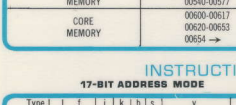
## MEMORY ADDRESS ASSIGNMENT

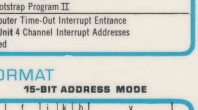
| OCTAL ADDRESS RANGE | | USE |
|---|---|---|
| CORE MEMORY | 00000 | Fault Entrance |
| | 00001-00017 | Unassigned |
| | 00020-00037 | External Interrupt Entrance |
| | 00040-00057 | Input Monitor Interrupt Entrance |
| | 00060-00077 | Output Monitor Interrupt Entrance |
| CONTROL MEMORY | 00100-00117 | Input Buffer Control Registers |
| | 00120-00137 | Output Buffer Control Registers |
| | 00140-00157 | External Function Buffer Control Registers |
| | 00160 | Real-Time Clock |
| | 00161-00167 | $B_1$ through $B_7$ Index Registers |
| | 00170-00177 | Unassigned Film Locations |
| | 00200-00217 | ESI Input Buffer Terminate or CDM Reload |
| | 00220-00237 | ESI Output Buffer Terminate or CDM Reload |
| | 00240-00257 | ESI External Function Buffer Terminate |
| | 00260-00277 | Unassigned |
| | 00300-00477 | Unassigned |
| CORE MEMORY | 00500-00517 | External Function Monitor Interrupt Entrance |
| | 00520-00537 | External Interrupt Code Storage |
| WIRED MEMORY | 00540-00577 | NDRO Bootstrap Program I |
| | 00540-00577 | NDRO Bootstrap Program II |
| CORE MEMORY | 00600-00617 | Intercomputer Time-Out Interrupt Entrance |
| | 00620-00653 | Memory Unit 4 Channel Interrupt Addresses |
| | 00654 → | Unassigned |

## INSTRUCTION WORD FORMAT

**17-BIT ADDRESS MODE**

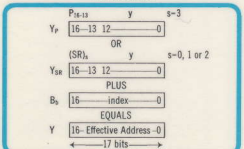



**15-BIT ADDRESS MODE**

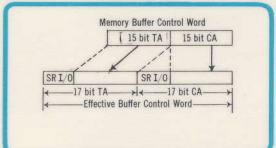



## EXPANDED MEMORY ADDRESSING

**OPERANDS**




**INPUT/OUTPUT**




# UNIVAC 1230 COMPUTER
## REPERTOIRE OF INSTRUCTIONS

| Code (Octal) f | INSTRUCTION | DESCRIPTION | Time μsec. |
|---|---|---|---|
| 01 | Right SHift-Q | Shift (Q) Right by Y | 2 |
| 02 | Right SHift-A | Shift (A) Right by Y | 2 |
| 03 | Right SHift-AQ | Shift (AQ) Right by Y | 2 |
| *04 | COMpare-A, -Q, -AQ | Compare Y with A, Q or A & Q; $A_i - A_f$ | 2 |
| 05 | Left SHift-Q | Shift (Q) Left by Y | 2 |
| 06 | Left SHift-A | Shift (A) Left by Y | 2 |
| 07 | Left SHift-AQ | Shift (AQ) Left by Y | 2 |
| 10 | ENTer-Q | Y→Q | 2 |
| 10 | CLear-Q | 0→Q | 2 |
| 11 | ENTer-A | Y→A | 2 |
| 11 | CLear-A | 0→A | 2 |
| 12 | ENTer-$B^1$ | Y→$B^1$ | 4** |
| 12 | CLear-$B^1$ | 0→$B^1$ | 4** |
| 12_ | NO-OPeration | 0→$B^1$ (do nothing operation) | 4** |
| ^13k0 | EXternal-COMmand-C'-W(Y)-MONITOR | (Y)→C' | 4 |
| ^13k1 | EXcom-COMmand-C'-W(Y)-MONFORCE | (Y)→C' (use on CP-642A equipment) | 4*** |
| ^13k2 | EXternal-COMmand-C'-W(Y) | (Y)→C' | 4 |
| ^13k3 | EXternal-COMmand-C'-W(Y)-FORCE | (Y)→C' (use on CP-642A equipment) | 4*** |
| 14 | SToRe-Q | (Q)→Y | 2 |
| 14k0 | ComPlement-Q | Q'→Q | 2 |
| 15 | SToRe-A | (A)→Y | 2 |
| 16 | SToRe-$B^1$ | ($B^1$)→Y | 2 |
| ^17k0 | JumP-Y-C'-COMACTIVE | Jump to Y if external buffer active | 4 |
| ^17k1 | JumP-L(Y)-C'-COMACTIVE | Jump to (Y)L if external function active | 4 |
| 17k2 | SToRe-C'-W(Y)-FORCE | Force C'→(Y)–(abnormal test mode) | 4 |
| ^17k3 | SToRe-C'-W(Y) | (00520+I)→(Y) | 4 |
| 20 | ADD-A | (A)+Y→A | 2 |
| 21 | SUBtract-A | (A)–Y→A | 2 |
| 22 | MULtiply | (Q)Y→AQ | 8** |
| *23 | DIVide | (AQ)/Y→Q; R→$A_f$ | 14** |
| *23k7 | SQuare RooT | $\sqrt{Y}$→Q; remainder→A | 8** |
| 24 | RePLace-A+Y | (A)+(Y)→Y&A | 4 |
| 25 | RePLace-A–Y | (A)–(Y)→Y&A | 4 |
| *26 | ADD-Q | (Q)+Y→Q | 2 |
| *27 | SUBtract-Q | (Q)–Y→Q | 2 |
| 30 | ENTer-Y+Q | Y+Q→A | 2 |
| 31 | ENTer-Y–Q | Y–Q→A | 2 |
| 32 | SToRe-A+Q | (A)+(Q)→Y&A | 4 |
| 33 | SToRe-A–Q | (A)–(Q)→Y&A | 4 |
| 34 | RePLace-Y+Q | (Y)+(Q)→Y&A | 4 |
| 35 | RePLace-Y–Q | (Y)–(Q)→Y&A | 4 |
| 36 | RePLace-Y+1 | (Y)+1→Y&A | 4 |
| 37 | RePLace-Y–1 | (Y)–1→Y&A | 4 |
| *40 | ENTer-LP | L[Y(Q)]→A | 2 |
| 41 | ADD-LP | (A)+L[Y(Q)]→A | 2 |
| 42 | SUBtract-LP | (A)–L[Y(Q)]→A | 2 |
| 43 | COMpare-MASK | (A)–L[Y(Q)]sense (j); $A - L[\underline{Y}(Q)]$; $(A)_i - (A)_f$ | 2 |
| *44 | RePLace-LP | L(Y)(Q)→Y&A | 4 |
| 45 | RePLace-A+LP | (A)+L(Y)(Q)→Y&A | 4 |
| 46 | RePLace-A–LP | (A)–L(Y)(Q)→Y&A | 4 |
| 47 | SToRe-LP | L(A)(Q)→Y; $(A)_i - (A)_f$ | 2 |
| 50 | SELective-SET | Set $(A)_j$ for $Y_{0-1}$ | 2 |

j, k | Special j and k designators
Y — The operand; Y or (Y)
**Execution time is constant
***Program held until transfer is completed.

# UNIVAC 1230 COMPUTER
## REPERTOIRE OF INSTRUCTIONS

| Code (Octal) f | INSTRUCTION | DESCRIPTION | Time μsec. |
|---|---|---|---|
| 51 | **SELective-ComPlement** | Complement $(A)_n$ for $\underline{Y}_{n-1}$ | 2 |
| 51k4 | **ComPlement-A** | If Y is 77777, A'→A | 2 |
| 52 | **SELective-CLear** | Clear $(A)_n$ for $\underline{Y}_{n-1}$ | 2 |
| 53 | **SELective-SUbstitute** | $\underline{Y}_{n}$→$(A)_n$ for $(Q)_{n-1}$ | 2 |
| 54 | **Replace SElective-SET** | [Set $(A)_n$ for $(Y)_{n-1}$]→Y&A | 4 |
| 55 | **Replace SElective-CP** | [Complement $(A)_n$ for $(Y)_{n-1}$]→Y&A | 4 |
| 56 | **Replace SElective-CL** | [Clear $(A)_n$ for $(Y)_{n-1}$]→Y&A | 4 |
| 57 | **Replace SElective-SU** | [$(Y)_n$→$(A)_n$ for $(Q)_{n-1}$]→Y&A | 4 |
| *60 | **JumP** (arithmetic) | Jump to Y if j-condition is satisfied | 4** |
| 60j0 | **Remove Interrupt Lockout** | Enable all interrupts not locked out by SIL-EX | 2** |
| 60j1 | **Remove Interrupt Lockout JumP-Y** | Enable interrupts and jump to Y | 4** |
| *61 | **JumP** (manual) | Jump to Y if j-condition is satisfied | 4** |
| ^62 | **JumP on-C^-ACTIVE INput** buffer | Jump to Y if $C^j$ input buffer active | 4** |
| ^63 | **JumP on-C^-ACTIVE OUTput** buffer | Jump to Y if $C^j$ output buffer active | 4** |
| *64 | **Return JumP** (arithmetic) | Jump to Y+1 and (P)+1→$Y_L$ if j-condition is satisfied (see JP and RJP j-designators) | 6** |
| *65 | **Return JumP** (manual) | | 6** |
| ^66_ | **TERMinate-C^-INPUT** | Termtnate input buffer on $C^j$ | 2 |
| ^66k1 | **Remove Interrupt Lockout-ALL** | Enable'all interrupts not locked out by SIL-EX | 2 |
| ^66k2 | **Remove Interrupt Lockout-EXternal-ALL** | Enable external interrupts; all channels | 2 |
| ^66k3 | **Remove Interrupt Lockout-EXternal-C^** | Enable external interrupts on $C^j$ | 2 |
| ^66k1b1 | **Set Interrupt Lockout-ALL** | Lockout all interrupts on all channels | 2 |
| ^66k2b1 | **Set Interrupt Lockout-EXternal-ALL** | Lockout external interrupts; all channels | 2 |
| ^66k3b1 | **Set Interrupt Lockout-EXternal-C^** | Lockout external interrupt on $C^j$ | 2 |
| ^67 | **TERMinate-C^-OUTPUT** | Terminate output buffer on $C^j$ | 2 |
| ^67k1 | **TERMinate-C^-COMmand** | Terminate external function buffer on $C^j$ | 2 |
| ^67k2 | **TERMinate-ALL** | Terminate ALL buffers | 2 |
| *70 | **RePeaT** | Execute NI Y times | 4** |
| 71 | **BSKip-B^** | (B)←Y, skip NI and clear (B)†; (B)≠Y, advance B^ and read NI | 4** |
| 72 | **BJumP-B^** | (B)=0, read NI; (B)≠ 0, (B)-1→B^ & jump to Y | 4** |
| ^73 | **INputC^** (without monitor mode) | Buffer in on $C^j$; Y→00100+j | 4** |
| ^74 | **OUTputC^** (without monitor mode) | Buffer out on $C^j$; Y→00120+j | 4** |
| ^74k2 | **EXternal-COMmand-MultiWord-C^-W(Y)** | Buffer commands out on $C^j$; (Y)→00140+j | 4** |
| ^75 | **INputC^** (with **MONITOR** mode) | Buffer in on $C^j$ with monitor; Y→00100+j | 4** |
| ^76 | **OUTputC^** (with **MONITOR** mode) | Buffer out on $C^j$ with monitor; Y→00120+j | 4** |
| ^76k2 | **EX-COM-MultiWord-C^-W(Y)-MONITOR** | Buffer commands out on $C^j$ with monitor; (Y)→00140+j | 4** |
| 7707 | **NORMalize-AQ** | Shift AQ Left until $A_{35} \neq A_{34}$; Shift Count→Y*** | 4 |
| 7760 | **ENTer SR^**; (n-0, 1, 2 or 3) | $\underline{Y}_{3-0}$→$SR_j$ | 2 |
| 7761 | **ENTer SR-C^-Y-INPUT** | $\underline{Y}_{1-0}$→Input SR-Channel j | 2 |
| 7762 | **ENTer SR-C^-Y-OUTPUT** | $\underline{Y}_{1-0}$→Output SR-Channel j | 2 |
| 7763 | **ENTer SR-C^-Y-EF** | $\underline{Y}_{1-0}$→External Function SR-Channel j | 2 |
| 7764 | **Enable CDM-C^-INPUT** | Enable Input Channel j CDM | 2 |
| 7765 | **Enable CDM-C^-OUTPUT** | Enable Output Channel j CDM | 2 |
| 7766 | **Disable CDM-C^-INPUT** | Disable Input Channel j CDM | 2 |
| 7767 | **Disable CDM-C^-OUTPUT** | Disable Output Channel j CDM | 2 |
| 7770 | **STORe SR^**; (n-0, 1, 2 or 3) | $(SR)_j$→$Y_{3-0}$ | 2 |
| 7771 | **STORe SR-C^-Y-INPUT** | $(SR)_j$→$Y_{3-0}$; for Channel j Input | 2 |
| 7772 | **STORe SR-C^-Y-OUTPUT** | $(SR)_j$→$Y_{3-0}$; for Channel j Output | 2 |
| 7773 | **STORe SR-C^-Y-EF** | $(SR)_j$→$Y_{3-0}$; for Channel j External function | 2 |
| 7774 | Disable EXPanded memory code | Disable 17-bit addressing mode | 2 |
| 7775 | Enable EXPanded memory mode | Enable 17-bit addressing mode | 2 |

$\underline{j}$ Special j and k designators
$\underline{Y}$—The operand; Y or (Y)
**Execution time is constant.
***Y-13 bits subject to s & b indicators.

# UNIVAC 1230 COMPUTER
## REPERTOIRE OF INSTRUCTIONS

### NORMAL j-DESIGNATORS

| j | (Not applicable on * or ^) Skip Code |
|---|---|
| 0 | (no skip) |
| 1 | **SKIP** |
| 2 | **Q POS** |
| 3 | **Q NEG** |
| 4 | **A ZERO** |
| 5 | **A NOT** Zero |
| 6 | **A POS** |
| 7 | **A NEG** |

### NORMAL k-DESIGNATORS

| k | READ Code | READ Origin | STORE Code | STORE Dest. | REPLACE Code | REPLACE Origin | REPLACE Dest. |
|---|---|---|---|---|---|---|---|
| 0 | 'blank' | $U_L$ | **Q** | Q | 'not used' | — | — |
| 1 | **L** | $M_L$ | **L** | $M_L$ | **L** | $M_L$ | $M_L$ |
| 2 | **U** | $M_U$ | **U** | $M_U$ | **U** | $M_U$ | $M_U$ |
| 3 | **W** | M | **W** | M | **W** | M | M |
| 4 | **X** | $XU_L$ | **A** | A | 'not used' | — | — |
| 5 | **LX** | $XM_L$ | **CPL** | Cpl $M_L$ | **LX** | $XM_L$ | $M_L$ |
| 6 | **UX** | $XM_U$ | **CPU** | Cpl $M_U$ | **UX** | $XM_U$ | $M_U$ |
| 7 | **A** | A | **CPW** | Cpl M | 'not used' | — | — |

### LEGEND

M—Memory word (30 bits)
$M_L$—Lower half memory word
$M_U$—Upper half memory word
X—Sign bit extended
Cpl—Complement
A—A-register
Q—Q-register
$U_L$—U-register, lower half

### SPECIAL †j-DESIGNATORS

| j | SKIP CONDITIONS: COM-A, -Q, -AQ f04 | DIV f23 | ADD-Q, SUB-Q f26 f27 | ENT-LP, RPL-LP f40 f44 | SQRT f23 k7 | ADDRESS MODIFICATIONS: RPT f70 |
|---|---|---|---|---|---|---|
| 0 | (no skip) | (no skip) | (no skip) | (no skip) | (no skip) | (no mod.) : Y of NE-Y |
| 1 | (unconditional skip) | **SKIP** | **SKIP** | **SKIP** | **SKIP** | **ADV** : Y of NE-Y+1 |
| 2 | **Y LESS** : $Y_{\leq}(Q)$ | **NO Over Flow** | **A POS** | **EVEN** parity | **REM** | **BACK** : Y of NE-Y-1 |
| 3 | **Y MORE** : $Y_{>}(Q)$ | **Over Flow** | **A NEG** | **ODD** parity | **NO REM** | **ADD B** : Y of NE-Y+$B^a$ |
| 4 | **Y IN** :$(Q)_S$ Y and $Y_{>}(A)$ | **A ZERO** | **Q ZERO** | **A ZERO** | 'not used' | **Rpl. Inc.** : Y of NE-Y[+$B^a$] ✓ |
| 5 | **Y OUT** :$(Q)_S$ Y or $Y_{\leq}(A)$ | **A NOT** Zero | **Q NOT** Zero | **A NOT** Zero | 'not used' | **ADV R** : Y of NE-Y+1[+$B^a$] ✓ |
| 6 | **Y LESS** : $Y_{\leq}(A)$ | **A POS** | **Q POS** | **A POS** | 'not used' | **BACK R** : Y of NE-Y-1[+$B^a$] ✓ |
| 7 | **Y MORE** : $Y_{>}(A)$ | **A NEG** | **Q NEG** | **A NEG** | 'not used' | **ADD BR** : Y of NE-Y+$B^a$[+$B^a$] ✓ |

✓ If NI is RPL class, $B^a$ increments Y address for the store portion of the replace.
NE—Next execution.

### †-DESIGNATORS & ^-DESIGNATORS

† is a 4-bit Input/Output Channel Designator or Special Register Designator.

^ is a 1 or 2-bit Operand Interpretation Designator.

Store Class Type II instructions
^-0: Store in $Q_{2-0}$
^-1: Store in $Y_{7-0}$; $(Yu)_{7-0}$ $(Yu)_L$

Read Class Type II instructions
^-0: $(U_{7-0})$→Register
^-1: $(Y_{7-0})$→Register

"s" is interpreted: in instructions 01-12, 14-16 and 20-57 when k ≠ 0, 4 or 7
in instructions 13, 17, 64, 65
in instructions 73, 76 when ^ ≠ 0
in Type II instructions when ^ ≠ 0

### JP & RJP †& ^j-DESIGNATORS

| j | JP f60 | RJP f64 | JP f61 | RJP f65 |
|---|---|---|---|---|
| 0 | (No j-mp)# | | (Uncond. Jump) | |
| 1 | (Uncond. Jump)# | | **KEY 1** | |
| 2 | **Q POS** | | **KEY 2** | |
| 3 | **Q NEG** | | **KEY 3** | |
| 4 | **A ZERO** | | **STOP** | |
| 5 | **A NOT** Zero | | **STOP 5** | |
| 6 | **A POS** | | **STOP 6** | |
| 7 | **A NEG** | | **STOP 7** | |
| † | 62† | | 63† | |
| 0-17$_8$ | **C^ ACTIVE IN** | | **C^ ACTIVE OUT** | |

#60 Clears interrupt & bootstrap modes.