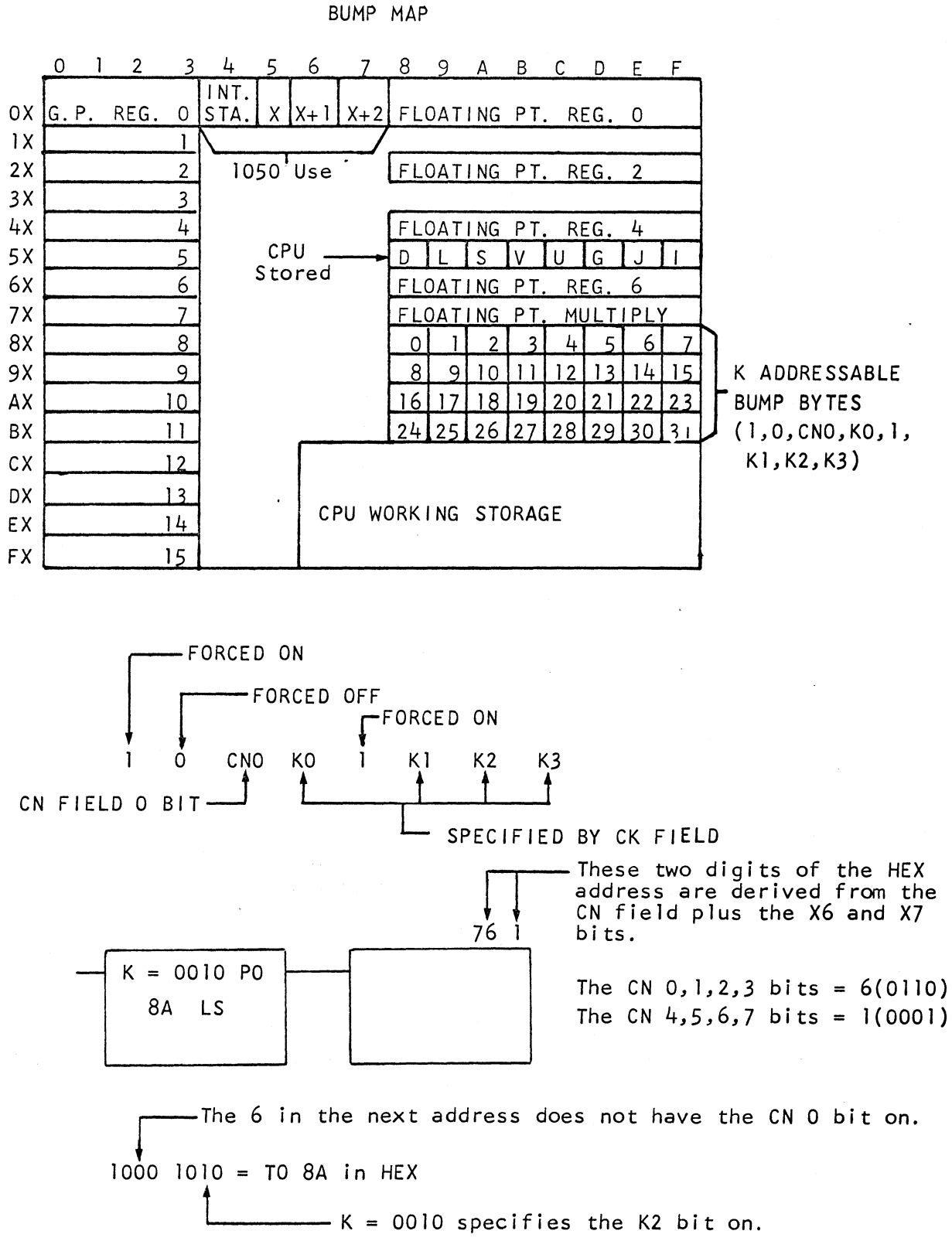
C. CM = 0011 through 0111



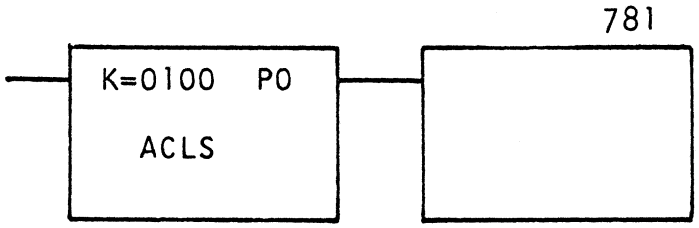
All of these instructions send a read call to memory and provide the memory address.

D. CM = 0110

This illustrates how the K addressable byte is formed.



CN field 0 bit = 0 indicates the K addressable bytes 0 through 15 that will be addressed.
CN field 0 bit = 1 indicates the K addressable bytes 16 through 31 that will be addressed.



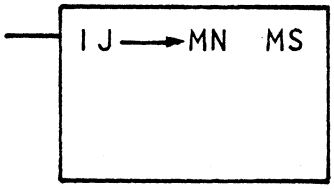
The 8 in the next address does have the CN 0 bit on.

10101100 = T0 AC in HEX

K = 0100 Specifies the K1 Bit on.

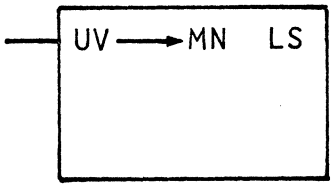
3 CU FIELD—AREA OF MEMORY TO BE ADDRESSED

A. CU = 0000



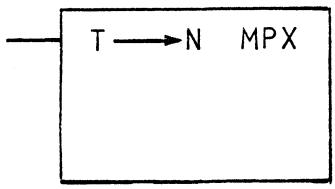
Signifies that main storage will be addressed.

B. CU = 0001



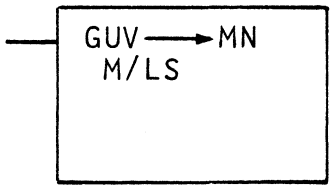
Signifies that local storage (CPU Bump) will be addressed.

C. CU = 0010



Signifies that multiplex storage (UCW Bump) will be addressed.

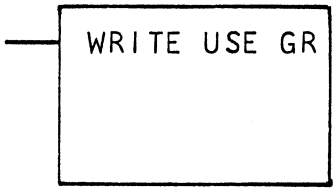
D. CU = 0011



Signifies that main storage will be addressed if the G register 0 and/or 1 bits are on (1), and that local storage (CPU Bump) will be addressed if the G register 0 and 1 bits are off (0).

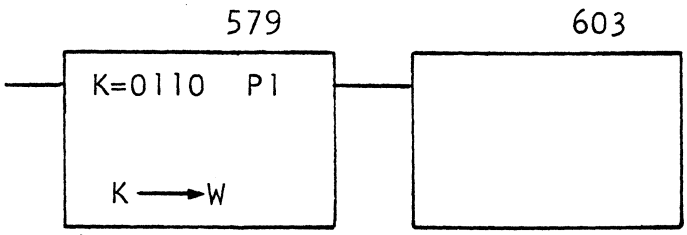
4 ALTERNATE CU FIELD

A. CU = 0001



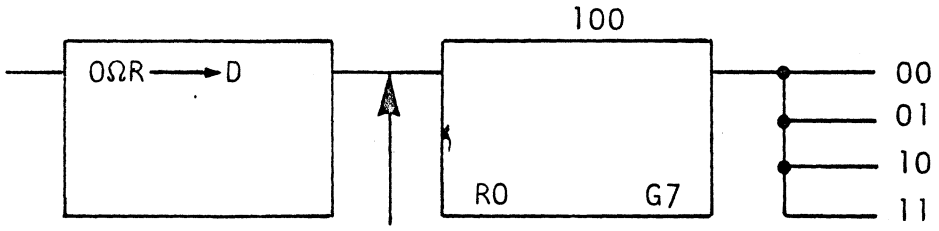
Signifies that the selector channel data register GR is the destination for data instead of the normal destination R register.

B. CU = 0010

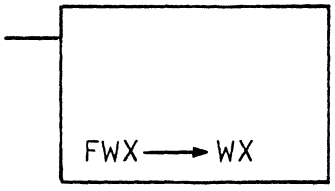


This signifies a module change (the K field value specifying that the address of the next microword is in the 600 block).

C. CU = 0011



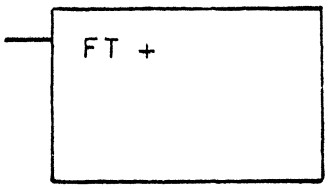
If a trap occurred at this point in a microprogram, the WX register which contains the address of the next microword (100) is set into the FWX register (backup ROSAR). The CH and CL fields are interrogated and set into the X6 and X7 buffer latches.



This instruction sets the FWX register (backup ROSAR) back into the WX register (100) and interrogates the X6 and X7 buffer latches to determine what the branching status was at the time of the trap, thus the instruction establishes the link address (100), which is the first word executed after the trap has been satisfied.

5 CA FIELD AND ALTERNATE CA FIELD—A REGISTER INPUTS

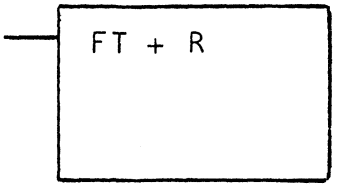
CA = 0000 through 1111



This signifies the source for the input data to the A register.

6 CB FIELD—B REGISTER INPUTS

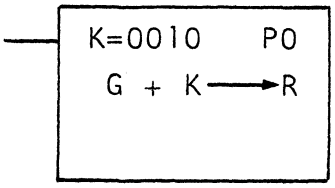
GB = 0000 through 0011



This signifies the source of the input data to the B register.

7 CK FIELD—CONSTANT GENERATOR

K = 0010



This signifies that a constant of 2 (0010) will be generated on the K buss, and is the source input data for the B register.

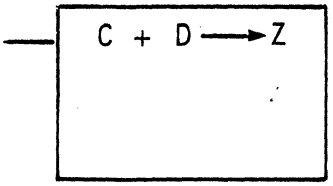
8 ALTERNATE CK FIELD

CK = 0001 through 0011

These are 64K options.

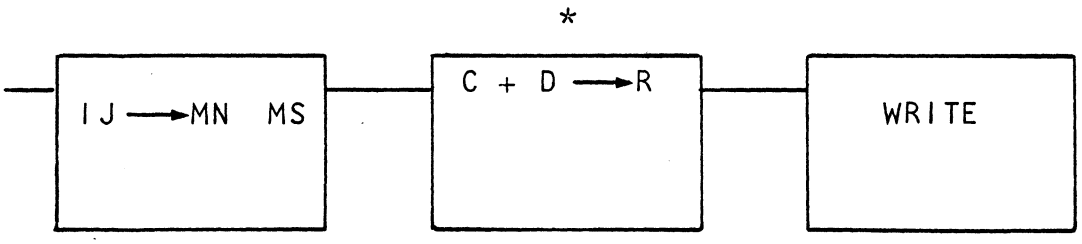
9 CD FIELD—DESTINATION FOR THE ALU OUTPUT

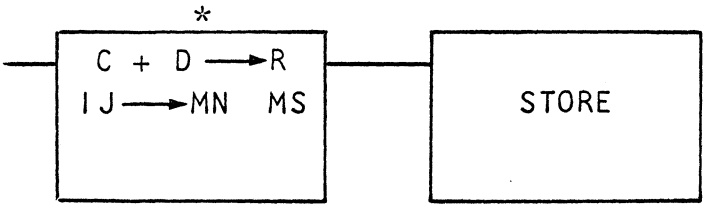
A. CD = 0000

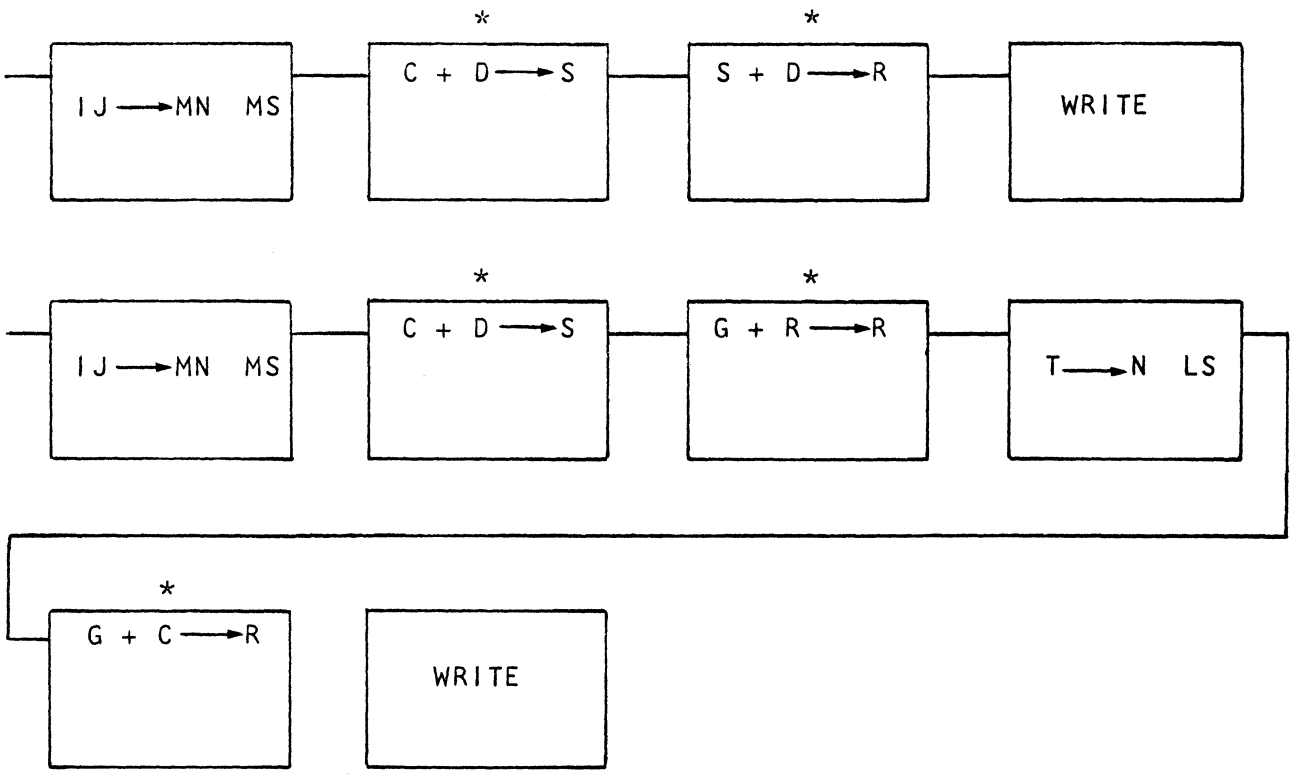


This signifies that the ALU output goes onto the Z buss but not into any register.

B. CD = 0111



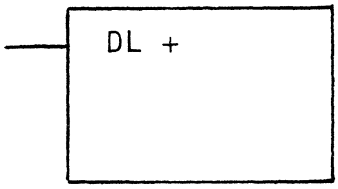




The * signifies the only time that the R register can be used as a destination.

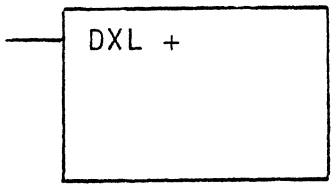
10 CF FIELD

A. CF = 0000 through 0011



This signifies that the ALU—A register entry can be blocked, just the Low 4 bits can be gated, just the High 4 bits can be gated, or all bits can be gated.

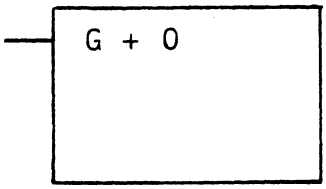
B. CF = 0101 through 0111



This signifies that the ALU—A register entry can be crossed, Low 4 bits to High 4 bits and vice versa, and just the low, or high, or all bits can be gated.

11 CG FIELD

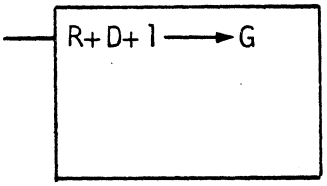
CG = 0000 through 0011



This signifies that the ALU—B register can be blocked, and that just the low, or high, or all bits can be gated.

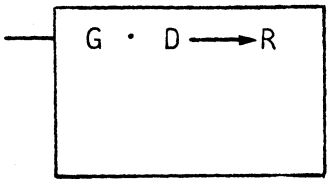
12 CC FIELD—CARRY INPUTS AND OUTPUTS

A. CC = 0001



The 1 in this statement forces the carry-input line to a 1.

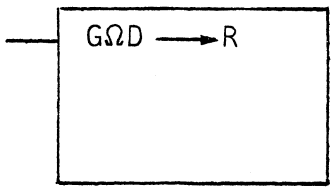
B. CC = 0010




```
G = 01001001
D = 01000001
-----------
R = 01000001   result
```

The AND function (.) requires a coincidence of bits to obtain an output.

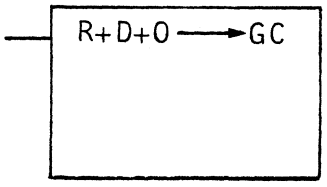
C. CC = 0011




```
G = 01001001
D = 01000001
--------------
R = 01001001   result
```

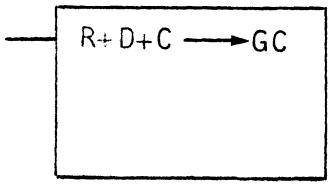
The OR function (Ω) requires a bit on either side to obtain an output.

D. CC = 0100



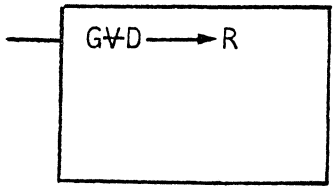
The 0 in this statement forces the carry-input line to 0, and the C indicates that if a carry out of the high-order position results, the S3 bit will be set to a 1.

E. CC = 0110



The first C in this statement forces the value of the carry latch (S3) onto the carry-input line. The second C indicates that if a carry out of the high order position results, the S3 bit will be set to a 1.

F. CC = 0111

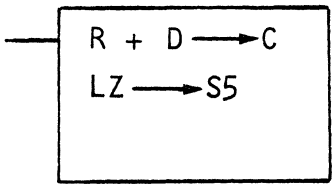



```
G = 01001001
D = 01000001
--------------
R = 00001000   result
```

The Exclusive OR function (⊻) requires no coincidence of bits to obtain an output.

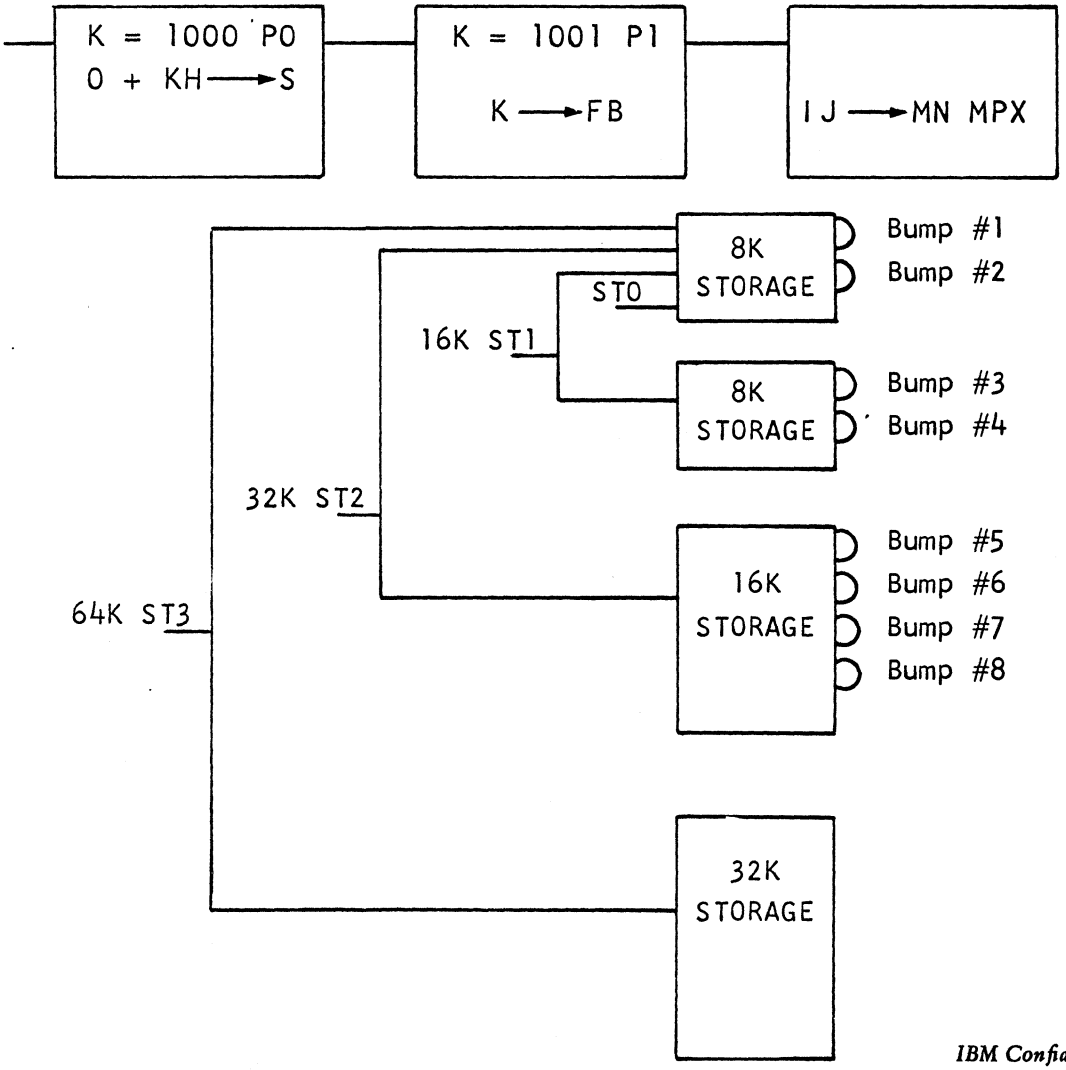
13 CS FIELD—STATUS

A. CS = 0001, 0010, 0011, and 1001



Must be tested in the same cycle as the arithmetic statement that generates the condition.

B. CS = 1110



8 K

IJ → MN MPX does not force any M register bits on. It also addresses Bump #1.

IJ → MN LS forces the M register 3 bit on, and addresses Bump #2.

16 K

IJ → MN MPX does not force any M register bits on. It also addresses Bump #1.

XL latch on—IJ → MN MPX forces the M register 3 bit on, and addresses Bump #2.

SH latch on—IJ → MN MPX forces the M register 2 bit on, and addresses Bump #3.

IJ → MN LS forces the M register 2 and 3 bits on, and addresses Bump #4.

32 or 64 K

IJ → MN MPX does not force any M register bits on. It also addresses Bump #1.

XL latch on—IJ → MN MPX forces the M register 3 bit on, and addresses Bump #2.

XH latch on—IJ → MN MPX forces the M register 2 bit on, and addresses Bump #3.

XL, XH latches on—IJ → MN MPX forces the M register 2 and 3 bits on, and addresses Bump #4.

XXH latch on—IJ → MN MPX forces the M register 1 bit on, and addresses Bump #5.

XL, XXH latches on—IJ → MN MPX forces the M register 1 and 3 bits on, and addresses Bump #6.

XH, XXH latches on—IJ → MN MPX forces the M register 7 and 2 bits on, and addresses Bump #7.

IJ → MN LS forces the M register 1, 2 and 3 bits on, and addresses Bump #8.

APPENDIX

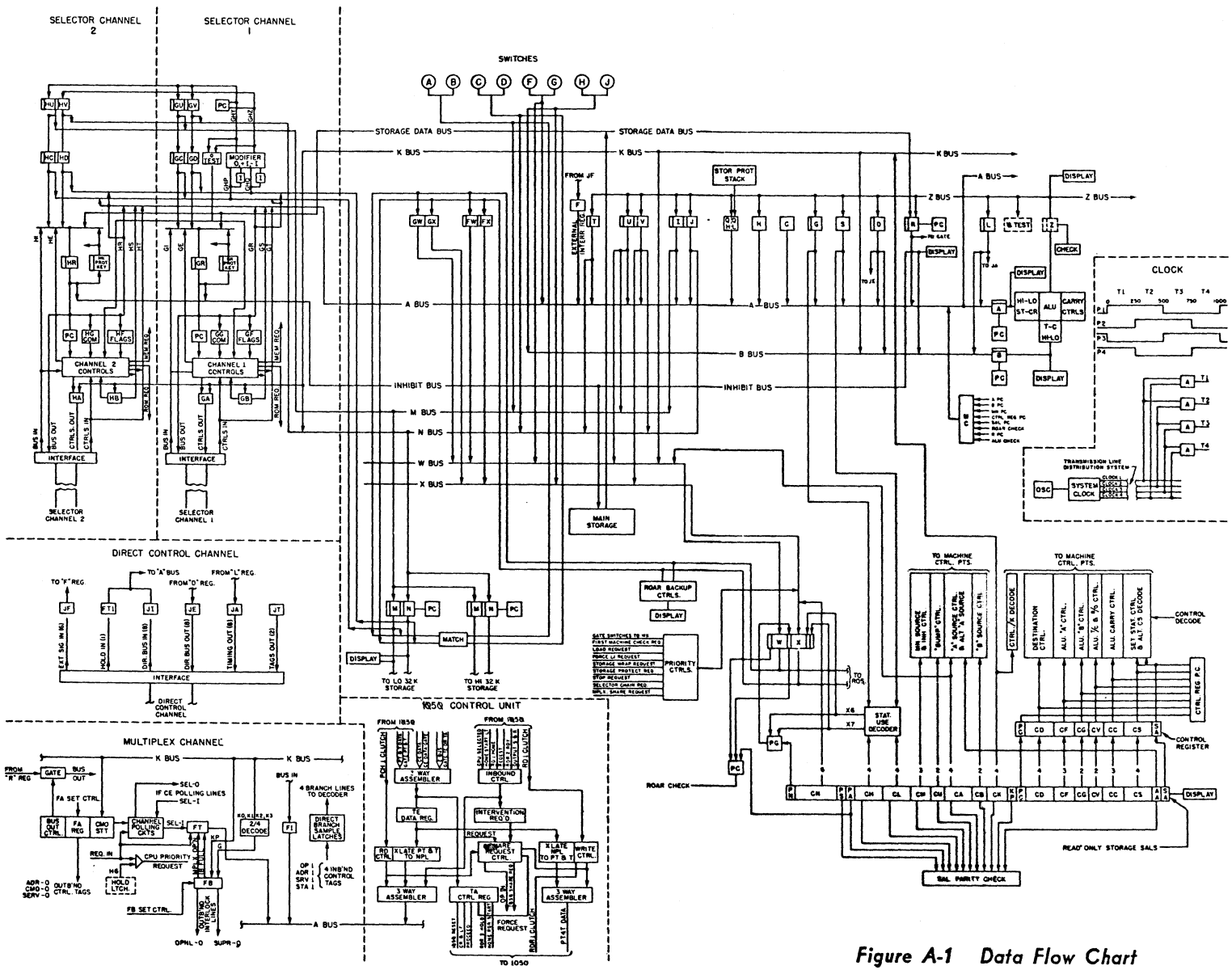


Figure A-1 Data Flow Chart

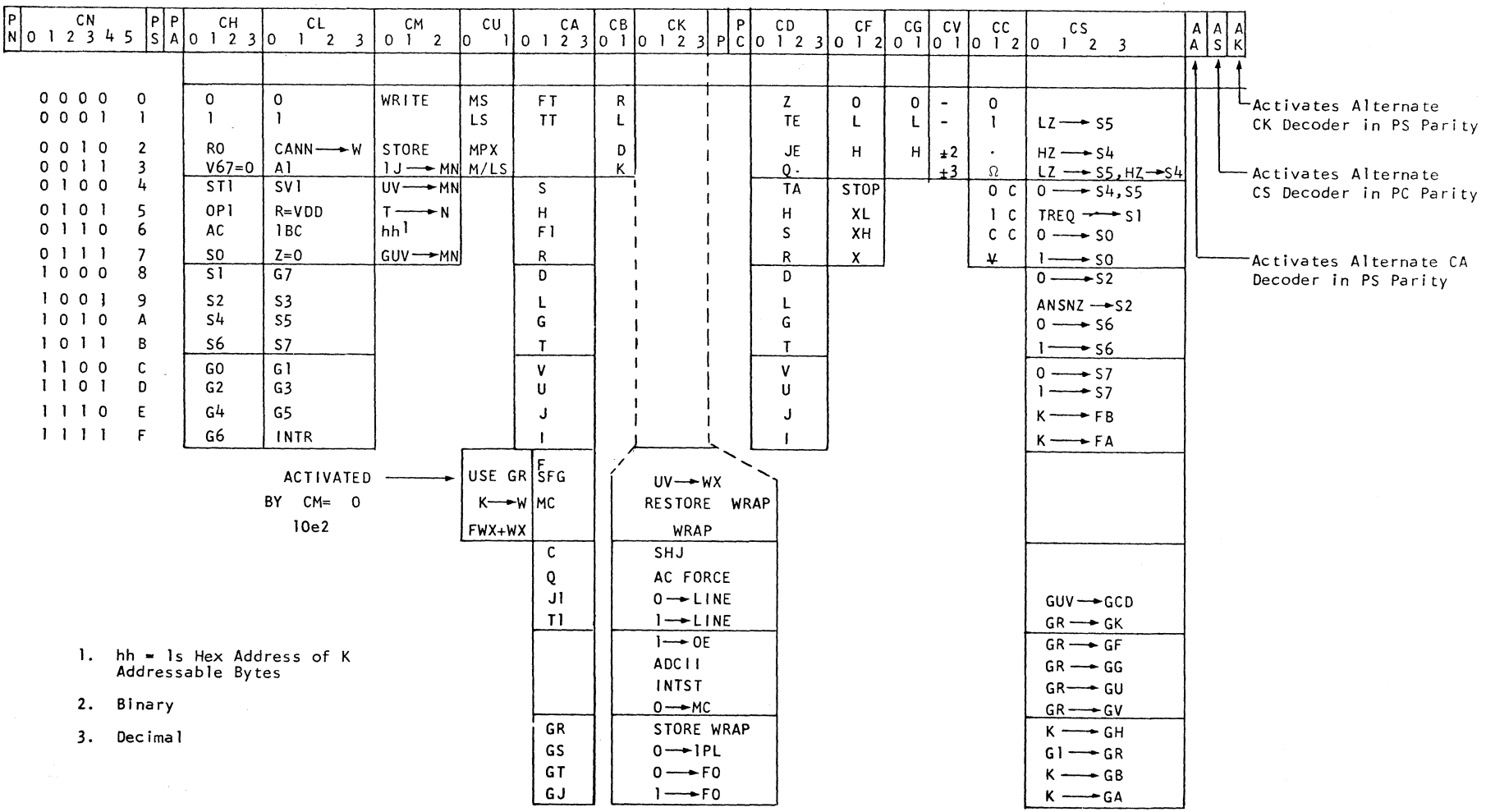


Figure A-2 Microword Format

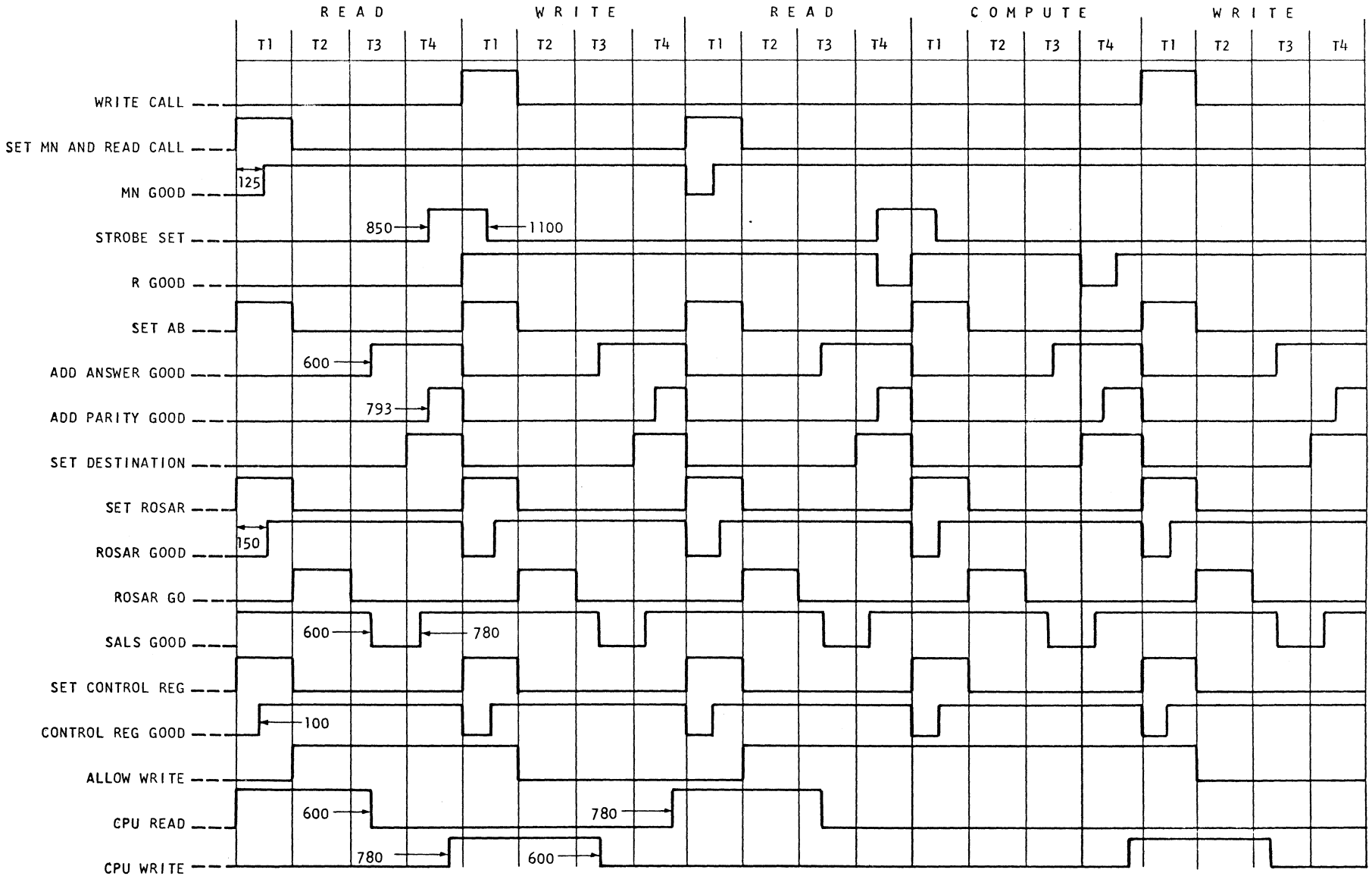


Figure A-3 Timing Chart

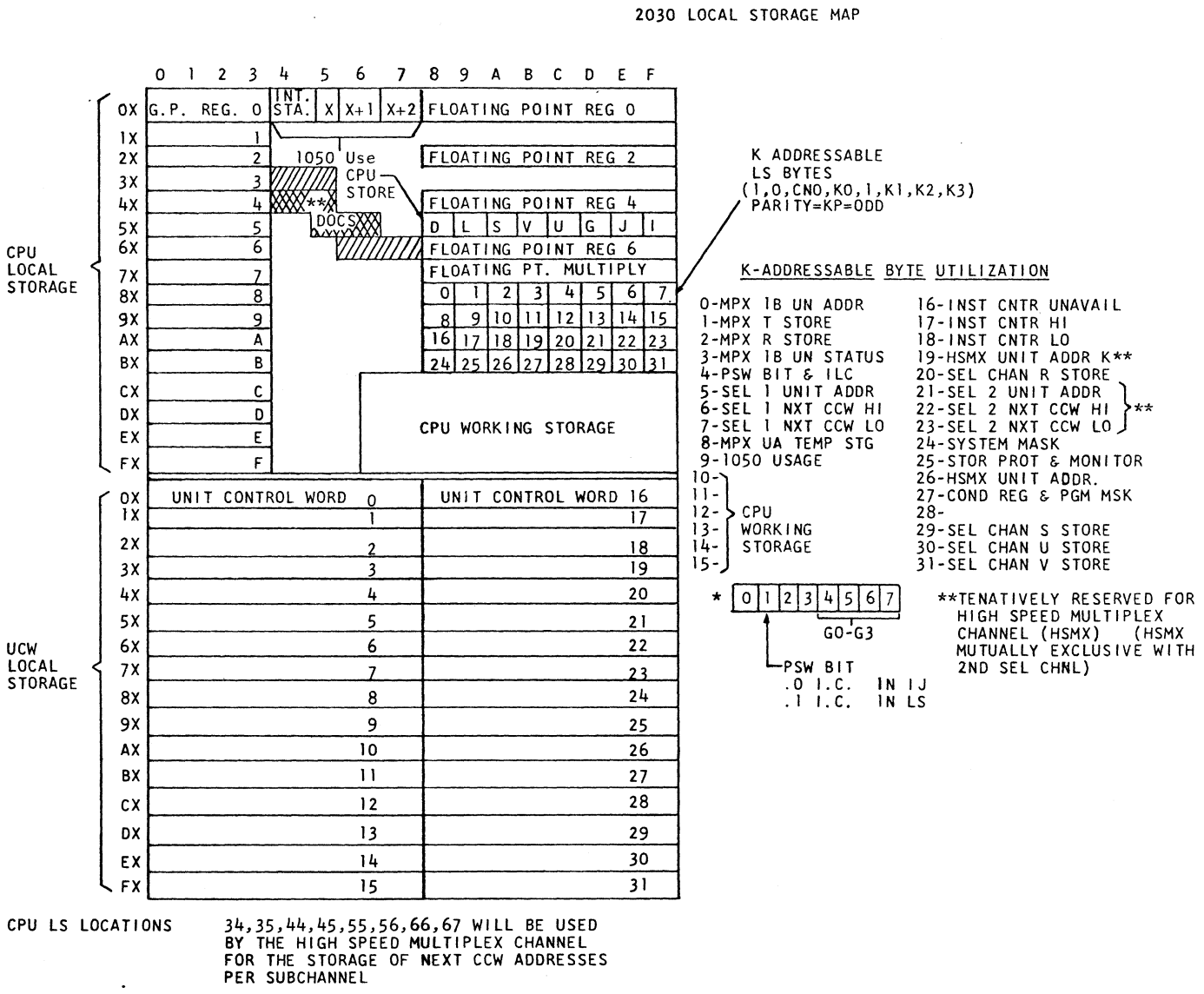


Figure A-4 2030 Local Storage Map

List of Abbreviations and Symbols

AC	Address Carry (ALU Carry)
AI	Address In (I/O Address)
ALT	Alternate
ALU	Arithmetic and Logic Unit
ANSNZ	Answer Nonzero
AP	Address Parity (ROS Address)
ASCII	American Standard Code for Information Interchange
C	Counter for Interval Timer
CAnn ---→ W	Gate CA Field to W Register
CCROS	Card Capacitor Read Only Storage
CP	Control Register Parity
F	External Interrupt Register
FA	Multiplex Channel Tags Out
FB	Multiplex Channel Tags Out
FI	Multiplex Channel Tags In
FT	Multiplex Channel Tags In
FWX	Multiplex Channel Backup Rosar
GA	Selector Channel Tags Out
GB	Selector Channel Tags Out
GEOG	Geographic Address in CCROS
GMWM	Group Mark—Word Mark
GR	Selector Channel Data Register
GS	Selector Channel Tags In
GT	Selector Channel Tags In
GUV	Selector Channel Address Register
H	Holding Register (Priority)
HEX	Hex-a-Decimal
HZ	High Z Buss Zero
INTR	Interrupt
JE	Direct Data Channel Buss Out
JI	Direct Data Channel Buss In
K	Constant Generator
LOAD	Reset
LS	Local Storage
LZ	Low Z Buss Zero
MC	Machine Check
M/LS	Main Storage or Local Storage
MPX	Multiplex
MS	Main Storage
NP	Next Address Parity
OPI	Operational in (I/O)

P	Priority Register
Q	Memory Protect Register
ROS	Read Only Storage
ROSAR	Read Only Storage Address Register
RVDD	R Register is a Valid Decimal Digit
SAB	Switches A and B (Console)
SAL	Sense Amplifier Latch
SCD	Switches C and D (Console)
SFG	Switches F and G (Console)
SP	Sense Amplifier Latches Parity
STI	Status In (I/O)
STOP	Stop
STORE	Write, But Don't Set R Register (Clear)
SVI	Service In (I/O)
TA	1050 Tags Out
TE	1050 Buss Out
TI	1050 Buss In
TREQ	1050 Request
TT	1050 Tags In
V67 = 0	V Register 6 and 7 Bits Equal Zero
WRITE	Write and Set R Register
Z	Z Buss
1BC	1 Bit Carry
± 2	Binary Add Under True/Complement Control
± 3	Decimal Add Under True/Complement Control
.	AND
Ω	OR
¥	Exclusive OR