This document accompanies the program hp45term and is a listing of the ROM code which hp45term runs. Please see the ‘read me’ document included with the release of hp45term, and the ‘about page’ of the program itself, for credits to the work of others who made this possible.

The detailed processing of the ROM contents can be understood from the comments included in the source code of hp45term, but here are a few notes with which to start.

The HP-45 has 8 ROMs, each with 256 10-bit bytes. Each byte holds an operation, and parameters for that operation if parameters are required. Broadly speaking, the operations can be split into the following groups, which are colour-coded in the listing.

register:	there are 7 internal/general registers, A to F and M.
pointer:	there is a single pointer variable used to select data within a register
status:	there is a single status register in which bits are set to record things like the shift key’s having been pressed, etc.
rom:	rom functions pass program flow from one ROM to another
constant:	this places constants in the C register
flow:	goto or jump-sub a specific byte in the current ROM
other:	including data operations using 10 data/storage registers
no oper:	no operation (execution skips to next byte)

Each internal/general register (and data/storage register) consists of 14 4-bit nibbles, numbered from 13 down to 0. Depending on the register, the value of each nibble can represent a digit, or a + or – sign, or the position of the decimal point, or a blank space on the display, etc. Operators operate on subsets of the 14 nibbles in the register. These subsets [and their nibbles] are:

P	Pointer	[nibble indicated by pointer]
M	Mantissa	[nibbles 12, 11, ..., 3]
X	Exponent	[nibbles 2, 1, 0]
W	Word (entire register)	[nibbles 13, 12, ..., 0]
WP	Word to pointer nibble	[P, P-1, ..., 0]
MS	Mantissa and Sign	[13, 12, ..., 3]
XS	Exponent Sign	[2]
S	Mantissa Sign	[13]

Amazingly, then, the HP-45 manages to cram all its functionality into only 2048 bytes, less than a (still legible) page's worth:

173 784 297 814 998 314 890 910 746 905 905 746 805 403 532 327 541 942 398 11 656 541 942 528 144 212 99 296 206 624 0 760 942 633 793 649 793 766 760 446 183 894 181 83 144 942 532 971 254 969 84 179 660 423 235 144 398 144 942 398 506 283 426 283 206 12 344 780 458 943 660 471 942 227 202 12 344 842 634 819 1023 297 909 46 818 814 270 917 452 805 206 216 408 780 633 537 503 490 482 527 206 140 280 942 7 144 452 276 463 686 478 614 459 254 665 68 87 144 532 427 541 68 660 235 975 942 400 362 362 452 354 660 635 651 676 452 292 388 680 446 571 260 510 583 420 206 490 84 771 276 903 484 404 523 391 683 660 651 708 100 46 663 708 100 144 144 708 254 100 164 46 1018 1018 506 506 74 715 942 934 422 731 942 550 74 763 654 1002 14 763 735 144 1023 404 903 484 276 523 511 424 296 48 708 144 144 550 654 1002 823 750 396 48 724 867 686 452 627 144 780 847 724 999 708 468 815 740 48 46 818 654 652 910 28 300 923 48 206 482 660 223 423 84 507 942 206 354 624 206 752 503 212 963 660 1019 528 656 272 574 975 814 193 760 942 193 760 942 596 51 942 340 107 126 71 548 222 665 752 661 609 181 760 16 942 665 660 875 750 994 294 934 362 658 442 135 722 490 151 718 654 752 558 283 558 268 891 752 942 418 215 174 398 138 815 39 16 340 119 254 958 79 658 894 255 510 818 466 814 302 850 259 942 760 590 958 814 274 752 1022 1022 175 206 42 726 713 866 760 268 657 396 621 524 621 140 536 652 621 569 621 817 270 621 142 813 817 686 660 471 596 471 942 254 617 817 686 16 817 686 686 597 686 941 817 652 625 569 524 629 140 536 396 625 268 625 625 814 590 844 344 1011 396 536 408 344 152 280 600 84 875 48 750 994 16 272 270 662 558 647 510 782 643 910 272 272 330 272 482 846 675 974 270 28 594 44 679 215 482 790 715 918 278 28 44 719 215 28 918 879 16 378 378 746 862 638 795 272 518 811 254 814 782 272 206 716 472 536 344 216 600 536 88 408 216 344 16 48 16 906 891 354 510 44 751 938 746 98 923 718 590 554 202 780 699 272 658 658 382 947 466 786 562 142 894 955 958 760 942 30 11 270 946 752 658 382 784 830 1022 598 274 75 424 665 398 532 267 750 838 3 718 382 3 510 302 601 866 71 818 926 7 460 437 524 629 588 625 1017 652 625 501 716 625 893 625 741 625 985 942 334 26 191 334 814 28 270 108 195 942 446 227 230 490 716 789 596 27 340 595 985 669 595 985 945 741 716 621 893 652 621 501 588 621 1017 524 621 621 621 396 754 844 558 942 408 571 148 379 1002 634 779 790 359 918 270 362 371 718 210 938 446 435 814 782 238 718 558 206 358 148 475 280 486 487 408 108 471 590 590 148 595 48 460 216 216 24 536 344 24 600 939 601 994 302 382 539 722 942 278 942 894 547 814 994 817 144 722 894 599 766 910 48 144 718 558 643 910 382 639 942 278 942 439 204 458 350 687 190 806 750 812 791 102 731 84 3 656 879 0 562 934 144 548 408 600 216 88 280 472 88 923 998 403 910 354 787 718 60 876 791 490 766 780 46 610 859 270 362 622 831 206 298 910 638 799 934 398 46 780 491 588 216 88 24 88 472 600 536 24 344 344 216 887 942 302 390 698 379 506 718 490 971 415 206 780 152 216 24 152 344 519 332 507 528 0 299 359 363 0 275 272 528 784 411 419 808 511 424 296 507 255 1006 1006 1006 107 528 541 270 331 1006 1006 1006 48 528 208 548 528 131 204 48 528 151 324 401 267 307 159 528 0 389 291 0 16 1006 1006 1006 75 528 0 665 519 751 775 46 912 296 669 519 389 612 199 389 750 994 295 528 528 612 95 401 580 324 199 270 1006 270 168 938 168 459 0 0 548 580 389 132 31 541 323 196 676 784 784 68 391 580 423 612 132 541 469 397 596 455 985 507 752 468 255 511 12 610 1023 459 541 469 528 0 0 942 669 452 533 981 859 254 676 547 718 414 548 595 506 516 340 587 490 559 40 20 563 36 28 812 599 552 532 583 270 356 660 127 528 0 210 370 218 906 671 206 52 398 780 298 394 442 711 170 378 647 938 276 39 810 42 533 812 543 266 763 260 724 115 718 946 795 718 276 531 946 250 398 442 815 218 170 844 278 362 638 947 630 819 202 533 726 414 812 895 142 494 76 274 60 418 879 942 236 919 202 388 795 404 931 28 658 793 28 490 2 939 708 726 934 276 847 677 519 758 468 999 296 452 206 366 190 510 558 48 0 531 371 803 355 784 389 363 611 267 548 67 131 16 389 1023 389 55 523 523 523 656 0 419 727 0 523 523 523 0 0 443 385 656 835 459 455 0 463 324 377 263 303 159 591 675 389 263 0 0 1006 1006 451 656 0 431 389 983 1006 255 0 0 878 12 315 400 385 548 808 296 362 362 532 655 647 377 400 400 270 60 876 315 168 930 168 400 0 0 324 400 541 400 676 400 68 391 196 644 784 784 784 644 196 399 388 260 471 420 260 471 388 467 1006 784 784 420 292 148 511 676 400 676 656 0 559 0 400 389 424 487 676 535 644 612 400 424 296 942 48 596 579 385 942 487 405 752 575 377 424 196 676 311 385 808 296 665 275 0 0 656 208 228 663 228 144 144 254 16 377 196 292 446 699 260 808 545 614 215 210 482 213 100 541 398 657 752 813 401 657 760 669 890 890 890 791 378 965 398 261 967 389 132 23 206 624 0 760 48 596 847 639 385 87 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 16 0 0 0 656 0 0 0 0 0 0 0 0 0 48 784 784 482 482 482 482 482 334 624 0 760 276 1023 942 532 671 667 400 0 0 0 0 915 260 676 657 9 749 181 398 13 143 0 0 0 267 749 201 17 749 206 482 398 5 749 67 206 624 0 760 48 808 296 398 48 548 612 661 0 0 0 0 0 0 0 0 0 400 676 292 13 398 657 5 126 3 661 9 942 665 296 5 942 206 482 665 424 661 261 296 13 398 5 645 292 468 387 296 398 17 296 398 13 507 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 749 942 400 0 0 0 881 596 471 400 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 371 228 663 228 144 144 254 16 0 0 0 0 0 0 0 0 210 370 218 272 707 210 370 218 906 751 206 398 780 298 394 442 787 170 378 727 950 760 942 752 48 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 276 903 404 667 671 404 663 659 206 382 140 152 168 206 750 780 354 624 942 296 752 942 482 482 947 507 206 408 939 0 0 0 0 0 0 48 755 31 52 511 506 505 977 206 52 324 398 680 942 103 726 1002 59 844 826 314 28 172 183 750 874 46 516 76 314 390 278 610 71 548 908 394 122 59 28 172 171 95 874 155 75 890 83 290 28 754 394 918 235 726 1022 1002 532 239 60 726 54 890 634 263 95 1018 814 994 994 278 890 279 754 882 726 814 532 343 938 42 442 339 170 378 938 340 1019 515 0 0 0 0 0 0 0 0 0 0 0 0 0 723 659 398 206 354 624 942 398 1007 0 0 0 0 0 0 519 473 52 144 468 487 296 206 48 0 676 401 400 101 400 270 270 473 890 543 499 890 567 152 344 280 507 890 615 280 344 216 344 600 152 216 472 362 505 216 472 536 344 280 88 88 472 536 280 507 12 270 60 812 663 766 942 624 164 760 942 740 46 84 1007 407 302 942 206 624 142 942 752 403 126 31 122 31 106 787 28 815 28 236 803 19 362 769 272 114 31 938 716 426 855 362 106 19 270 942 750 994 174 811 942 590 510 780 310 974 895 846 270 974 911 1022 814 985 716 985 142 50 654 814 918 887 942 554 490 809 548 272 2 1003 658 1002 48 660 1019 528 400 0 23 651 23 23 23 55 23 71 23 671 23 23 23 659 23 600 591 663 195 127 280 503 23 536 591 675 491 499 88 503 23 344 591 159 579 839 24 503 23 39 23 183 23 23 23 191 23 223 408 591 63 95 472 503 23 1015 750 53 467 571 459 400 596 247 52 126 1019 362 362 442 1019 295 582 490 291 942 680 88 216 70 1019 408 652 66 1019 524 722 460 408 460 66 1019 168 332 722 722 722 722 814 882 746 524 866 716 994 994 894 814 36 165 165 20 563 667 644 659 10 547 42 31 0 0 216 591 152 591 400 20 7 532 435 552 548 780 206 208 810 874 810 663 516 135 660 459 676 624 603 624 660 623 942 398 582 752 667 760 780 418 643 206 942 278 181 189 189 189 189 221 680 552 40 660 511 76 1010 511 396 994 651 460 994 66 739 655 754 588 994 55 652 994 66 775 663 210 754 716 994 811 750 780 994 671 70 823 39 750 716 994 103 680 210 140 10 867 280 871 408 168 206 780 46 624 558 760 98 915 941 752 558 482 887 710 942 941 507 434 1015 942 270 524 274 396 274 274 274 780 1002 610 1011 874 262 991 942 48 206 301

I have changed only one byte from the original ROM listing, to fix a 'bug' which I believe was present in very early machines but soon corrected. Details are given below.

Each line of the listing has the following columns:

Line Number	
ROM Number	
Byte Number	
Label	if one exists
Contents in Binary (Contents in Decimal)	
[Operation Type]	
(Operation Subtype) or (Condition Text)	if applicable
Operation	
Destination Label	if flow operator
Note Indicator	'N' if there is a note
Original Patent Text	prefixed with "p:"

The patent text is from the original HP-45 patent.

Notes

L4146 (r4 b146)

This is the only difference between the ROM codes used in the program and the listing in the patent. Sending program flow to L4142 (label: am11) causes internal registers A and C to be swapped. This swap results in incorrect results for some combined STO operations: STO ÷ gives the reciprocal of the required result and STO – gives the negative of the required result. Changing this line so that program flow goes instead to L4143 misses out this swap and corrects this error: the functionality of the simulator then matches what is described in the documentation available.

L7173 (r7 b173) Increments 100ths of a second in timer mode.
L7176 (r7 b176) Increments seconds in timer mode.
L7185 (r7 b185) Increments minutes in timer mode.
L7195 (r7 b195) Increments hours in timer mode.

all other note indicators

The label is unclear in available scans of the HP-45 patent, but informed guesses have been made; the choice of label text does not affect program flow.

Sarah K. Libman
December 2021

```
L0000   r0 b000   pwo1       0010101101 (0173)     [flow    ] jumpSub : 43                        → pwo2
L0001   r0 b001   tms5       1100010000 (0784)     [rom     ] selectRom : 6
L0002   r0 b002   tms2       0100101001 (0297)     [flow    ] jumpSub : 74                        → frac
L0003   r0 b003              1100101110 (0814)     [register] using W : exchange A B
L0004   r0 b004              1111100110 (0998)     [register] using M : increment A
L0005   r0 b005              0100111010 (0314)     [register] using XS : A → B
L0006   r0 b006              1101111010 (0890)     [register] using XS : decrement A
L0007   r0 b007              1110001110 (0910)     [register] using W : A + B → A
L0008   r0 b008              1011101010 (0746)     [register] using X : clear A
L0009   r0 b009              1110001001 (0905)     [flow    ] jumpSub : 226                       → mlop
L0010   r0 b010              1110001001 (0905)     [flow    ] jumpSub : 226                       → mlop
L0011   r0 b011              1011101010 (0746)     [register] using X : clear A
L0012   r0 b012              1100100101 (0805)     [flow    ] jumpSub : 201                       → norm
L0013   r0 b013              0110010011 (0403)     [flow    ] goTo : 100                          → tms4
L0014   r0 b014   dmdtz4     1000010100 (0532)     [status  ] if statusBit ≠ 1 : bit 8
L0015   r0 b015              0101000111 (0327)     [flow    ]    (then) goTo : 81                 → dmt2
L0016   r0 b016   tdms       1000011101 (0541)     [flow    ] jumpSub : 135                       → mode
L0017   r0 b017              1110101110 (0942)     [register] using W : exchange A C
L0018   r0 b018              0110001110 (0398)     [register] using W : C → A
L0019   r0 b019              0000001011 (0011)     [flow    ] goTo : 2                            → tms2
L0020   r0 b020   pwo3       1010010000 (0656)     [rom     ] selectRom : 5
L0021   r0 b021   tpl4       1000011101 (0541)     [flow    ] jumpSub : 135                       → mode
L0022   r0 b022              1110101110 (0942)     [register] using W : exchange A C
L0023   r0 b023              1000010000 (0528)     [rom     ] selectRom : 4
L0024   r0 b024   ret1       0010010000 (0144)     [rom     ] selectRom : 1
L0025   r0 b025   trc1z1     0011010100 (0212)     [status  ] if statusBit ≠ 1 : bit 3
L0026   r0 b026              0001100011 (0099)     [flow    ]    (then) goTo : 24                 → ret1
L0027   r0 b027              0100101000 (0296)     [other   ] (stack) M → stack
L0028   r0 b028              0011001110 (0206)     [register] using W : clear c
L0029   r0 b029              1001110000 (0624)     [other   ] (data) C → dataAddress
L0030   r0 b030              0000000000 (0000)     [no oper ] nOp
L0031   r0 b031              1011111000 (0760)     [other   ] (data) data → C
L0032   r0 b032              1110101110 (0942)     [register] using W : exchange A C
L0033   r0 b033              1001111001 (0633)     [flow    ] jumpSub : 158                       → div1
L0034   r0 b034              1100011001 (0793)     [flow    ] jumpSub : 198                       → exch
L0035   r0 b035              1010001001 (0649)     [flow    ] jumpSub : 162                       → mul1
L0036   r0 b036              1100011001 (0793)     [flow    ] jumpSub : 198                       → exch
L0037   r0 b037              1011111110 (0766)     [register] using S : clear A
L0038   r0 b038              1011111000 (0760)     [other   ] (data) data → C
L0039   r0 b039              0110111110 (0446)     [register] using S : if C = 0
L0040   r0 b040              0010110111 (0183)     [flow    ]    (then) goTo : 45                 → trc2
L0041   r0 b041              1101111110 (0894)     [register] using S : decrement A
L0042   r0 b042              0010110101 (0181)     [flow    ] (if no carry then) jumpSub : 45    → trc2
L0043   r0 b043   pwo2       0001010011 (0083)     [flow    ] goTo : 20                           → pwo3
L0044   r0 b044   sqt1       0010010000 (0144)     [rom     ] selectRom : 1
```

```
p: L0000:  ..1.1.11.1  -> L0053          PWO1  :  JSB PWO2
p: L0001:  11...1....  -> L6002   *****  TMS5  :  SELECT ROM 6
p: L0002:  .1..1.1..1  -> L0112          TMS2  :  JSB FRAC
p: L0003:  11..1.111.                            :  A EXCHANGE B[W]
p: L0004:  11111..11.                            :  A + 1 -> A[M]
p: L0005:  .1..111.1.                            :  A -> B[XS]
p: L0006:  11.1111.1.                            :  A - 1 -> A[XS]
p: L0007:  111...111.                            :  A + B -> A[W]
p: L0010:  1.111.1.1.                            :  0 -> A[X]
p: L0011:  111...1..1  -> L0342                  :  JSB MLOP
p: L0012:  111...1..1  -> L0342                  :  JSB MLOP
p: L0013:  1.111.1.1.                            :  0 -> A[X]
p: L0014:  11..1..1.1  -> L0311                  :  JSB NORM
p: L0015:  .11..1..11  -> L0144                  :  GO TO TMS4
p: L0016:  1....1.1..                    DMDTZ4:  IF S8 # 1
p: L0017:  .1.1...111  -> L0121                  :       THEN GO TO DMT2
p: L0020:  1....111.1  -> L0207          TDMS  :  JSB MODE
p: L0021:  111.1.111.                            :  A EXCHANGE C[W]
p: L0022:  .11...111.                            :  C -> A[W]
p: L0023:  ......1.11  -> L0002                  :  GO TO TMS2
p: L0024:  1.1..1....  -> L5025   *****  PWO3  :  SELECT ROM 5
p: L0025:  1....111.1  -> L0207          TPL4  :  JSB MODE
p: L0026:  111.1.111.                            :  A EXCHANGE C[W]
p: L0027:  1....1....  -> L4030   *****          :  SELECT ROM 4
p: L0030:  ..1..1....  -> L1031   *****  RET1  :  SELECT ROM 1
p: L0031:  ..11.1.1..                    TRC1Z1:  IF S3 # 1
p: L0032:  ...11...11  -> L0030                  :       THEN GO TO RET1
p: L0033:  .1..1.1...                            :  C -> STACK
p: L0034:  ..11..111.                            :  0 -> C[W]
p: L0035:  1..111....                            :  C -> DATA ADDRESS
p: L0036:  ..........                            :  NO OPERATION
p: L0037:  1.11111...                            :  DATA -> C
p: L0040:  111.1.111.                            :  A EXCHANGE C[W]
p: L0041:  1..1111..1  -> L0236                  :  JSB DIV1
p: L0042:  11...11..1  -> L0306                  :  JSB EXCH
p: L0043:  1.1...1..1  -> L0242                  :  JSB MUL1
p: L0044:  11...11..1  -> L0306                  :  JSB EXCH
p: L0045:  1.1111111.                            :  0 -> A[S]
p: L0046:  1.11111...                            :  DATA -> C
p: L0047:  .11.11111.                            :  IF C[S] = 0
p: L0050:  ..1.11.111  -> L0055                  :       THEN GO TO TRC2
p: L0051:  11.111111.                            :  A - 1 -> A[S]
p: L0052:  ..1.11.1.1  -> L0055                  :  JSB TRC2
p: L0053:  ...1.1..11  -> L0024          PWO2  :  GO TO PWO3
p: L0054:  ..1..1....  -> L1055   *****  SQT1  :  SELECT ROM 1
```

```
L0045   r0 b045   trc2       1110101110 (0942)     [register] using W : exchange A C
L0046   r0 b046              1000010100 (0532)     [status  ] if statusBit ≠ 1 : bit 8
L0047   r0 b047              1111001011 (0971)     [flow    ]    (then) goTo : 242                → reg9z4
L0048   r0 b048              0011111110 (0254)     [register] using S : 0 − C − 1 → C
L0049   r0 b049              1111001001 (0969)     [flow    ] jumpSub : 242                      → reg9z4
L0050   r0 b050   sqrtz3     0001010100 (0084)     [status  ] if statusBit ≠ 1 : bit 1
L0051   r0 b051              0010110011 (0179)     [flow    ]    (then) goTo : 44                 → sqt1
L0052   r0 b052              1010010100 (0660)     [status  ] if statusBit ≠ 1 : bit 10
L0053   r0 b053              0110100111 (0423)     [flow    ]    (then) goTo : 105                → tn12
L0054   r0 b054              0011101011 (0235)     [flow    ] goTo : 58                          → mag1z1
L0055   r0 b055   tanx       0010010000 (0144)     [rom     ] selectRom : 1
L0056   r0 b056   sn12       0110001110 (0398)     [register] using W : C → A
L0057   r0 b057              0010010000 (0144)     [rom     ] selectRom : 1
L0058   r0 b058   mag1z1     1110101110 (0942)     [register] using W : exchange A C
L0059   r0 b059              0110001110 (0398)     [register] using W : C → A
L0060   r0 b060              0111111010 (0506)     [register] using XS : increment C
L0061   r0 b061              0100011011 (0283)     [flow    ] (if no carry then) goTo : 70       → mag3
L0062   r0 b062              0110101010 (0426)     [register] using X : if C = 0
L0063   r0 b063              0100011011 (0283)     [flow    ]    (then) goTo : 70                 → mag3
L0064   r0 b064              0011001110 (0206)     [register] using W : clear c
L0065   r0 b065              0000001100 (0012)     [pointer ] value → P : value 0
L0066   r0 b066              0101011000 (0344)     [constant] loadConstant : 5
L0067   r0 b067              1100001100 (0780)     [pointer ] value → P : value 12
L0068   r0 b068              0111001010 (0458)     [register] using X : A + C → C
L0069   r0 b069              1110101111 (0943)     [flow    ] (if no carry then) goTo : 235      → mag4
L0070   r0 b070   mag3       1010010100 (0660)     [status  ] if statusBit ≠ 1 : bit 10
L0071   r0 b071              0111010111 (0471)     [flow    ]    (then) goTo : 117                → rom1
L0072   r0 b072              1110101110 (0942)     [register] using W : exchange A C
L0073   r0 b073              0011100011 (0227)     [flow    ] goTo : 56                          → sn12
L0074   r0 b074   frac       0011001010 (0202)     [register] using X : clear c
L0075   r0 b075              0000001100 (0012)     [pointer ] value → P : value 0
L0076   r0 b076              0101011000 (0344)     [constant] loadConstant : 5
L0077   r0 b077              1101001010 (0842)     [register] using X : A − C → A
L0078   r0 b078              1001111010 (0634)     [register] using XS : if A ≥ 1
L0079   r0 b079              1100110011 (0819)     [flow    ]    (then) goTo : 204                → frc1
L0080   r0 b080              1111111111 (1023)     [flow    ] goTo : 255                         → err2
L0081   r0 b081   dmt2       0100101001 (0297)     [flow    ] jumpSub : 74                       → frac
L0082   r0 b082              1110001101 (0909)     [flow    ] jumpSub : 227                      → mlp0
L0083   r0 b083              0000101110 (0046)     [register] using W : clear B
L0084   r0 b084              1100110010 (0818)     [register] using WP : exchange A B
L0085   r0 b085              1100101110 (0814)     [register] using W : exchange A B
L0086   r0 b086              0100001110 (0270)     [register] using W : leftShift A
L0087   r0 b087              1110010101 (0917)     [flow    ] jumpSub : 229                      → mlp2
L0088   r0 b088              0111000100 (0452)     [status  ] set statusBit : bit 7
L0089   r0 b089              1100100101 (0805)     [flow    ] jumpSub : 201                      → norm
```

```
p: L0055:  111.1.111.                    TRC2  :  A EXCHANGE C[W]
p: L0056:  1....1.1..                            :  IF S8 # 1
p: L0057:  1111..1.11  -> L0362                  :       THEN GO TO REG9Z4
p: L0060:  ..1111111.                            :  0 − C − 1 -> C[S]
p: L0061:  1111..1..1  -> L0362                  :  JSB REG9Z4
p: L0062:  ...1.1.1..                    SQRTZ3:  IF S1 #  1
p: L0063:  ..1.11..11  -> L0054                  :       THEN GO TO SQT1
p: L0064:  1.1..1.1..                            :  IF S10 # 1
p: L0065:  .11.1..111  -> L0151                  :       THEN GO TO TN12
p: L0066:  ..111.1.11  -> L0072                  :  GO TO MAG1Z1
p: L0067:  ..1..1....  -> L1070   *****  TANX  :  SELECT ROM 1
p: L0070:  .11...111.                    SN12  :  C -> A[W]
p: L0071:  ..1..1....  -> L1072   *****         :  SELECT ROM 1
p: L0072:  111.1.111.                    MAG1Z1:  A EXCHANGE C[W]
p: L0073:  .11...111.                            :  C -> A[W]
p: L0074:  .111111.1.                            :  C + 1 -> C[XS]
p: L0075:  .1...11.11  -> L0106                  :  IF NO CARRY GO TO MAG3
p: L0076:  .11.1.1.1.                            :  IF C[X] = 0
p: L0077:  .1...11.11  -> L0106                  :       THEN GO TO MAG3
p: L0100:  ..11..111.                            :  0 -> C[W]
p: L0101:  ......11..                            :  0 -> P
p: L0102:  .1.1.11...                            :  LOAD CONSTANT 5
p: L0103:  11....11..                            :  12 -> P
p: L0104:  .111..1.1.                            :  A + C -> C[X]
p: L0105:  111.1.1111  -> L0353                  :  IF NO CARRY GO TO MAG4
p: L0106:  1.1..1.1..                    MAG3  :  IF S10 # 1
p: L0107:  .111.1.111  -> L0165                  :       THEN GO TO ROM1
p: L0110:  111.1.111.                            :  A EXCHANGE C[W]
p: L0111:  ..111...11  -> L0070                  :  GO TO SN12
p: L0112:  ..11..1.1.                    FRAC  :  0 -> C[X]
p: L0113:  ......11..                            :  0 -> P
p: L0114:  .1.1.11...                            :  LOAD CONSTANT 5
p: L0115:  11.1..1.1.                            :  A − C -> A[X]
p: L0116:  1..1111.1.                            :  IF A[XS] >= 1
p: L0117:  11..11..11  -> L0314                  :       THEN GO TO FRC1
p: L0120:  1111111111  -> L0377                  :  GO TO ERR2
p: L0121:  .1..1.1..1  -> L0112          DMT2  :  JSB FRAC
p: L0122:  111...11.1  -> L0343                  :  JSB MLP0
p: L0123:  ....1.111.                            :  0 -> B[W]
p: L0124:  11..11..1.                            :  A EXCHANGE B[WP]
p: L0125:  11..1.111.                            :  A EXCHANGE B[W]
p: L0126:  .1....111.                            :  SHIFT LEFT A[W]
p: L0127:  111..1.1.1  -> L0345                  :  JSB MLP2
p: L0130:  .111...1..                            :  1 -> S7
p: L0131:  11..1..1.1  -> L0311                  :  JSB NORM
```

```
L0090   r0 b090             0011001110 (0206)     [register] using W : clear c
L0091   r0 b091             0011011000 (0216)     [constant] loadConstant : 3
L0092   r0 b092             0110011000 (0408)     [constant] loadConstant : 6
L0093   r0 b093             1100001100 (0780)     [pointer ] value → P : value 12
L0094   r0 b094             1001111001 (0633)     [flow    ] jumpSub : 158                    → div1
L0095   r0 b095             1000011001 (0537)     [flow    ] jumpSub : 134                    → mod0
L0096   r0 b096             0111110111 (0503)     [flow    ] goTo : 125                       → fst0
L0097   r0 b097   rtfg       0111101010 (0490)     [register] using X : increment C
L0098   r0 b098             0111100010 (0482)     [register] using P : increment C
L0099   r0 b099             1000001111 (0527)     [flow    ] (if no carry then) goTo : 131   → dvml
L0100   r0 b100   tms4       0011001110 (0206)     [register] using W : clear c
L0101   r0 b101             0010001100 (0140)     [pointer ] value → P : value 2
L0102   r0 b102             0100011000 (0280)     [constant] loadConstant : 4
L0103   r0 b103             1110101110 (0942)     [register] using W : exchange A C
L0104   r0 b104             0000000111 (0007)     [flow    ] goTo : 1                         → tms5
L0105   r0 b105   tn12       0010010000 (0144)     [rom     ] selectRom : 1
L0106   r0 b106   tpol       0111000100 (0452)     [status  ] set statusBit : bit 7
L0107   r0 b107             0100010100 (0276)     [status  ] if statusBit ≠ 1 : bit 4
L0108   r0 b108             0111001111 (0463)     [flow    ]    (then) goTo : 115             → tpl3
L0109   r0 b109             1010101110 (0686)     [register] using W : C + C → C
L0110   r0 b110             0111011110 (0478)     [register] using S : A + C → C
L0111   r0 b111             1001100110 (0614)     [register] using M : if A ≥ 1
L0112   r0 b112             0111001011 (0459)     [flow    ]    (then) goTo : 114             → tpl2
L0113   r0 b113             0011111110 (0254)     [register] using S : 0 − C − 1 → C
L0114   r0 b114   tpl2       1010011001 (0665)     [flow    ] jumpSub : 166                    → sub1
L0115   r0 b115   tpl3       0001000100 (0068)     [status  ] set statusBit : bit 1
L0116   r0 b116             0001010111 (0087)     [flow    ] goTo : 21                        → tpl4
L0117   r0 b117   rom1       0010010000 (0144)     [rom     ] selectRom : 1
L0118   r0 b118   drg1z1     1000010100 (0532)     [status  ] if statusBit ≠ 1 : bit 8
L0119   r0 b119             0110101011 (0427)     [flow    ]    (then) goTo : 106             → tpol
L0120   r0 b120   drg0       1000011101 (0541)     [flow    ] jumpSub : 135                    → mode
L0121   r0 b121             0001000100 (0068)     [status  ] set statusBit : bit 1
L0122   r0 b122             1010010100 (0660)     [status  ] if statusBit ≠ 1 : bit 10
L0123   r0 b123             0011101011 (0235)     [flow    ]    (then) goTo : 58              → mag1z1
L0124   r0 b124             1111001111 (0975)     [flow    ] goTo : 243                       → regx
L0125   r0 b125   fst0       1110101110 (0942)     [register] using W : exchange A C
L0126   r0 b126   fst1       0110010000 (0400)     [rom     ] selectRom : 3
L0127   r0 b127   gtfd       0101101010 (0362)     [register] using X : decrement C
L0128   r0 b128             0101101010 (0362)     [register] using X : decrement C
L0129   r0 b129             0111000100 (0452)     [status  ] set statusBit : bit 7
L0130   r0 b130   dtfr       0101100010 (0354)     [register] using P : decrement C
L0131   r0 b131   dvml       1010010100 (0660)     [status  ] if statusBit ≠ 1 : bit 10
L0132   r0 b132             1001111011 (0635)     [flow    ]    (then) goTo : 158             → div1
L0133   r0 b133             1010001011 (0651)     [flow    ] goTo : 162                       → mul1
L0134   r0 b134   mod0       1010100100 (0676)     [status  ] clear statusBit : bit 10
```

```
p: L0132:  ..11..111.                          :  0 -> C[W]
p: L0133:  ..11.11...                          :  LOAD CONSTANT 3
p: L0134:  .11..11...                          :  LOAD CONSTANT 6
p: L0135:  11....11..                          :  12 -> P
p: L0136:  1..1111..1  -> L0236                :  JSB DIV1
p: L0137:  1....11..1  -> L0206                :  JSB MOD0
p: L0140:  .11111.111  -> L0175                :  GO TO FST0
p: L0141:  .1111.1.1.                    RTFG  :  C + 1 -> C[X]
p: L0142:  .1111...1.                          :  C + 1 -> C[P]
p: L0143:  1.....1111  -> L0203                :  IF NO CARRY GO TO DVML
p: L0144:  ..11..111.                    TMS4  :  0 -> C[W]
p: L0145:  ..1...11..                          :  2 -> P
p: L0146:  .1...11...                          :  LOAD CONSTANT 4
p: L0147:  111.1.111.                          :  A EXCHANGE C[W]
p: L0150:  .......111  -> L0001                :  GO TO TMS5
p: L0151:  ..1..1....  -> L1152   *****  TN12  :  SELECT ROM 1
p: L0152:  .111...1..                    TPOL  :  1 -> S7
p: L0153:  .1...1.1..                          :  IF S4 # 1
p: L0154:  .111..1111  -> L0163                :       THEN GO TO TPL3
p: L0155:  1.1.1.111.                          :  C + C -> C[W]
p: L0156:  .111.1111.                          :  A + C -> C[S]
p: L0157:  1..11..11.                          :  IF A[M] >= 1
p: L0160:  .111..1.11  -> L0162                :       THEN GO TO TPL2
p: L0161:  ..1111111.                          :  0 - C - 1 -> C[S]
p: L0162:  1.1..11..1  -> L0246          TPL2  :  JSB SUB1
p: L0163:  ...1...1..                    TPL3  :  1 -> S1
p: L0164:  ...1.1.111  -> L0025                :  GO TO TPL4
p: L0165:  ..1..1....  -> L1166   *****  ROM1  :  SELECT ROM 1
p: L0166:  1....1.1..                    DRG1Z1:  IF S8 # 1
p: L0167:  .11.1.1.11  -> L0152                :       THEN GO TO TPOL
p: L0170:  1....111.1  -> L0207          DRG0  :  JSB MODE
p: L0171:  ...1...1..                          :  1 -> S1
p: L0172:  1.1..1.1..                          :  IF S10 # 1
p: L0173:  ..111.1.11  -> L0072                :       THEN GO TO MAG1Z1
p: L0174:  1111..1111  -> L0363                :  GO TO REGX
p: L0175:  111.1.111.                    FST0  :  A EXCHANGE C[W]
p: L0176:  .11..1....  -> L3177   *****  FST1  :  SELECT ROM 3
p: L0177:  .1.11.1.1.                    GTFD  :  C - 1 -> C[X]
p: L0200:  .1.11.1.1.                          :  C - 1 -> C[X]
p: L0201:  .111...1..                          :  1 -> S7
p: L0202:  .1.11...1.                    DTFR  :  C - 1 -> C[P]
p: L0203:  1.1..1.1..                    DVML  :  IF S10 # 1
p: L0204:  1..1111.11  -> L0236                :       THEN GO TO DIV1
p: L0205:  1.1...1.11  -> L0242                :  GO TO MUL1
p: L0206:  1.1.1..1..                    MOD0  :  0 -> S10
```

```
L0135   r0 b135   mode       0111000100 (0452)     [status  ] set statusBit : bit 7
L0136   r0 b136              0100100100 (0292)     [status  ] clear statusBit : bit 4
L0137   r0 b137              0110000100 (0388)     [status  ] set statusBit : bit 6
L0138   r0 b138              1010101000 (0680)     [other   ] (register) M → C
L0139   r0 b139              0110111110 (0446)     [register] using S : if C = 0
L0140   r0 b140              1000111011 (0571)     [flow    ]    (then) goTo : 142                → mod2
L0141   r0 b141              0100000100 (0260)     [status  ] set statusBit : bit 4
L0142   r0 b142   mod2       0111111110 (0510)     [register] using S : increment C
L0143   r0 b143              1001000111 (0583)     [flow    ] (if no carry then) goTo : 145       → mod3
L0144   r0 b144              0110100100 (0420)     [status  ] clear statusBit : bit 6
L0145   r0 b145   mod3       0011001110 (0206)     [register] using W : clear c
L0146   r0 b146              0111101010 (0490)     [register] using X : increment C
L0147   r0 b147              0001010100 (0084)     [status  ] if statusBit ≠ 1 : bit 1
L0148   r0 b148              1100000011 (0771)     [flow    ]    (then) goTo : 192                → degr
L0149   r0 b149              0100010100 (0276)     [status  ] if statusBit ≠ 1 : bit 4
L0150   r0 b150              1110000111 (0903)     [flow    ]    (then) goTo : 225                → ret0
L0151   r0 b151              0111100100 (0484)     [status  ] clear statusBit : bit 7
L0152   r0 b152              0110010100 (0404)     [status  ] if statusBit ≠ 1 : bit 6
L0153   r0 b153              1000001011 (0523)     [flow    ]    (then) goTo : 130                → dtfr
L0154   r0 b154              0110000111 (0391)     [flow    ] goTo : 97                           → rtfg
L0155   r0 b155   add3z1     1010101011 (0683)     [flow    ] goTo : 170                          → add3
L0156   r0 b156   mldv       1010010100 (0660)     [status  ] if statusBit ≠ 1 : bit 10
L0157   r0 b157              1010001011 (0651)     [flow    ]    (then) goTo : 162                → mul1
L0158   r0 b158   div1       1011000100 (0708)     [status  ] set statusBit : bit 11
L0159   r0 b159   div0       0001100100 (0100)     [status  ] clear statusBit : bit 1
L0160   r0 b160              0000101110 (0046)     [register] using W : clear B
L0161   r0 b161              1010010111 (0663)     [flow    ] goTo : 165                          → divx
L0162   r0 b162   mul1       1011000100 (0708)     [status  ] set statusBit : bit 11
L0163   r0 b163   mul0       0001100100 (0100)     [status  ] clear statusBit : bit 1
L0164   r0 b164   mulx       0010010000 (0144)     [rom     ] selectRom : 1
L0165   r0 b165   divx       0010010000 (0144)     [rom     ] selectRom : 1
L0166   r0 b166   sub1       1011000100 (0708)     [status  ] set statusBit : bit 11
L0167   r0 b167   sub0       0011111110 (0254)     [register] using S : 0 − C − 1 → C
L0168   r0 b168   add0zx     0001100100 (0100)     [status  ] clear statusBit : bit 1
L0169   r0 b169              0010100100 (0164)     [status  ] clear statusBit : bit 2
L0170   r0 b170   add3       0000101110 (0046)     [register] using W : clear B
L0171   r0 b171              1111111010 (1018)     [register] using XS : increment A
L0172   r0 b172              1111111010 (1018)     [register] using XS : increment A
L0173   r0 b173              0111111010 (0506)     [register] using XS : increment C
L0174   r0 b174              0111111010 (0506)     [register] using XS : increment C
L0175   r0 b175              0001001010 (0074)     [register] using X : if A ≥ C
L0176   r0 b176              1011001011 (0715)     [flow    ]    (then) goTo : 178                → add4
L0177   r0 b177              1110101110 (0942)     [register] using W : exchange A C
L0178   r0 b178   add4       1110100110 (0934)     [register] using M : exchange A C
L0179   r0 b179              0110100110 (0422)     [register] using M : if C = 0
```

```
p: L0207:  .111...1..                    MODE  :  1 -> S7
p: L0210:  .1..1..1..                            :  0 -> S4
p: L0211:  .11....1..                            :  1 -> S6
p: L0212:  1.1.1.1...                            :  M -> C
p: L0213:  .11.11111.                            :  IF C[S] = 0
p: L0214:  1...111.11  -> L0216                  :       THEN GO TO MOD2
p: L0215:  .1.....1..                            :  1 -> S4
p: L0216:  .11111111.                    MOD2  :  C + 1 -> C[S]
p: L0217:  1..1...111  -> L0221                  :  IF NO CARRY GO TO MOD3
p: L0220:  .11.1..1..                            :  0 -> S6
p: L0221:  ..11..111.                    MOD3  :  0 -> C[W]
p: L0222:  .1111.1.1.                            :  C + 1 -> C[X]
p: L0223:  ...1.1.1..                            :  IF S1 # 1
p: L0224:  11......11  -> L0300                  :       THEN GO TO DEGR
p: L0225:  .1...1.1..                            :  IF S4 # 1
p: L0226:  111....111  -> L0341                  :       THEN GO TO RET0
p: L0227:  .1111..1..                            :  0 -> S7
p: L0230:  .11..1.1..                            :  IF S6 # 1
p: L0231:  1.....1.11  -> L0202                  :       THEN GO TO DTFR
p: L0232:  .11....111  -> L0141                  :  GO TO RTFG
p: L0233:  1.1.1.1.11  -> L0252          ADD3Z1:  GO TO ADD3
p: L0234:  1.1..1.1..                    MLDV  :  IF S10 # 1
p: L0235:  1.1...1.11  -> L0242                  :       THEN GO TO MUL1
p: L0236:  1.11...1..                    DIV1  :  1 -> S11
p: L0237:  ...11..1..                    DIV0  :  0 -> S1
p: L0240:  ....1.111.                            :  0 -> B[W]
p: L0241:  1.1..1.111  -> L0245                  :  GO TO DIVX
p: L0242:  1.11...1..                    MUL1  :  1 -> S11
p: L0243:  ...11..1..                    MUL0  :  0 -> S1
p: L0244:  ..1..1....  -> L1245   *****  MULX  :  SELECT ROM 1
p: L0245:  ..1..1....  -> L1246   *****  DIVX  :  SELECT ROM 1
p: L0246:  1.11...1..                    SUB1  :  1 -> S11
p: L0247:  ..1111111.                    SUB0  :  0 − C − 1 -> C[S]
p: L0250:  ...11..1..                    ADD0ZX:  0 -> S1
p: L0251:  ..1.1..1..                            :  0 -> S2
p: L0252:  ....1.111.                    ADD3  :  0 -> B[W]
p: L0253:  1111111.1.                            :  A + 1 -> A[XS]
p: L0254:  1111111.1.                            :  A + 1 -> A[XS]
p: L0255:  .111111.1.                            :  C + 1 -> C[XS]
p: L0256:  .111111.1.                            :  C + 1 -> C[XS]
p: L0257:  ...1..1.1.                            :  IF A >= C[X]
p: L0260:  1.11..1.11  -> L0262                  :       THEN GO TO ADD4
p: L0261:  111.1.111.                            :  A EXCHANGE C[W]
p: L0262:  111.1..11.                    ADD4  :  A EXCHANGE C[M]
p: L0263:  .11.1..11.                            :  IF C[M] = 0
```

```
L0180   r0 b180             1011011011 (0731)   [flow    ]    (then) goTo : 182                → add5
L0181   r0 b181             1110101110 (0942)   [register] using W : exchange A C
L0182   r0 b182   add5       1000100110 (0550)   [register] using M : exchange B C
L0183   r0 b183   add6       0001001010 (0074)   [register] using X : if A ≥ C
L0184   r0 b184             1011111011 (0763)   [flow    ]    (then) goTo : 190                → add7
L0185   r0 b185   add8       1010001110 (0654)   [register] using W : rightShift B
L0186   r0 b186             1111101010 (1002)   [register] using X : increment A
L0187   r0 b187             0000001110 (0014)   [register] using W : if B = 0
L0188   r0 b188             1011111011 (0763)   [flow    ]    (then) goTo : 190                → add7
L0189   r0 b189             1011011111 (0735)   [flow    ] goTo : 183                         → add6
L0190   r0 b190   add7       0010010000 (0144)   [rom     ] selectRom : 1
L0191   r0 b191   err2z1     1111111111 (1023)   [flow    ] goTo : 255                         → err2
L0192   r0 b192   degr       0110010100 (0404)   [status  ] if statusBit ≠ 1 : bit 6
L0193   r0 b193             1110000111 (0903)   [flow    ]    (then) goTo : 225                → ret0
L0194   r0 b194             0111100100 (0484)   [status  ] clear statusBit : bit 7
L0195   r0 b195             0100010100 (0276)   [status  ] if statusBit ≠ 1 : bit 4
L0196   r0 b196             1000001011 (0523)   [flow    ]    (then) goTo : 130                → dtfr
L0197   r0 b197             0111111111 (0511)   [flow    ] goTo : 127                         → gtfd
L0198   r0 b198   exch       0110101000 (0424)   [other   ] (stack) stack → A
L0199   r0 b199             0100101000 (0296)   [other   ] (stack) M → stack
L0200   r0 b200             0000110000 (0048)   [other   ] (flow) return
L0201   r0 b201   norm       1011000100 (0708)   [status  ] set statusBit : bit 11
L0202   r0 b202             0010010000 (0144)   [rom     ] selectRom : 1
L0203   r0 b203   pii2       0010010000 (0144)   [rom     ] selectRom : 1
L0204   r0 b204   frc1       1000100110 (0550)   [register] using M : exchange B C
L0205   r0 b205   frc2       1010001110 (0654)   [register] using W : rightShift B
L0206   r0 b206             1111101010 (1002)   [register] using X : increment A
L0207   r0 b207             1100110111 (0823)   [flow    ] (if no carry then) goTo : 205  → frc2
L0208   r0 b208             1011101110 (0750)   [register] using W : clear A
L0209   r0 b209             0110001100 (0396)   [pointer ] value → P : value 6
L0210   r0 b210             0000110000 (0048)   [other   ] (flow) return
L0211   r0 b211   pii4       1011010100 (0724)   [status  ] if statusBit ≠ 1 : bit 11
L0212   r0 b212             1101100011 (0867)   [flow    ]    (then) goTo : 216                → pirt
L0213   r0 b213             1010101110 (0686)   [register] using W : C + C → C
L0214   r0 b214             0111000100 (0452)   [status  ] set statusBit : bit 7
L0215   r0 b215             1001110011 (0627)   [flow    ] goTo : 156                         → mldv
L0216   r0 b216   pirt       0010010000 (0144)   [rom     ] selectRom : 1
L0217   r0 b217   pii4z1     1100001100 (0780)   [pointer ] value → P : value 12
L0218   r0 b218             1101001111 (0847)   [flow    ] goTo : 211                         → pii4
L0219   r0 b219   retnz1     1011010100 (0724)   [status  ] if statusBit ≠ 1 : bit 11
L0220   r0 b220             1111100111 (0999)   [flow    ]    (then) goTo : 249                → rtrn
L0221   r0 b221             1011000100 (0708)   [status  ] set statusBit : bit 11
L0222   r0 b222             0111010100 (0468)   [status  ] if statusBit ≠ 1 : bit 7
L0223   r0 b223             1100101111 (0815)   [flow    ]    (then) goTo : 203                → pii2
L0224   r0 b224             1011100100 (0740)   [status  ] clear statusBit : bit 11
```

```
p: L0264:  1.11.11.11  -> L0266                :       THEN GO TO ADD5
p: L0265:  111.1.111.                          :  A EXCHANGE C[W]
p: L0266:  1...1..11.                  ADD5    :  B EXCHANGE C[M]
p: L0267:  ...1..1.1.                  ADD6    :  IF A >= C[X]
p: L0270:  1.11111.11  -> L0276                :       THEN GO TO ADD7
p: L0271:  1.1...111.                  ADD8    :  SHIFT RIGHT B[W]
p: L0272:  11111.1.1.                          :  A + 1 -> A[X]
p: L0273:  ......111.                          :  IF B[W] = 0
p: L0274:  1.11111.11  -> L0276                :       THEN GO TO ADD7
p: L0275:  1.11.11111  -> L0267                :  GO TO ADD6
p: L0276:  ..1..1....  -> L1277   *****  ADD7  :  SELECT ROM 1
p: L0277:  1111111111  -> L0377          ERR2Z1:  GO TO ERR2
p: L0300:  .11..1.1..                    DEGR  :  IF S6 # 1
p: L0301:  111....111  -> L0341                :       THEN GO TO RET0
p: L0302:  .1111..1..                          :  0 -> S7
p: L0303:  .1...1.1..                          :  IF S4 # 1
p: L0304:  1.....1.11  -> L0202                :       THEN GO TO DTFR
p: L0305:  0.111111111 -> L0177                :  GO TO GTFD
p: L0306:  .11.1.1...                    EXCH  :  STACK -> A
p: L0307:  .1..1.1...                          :  C -> STACK
p: L0310:  ....11....                          :  RETURN
p: L0311:  1.11...1..                    NORM  :  1 -> S11
p: L0312:  ..1..1....  -> L1313   *****        :  SELECT ROM 1
p: L0313:  ..1..1....  -> L1314   *****  PII2  :  SELECT ROM 1
p: L0314:  1...1..11.                    FRC1  :  B EXCHANGE C[M]
p: L0315:  1.1...111.                    FRC2  :  SHIFT RIGHT B[W]
p: L0316:  11111.1.1.                          :  A + 1 -> A[X]
p: L0317:  11..11.111  -> L0315                :  IF NO CARRY GO TO FRC2
p: L0320:  1.111.111.                          :  0 -> A[W]
p: L0321:  .11...11..                          :  6 -> P
p: L0322:  ....11....                          :  RETURN
p: L0323:  1.11.1.1..                    PII4  :  IF S11 # 1
p: L0324:  11.11...11  -> L0330                :       THEN GO TO PIRT
p: L0325:  1.1.1.111.                          :  C + C -> C[W]
p: L0326:  .111...1..                          :  1 -> S7
p: L0327:  1..111..11  -> L0234                :  GO TO MLDV
p: L0330:  ..1..1....  -> L1331   *****  PIRT  :  SELECT ROM 1
p: L0331:  11....11..                    PII4Z1:  12 -> P
p: L0332:  11.1..1111  -> L0323                :  GO TO PII4
p: L0333:  1.11.1.1..                    RETNZ1:  IF S11 # 1
p: L0334:  11111..111  -> L0371                :       THEN GO TO RTRN
p: L0335:  1.11...1..                          :  1 -> S11
p: L0336:  .111.1.1..                          :  IF S7 # 1
p: L0337:  11..1.1111  -> L0313                :       THEN GO TO PII2
p: L0340:  1.111..1..                          :  0 -> S11
```

```
L0225   r0 b225   ret0       0000110000 (0048)     [other   ] (flow) return
L0226   r0 b226   mlop       0000101110 (0046)     [register] using W : clear B
L0227   r0 b227   mlp0       1100110010 (0818)     [register] using WP : exchange A B
L0228   r0 b228              1010001110 (0654)     [register] using W : rightShift B
L0229   r0 b229   mlp2       1010001100 (0652)     [pointer ] value → P : value 10
L0230   r0 b230   mlp3       1110001110 (0910)     [register] using W : A + B → A
L0231   r0 b231              0000011100 (0028)     [pointer ] decrement P
L0232   r0 b232              0100101100 (0300)     [pointer ] if P ≠ value : value 4
L0233   r0 b233              1110011011 (0923)     [flow    ]    (then) goTo : 230                → mlp3
L0234   r0 b234              0000110000 (0048)     [other   ] (flow) return
L0235   r0 b235   mag4       0011001110 (0206)     [register] using W : clear c
L0236   r0 b236              0111100010 (0482)     [register] using P : increment C
L0237   r0 b237              1010010100 (0660)     [status  ] if statusBit ≠ 1 : bit 10
L0238   r0 b238              0011011111 (0223)     [flow    ]    (then) goTo : 55                 → tanx
L0239   r0 b239              0110100111 (0423)     [flow    ] goTo : 105                          → tn12
L0240   r0 b240   reg9       0001010100 (0084)     [status  ] if statusBit ≠ 1 : bit 1
L0241   r0 b241              0111111011 (0507)     [flow    ]    (then) goTo : 126                → fst1
L0242   r0 b242   reg9z4     1110101110 (0942)     [register] using W : exchange A C
L0243   r0 b243   regx       0011001110 (0206)     [register] using W : clear c
L0244   r0 b244              0101100010 (0354)     [register] using P : decrement C
L0245   r0 b245              1001110000 (0624)     [other   ] (data) C → dataAddress
L0246   r0 b246              0011001110 (0206)     [register] using W : clear c
L0247   r0 b247              1011110000 (0752)     [other   ] (data) C → data
L0248   r0 b248              0111110111 (0503)     [flow    ] goTo : 125                          → fst0
L0249   r0 b249   rtrn       0011010100 (0212)     [status  ] if statusBit ≠ 1 : bit 3
L0250   r0 b250              1111000011 (0963)     [flow    ]    (then) goTo : 240                → reg9
L0251   r0 b251              1010010100 (0660)     [status  ] if statusBit ≠ 1 : bit 10
L0252   r0 b252              1111111011 (1019)     [flow    ]    (then) goTo : 254                → ret5
L0253   r0 b253   ret4       1000010000 (0528)     [rom     ] selectRom : 4
L0254   r0 b254   ret5       1010010000 (0656)     [rom     ] selectRom : 5
L0255   r0 b255   err2       0100010000 (0272)     [rom     ] selectRom : 2
```

```
p: L0341:  ....11....                    RET0  :  RETURN
p: L0342:  ....1.111.                    MLOP  :  0 -> B[W]
p: L0343:  11..11..1.                    MLP0  :  A EXCHANGE B[WP]
p: L0344:  1.1...111.                          :  SHIFT RIGHT B[W]
p: L0345:  1.1...11..                    MLP2  :  10 -> P
p: L0346:  111...111.                    MLP3  :  A + B -> A[W]
p: L0347:  .....111..                          :  P - 1 -> P
p: L0350:  .1..1.11..                          :  IF P # 4
p: L0351:  111..11.11  -> L0346                :      THEN GO TO MLP3
p: L0352:  ....11....                          :  RETURN
p: L0353:  ..11..111.                    MAG4  :  0 -> C[W]
p: L0354:  .1111...1.                          :  C + 1 -> C[P]
p: L0355:  1.1..1.1..                          :  IF S10 # 1
p: L0356:  ..11.11111  -> L0067                :      THEN GO TO TANX
p: L0357:  .11.1..111  -> L0151                :  GO TO TN12
p: L0360:  ...1.1.1..                    REG9  :  IF S1 # 1
p: L0361:  .111111.11  -> L0176                :      THEN GO TO FST1
p: L0362:  111.1.111.                    REG9Z4:  A EXCHANGE C[W]
p: L0363:  ..11..111.                    REGX  :  0 -> C[W]
p: L0364:  .1.11...1.                          :  C - 1 -> C[P]
p: L0365:  1..111....                          :  C -> DATA ADDRESS
p: L0366:  ..11..111.                          :  0 -> C[W]
p: L0367:  1.1111....                          :  C -> DATA
p: L0370:  .11111.111  -> L0175                :  GO TO FST0
p: L0371:  ..11.1.1..                    RTRN  :  IF S3 # 1
p: L0372:  1111....11  -> L0360                :      THEN GO TO REG9
p: L0373:  1.1..1.1..                          :  IF S10 # 1
p: L0374:  1111111.11  -> L0376                :      THEN GO TO RET5
p: L0375:  1....1....  -> L4376   *****  RET4  :  SELECT ROM 4
p: L0376:  1.1..1....  -> L5377   *****  RET5  :  SELECT ROM 5
p: L0377:  .1...1....  -> L2000   *****  ERR2  :  SELECT ROM 2
```

```
L1000   r1 b000             1000111110 (0574)     [register] using S : exchange B C
L1001   r1 b001             1111001111 (0975)     [flow    ] goTo : 243                          → tan13
L1002   r1 b002   tan15      1100101110 (0814)     [register] using W : exchange A B
L1003   r1 b003             0011000001 (0193)     [flow    ] jumpSub : 48                        → tnm11
L1004   r1 b004             1011111000 (0760)     [other   ] (data) data → C
L1005   r1 b005             1110101110 (0942)     [register] using W : exchange A C
L1006   r1 b006             0011000001 (0193)     [flow    ] jumpSub : 48                        → tnm11
L1007   r1 b007             1011111000 (0760)     [other   ] (data) data → C
L1008   r1 b008             1110101110 (0942)     [register] using W : exchange A C
L1009   r1 b009   tanx       1001010100 (0596)     [status  ] if statusBit ≠ 1 : bit 9
L1010   r1 b010             0000110011 (0051)     [flow    ]    (then) goTo : 12                 → tan16
L1011   r1 b011             1110101110 (0942)     [register] using W : exchange A C
L1012   r1 b012   tan16      0101010100 (0340)     [status  ] if statusBit ≠ 1 : bit 5
L1013   r1 b013             0001101011 (0107)     [flow    ]    (then) goTo : 26                 → asn12
L1014   r1 b014             0001111110 (0126)     [register] using S : if C ≥ 1
L1015   r1 b015             0001000111 (0071)     [flow    ]    (then) goTo : 17                 → tan17
L1016   r1 b016             1000100100 (0548)     [status  ] clear statusBit : bit 8
L1017   r1 b017   tan17      0011011110 (0222)     [register] using S : clear c
L1018   r1 b018             1010011001 (0665)     [flow    ] jumpSub : 166                       → div11
L1019   r1 b019   asn11      1011110000 (0752)     [other   ] (data) C → data
L1020   r1 b020             1010010101 (0661)     [flow    ] jumpSub : 165                       → mpy11
L1021   r1 b021             1001100001 (0609)     [flow    ] jumpSub : 152                       → add10
L1022   r1 b022             0010110101 (0181)     [flow    ] jumpSub : 45                        → sqt11
L1023   r1 b023             1011111000 (0760)     [other   ] (data) data → C
L1024   r1 b024             0000010000 (0016)     [rom     ] selectRom : 0
L1025   r1 b025   asn1z0     1110101110 (0942)     [register] using W : exchange A C
L1026   r1 b026   asn12      1010011001 (0665)     [flow    ] jumpSub : 166                       → div11
L1027   r1 b027             1010010100 (0660)     [status  ] if statusBit ≠ 1 : bit 10
L1028   r1 b028             1101101011 (0875)     [flow    ]    (then) goTo : 218                → rtn12
L1029   r1 b029   atn11      1011101110 (0750)     [register] using W : clear A
L1030   r1 b030             1111100010 (0994)     [register] using P : increment A
L1031   r1 b031             0100100110 (0294)     [register] using M : A → B
L1032   r1 b032             1110100110 (0934)     [register] using M : exchange A C
L1033   r1 b033   atn12      0101101010 (0362)     [register] using X : decrement C
L1034   r1 b034             1010010010 (0658)     [register] using WP : rightShift B
L1035   r1 b035             0110111010 (0442)     [register] using XS : if C = 0
L1036   r1 b036             0010000111 (0135)     [flow    ]    (then) goTo : 33                 → atn12
L1037   r1 b037   atn13      1011010010 (0722)     [register] using WP : rightShift A
L1038   r1 b038             0111101010 (0490)     [register] using X : increment C
L1039   r1 b039             0010010111 (0151)     [flow    ] (if no carry then) goTo : 37       → atn13
L1040   r1 b040             1011001110 (0718)     [register] using W : rightShift A
L1041   r1 b041             1010001110 (0654)     [register] using W : rightShift B
L1042   r1 b042             1011110000 (0752)     [other   ] (data) C → data
L1043   r1 b043   atn14      1000101110 (0558)     [register] using W : exchange B C
L1044   r1 b044             0100011011 (0283)     [flow    ] goTo : 70                           → atn18
```

```
p: L1000:  1...11111.                        :  B EXCHANGE C[S]
p: L1001:  1111..1111  -> L1363              :  GO TO TAN13
p: L1002:  11..1.111.                 TAN15 :  A EXCHANGE B[W]
p: L1003:  ..11.....1  -> L1060              :  JSB TNM11
p: L1004:  1.11111...                        :  DATA -> C
p: L1005:  111.1.111.                        :  A EXCHANGE C[W]
p: L1006:  ..11.....1  -> L1060              :  JSB TNM11
p: L1007:  1.11111...                        :  DATA -> C
p: L1010:  111.1.111.                        :  A EXCHANGE C[W]
p: L1011:  1..1.1.1..                 TANX   :  IF S9 # 1
p: L1012:  ....11..11  -> L1014              :       THEN GO TO TAN16
p: L1013:  111.1.111.                        :  A EXCHANGE C[W]
p: L1014:  .1.1.1.1..                 TAN16  :  IF S5 # 1
p: L1015:  ...11.1.11  -> L1032              :       THEN GO TO ASN12
p: L1016:  ...111111.                        :  IF C[S] >= 1
p: L1017:  ...1...111  -> L1021              :       THEN GO TO TAN17
p: L1020:  1...1..1..                        :  0 -> S8
p: L1021:  ..11.1111.                 TAN17  :  0 -> C[S]
p: L1022:  1.1..11..1  -> L1246              :  JSB DIV11
p: L1023:  1.1111....                 ASN11  :  C -> DATA
p: L1024:  1.1..1.1.1  -> L1245              :  JSB MPY11
p: L1025:  1..11....1  -> L1230              :  JSB ADD10
p: L1026:  ..1.11.1.1  -> L1055              :  JSB SQT11
p: L1027:  1.11111...                        :  DATA -> C
p: L1030:  .....1....  -> L0031   *****      :  SELECT ROM 0
p: L1031:  111.1.111.                 ASN1Z0 :  A EXCHANGE C[W]
p: L1032:  1.1..11..1  -> L1246       ASN12  :  JSB DIV11
p: L1033:  1.1..1.1..                        :  IF S10 # 1
p: L1034:  11.11.1.11  -> L1332              :       THEN GO TO RTN12
p: L1035:  1.111.111.                 ATN11  :  0 -> A[W]
p: L1036:  11111...1.                        :  A + 1 -> A[P]
p: L1037:  .1..1..11.                        :  A -> B[M]
p: L1040:  111.1..11.                        :  A EXCHANGE C[M]
p: L1041:  .1.11.1.1.                 ATN12  :  C - 1 -> C[X]
p: L1042:  1.1..1..1.                        :  SHIFT RIGHT B[WP]
p: L1043:  .11.111.1.                        :  IF C[XS] = 0
p: L1044:  ..1....111  -> L1041              :       THEN GO TO ATN12
p: L1045:  1.11.1..1.                 ATN13  :  SHIFT RIGHT A[WP]
p: L1046:  .1111.1.1.                        :  C + 1 -> C[X]
p: L1047:  ..1..1.111  -> L1045              :  IF NO CARRY GO TO ATN13
p: L1050:  1.11..111.                        :  SHIFT RIGHT A[W]
p: L1051:  1.1...111.                        :  SHIFT RIGHT B[W]
p: L1052:  1.1111....                        :  C -> DATA
p: L1053:  1...1.111.                 ATN14  :  B EXCHANGE C[W]
p: L1054:  .1...11.11  -> L1106              :  GO TO ATN18
```

```
L1045   r1 b045   sqt11      1000101110 (0558)     [register] using W : exchange B C                                  p: L1055:  1...1.111.                    SQT11 :  B EXCHANGE C[W]
L1046   r1 b046              0100001100 (0268)     [pointer ] value → P : value 4                                     p: L1056:  .1....11..                           :  4 -> P
L1047   r1 b047              1101111011 (0891)     [flow    ] goTo : 222                              → sqt14          p: L1057:  11.1111.11  -> L1336           :  GO TO SQT14
L1048   r1 b048   tnm11      1011110000 (0752)     [other   ] (data) C → data                                         p: L1060:  1.1111....                    TNM11 :  C -> DATA
L1049   r1 b049              1110101110 (0942)     [register] using W : exchange A C                                  p: L1061:  111.1.111.                           :  A EXCHANGE C[W]
L1050   r1 b050              0110100010 (0418)     [register] using P : if C = 0                                      p: L1062:  .11.1...1.                           :  IF C[P] = 0
L1051   r1 b051              0011010111 (0215)     [flow    ]    (then) goTo : 53                     → tnm12          p: L1063:  ..11.1.111  -> L1065           :       THEN GO TO TNM12
L1052   r1 b052              0010101110 (0174)     [register] using W : 0 − C → C                                     p: L1064:  ..1.1.111.                           :  0 − C -> C[W]
L1053   r1 b053   tnm12      0110001110 (0398)     [register] using W : C → A                                         p: L1065:  .11...111.                    TNM12 :  C -> A[W]
L1054   r1 b054              0010001010 (0138)     [register] using X : B → C                                         p: L1066:  ..1...1.1.                           :  B -> C[X]
L1055   r1 b055              1100101111 (0815)     [flow    ] goTo : 203                              → add15          p: L1067:  11..1.1111  -> L1313           :  GO TO ADD15
L1056   r1 b056   tanxz0     0000100111 (0039)     [flow    ] goTo : 9                                → tanx           p: L1070:  ....1..111  -> L1011           TANXZ0:  GO TO TANX
L1057   r1 b057   tploxj     0000010000 (0016)     [rom     ] selectRom : 0                                    p: L1071:  .....1....  -> L0072   *****  TPLOXJ:  SELECT ROM 0
L1058   r1 b058   sin12      0101010100 (0340)     [status  ] if statusBit ≠ 1 : bit 5                                 p: L1072:  .1.1.1.1..                    SIN12 :  IF S5 # 1
L1059   r1 b059              0001110111 (0119)     [flow    ]    (then) goTo : 29                     → atn11          p: L1073:  ...111.111  -> L1035           :       THEN GO TO ATN11
L1060   r1 b060              0011111110 (0254)     [register] using S : 0 − C − 1 → C                                 p: L1074:  ..1111111.                           :  0 − C − 1 -> C[S]
L1061   r1 b061              1110111110 (0958)     [register] using S : exchange A C                                  p: L1075:  111.11111.                           :  A EXCHANGE C[S]
L1062   r1 b062              0001001111 (0079)     [flow    ] goTo : 19                               → asn11          p: L1076:  ...1..1111  -> L1023           :  GO TO ASN11
L1063   r1 b063   atn15      1010010010 (0658)     [register] using WP : rightShift B                                 p: L1077:  1.1..1..1.                    ATN15 :  SHIFT RIGHT B[WP]
L1064   r1 b064   atn16      1101111110 (0894)     [register] using S : decrement A                                   p: L1100:  11.111111.                    ATN16 :  A − 1 -> A[S]
L1065   r1 b065              0011111111 (0255)     [flow    ] (if no carry then) goTo : 63           → atn15          p: L1101:  ..11111111  -> L1077           :  IF NO CARRY GO TO ATN15
L1066   r1 b066              0111111110 (0510)     [register] using S : increment C                                   p: L1102:  .11111111.                           :  C + 1 -> C[S]
L1067   r1 b067              1100110010 (0818)     [register] using WP : exchange A B                                 p: L1103:  11..11..1.                           :  A EXCHANGE B[WP]
L1068   r1 b068              0111010010 (0466)     [register] using WP : A + C → C                                    p: L1104:  .111.1..1.                           :  A + C -> C[WP]
L1069   r1 b069              1100101110 (0814)     [register] using W : exchange A B                                  p: L1105:  11..1.111.                           :  A EXCHANGE B[W]
L1070   r1 b070   atn18      0100101110 (0302)     [register] using W : A → B                                         p: L1106:  .1..1.111.                    ATN18 :  A -> B[W]
L1071   r1 b071              1101010010 (0850)     [register] using WP : A − C → A                                    p: L1107:  11.1.1..1.                           :  A − C -> A[WP]
L1072   r1 b072              0100000011 (0259)     [flow    ] (if no carry then) goTo : 64           → atn16          p: L1110:  .1......11  -> L1100           :  IF NO CARRY GO TO ATN16
L1073   r1 b073              1110101110 (0942)     [register] using W : exchange A C                                  p: L1111:  111.1.111.                           :  A EXCHANGE C[W]
L1074   r1 b074              1011111000 (0760)     [other   ] (data) data → C                                         p: L1112:  1.11111...                           :  DATA -> C
L1075   r1 b075              1001001110 (0590)     [register] using W : rightShift C                                  p: L1113:  1..1..111.                           :  SHIFT RIGHT C[W]
L1076   r1 b076              1110111110 (0958)     [register] using S : exchange A C                                  p: L1114:  111.11111.                           :  A EXCHANGE C[S]
L1077   r1 b077              1100101110 (0814)     [register] using W : exchange A B                                  p: L1115:  11..1.111.                           :  A EXCHANGE B[W]
L1078   r1 b078              0100010010 (0274)     [register] using WP : leftShift A                                  p: L1116:  .1...1..1.                           :  SHIFT LEFT A[WP]
L1079   r1 b079              1011110000 (0752)     [other   ] (data) C → data                                         p: L1117:  1.1111....                           :  C -> DATA
L1080   r1 b080              1111111110 (1022)     [register] using S : increment A                                   p: L1120:  111111111.                           :  A + 1 -> A[S]
L1081   r1 b081              1111111110 (1022)     [register] using S : increment A                                   p: L1121:  111111111.                           :  A + 1 -> A[S]
L1082   r1 b082              0010101111 (0175)     [flow    ] (if no carry then) goTo : 43           → atn14          p: L1122:  ..1.1.1111  -> L1053           :  IF NO CARRY GO TO ATN14
L1083   r1 b083              0011001110 (0206)     [register] using W : clear c                                       p: L1123:  ..11..111.                           :  0 -> C[W]
L1084   r1 b084              0000101010 (0042)     [register] using X : clear B                                       p: L1124:  ....1.1.1.                           :  0 -> B[X]
L1085   r1 b085              1011010110 (0726)     [register] using MS : rightShift A                                 p: L1125:  1.11.1.11.                           :  SHIFT RIGHT A[MS]
L1086   r1 b086              1011001001 (0713)     [flow    ] jumpSub : 178                           → div14          p: L1126:  1.11..1..1  -> L1262           :  JSB DIV14
L1087   r1 b087              1101100010 (0866)     [register] using P : decrement A                                   p: L1127:  11.11...1.                           :  A − 1 -> A[P]
L1088   r1 b088              1011111000 (0760)     [other   ] (data) data → C                                         p: L1130:  1.11111...                           :  DATA -> C
L1089   r1 b089              0100001100 (0268)     [pointer ] value → P : value 4                                     p: L1131:  .1....11..                           :  4 -> P
```

```
L1090   r1 b090   atn17   1010010001 (0657)   [flow    ] jumpSub : 164                    → pqo13
L1091   r1 b091           0110001100 (0396)   [pointer ] value → P : value 6
L1092   r1 b092           1001101101 (0621)   [flow    ] jumpSub : 155                    → pmu11
L1093   r1 b093           1000001100 (0524)   [pointer ] value → P : value 8
L1094   r1 b094           1001101101 (0621)   [flow    ] jumpSub : 155                    → pmu11
L1095   r1 b095           0010001100 (0140)   [pointer ] value → P : value 2
L1096   r1 b096           1000011000 (0536)   [constant] loadConstant : 8
L1097   r1 b097           1010001100 (0652)   [pointer ] value → P : value 10
L1098   r1 b098           1001101101 (0621)   [flow    ] jumpSub : 155                    → pmu11
L1099   r1 b099           1000111001 (0569)   [flow    ] jumpSub : 142                    → atcd1
L1100   r1 b100           1001101101 (0621)   [flow    ] jumpSub : 155                    → pmu11
L1101   r1 b101           1100110001 (0817)   [flow    ] jumpSub : 204                    → atc1
L1102   r1 b102           0100001110 (0270)   [register] using W : leftShift A
L1103   r1 b103           1001101101 (0621)   [flow    ] jumpSub : 155                    → pmu11
L1104   r1 b104           0010001110 (0142)   [register] using W : B → C
L1105   r1 b105           1100101101 (0813)   [flow    ] jumpSub : 203                    → add15
L1106   r1 b106   tan12   1100110001 (0817)   [flow    ] jumpSub : 204                    → atc1
L1107   r1 b107           1010101110 (0686)   [register] using W : C + C → C
L1108   r1 b108           1010010100 (0660)   [status  ] if statusBit ≠ 1 : bit 10
L1109   r1 b109           0111010111 (0471)   [flow    ]    (then) goTo : 117             → rom0
L1110   r1 b110           1001010100 (0596)   [status  ] if statusBit ≠ 1 : bit 9
L1111   r1 b111           0111010111 (0471)   [flow    ]    (then) goTo : 117             → rom0
L1112   r1 b112           1110101110 (0942)   [register] using W : exchange A C
L1113   r1 b113           0011111110 (0254)   [register] using S : 0 − C − 1 → C
L1114   r1 b114           1001101001 (0617)   [flow    ] jumpSub : 154                    → add11
L1115   r1 b115           1100110001 (0817)   [flow    ] jumpSub : 204                    → atc1
L1116   r1 b116           1010101110 (0686)   [register] using W : C + C → C
L1117   r1 b117   rom0     0000010000 (0016)   [rom     ] selectRom : 0
L1118   r1 b118   lpi11    1100110001 (0817)   [flow    ] jumpSub : 204                    → atc1
L1119   r1 b119           1010101110 (0686)   [register] using W : C + C → C
L1120   r1 b120           1010101110 (0686)   [register] using W : C + C → C
L1121   r1 b121           1001010101 (0597)   [flow    ] (if no carry then) jumpSub : 149   → rtn11
L1122   r1 b122           1010101110 (0686)   [register] using W : C + C → C
L1123   r1 b123           1110101101 (0941)   [flow    ] (if no carry then) jumpSub : 235   → pre11
L1124   r1 b124           1100110001 (0817)   [flow    ] jumpSub : 204                    → atc1
L1125   r1 b125           1010001100 (0652)   [pointer ] value → P : value 10
L1126   r1 b126           1001110001 (0625)   [flow    ] jumpSub : 156                    → pqo11
L1127   r1 b127           1000111001 (0569)   [flow    ] jumpSub : 142                    → atcd1
L1128   r1 b128           1000001100 (0524)   [pointer ] value → P : value 8
L1129   r1 b129           1001110101 (0629)   [flow    ] jumpSub : 157                    → pqo12
L1130   r1 b130           0010001100 (0140)   [pointer ] value → P : value 2
L1131   r1 b131           1000011000 (0536)   [constant] loadConstant : 8
L1132   r1 b132           0110001100 (0396)   [pointer ] value → P : value 6
L1133   r1 b133           1001110001 (0625)   [flow    ] jumpSub : 156                    → pqo11
L1134   r1 b134           0100001100 (0268)   [pointer ] value → P : value 4
```

```
p: L1132:  1.1..1...1  -> L1244          ATN17 :  JSB PQO13
p: L1133:  .11...11..                          :  6 -> P
p: L1134:  1..11.11.1  -> L1233                :  JSB PMU11
p: L1135:  1.....11..                          :  8 -> P
p: L1136:  1..11.11.1  -> L1233                :  JSB PMU11
p: L1137:  ..1...11..                          :  2 -> P
p: L1140:  1....11...                          :  LOAD CONSTANT 8
p: L1141:  1.1...11..                          :  10 -> P
p: L1142:  1..11.11.1  -> L1233                :  JSB PMU11
p: L1143:  1...111..1  -> L1216                :  JSB ATCD1
p: L1144:  1..11.11.1  -> L1233                :  JSB PMU11
p: L1145:  11..11...1  -> L1314                :  JSB ATC1
p: L1146:  .1....111.                          :  SHIFT LEFT A[W]
p: L1147:  1..11.11.1  -> L1233                :  JSB PMU11
p: L1150:  ..1...111.                          :  B -> C[W]
p: L1151:  11..1.11.1  -> L1313                :  JSB ADD15
p: L1152:  11..11...1  -> L1314          TAN12 :  JSB ATC1
p: L1153:  1.1.1.111.                          :  C + C -> C[W]
p: L1154:  1.1..1.1..                          :  IF S10 # 1
p: L1155:  .111.1.111  -> L1165                :       THEN GO TO ROM0
p: L1156:  1..1.1.1..                          :  IF S9 # 1
p: L1157:  .111.1.111  -> L1165                :       THEN GO TO ROM0
p: L1160:  111.1.111.                          :  A EXCHANGE C[W]
p: L1161:  ..1111111.                          :  0 − C − 1 -> C[S]
p: L1162:  1..11.1..1  -> L1232                :  JSB ADD11
p: L1163:  11..11...1  -> L1314                :  JSB ATC1
p: L1164:  1.1.1.111.                          :  C + C -> C[W]
p: L1165:  .....1....  -> L0166   *****  ROM0  :  SELECT ROM 0
p: L1166:  11..11...1  -> L1314          LPI11 :  JSB ATC1
p: L1167:  1.1.1.111.                          :  C + C -> C[W]
p: L1170:  1.1.1.111.                          :  C + C -> C[W]
p: L1171:  1..1.1.1.1  -> L1225                :  JSB RTN11
p: L1172:  1.1.1.111.                          :  C + C -> C[W]
p: L1173:  111.1.11.1  -> L1353                :  JSB PRE11
p: L1174:  11..11...1  -> L1314                :  JSB ATC1
p: L1175:  1.1...11..                          :  10 -> P
p: L1176:  1..111...1  -> L1234                :  JSB PQO11
p: L1177:  1...111..1  -> L1216                :  JSB ATCD1
p: L1200:  1.....11..                          :  8 -> P
p: L1201:  1..111.1.1  -> L1235                :  JSB PQO12
p: L1202:  ..1...11..                          :  2 -> P
p: L1203:  1....11...                          :  LOAD CONSTANT 8
p: L1204:  .11...11..                          :  6 -> P
p: L1205:  1..111...1  -> L1234                :  JSB PQO11
p: L1206:  .1....11..                          :  4 -> P
```

```
L1135   r1 b135             1001110001 (0625)     [flow    ] jumpSub : 156                    → pqo11
L1136   r1 b136             1001110001 (0625)     [flow    ] jumpSub : 156                    → pqo11
L1137   r1 b137             1100101110 (0814)     [register] using W : exchange A B
L1138   r1 b138             1001001110 (0590)     [register] using W : rightShift C
L1139   r1 b139             1101001100 (0844)     [pointer ] value → P : value 13
L1140   r1 b140             0101011000 (0344)     [constant] loadConstant : 5
L1141   r1 b141             1111110011 (1011)     [flow    ] goTo : 252                       → tan14
L1142   r1 b142   atcd1      0110001100 (0396)     [pointer ] value → P : value 6
L1143   r1 b143             1000011000 (0536)     [constant] loadConstant : 8
L1144   r1 b144             0110011000 (0408)     [constant] loadConstant : 6
L1145   r1 b145             0101011000 (0344)     [constant] loadConstant : 5
L1146   r1 b146             0010011000 (0152)     [constant] loadConstant : 2
L1147   r1 b147             0100011000 (0280)     [constant] loadConstant : 4
L1148   r1 b148             1001011000 (0600)     [constant] loadConstant : 9
L1149   r1 b149   rtn11      0001010100 (0084)     [status  ] if statusBit ≠ 1 : bit 1
L1150   r1 b150             1101101011 (0875)     [flow    ]    (then) goTo : 218             → rtn12
L1151   r1 b151             0000110000 (0048)     [other   ] (flow) return
L1152   r1 b152   add10      1011101110 (0750)     [register] using W : clear A
L1153   r1 b153             1111100010 (0994)     [register] using P : increment A
L1154   r1 b154   add11      0000010000 (0016)     [rom     ] selectRom : 0
L1155   r1 b155   pmu11      0100010000 (0272)     [rom     ] selectRom : 2
L1156   r1 b156   pqo11      0100001110 (0270)     [register] using W : leftShift A
L1157   r1 b157   pqo12      1010010110 (0662)     [register] using MS : rightShift B
L1158   r1 b158             1000101110 (0558)     [register] using W : exchange B C
L1159   r1 b159             1010000111 (0647)     [flow    ] goTo : 161                       → pqo16
L1160   r1 b160   pqo15      0111111110 (0510)     [register] using S : increment C
L1161   r1 b161   pqo16      1100001110 (0782)     [register] using W : A − B → A
L1162   r1 b162             1010000011 (0643)     [flow    ] (if no carry then) goTo : 160   → pqo15
L1163   r1 b163             1110001110 (0910)     [register] using W : A + B → A
L1164   r1 b164   pqo13      0100010000 (0272)     [rom     ] selectRom : 2
L1165   r1 b165   mpy11      0100010000 (0272)     [rom     ] selectRom : 2
L1166   r1 b166   div11      0101001010 (0330)     [register] using X : A − C → C
L1167   r1 b167             0100010000 (0272)     [rom     ] selectRom : 2
L1168   r1 b168   sqt15      0111100010 (0482)     [register] using P : increment C
L1169   r1 b169   sqt16      1101001110 (0846)     [register] using W : A − C → A
L1170   r1 b170             1010100011 (0675)     [flow    ] (if no carry then) goTo : 168   → sqt15
L1171   r1 b171             1111001110 (0974)     [register] using W : A + C → A
L1172   r1 b172             0100001110 (0270)     [register] using W : leftShift A
L1173   r1 b173             0000011100 (0028)     [pointer ] decrement P
L1174   r1 b174   sqt17      1001010010 (0594)     [register] using WP : rightShift C
L1175   r1 b175             0000101100 (0044)     [pointer ] if P ≠ value : value 0
L1176   r1 b176             1010100111 (0679)     [flow    ]    (then) goTo : 169             → sqt16
L1177   r1 b177             0011010111 (0215)     [flow    ] goTo : 53                        → tnm12
L1178   r1 b178   div14      0111100010 (0482)     [register] using P : increment C
L1179   r1 b179   div15      1100010110 (0790)     [register] using MS : A − B → A
```

```
p: L1207:  1..111...1  -> L1234              :  JSB PQO11
p: L1210:  1..111...1  -> L1234              :  JSB PQO11
p: L1211:  11..1.111.                        :  A EXCHANGE B[W]
p: L1212:  1..1..111.                        :  SHIFT RIGHT C[W]
p: L1213:  11.1..11..                        :  13 -> P
p: L1214:  .1.1.11...                        :  LOAD CONSTANT 5
p: L1215:  111111..11  -> L1374              :  GO TO TAN14
p: L1216:  .11...11..                  ATCD1 :  6 -> P
p: L1217:  1....11...                        :  LOAD CONSTANT 8
p: L1220:  .11..11...                        :  LOAD CONSTANT 6
p: L1221:  .1.1.11...                        :  LOAD CONSTANT 5
p: L1222:  ..1..11...                        :  LOAD CONSTANT 2
p: L1223:  .1...11...                        :  LOAD CONSTANT 4
p: L1224:  1..1.11...                        :  LOAD CONSTANT 9
p: L1225:  ...1.1.1..                  RTN11 :  IF S1 # 1
p: L1226:  11.11.1.11  -> L1332              :       THEN GO TO RTN12
p: L1227:  ....11....                        :  RETURN
p: L1230:  1.111.111.                  ADD10 :  0 -> A[W]
p: L1231:  11111...1.                        :  A + 1 -> A[P]
p: L1232:  .....1....  -> L0233   *****  ADD11 :  SELECT ROM 0
p: L1233:  .1...1....  -> L2234   *****  PMU11 :  SELECT ROM 2
p: L1234:  .1....111.                  PQO11 :  SHIFT LEFT A[W]
p: L1235:  1.1..1.11.                  PQO12 :  SHIFT RIGHT B[MS]
p: L1236:  1...1.111.                        :  B EXCHANGE C[W]
p: L1237:  1.1....111  -> L1241              :  GO TO PQO16
p: L1240:  .11111111.                  PQO15 :  C + 1 -> C[S]
p: L1241:  11....111.                  PQO16 :  A − B -> A[W]
p: L1242:  1.1.....11  -> L1240              :  IF NO CARRY GO TO PQO15
p: L1243:  111...111.                        :  A + B -> A[W]
p: L1244:  .1...1....  -> L2245   *****  PQO13 :  SELECT ROM 2
p: L1245:  .1...1....  -> L2246   *****  MPY11 :  SELECT ROM 2
p: L1246:  .1.1..1.1.                  DIV11 :  A − C -> C[X]
p: L1247:  .1...1....  -> L2250   *****        :  SELECT ROM 2
p: L1250:  .1111...1.                  SQT15 :  C + 1 -> C[P]
p: L1251:  11.1..111.                  SQT16 :  A − C -> A[W]
p: L1252:  1.1.1...11  -> L1250              :  IF NO CARRY GO TO SQT15
p: L1253:  1111..111.                        :  A + C -> A[W]
p: L1254:  .1....111.                        :  SHIFT LEFT A[W]
p: L1255:  .....111..                        :  P − 1 -> P
p: L1256:  1..1.1..1.                  SQT17 :  SHIFT RIGHT C[WP]
p: L1257:  ....1.11..                        :  IF P # 0
p: L1260:  1.1.1..111  -> L1251              :       THEN GO TO SQT16
p: L1261:  ..11.1.111  -> L1065              :  GO TO TNM12
p: L1262:  .1111...1.                  DIV14 :  C + 1 -> C[P]
p: L1263:  11...1.11.                  DIV15 :  A − B -> A[MS]
```

```
L1180   r1 b180            1011001011 (0715)   [flow    ] (if no carry then) goTo : 178      → div14
L1181   r1 b181            1110010110 (0918)   [register] using MS : A + B → A
L1182   r1 b182            0100010110 (0278)   [register] using MS : leftShift A
L1183   r1 b183   div16    0000011100 (0028)   [pointer ] decrement P
L1184   r1 b184            0000101100 (0044)   [pointer ] if P ≠ value : value 0
L1185   r1 b185            1011001111 (0719)   [flow    ]    (then) goTo : 179               → div15
L1186   r1 b186            0011010111 (0215)   [flow    ] goTo : 53                          → tnm12
L1187   r1 b187   sqt12    0000011100 (0028)   [pointer ] decrement P
L1188   r1 b188            1110010110 (0918)   [register] using MS : A + B → A
L1189   r1 b189            1101101111 (0879)   [flow    ] (if no carry then) goTo : 219      → sqt18
L1190   r1 b190            0000010000 (0016)   [rom     ] selectRom : 0
L1191   r1 b191   add12    0101111010 (0378)   [register] using XS : decrement C
L1192   r1 b192            0101111010 (0378)   [register] using XS : decrement C
L1193   r1 b193            1011101010 (0746)   [register] using X : clear A
L1194   r1 b194            1101011110 (0862)   [register] using S : A − C → A
L1195   r1 b195            1001111110 (0638)   [register] using S : if A ≥ 1
L1196   r1 b196            1100011011 (0795)   [flow    ]    (then) goTo : 198               → add13
L1197   r1 b197            0100010000 (0272)   [rom     ] selectRom : 2
L1198   r1 b198   add13    1000000110 (0518)   [register] using M : if A ≥ B
L1199   r1 b199            1100101011 (0811)   [flow    ]    (then) goTo : 202               → add14
L1200   r1 b200            0011111110 (0254)   [register] using S : 0 − C − 1 → C
L1201   r1 b201            1100101110 (0814)   [register] using W : exchange A B
L1202   r1 b202   add14    1100001110 (0782)   [register] using W : A − B → A
L1203   r1 b203   add15    0100010000 (0272)   [rom     ] selectRom : 2
L1204   r1 b204   atc1     0011001110 (0206)   [register] using W : clear c
L1205   r1 b205            1011001100 (0716)   [pointer ] value → P : value 11
L1206   r1 b206            0111011000 (0472)   [constant] loadConstant : 7
L1207   r1 b207            1000011000 (0536)   [constant] loadConstant : 8
L1208   r1 b208            0101011000 (0344)   [constant] loadConstant : 5
L1209   r1 b209            0011011000 (0216)   [constant] loadConstant : 3
L1210   r1 b210            1001011000 (0600)   [constant] loadConstant : 9
L1211   r1 b211            1000011000 (0536)   [constant] loadConstant : 8
L1212   r1 b212            0001011000 (0088)   [constant] loadConstant : 1
L1213   r1 b213            0110011000 (0408)   [constant] loadConstant : 6
L1214   r1 b214            0011011000 (0216)   [constant] loadConstant : 3
L1215   r1 b215            0101011000 (0344)   [constant] loadConstant : 5
L1216   r1 b216            0000010000 (0016)   [rom     ] selectRom : 0
L1217   r1 b217            0000110000 (0048)   [other   ] (flow) return
L1218   r1 b218   rtn12    0000010000 (0016)   [rom     ] selectRom : 0
L1219   r1 b219   sqt18    1110001010 (0906)   [register] using X : A + B → A
L1220   r1 b220            1101111011 (0891)   [flow    ] (if no carry then) goTo : 222      → sqt14
L1221   r1 b221            0101100010 (0354)   [register] using P : decrement C
L1222   r1 b222   sqt14    0111111110 (0510)   [register] using S : increment C
L1223   r1 b223            0000101100 (0044)   [pointer ] if P ≠ value : value 0
L1224   r1 b224            1011101111 (0751)   [flow    ]    (then) goTo : 187               → sqt12

p: L1264:  1.11..1.11  -> L1262                :  IF NO CARRY GO TO DIV14
p: L1265:  111..1.11.                          :  A + B -> A[MS]
p: L1266:  .1...1.11.                          :  SHIFT LEFT A[MS]
p: L1267:  .....111..                  DIV16 :  P − 1 -> P
p: L1270:  ....1.11..                          :  IF P # 0
p: L1271:  1.11..1111  -> L1263                :       THEN GO TO DIV15
p: L1272:  ..11.1.111  -> L1065                :  GO TO TNM12
p: L1273:  .....111..                  SQT12 :  P − 1 -> P
p: L1274:  111..1.11.                          :  A + B -> A[MS]
p: L1275:  11.11.1111  -> L1333                :  IF NO CARRY GO TO SQT18
p: L1276:  .....1....  -> L0277   *****        :  SELECT ROM 0
p: L1277:  .1.1111.1.                  ADD12 :  C − 1 -> C[XS]
p: L1300:  .1.1111.1.                          :  C − 1 -> C[XS]
p: L1301:  1.111.1.1.                          :  0 -> A[X]
p: L1302:  11.1.1111.                          :  A − C -> A[S]
p: L1303:  1..111111.                          :  IF A[S] >= 1
p: L1304:  11...11.11  -> L1306                :       THEN GO TO ADD13
p: L1305:  .1...1....  -> L2306   *****        :  SELECT ROM 2
p: L1306:  1......11.                  ADD13 :  IF A >= B[M]
p: L1307:  11..1.1.11  -> L1312                :       THEN GO TO ADD14
p: L1310:  ..1111111.                          :  0 − C − 1 -> C[S]
p: L1311:  11..1.111.                          :  A EXCHANGE B[W]
p: L1312:  11....111.                  ADD14 :  A − B -> A[W]
p: L1313:  .1...1....  -> L2314   *****  ADD15 :  SELECT ROM 2
p: L1314:  ..11..111.                  ATC1  :  0 -> C[W]
p: L1315:  1.11..11..                          :  11 -> P
p: L1316:  .111.11...                          :  LOAD CONSTANT 7
p: L1317:  1....11...                          :  LOAD CONSTANT 8
p: L1320:  .1.1.11...                          :  LOAD CONSTANT 5
p: L1321:  ..11.11...                          :  LOAD CONSTANT 3
p: L1322:  1..1.11...                          :  LOAD CONSTANT 9
p: L1323:  1....11...                          :  LOAD CONSTANT 8
p: L1324:  ...1.11...                          :  LOAD CONSTANT 1
p: L1325:  .11..11...                          :  LOAD CONSTANT 6
p: L1326:  ..11.11...                          :  LOAD CONSTANT 3
p: L1327:  .1.1.11...                          :  LOAD CONSTANT 5
p: L1330:  .....1....  -> L0331   *****        :  SELECT ROM 0
p: L1331:  ....11....                          :  RETURN
p: L1332:  .....1....  -> L0333   *****  RTN12 :  SELECT ROM 0
p: L1333:  111...1.1.                  SQT18 :  A + B -> A[X]
p: L1334:  11.1111.11  -> L1336                :  IF NO CARRY GO TO SQT14
p: L1335:  .1.11...1.                          :  C − 1 -> C[P]
p: L1336:  .11111111.                  SQT14 :  C + 1 -> C[S]
p: L1337:  ....1.11..                          :  IF P # 0
p: L1340:  1.111.1111  -> L1273                :       THEN GO TO SQT12
```

```
L1225   r1 b225              1110101010 (0938)     [register] using X : exchange A C                                  p: L1341:  111.1.1.1.                        :  A EXCHANGE C[X]
L1226   r1 b226              1011101010 (0746)     [register] using X : clear A                                       p: L1342:  1.111.1.1.                        :  0 -> A[X]
L1227   r1 b227              0001100010 (0098)     [register] using P : if C ≥ 1                                      p: L1343:  ...11...1.                        :  IF C[P] >= 1
L1228   r1 b228              1110011011 (0923)     [flow    ]    (then) goTo : 230                 → sqt13             p: L1344:  111..11.11  -> L1346               :       THEN GO TO SQT13
L1229   r1 b229              1011001110 (0718)     [register] using W : rightShift A                                  p: L1345:  1.11..111.                        :  SHIFT RIGHT A[W]
L1230   r1 b230   sqt13      1001001110 (0590)     [register] using W : rightShift C                                  p: L1346:  1..1..111.                  SQT13 :  SHIFT RIGHT C[W]
L1231   r1 b231              1000101010 (0554)     [register] using X : exchange B C                                  p: L1347:  1...1.1.1.                        :  B EXCHANGE C[X]
L1232   r1 b232              0011001010 (0202)     [register] using X : clear c                                       p: L1350:  ..11..1.1.                        :  0 -> C[X]
L1233   r1 b233              1100001100 (0780)     [pointer ] value → P : value 12                                    p: L1351:  11....11..                        :  12 -> P
L1234   r1 b234              1010111011 (0699)     [flow    ] goTo : 174                     → sqt17             p: L1352:  1.1.111.11  -> L1256               :  GO TO SQT17
L1235   r1 b235   pre11      0100010000 (0272)     [rom     ] selectRom : 2                                           p: L1353:  .1...1....  -> L2354   *****  PRE11 :  SELECT ROM 2
L1236   r1 b236   tan18      1010010010 (0658)     [register] using WP : rightShift B                                 p: L1354:  1.1..1..1.                  TAN18 :  SHIFT RIGHT B[WP]
L1237   r1 b237              1010010010 (0658)     [register] using WP : rightShift B                                 p: L1355:  1.1..1..1.                        :  SHIFT RIGHT B[WP]
L1238   r1 b238   tan19      0101111110 (0382)     [register] using S : decrement C                                   p: L1356:  .1.111111.                  TAN19 :  C - 1 -> C[S]
L1239   r1 b239              1110110011 (0947)     [flow    ] (if no carry then) goTo : 236  → tan18             p: L1357:  111.11..11  -> L1354               :  IF NO CARRY GO TO TAN18
L1240   r1 b240              0111010010 (0466)     [register] using WP : A + C → C                                    p: L1360:  .111.1..1.                        :  A + C -> C[WP]
L1241   r1 b241              1100010010 (0786)     [register] using WP : A − B → A                                    p: L1361:  11...1..1.                        :  A - B -> A[WP]
L1242   r1 b242              1000110010 (0562)     [register] using WP : exchange B C                                 p: L1362:  1...11..1.                        :  B EXCHANGE C[WP]
L1243   r1 b243   tan13      0010001110 (0142)     [register] using W : B → C                                         p: L1363:  ..1...111.                  TAN13 :  B -> C[W]
L1244   r1 b244              1101111110 (0894)     [register] using S : decrement A                                   p: L1364:  11.111111.                        :  A - 1 -> A[S]
L1245   r1 b245              1110111011 (0955)     [flow    ] (if no carry then) goTo : 238  → tan19             p: L1365:  111.111.11  -> L1356               :  IF NO CARRY GO TO TAN19
L1246   r1 b246              1110111110 (0958)     [register] using S : exchange A C                                  p: L1366:  111.11111.                        :  A EXCHANGE C[S]
L1247   r1 b247              1011111000 (0760)     [other   ] (data) data → C                                         p: L1367:  1.11111...                        :  DATA -> C
L1248   r1 b248              1110101110 (0942)     [register] using W : exchange A C                                  p: L1370:  111.1.111.                        :  A EXCHANGE C[W]
L1249   r1 b249              0000011110 (0030)     [register] using S : if B = 0                                      p: L1371:  .....1111.                        :  IF B[S] = 0
L1250   r1 b250              0000001011 (0011)     [flow    ]    (then) goTo : 2                   → tan15             p: L1372:  ......1.11  -> L1002               :       THEN GO TO TAN15
L1251   r1 b251              0100001110 (0270)     [register] using W : leftShift A                                   p: L1373:  .1....111.                        :  SHIFT LEFT A[W]
L1252   r1 b252   tan14      1110110010 (0946)     [register] using WP : exchange A C                                 p: L1374:  111.11..1.                  TAN14 :  A EXCHANGE C[WP]
L1253   r1 b253              1011110000 (0752)     [other   ] (data) C → data                                         p: L1375:  1.1111....                        :  C -> DATA
L1254   r1 b254              1010010010 (0658)     [register] using WP : rightShift B                                 p: L1376:  1.1..1..1.                        :  SHIFT RIGHT B[WP]
L1255   r1 b255              0101111110 (0382)     [register] using S : decrement C                                   p: L1377:  .1.111111.                        :  C - 1 -> C[S]
```

```
L2000   r2 b000   err21      1100010000 (0784)    [rom     ] selectRom : 6
L2001   r2 b001   ln24       1100111110 (0830)    [register] using S : exchange A B
L2002   r2 b002              1111111110 (1022)    [register] using S : increment A
L2003   r2 b003              1001010110 (0598)    [register] using MS : rightShift C
L2004   r2 b004              0100010010 (0274)    [register] using WP : leftShift A
L2005   r2 b005              0001001011 (0075)    [flow    ] goTo : 18                          → ln26
L2006   r2 b006   xty22      0110101000 (0424)    [other   ] (stack) stack → A
L2007   r2 b007              1010011001 (0665)    [flow    ] jumpSub : 166                      → mpy21
L2008   r2 b008   xty21      0110001110 (0398)    [register] using W : C → A
L2009   r2 b009              1000010100 (0532)    [status  ] if statusBit ≠ 1 : bit 8
L2010   r2 b010              0100001011 (0267)    [flow    ]    (then) goTo : 66                → exp21
L2011   r2 b011   ln22       1011101110 (0750)    [register] using W : clear A
L2012   r2 b012              1101000110 (0838)    [register] using M : A − C → A
L2013   r2 b013              0000000011 (0003)    [flow    ] (if no carry then) goTo : 0       → err21
L2014   r2 b014              1011001110 (0718)    [register] using W : rightShift A
L2015   r2 b015              0101111110 (0382)    [register] using S : decrement C
L2016   r2 b016              0000000011 (0003)    [flow    ] (if no carry then) goTo : 0       → err21
L2017   r2 b017   ln25       0111111110 (0510)    [register] using S : increment C
L2018   r2 b018   ln26       0100101110 (0302)    [register] using W : A → B
L2019   r2 b019              1001011001 (0601)    [flow    ] jumpSub : 150                     → eca22
L2020   r2 b020              1101100010 (0866)    [register] using P : decrement A
L2021   r2 b021              0001000111 (0071)    [flow    ] (if no carry then) goTo : 17      → ln25
L2022   r2 b022              1100110010 (0818)    [register] using WP : exchange A B
L2023   r2 b023              1110011110 (0926)    [register] using S : A + B → A
L2024   r2 b024              0000000111 (0007)    [flow    ] (if no carry then) goTo : 1       → ln24
L2025   r2 b025              0111001100 (0460)    [pointer ] value → P : value 7
L2026   r2 b026              0110110101 (0437)    [flow    ] jumpSub : 109                     → pqo23
L2027   r2 b027              1000001100 (0524)    [pointer ] value → P : value 8
L2028   r2 b028              1001110101 (0629)    [flow    ] jumpSub : 157                     → pmu22
L2029   r2 b029              1001001100 (0588)    [pointer ] value → P : value 9
L2030   r2 b030              1001110001 (0625)    [flow    ] jumpSub : 156                     → pmu21
L2031   r2 b031              1111111001 (1017)    [flow    ] jumpSub : 254                     → lncd3
L2032   r2 b032              1010001100 (0652)    [pointer ] value → P : value 10
L2033   r2 b033              1001110001 (0625)    [flow    ] jumpSub : 156                     → pmu21
L2034   r2 b034              0111110101 (0501)    [flow    ] jumpSub : 125                     → lncd2
L2035   r2 b035              1011001100 (0716)    [pointer ] value → P : value 11
L2036   r2 b036              1001110001 (0625)    [flow    ] jumpSub : 156                     → pmu21
L2037   r2 b037              1101111101 (0893)    [flow    ] jumpSub : 223                     → lncd1
L2038   r2 b038              1001110001 (0625)    [flow    ] jumpSub : 156                     → pmu21
L2039   r2 b039              1011100101 (0741)    [flow    ] jumpSub : 185                     → lnc2
L2040   r2 b040              1001110001 (0625)    [flow    ] jumpSub : 156                     → pmu21
L2041   r2 b041              1111011001 (0985)    [flow    ] jumpSub : 246                     → lnc10
L2042   r2 b042              1110101110 (0942)    [register] using W : exchange A C
L2043   r2 b043              0101001110 (0334)    [register] using W : A − C → C
L2044   r2 b044              0000011010 (0026)    [register] using XS : if B = 0
```

```
p: L2000:  11...1....  -> L6001   *****  ERR21 :  SELECT ROM 6
p: L2001:  11..11111.                    LN24  :  A EXCHANGE B[S]
p: L2002:  111111111                           :  A + 1 -> A[S]
p: L2003:  1..1.1.11.                          :  SHIFT RIGHT C[MS]
p: L2004:  .1...1..1.                          :  SHIFT LEFT A[WP]
p: L2005:  ...1..1.11  -> L2022                :  GO TO LN26
p: L2006:  .11.1.1...                    XTY22 :  STACK -> A
p: L2007:  1.1..11..1  -> L2246                :  JSB MPY21
p: L2010:  .11...111.                    XTY21 :  C -> A[W]
p: L2011:  1....1.1..                          :  IF S8 # 1
p: L2012:  .1....1.11  -> L2102                :        THEN GO TO EXP21
p: L2013:  1.111.111.                    LN22  :  0 -> A[W]
p: L2014:  11.1...11.                          :  A − C -> A[M]
p: L2015:  ........11  -> L2000                :  IF NO CARRY GO TO ERR21
p: L2016:  1.11..111.                          :  SHIFT RIGHT A[W]
p: L2017:  .1.111111.                          :  C − 1 -> C[S]
p: L2020:  ........11  -> L2000                :  IF NO CARRY GO TO ERR21
p: L2021:  .11111111.                    LN25  :  C + 1 -> C[S]
p: L2022:  .1..1.111.                    LN26  :  A -> B[W]
p: L2023:  1..1.11..1  -> L2226                :  JSB ECA22
p: L2024:  11.11...1.                          :  A − 1 -> A[P]
p: L2025:  ...1...111  -> L2021                :  IF NO CARRY GO TO LN25
p: L2026:  11..11..1.                          :  A EXCHANGE B[WP]
p: L2027:  111..1111.                          :  A + B -> A[S]
p: L2030:  .......111  -> L2001                :  IF NO CARRY GO TO LN24
p: L2031:  .111..11..                          :  7 -> P
p: L2032:  .11.11.1.1  -> L2155                :  JSB PQO23
p: L2033:  1.....11..                          :  8 -> P
p: L2034:  1..111.1.1  -> L2235                :  JSB PMU22
p: L2035:  1..1..11..                          :  9 -> P
p: L2036:  1..111...1  -> L2234                :  JSB PMU21
p: L2037:  1111111..1  -> L2376                :  JSB LNCD3
p: L2040:  1.1...11..                          :  10 -> P
p: L2041:  1..111...1  -> L2234                :  JSB PMU21
p: L2042:  .11111.1.1  -> L2175                :  JSB LNCD2
p: L2043:  1.11..11..                          :  11 -> P
p: L2044:  1..111...1  -> L2234                :  JSB PMU21
p: L2045:  11.11111.1  -> L2337                :  JSB LNCD1
p: L2046:  1..111...1  -> L2234                :  JSB PMU21
p: L2047:  1.111..1.1  -> L2271                :  JSB LNC2
p: L2050:  1..111...1  -> L2234                :  JSB PMU21
p: L2051:  1111.11..1  -> L2366                :  JSB LNC10
p: L2052:  111.1.111.                          :  A EXCHANGE C[W]
p: L2053:  .1.1..111.                          :  A − C -> C[W]
p: L2054:  .....11.1.                          :  IF B[XS] = 0
```

```
L2045   r2 b045          0010111111 (0191)   [flow    ]    (then) goTo : 47                → ln27
L2046   r2 b046          0101001110 (0334)   [register] using W : A − C → C
L2047   r2 b047   ln27    1100101110 (0814)   [register] using W : exchange A B
L2048   r2 b048   ln28    0000011100 (0028)   [pointer ] decrement P
L2049   r2 b049          0100001110 (0270)   [register] using W : leftShift A
L2050   r2 b050          0001101100 (0108)   [pointer ] if P ≠ value : value 1
L2051   r2 b051          0011000011 (0195)   [flow    ]    (then) goTo : 48                → ln28
L2052   r2 b052          1110101110 (0942)   [register] using W : exchange A C
L2053   r2 b053          0110111110 (0446)   [register] using S : if C = 0
L2054   r2 b054          0011100011 (0227)   [flow    ]    (then) goTo : 56                → ln29
L2055   r2 b055          0011100110 (0230)   [register] using M : 0 − C − 1 → C
L2056   r2 b056   ln29    0111101010 (0490)   [register] using X : increment C
L2057   r2 b057          1011001100 (0716)   [pointer ] value → P : value 11
L2058   r2 b058          1100010101 (0789)   [flow    ] jumpSub : 197                 → mpy27
L2059   r2 b059          1001010100 (0596)   [status  ] if statusBit ≠ 1 : bit 9
L2060   r2 b060          0000011011 (0027)   [flow    ]    (then) goTo : 6                 → xty22
L2061   r2 b061          0101010100 (0340)   [status  ] if statusBit ≠ 1 : bit 5
L2062   r2 b062          1001010011 (0595)   [flow    ]    (then) goTo : 148               → rtn21
L2063   r2 b063          1111011001 (0985)   [flow    ] jumpSub : 246                 → lnc10
L2064   r2 b064          1010011101 (0669)   [flow    ] jumpSub : 167                 → mpy22
L2065   r2 b065          1001010011 (0595)   [flow    ] goTo : 148                    → rtn21
L2066   r2 b066   exp21   1111011001 (0985)   [flow    ] jumpSub : 246                 → lnc10
L2067   r2 b067          1110110001 (0945)   [flow    ] jumpSub : 236                 → pre21
L2068   r2 b068          1011100101 (0741)   [flow    ] jumpSub : 185                 → lnc2
L2069   r2 b069          1011001100 (0716)   [pointer ] value → P : value 11
L2070   r2 b070          1001101101 (0621)   [flow    ] jumpSub : 155                 → pqo21
L2071   r2 b071          1101111101 (0893)   [flow    ] jumpSub : 223                 → lncd1
L2072   r2 b072          1010001100 (0652)   [pointer ] value → P : value 10
L2073   r2 b073          1001101101 (0621)   [flow    ] jumpSub : 155                 → pqo21
L2074   r2 b074          0111110101 (0501)   [flow    ] jumpSub : 125                 → lncd2
L2075   r2 b075          1001001100 (0588)   [pointer ] value → P : value 9
L2076   r2 b076          1001101101 (0621)   [flow    ] jumpSub : 155                 → pqo21
L2077   r2 b077          1111111001 (1017)   [flow    ] jumpSub : 254                 → lncd3
L2078   r2 b078          1000001100 (0524)   [pointer ] value → P : value 8
L2079   r2 b079          1001101101 (0621)   [flow    ] jumpSub : 155                 → pqo21
L2080   r2 b080          1001101101 (0621)   [flow    ] jumpSub : 155                 → pqo21
L2081   r2 b081          1001101101 (0621)   [flow    ] jumpSub : 155                 → pqo21
L2082   r2 b082          0110001100 (0396)   [pointer ] value → P : value 6
L2083   r2 b083          1011110010 (0754)   [register] using WP : clear A
L2084   r2 b084          1101001100 (0844)   [pointer ] value → P : value 13
L2085   r2 b085          1000101110 (0558)   [register] using W : exchange B C
L2086   r2 b086          1110101110 (0942)   [register] using W : exchange A C
L2087   r2 b087          0110011000 (0408)   [constant] loadConstant : 6
L2088   r2 b088          1000111011 (0571)   [flow    ] goTo : 142                    → exp23
L2089   r2 b089   pre23   0010010100 (0148)   [status  ] if statusBit ≠ 1 : bit 2

p: L2055:  ..1.111111  -> L2057          :       THEN GO TO LN27
p: L2056:  .1.1..111.                    :  A − C -> C[W]
p: L2057:  11..1.111.                LN27  :  A EXCHANGE B[W]
p: L2060:  .....111..                LN28  :  P − 1 -> P
p: L2061:  .1....111.                      :  SHIFT LEFT A[W]
p: L2062:  ...11.11..                      :  IF P # 1
p: L2063:  ..11....11  -> L2060            :       THEN GO TO LN28
p: L2064:  111.1.111.                      :  A EXCHANGE C[W]
p: L2065:  .11.11111.                      :  IF C[S] = 0
p: L2066:  ..111...11  -> L2070            :       THEN GO TO LN29
p: L2067:  ..111..11.                      :  0 − C − 1 -> C[M]
p: L2070:  .1111.1.1.                LN29  :  C + 1 -> C[X]
p: L2071:  1.11..11..                      :  11 -> P
p: L2072:  11...1.1.1  -> L2305            :  JSB MPY27
p: L2073:  1..1.1.1..                      :  IF S9 # 1
p: L2074:  .....11.11  -> L2006            :       THEN GO TO XTY22
p: L2075:  .1.1.1.1..                      :  IF S5 # 1
p: L2076:  1..1.1..11  -> L2224            :       THEN GO TO RTN21
p: L2077:  1111.11..1  -> L2366            :  JSB LNC10
p: L2100:  1.1..111.1  -> L2247            :  JSB MPY22
p: L2101:  1..1.1..11  -> L2224            :  GO TO RTN21
p: L2102:  1111.11..1  -> L2366      EXP21 :  JSB LNC10
p: L2103:  111.11...1  -> L2354            :  JSB PRE21
p: L2104:  1.111..1.1  -> L2271            :  JSB LNC2
p: L2105:  1.11..11..                      :  11 -> P
p: L2106:  1..11.11.1  -> L2233            :  JSB PQO21
p: L2107:  11.11111.1  -> L2337            :  JSB LNCD1
p: L2110:  1.1...11..                      :  10 -> P
p: L2111:  1..11.11.1  -> L2233            :  JSB PQO21
p: L2112:  .11111.1.1  -> L2175            :  JSB LNCD2
p: L2113:  1..1..11..                      :  9 -> P
p: L2114:  1..11.11.1  -> L2233            :  JSB PQO21
p: L2115:  1111111..1  -> L2376            :  JSB LNCD3
p: L2116:  1.....11..                      :  8 -> P
p: L2117:  1..11.11.1  -> L2233            :  JSB PQO21
p: L2120:  1..11.11.1  -> L2233            :  JSB PQO21
p: L2121:  1..11.11.1  -> L2233            :  JSB PQO21
p: L2122:  .11...11..                      :  6 -> P
p: L2123:  1.1111..1.                      :  0 -> A[WP]
p: L2124:  11.1..11..                      :  13 -> P
p: L2125:  1...1.111.                      :  B EXCHANGE C[W]
p: L2126:  111.1.111.                      :  A EXCHANGE C[W]
p: L2127:  .11..11...                      :  LOAD CONSTANT 6
p: L2130:  1...111.11  -> L2216            :  GO TO EXP23
p: L2131:  ..1..1.1..                PRE23 :  IF S2 # 1
```

```
L2090   r2 b090          0101111011 (0379)   [flow    ]    (then) goTo : 94              → pre24
L2091   r2 b091          1111101010 (1002)   [register] using X : increment A
L2092   r2 b092   pre29  1001111010 (0634)   [register] using XS : if A ≥ 1
L2093   r2 b093          1100001011 (0779)   [flow    ]    (then) goTo : 194             → pre27
L2094   r2 b094   pre24  1100010110 (0790)   [register] using MS : A − B → A
L2095   r2 b095          0101100111 (0359)   [flow    ] (if no carry then) goTo : 89    → pre23
L2096   r2 b096          1110010110 (0918)   [register] using MS : A + B → A
L2097   r2 b097          0100001110 (0270)   [register] using W : leftShift A
L2098   r2 b098          0101101010 (0362)   [register] using X : decrement C
L2099   r2 b099          0101110011 (0371)   [flow    ] (if no carry then) goTo : 92    → pre29
L2100   r2 b100   pre25  1011001110 (0718)   [register] using W : rightShift A
L2101   r2 b101          0011010010 (0210)   [register] using WP : clear c
L2102   r2 b102          1110101010 (0938)   [register] using X : exchange A C
L2103   r2 b103   pre26  0110111110 (0446)   [register] using S : if C = 0
L2104   r2 b104          0110110011 (0435)   [flow    ]    (then) goTo : 108             → pre28
L2105   r2 b105          1100101110 (0814)   [register] using W : exchange A B
L2106   r2 b106          1100001110 (0782)   [register] using W : A − B → A
L2107   r2 b107          0011101110 (0238)   [register] using W : 0 − C − 1 → C
L2108   r2 b108   pre28  1011001110 (0718)   [register] using W : rightShift A
L2109   r2 b109   pqo23  1000101110 (0558)   [register] using W : exchange B C
L2110   r2 b110          0011001110 (0206)   [register] using W : clear c
L2111   r2 b111          0101100110 (0358)   [register] using M : decrement C
L2112   r2 b112          0010010100 (0148)   [status  ] if statusBit ≠ 1 : bit 2
L2113   r2 b113          0111011011 (0475)   [flow    ]    (then) goTo : 118             → pqo28
L2114   r2 b114          0100011000 (0280)   [constant] loadConstant : 4
L2115   r2 b115          0111100110 (0486)   [register] using M : increment C
L2116   r2 b116          0111100111 (0487)   [flow    ] (if no carry then) goTo : 121   → pqo24
L2117   r2 b117   pqo27  0110011000 (0408)   [constant] loadConstant : 6
L2118   r2 b118   pqo28  0001101100 (0108)   [pointer ] if P ≠ value : value 1
L2119   r2 b119          0111010111 (0471)   [flow    ]    (then) goTo : 117             → pqo27
L2120   r2 b120          1001001110 (0590)   [register] using W : rightShift C
L2121   r2 b121   pqo24  1001001110 (0590)   [register] using W : rightShift C
L2122   r2 b122   nrm26  0010010100 (0148)   [status  ] if statusBit ≠ 1 : bit 2
L2123   r2 b123          1001010011 (0595)   [flow    ]    (then) goTo : 148             → rtn21
L2124   r2 b124          0000110000 (0048)   [other   ] (flow) return
L2125   r2 b125   lncd2  0111001100 (0460)   [pointer ] value → P : value 7
L2126   r2 b126   lnc6   0011011000 (0216)   [constant] loadConstant : 3
L2127   r2 b127          0011011000 (0216)   [constant] loadConstant : 3
L2128   r2 b128          0000011000 (0024)   [constant] loadConstant : 0
L2129   r2 b129   lnc7   1000011000 (0536)   [constant] loadConstant : 8
L2130   r2 b130          0101011000 (0344)   [constant] loadConstant : 5
L2131   r2 b131          0000011000 (0024)   [constant] loadConstant : 0
L2132   r2 b132          1001011000 (0600)   [constant] loadConstant : 9
L2133   r2 b133          1110101011 (0939)   [flow    ] goTo : 234                  → lnc9
L2134   r2 b134   exp29  1001011001 (0601)   [flow    ] jumpSub : 150               → eca22
```

```
p: L2132:  .1.1111.11  -> L2136          :       THEN GO TO PRE24
p: L2133:  11111.1.1.                    :  A + 1 -> A[X]
p: L2134:  1..1111.1.                PRE29 :  IF A[XS] >= 1
p: L2135:  11....1.11  -> L2302          :       THEN GO TO PRE27
p: L2136:  11...1.11.                PRE24 :  A − B -> A[MS]
p: L2137:  .1.11..111  -> L2131          :  IF NO CARRY GO TO PRE23
p: L2140:  111..1.11.                    :  A + B -> A[MS]
p: L2141:  .1....111.                    :  SHIFT LEFT A[W]
p: L2142:  .1.11.1.1.                    :  C − 1 -> C[X]
p: L2143:  .1.111..11  -> L2134          :  IF NO CARRY GO TO PRE29
p: L2144:  1.11..111.                PRE25 :  SHIFT RIGHT A[W]
p: L2145:  ..11.1..1.                    :  0 -> C[WP]
p: L2146:  111.1.1.1.                    :  A EXCHANGE C[X]
p: L2147:  .11.11111.                PRE26 :  IF C[S] = 0
p: L2150:  .11.11..11  -> L2154          :       THEN GO TO PRE28
p: L2151:  11..1.111.                    :  A EXCHANGE B[W]
p: L2152:  11....111.                    :  A − B -> A[W]
p: L2153:  ..111.111.                    :  0 − C − 1 -> C[W]
p: L2154:  1.11..111.                PRE28 :  SHIFT RIGHT A[W]
p: L2155:  1...1.111.                PQO23 :  B EXCHANGE C[W]
p: L2156:  ..11..111.                    :  0 -> C[W]
p: L2157:  .1.11..11.                    :  C − 1 -> C[M]
p: L2160:  ..1..1.1..                    :  IF S2 # 1
p: L2161:  .111.11.11  -> L2166          :       THEN GO TO PQO28
p: L2162:  .1...11...                    :  LOAD CONSTANT 4
p: L2163:  .1111..11.                    :  C + 1 -> C[M]
p: L2164:  .1111..111  -> L2171          :  IF NO CARRY GO TO PQO24
p: L2165:  .11..11...                PQO27 :  LOAD CONSTANT 6
p: L2166:  ...11.11..                PQO28 :  IF P # 1
p: L2167:  .111.1.111  -> L2165          :       THEN GO TO PQO27
p: L2170:  1..1..111.                    :  SHIFT RIGHT C[W]
p: L2171:  1..1..111.                PQO24 :  SHIFT RIGHT C[W]
p: L2172:  ..1..1.1..                NRM26 :  IF S2 # 1
p: L2173:  1..1.1..11  -> L2224          :       THEN GO TO RTN21
p: L2174:  ....11....                    :  RETURN
p: L2175:  .111..11..                LNCD2 :  7 -> P
p: L2176:  ..11.11...                LNC6  :  LOAD CONSTANT 3
p: L2177:  ..11.11...                    :  LOAD CONSTANT 3
p: L2200:  .....11...                    :  LOAD CONSTANT 0
p: L2201:  1....11...                LNC7  :  LOAD CONSTANT 8
p: L2202:  .1.1.11...                    :  LOAD CONSTANT 5
p: L2203:  .....11...                    :  LOAD CONSTANT 0
p: L2204:  1..1.11...                    :  LOAD CONSTANT 9
p: L2205:  111.1.1.11  -> L2352          :  GO TO LNC9
p: L2206:  1..1.11..1  -> L2226      EXP29 :  JSB ECA22
```

```
L2135   r2 b135             1111100010 (0994)     [register] using P : increment A                                  p: L2207:  11111...1.                      :  A + 1 -> A[P]
L2136   r2 b136   exp22     0100101110 (0302)     [register] using W : A → B                                        p: L2210:  .1..1.111.                  EXP22 :  A -> B[W]
L2137   r2 b137             0101111110 (0382)     [register] using S : decrement C                                  p: L2211:  .1.111111.                      :  C - 1 -> C[S]
L2138   r2 b138             1000011011 (0539)     [flow    ] (if no carry then) goTo : 134       → exp29     p: L2212:  1....11.11  -> L2206           :  IF NO CARRY GO TO EXP29
L2139   r2 b139             1011010010 (0722)     [register] using WP : rightShift A                                p: L2213:  1.11.1..1.                      :  SHIFT RIGHT A[WP]
L2140   r2 b140             1110101110 (0942)     [register] using W : exchange A C                                 p: L2214:  111.1.111.                      :  A EXCHANGE C[W]
L2141   r2 b141             0100010110 (0278)     [register] using MS : leftShift A                                 p: L2215:  .1...1.11.                      :  SHIFT LEFT A[MS]
L2142   r2 b142   exp23     1110101110 (0942)     [register] using W : exchange A C                                 p: L2216:  111.1.111.                  EXP23 :  A EXCHANGE C[W]
L2143   r2 b143             1101111110 (0894)     [register] using S : decrement A                                  p: L2217:  11.111111.                      :  A - 1 -> A[S]
L2144   r2 b144             1000100011 (0547)     [flow    ] (if no carry then) goTo : 136       → exp22     p: L2220:  1...1...11  -> L2210           :  IF NO CARRY GO TO EXP22
L2145   r2 b145             1100101110 (0814)     [register] using W : exchange A B                                 p: L2221:  11..1.111.                      :  A EXCHANGE B[W]
L2146   r2 b146             1111100010 (0994)     [register] using P : increment A                                  p: L2222:  11111...1.                      :  A + 1 -> A[P]
L2147   r2 b147             1100110001 (0817)     [flow    ] (if no carry then) jumpSub : 204    → nrm21     p: L2223:  11..11...1  -> L2314           :  JSB NRM21
L2148   r2 b148   rtn21     0010010000 (0144)     [rom     ] selectRom : 1                                          p: L2224:  ..1..1....  -> L1225  *****  RTN21 :  SELECT ROM 1
L2149   r2 b149   eca21     1011010010 (0722)     [register] using WP : rightShift A                                p: L2225:  1.11.1..1.                  ECA21 :  SHIFT RIGHT A[WP]
L2150   r2 b150   eca22     1101111110 (0894)     [register] using S : decrement A                                  p: L2226:  11.111111.                  ECA22 :  A - 1 -> A[S]
L2151   r2 b151             1001010111 (0599)     [flow    ] (if no carry then) goTo : 149       → eca21     p: L2227:  1..1.1.111  -> L2225           :  IF NO CARRY GO TO ECA21
L2152   r2 b152             1011111110 (0766)     [register] using S : clear A                                      p: L2230:  1.1111111.                      :  0 -> A[S]
L2153   r2 b153             1110001110 (0910)     [register] using W : A + B → A                                    p: L2231:  111...111.                      :  A + B -> A[W]
L2154   r2 b154             0000110000 (0048)     [other   ] (if no carry then) (flow) return                       p: L2232:  ....11....                      :  RETURN
L2155   r2 b155   pqo21     0010010000 (0144)     [rom     ] selectRom : 1                                          p: L2233:  ..1..1....  -> L1234  *****  PQO21 :  SELECT ROM 1
L2156   r2 b156   pmu21     1011001110 (0718)     [register] using W : rightShift A                                 p: L2234:  1.11..111.                  PMU21 :  SHIFT RIGHT A[W]
L2157   r2 b157   pmu22     1000101110 (0558)     [register] using W : exchange B C                                 p: L2235:  1...1.111.                  PMU22 :  B EXCHANGE C[W]
L2158   r2 b158             1010000011 (0643)     [flow    ] goTo : 160                  → pmu24             p: L2236:  1.1.....11  -> L2240           :  GO TO PMU24
L2159   r2 b159   pmu23     1110001110 (0910)     [register] using W : A + B → A                                    p: L2237:  111...111.                  PMU23 :  A + B -> A[W]
L2160   r2 b160   pmu24     0101111110 (0382)     [register] using S : decrement C                                  p: L2240:  .1.111111.                  PMU24 :  C - 1 -> C[S]
L2161   r2 b161             1001111111 (0639)     [flow    ] (if no carry then) goTo : 159       → pmu23     p: L2241:  1..1111111  -> L2237           :  IF NO CARRY GO TO PMU23
L2162   r2 b162             1110101110 (0942)     [register] using W : exchange A C                                 p: L2242:  111.1.111.                      :  A EXCHANGE C[W]
L2163   r2 b163             0100010110 (0278)     [register] using MS : leftShift A                                 p: L2243:  .1...1.11.                      :  SHIFT LEFT A[MS]
L2164   r2 b164             1110101110 (0942)     [register] using W : exchange A C                                 p: L2244:  111.1.111.                      :  A EXCHANGE C[W]
L2165   r2 b165             0110110111 (0439)     [flow    ] goTo : 109                  → pqo23             p: L2245:  .11.11.111  -> L2155           :  GO TO PQO23
L2166   r2 b166   mpy21     0011001100 (0204)     [pointer ] value → P : value 3                                    p: L2246:  ..11..11..                  MPY21 :  3 -> P
L2167   r2 b167   mpy22     0111001010 (0458)     [register] using X : A + C → C                                    p: L2247:  .111..1.1.                  MPY22 :  A + C -> C[X]
L2168   r2 b168   div21     0101011110 (0350)     [register] using S : A - C → C                                    p: L2250:  .1.1.1111.                  DIV21 :  A - C -> C[S]
L2169   r2 b169             1010101111 (0687)     [flow    ] (if no carry then) goTo : 171       → div22     p: L2251:  1.1.1.1111  -> L2253           :  IF NO CARRY GO TO DIV22
L2170   r2 b170             0010111110 (0190)     [register] using S : 0 - C → C                                    p: L2252:  ..1.11111.                      :  0 - C -> C[S]
L2171   r2 b171   div22     1100100110 (0806)     [register] using M : exchange A B                                 p: L2253:  11..1..11.                  DIV22 :  A EXCHANGE B[M]
L2172   r2 b172             1011101110 (0750)     [register] using W : clear A                                      p: L2254:  1.111.111.                      :  0 -> A[W]
L2173   r2 b173             1100101100 (0812)     [pointer ] if P ≠ value : value 12                                p: L2255:  11..1.11..                      :  IF P # 12
L2174   r2 b174             1100010111 (0791)     [flow    ]    (then) goTo : 197                → mpy27     p: L2256:  11...1.111  -> L2305           :      THEN GO TO MPY27
L2175   r2 b175             0001100110 (0102)     [register] using M : if C ≥ 1                                     p: L2257:  ...11..11.                      :  IF C[M] >= 1
L2176   r2 b176             1011011011 (0731)     [flow    ]    (then) goTo : 182                → div23     p: L2260:  1.11.11.11  -> L2266           :      THEN GO TO DIV23
L2177   r2 b177             0001010100 (0084)     [status  ] if statusBit ≠ 1 : bit 1                               p: L2261:  ...1.1.1..                      :  IF S1 # 1
L2178   r2 b178             0000000011 (0003)     [flow    ]    (then) goTo : 0                  → err21     p: L2262:  ........11  -> L2000           :      THEN GO TO ERR21
L2179   r2 b179             1010010000 (0656)     [rom     ] selectRom : 5                                          p: L2263:  1.1..1....  -> L5264  *****         :  SELECT ROM 5
```

```
L2180   r2 b180            1101101111 (0879)   [flow    ] goTo : 219                          → nrm25
L2181   r2 b181            0000000000 (0000)   [no oper ] nOp
L2182   r2 b182   div23    1000110010 (0562)   [register] using WP : exchange B C
L2183   r2 b183            1110100110 (0934)   [register] using M : exchange A C
L2184   r2 b184            0010010000 (0144)   [rom     ] selectRom : 1
L2185   r2 b185   lnc2     1000100100 (0548)   [status  ] clear statusBit : bit 8
L2186   r2 b186            0110011000 (0408)   [constant] loadConstant : 6
L2187   r2 b187            1001011000 (0600)   [constant] loadConstant : 9
L2188   r2 b188            0011011000 (0216)   [constant] loadConstant : 3
L2189   r2 b189            0001011000 (0088)   [constant] loadConstant : 1
L2190   r2 b190            0100011000 (0280)   [constant] loadConstant : 4
L2191   r2 b191            0111011000 (0472)   [constant] loadConstant : 7
L2192   r2 b192            0001011000 (0088)   [constant] loadConstant : 1
L2193   r2 b193            1110011011 (0923)   [flow    ] goTo : 230                          → lnc8
L2194   r2 b194   pre27    1111100110 (0998)   [register] using M : increment A
L2195   r2 b195            0110010011 (0403)   [flow    ] (if no carry then) goTo : 100       → pre25
L2196   r2 b196   mpy26    1110001110 (0910)   [register] using W : A + B → A
L2197   r2 b197   mpy27    0101100010 (0354)   [register] using P : decrement C
L2198   r2 b198            1100010011 (0787)   [flow    ] (if no carry then) goTo : 196       → mpy26
L2199   r2 b199   mpy28    1011001110 (0718)   [register] using W : rightShift A
L2200   r2 b200            0000111100 (0060)   [pointer ] increment P
L2201   r2 b201            1101101100 (0876)   [pointer ] if P ≠ value : value 13
L2202   r2 b202            1100010111 (0791)   [flow    ]    (then) goTo : 197                → mpy27
L2203   r2 b203   nrm20    0111101010 (0490)   [register] using X : increment C
L2204   r2 b204   nrm21    1011111110 (0766)   [register] using S : clear A
L2205   r2 b205            1100001100 (0780)   [pointer ] value → P : value 12
L2206   r2 b206            0000101110 (0046)   [register] using W : clear B
L2207   r2 b207   nrm23    1001100010 (0610)   [register] using P : if A ≥ 1
L2208   r2 b208            1101011011 (0859)   [flow    ]    (then) goTo : 214                → nrm24
L2209   r2 b209            0100001110 (0270)   [register] using W : leftShift A
L2210   r2 b210            0101101010 (0362)   [register] using X : decrement C
L2211   r2 b211            1001101110 (0622)   [register] using W : if A ≥ 1
L2212   r2 b212            1100111111 (0831)   [flow    ]    (then) goTo : 207                → nrm23
L2213   r2 b213            0011001110 (0206)   [register] using W : clear c
L2214   r2 b214   nrm24    0100101010 (0298)   [register] using X : A → B
L2215   r2 b215            1110001110 (0910)   [register] using W : A + B → A
L2216   r2 b216            1001111110 (0638)   [register] using S : if A ≥ 1
L2217   r2 b217            1100011111 (0799)   [flow    ]    (then) goTo : 199                → mpy28
L2218   r2 b218            1110100110 (0934)   [register] using M : exchange A C
L2219   r2 b219   nrm25    0110001110 (0398)   [register] using W : C → A
L2220   r2 b220            0000101110 (0046)   [register] using W : clear B
L2221   r2 b221   nrm27    1100001100 (0780)   [pointer ] value → P : value 12
L2222   r2 b222            0111101011 (0491)   [flow    ] goTo : 122                          → nrm26
L2223   r2 b223   lncd1    1001001100 (0588)   [pointer ] value → P : value 9
L2224   r2 b224            0011011000 (0216)   [constant] loadConstant : 3
```

```
p: L2264:  11.11.1111  -> L2333                :  GO TO NRM25
p: L2265:  ..........                          :  NO OPERATION
p: L2266:  1...11..1.                    DIV23 :  B EXCHANGE C[WP]
p: L2267:  111.1..11.                          :  A EXCHANGE C[M]
p: L2270:  ..1..1....  -> L1271   *****        :  SELECT ROM 1
p: L2271:  1...1..1..                    LNC2  :  0 -> S8
p: L2272:  .11..11...                          :  LOAD CONSTANT 6
p: L2273:  1..1.11...                          :  LOAD CONSTANT 9
p: L2274:  ..11.11...                          :  LOAD CONSTANT 3
p: L2275:  ...1.11...                          :  LOAD CONSTANT 1
p: L2276:  .1...11...                          :  LOAD CONSTANT 4
p: L2277:  .111.11...                          :  LOAD CONSTANT 7
p: L2300:  ...1.11...                          :  LOAD CONSTANT 1
p: L2301:  111..11.11  -> L2346                :  GO TO LNC8
p: L2302:  11111..11.                    PRE27 :  A + 1 -> A[M]
p: L2303:  .11..1..11  -> L2144                :  IF NO CARRY GO TO PRE25
p: L2304:  111...111.                    MPY26 :  A + B -> A[W]
p: L2305:  .1.11...1.                    MPY27 :  C - 1 -> C[P]
p: L2306:  11...1..11  -> L2304                :  IF NO CARRY GO TO MPY26
p: L2307:  1.11..111.                    MPY28 :  SHIFT RIGHT A[W]
p: L2310:  ....1111..                          :  P + 1 -> P
p: L2311:  11.11.11..                          :  IF P # 13
p: L2312:  11...1.111  -> L2305                :      THEN GO TO MPY27
p: L2313:  .1111.1.1.                    NRM20 :  C + 1 -> C[X]
p: L2314:  1.1111111.                    NRM21 :  0 -> A[S]
p: L2315:  11....11..                          :  12 -> P
p: L2316:  ....1.111.                          :  0 -> B[W]
p: L2317:  1..11...1.                    NRM23 :  IF A[P] >= 1
p: L2320:  11.1.11.11  -> L2326                :      THEN GO TO NRM24
p: L2321:  .1....111.                          :  SHIFT LEFT A[W]
p: L2322:  .1.11.1.1.                          :  C - 1 -> C[X]
p: L2323:  1..11.111.                          :  IF A[W] >= 1
p: L2324:  11..111111  -> L2317                :      THEN GO TO NRM23
p: L2325:  ..11..111.                          :  0 -> C[W]
p: L2326:  .1..1.1.1.                    NRM24 :  A -> B[X]
p: L2327:  111...111.                          :  A + B -> A[W]
p: L2330:  1..111111.                          :  IF A[S] >= 1
p: L2331:  11...11111  -> L2307                :      THEN GO TO MPY28
p: L2332:  111.1..11.                          :  A EXCHANGE C[M]
p: L2333:  .11...111.                    NRM25 :  C -> A[W]
p: L2334:  ....1.111.                          :  0 -> B[W]
p: L2335:  11....11..                    NRM27 :  12 -> P
p: L2336:  .1111.1.11  -> L2172                :  GO TO NRM26
p: L2337:  1..1..11..                    LNCD1 :  9 -> P
p: L2340:  ..11.11...                          :  LOAD CONSTANT 3
```

```
L2225   r2 b225             0001011000 (0088)     [constant] loadConstant : 1                                  p: L2341:  ...1.11...                        :  LOAD CONSTANT 1
L2226   r2 b226             0000011000 (0024)     [constant] loadConstant : 0                                  p: L2342:  .....11...                        :  LOAD CONSTANT 0
L2227   r2 b227             0001011000 (0088)     [constant] loadConstant : 1                                  p: L2343:  ...1.11...                        :  LOAD CONSTANT 1
L2228   r2 b228             0111011000 (0472)     [constant] loadConstant : 7                                  p: L2344:  .111.11...                        :  LOAD CONSTANT 7
L2229   r2 b229             1001011000 (0600)     [constant] loadConstant : 9                                  p: L2345:  1..1.11...                        :  LOAD CONSTANT 9
L2230   r2 b230   lnc8      1000011000 (0536)     [constant] loadConstant : 8                                  p: L2346:  1....11...                  LNC8  :  LOAD CONSTANT 8
L2231   r2 b231             0000011000 (0024)     [constant] loadConstant : 0                                  p: L2347:  .....11...                        :  LOAD CONSTANT 0
L2232   r2 b232             0101011000 (0344)     [constant] loadConstant : 5                                  p: L2350:  .1.1.11...                        :  LOAD CONSTANT 5
L2233   r2 b233             0101011000 (0344)     [constant] loadConstant : 5                                  p: L2351:  .1.1.11...                        :  LOAD CONSTANT 5
L2234   r2 b234   lnc9      0011011000 (0216)     [constant] loadConstant : 3                                  p: L2352:  ..11.11...                  LNC9  :  LOAD CONSTANT 3
L2235   r2 b235             1101110111 (0887)     [flow    ] goTo : 221                        → nrm27         p: L2353:  11.111.111  -> L2335               :  GO TO NRM27
L2236   r2 b236   pre21     1110101110 (0942)     [register] using W : exchange A C                            p: L2354:  111.1.111.                  PRE21 :  A EXCHANGE C[W]
L2237   r2 b237             0100101110 (0302)     [register] using W : A → B                                   p: L2355:  .1..1.111.                        :  A -> B[W]
L2238   r2 b238             0110000110 (0390)     [register] using M : C → A                                   p: L2356:  .11....11.                        :  C -> A[M]
L2239   r2 b239             1010111010 (0698)     [register] using XS : C + C → C                              p: L2357:  1.1.111.1.                        :  C + C -> C[XS]
L2240   r2 b240             0101111011 (0379)     [flow    ] (if no carry then) goTo : 94      → pre24         p: L2360:  .1.1111.11  -> L2136              :  IF NO CARRY GO TO PRE24
L2241   r2 b241             0111111010 (0506)     [register] using XS : increment C                            p: L2361:  .111111.1.                        :  C + 1 -> C[XS]
L2242   r2 b242   pre22     1011001110 (0718)     [register] using W : rightShift A                            p: L2362:  1.11..111.                  PRE22 :  SHIFT RIGHT A[W]
L2243   r2 b243             0111101010 (0490)     [register] using X : increment C                             p: L2363:  .1111.1.1.                        :  C + 1 -> C[X]
L2244   r2 b244             1111001011 (0971)     [flow    ] (if no carry then) goTo : 242     → pre22         p: L2364:  1111..1.11  -> L2362              :  IF NO CARRY GO TO PRE22
L2245   r2 b245             0110011111 (0415)     [flow    ] goTo : 103                        → pre26         p: L2365:  .11..11111  -> L2147              :  GO TO PRE26
L2246   r2 b246   lnc10     0011001110 (0206)     [register] using W : clear c                                 p: L2366:  ..11..111.                  LNC10 :  0 -> C[W]
L2247   r2 b247             1100001100 (0780)     [pointer ] value → P : value 12                              p: L2367:  11....11..                        :  12 -> P
L2248   r2 b248             0010011000 (0152)     [constant] loadConstant : 2                                  p: L2370:  ..1..11...                        :  LOAD CONSTANT 2
L2249   r2 b249             0011011000 (0216)     [constant] loadConstant : 3                                  p: L2371:  ..11.11...                        :  LOAD CONSTANT 3
L2250   r2 b250             0000011000 (0024)     [constant] loadConstant : 0                                  p: L2372:  .....11...                        :  LOAD CONSTANT 0
L2251   r2 b251             0010011000 (0152)     [constant] loadConstant : 2                                  p: L2373:  ..1..11...                        :  LOAD CONSTANT 2
L2252   r2 b252             0101011000 (0344)     [constant] loadConstant : 5                                  p: L2374:  .1.1.11...                        :  LOAD CONSTANT 5
L2253   r2 b253             1000000111 (0519)     [flow    ] goTo : 129                        → lnc7          p: L2375:  1......111  -> L2201             :  GO TO LNC7
L2254   r2 b254   lncd3     0101001100 (0332)     [pointer ] value → P : value 5                               p: L2376:  .1.1..11..                  LNCD3 :  5 -> P
L2255   r2 b255             0111111011 (0507)     [flow    ] goTo : 126                        → lnc6          p: L2377:  .111111.11  -> L2176              :  GO TO LNC6
```

L3000	r3 b000	prfx	1000010000 (0528)	[rom]	selectRom : 4		N
L3001	r3 b001		0000000000 (0000)	[no oper]	nOp		
L3002	r3 b002	fix1	0100101011 (0299)	[flow]	goTo : 74	→ fix2	N
L3003	r3 b003	expn	0101100111 (0359)	[flow]	goTo : 89	→ exp1	N
L3004	r3 b004	lnnn	0101101011 (0363)	[flow]	goTo : 90	→ lnn2z4	N
L3005	r3 b005		0000000000 (0000)	[no oper]	nOp		
L3006	r3 b006	invx	0100010011 (0275)	[flow]	goTo : 68	→ inv1	N
L3007	r3 b007	lexx	0100010000 (0272)	[rom]	selectRom : 2		N
L3008	r3 b008	perc	1000010000 (0528)	[rom]	selectRom : 4		N
L3009	r3 b009	rnd0	1100010000 (0784)	[rom]	selectRom : 6		N
L3010	r3 b010	rcal	0110011011 (0411)	[flow]	goTo : 102	→ rcl0	N
L3011	r3 b011	stor	0110100011 (0419)	[flow]	goTo : 104	→ str0	
L3012	r3 b012	rold	1100101000 (0808)	[other]	(stack) rotateDown		
L3013	r3 b013		0111111111 (0511)	[flow]	goTo : 127	→ fst1zx	
L3014	r3 b014	exc1	0110101000 (0424)	[other]	(stack) stack → A		
L3015	r3 b015		0100101000 (0296)	[other]	(stack) M → stack		
L3016	r3 b016		0111111011 (0507)	[flow]	goTo : 126	→ fstxzj	
L3017	r3 b017	fst2z5	0011111111 (0255)	[flow]	goTo : 63	→ ent2	
L3018	r3 b018	dig6	1111101110 (1006)	[register]	using W : increment A		
L3019	r3 b019	dig5	1111101110 (1006)	[register]	using W : increment A		
L3020	r3 b020	dig4	1111101110 (1006)	[register]	using W : increment A		
L3021	r3 b021		0001101011 (0107)	[flow]	(if no carry then) goTo : 26	→ dig3	
L3022	r3 b022	addd	1000010000 (0528)	[rom]	selectRom : 4		
L3023	r3 b023	fix3	1000011101 (0541)	[flow]	jumpSub : 135	→ dsp0z4	
L3024	r3 b024		0100001110 (0270)	[register]	using W : leftShift A		
L3025	r3 b025		0101001011 (0331)	[flow]	goTo : 82	→ fmt1	
L3026	r3 b026	dig3	1111101110 (1006)	[register]	using W : increment A		
L3027	r3 b027	dig2	1111101110 (1006)	[register]	using W : increment A		
L3028	r3 b028	dig1	1111101110 (1006)	[register]	using W : increment A		
L3029	r3 b029		0000110000 (0048)	[other]	(if no carry then) (flow) return		
L3030	r3 b030	mult	1000010000 (0528)	[rom]	selectRom : 4		
L3031	r3 b031	tkra	0011010000 (0208)	[rom]	keys → romAddress : 1		
L3032	r3 b032	sig1	1000100100 (0548)	[status]	clear statusBit : bit 8		
L3033	r3 b033		1000010000 (0528)	[rom]	selectRom : 4		
L3034	r3 b034	sigp	0010000011 (0131)	[flow]	goTo : 32	→ sig1	
L3035	r3 b035	dcpt	0011001100 (0204)	[pointer]	value → P : value 3		
L3036	r3 b036	dig0	0000110000 (0048)	[other]	(flow) return		
L3037	r3 b037	dvid	1000010000 (0528)	[rom]	selectRom : 4		
L3038	r3 b038	divd	0010010111 (0151)	[flow]	goTo : 37	→ dvid	
L3039	r3 b039	tan2	0101000100 (0324)	[status]	set statusBit : bit 5		
L3040	r3 b040	tang	0110010001 (0401)	[flow]	jumpSub : 100	→ sav9	
L3041	r3 b041		0100001011 (0267)	[flow]	goTo : 66	→ sqt1z4	
L3042	r3 b042	coss	0100110011 (0307)	[flow]	goTo : 76	→ cos2	
L3043	r3 b043	sinn	0010011111 (0159)	[flow]	goTo : 39	→ tan2	
L3044	r3 b044	tpol	1000010000 (0528)	[rom]	selectRom : 4		

```
p: L3000:  1....1....  -> L4001   *****  PRFX  :  SELECT ROM 4
p: L3001:  ..........                          :  NO OPERATION
p: L3002:  .1..1.1.11  -> L3112          FIX1  :  GO TO FIX2
p: L3003:  .1.11..111  -> L3131          EXPN  :  GO TO EXP1
p: L3004:  .1.11.1.11  -> L3132          LNNN  :  GO TO LNN2Z4
p: L3005:  ..........                          :  NO OPERATION
p: L3006:  .1...1..11  -> L3104          INVX  :  GO TO INV1
p: L3007:  .1...1....  -> L2010   *****  LEXX  :  SELECT ROM 2
p: L3010:  1....1....  -> L4011   *****  PERC  :  SELECT ROM 4
p: L3011:  11...1....  -> L6012   *****  RND0  :  SELECT ROM 6
p: L3012:  .11..11.11  -> L3146          RCAL  :  GO TO RCL0
p: L3013:  .11.1...11  -> L3150          STOR  :  GO TO STR0
p: L3014:  11..1.1...                    ROLD  :  DOWN ROTATE
p: L3015:  0.111111111 -> L3177                :  GO TO FST1ZX
p: L3016:  .11.1.1...                    EXC1  :  STACK -> A
p: L3017:  .1..1.1...                          :  C -> STACK
p: L3020:  .111111.11  -> L3176                :  GO TO FSTXZJ
p: L3021:  ..11111111  -> L3077          FST2Z5:  GO TO ENT2
p: L3022:  11111.111.                    DIG6  :  A + 1 -> A[W]
p: L3023:  11111.111.                    DIG5  :  A + 1 -> A[W]
p: L3024:  11111.111.                    DIG4  :  A + 1 -> A[W]
p: L3025:  ...11.1.11  -> L3032                :  IF NO CARRY GO TO DIG3
p: L3026:  1....1....  -> L4027   *****  ADDD  :  SELECT ROM 4
p: L3027:  1....111.1  -> L3207          FIX3  :  JSB DSP0Z4
p: L3030:  .1....111.                          :  SHIFT LEFT A[W]
p: L3031:  .1.1..1.11  -> L3122                :  GO TO FMT1
p: L3032:  11111.111.                    DIG3  :  A + 1 -> A[W]
p: L3033:  11111.111.                    DIG2  :  A + 1 -> A[W]
p: L3034:  11111.111.                    DIG1  :  A + 1 -> A[W]
p: L3035:  ....11....                          :  RETURN
p: L3036:  1....1....  -> L4037   *****  MULT  :  SELECT ROM 4
p: L3037:  ..11.1....                    TKRA  :  KEYS -> ROM ADDRESS
p: L3040:  1...1..1..                    SIG1  :  0 -> S8
p: L3041:  1....1....  -> L4042   *****        :  SELECT ROM 4
p: L3042:  ..1.....11  -> L3040          SIGP  :  GO TO SIG1
p: L3043:  ..11..11..                    DCPT  :  3 -> P
p: L3044:  ....11....                    DIG0  :  RETURN
p: L3045:  1....1....  -> L4046   *****  DVID  :  SELECT ROM 4
p: L3046:  ..1..1.111  -> L3045          DIVD  :  GO TO DVID
p: L3047:  .1.1...1..                    TAN2  :  1 -> S5
p: L3050:  .11..1...1  -> L3144          TANG  :  JSB SAV9
p: L3051:  .1....1.11  -> L3102                :  GO TO SQT1Z4
p: L3052:  .1..11..11  -> L3114          COSS  :  GO TO COS2
p: L3053:  ..1..11111  -> L3047          SINN  :  GO TO TAN2
p: L3054:  1....1....  -> L4055   *****  TPOL  :  SELECT ROM 4
```

```
L3045   r3 b045            0000000000 (0000)    [no oper ] nOp
L3046   r3 b046   sqar     0110000101 (0389)    [flow    ] jumpSub : 97                    → save
L3047   r3 b047            0100100011 (0291)    [flow    ] goTo : 72                       → mul0
L3048   r3 b048            0000000000 (0000)    [no oper ] nOp
L3049   r3 b049   sqt2     0000010000 (0016)    [rom     ] selectRom : 0
L3050   r3 b050   dig9     1111101110 (1006)    [register] using W : increment A
L3051   r3 b051   dig8     1111101110 (1006)    [register] using W : increment A
L3052   r3 b052   dig7     1111101110 (1006)    [register] using W : increment A
L3053   r3 b053            0001001011 (0075)    [flow    ] (if no carry then) goTo : 18    → dig6
L3054   r3 b054   subt     1000010000 (0528)    [rom     ] selectRom : 4
L3055   r3 b055            0000000000 (0000)    [no oper ] nOp
L3056   r3 b056   clrx     1010011001 (0665)    [flow    ] jumpSub : 166                   → ofl2
L3057   r3 b057            1000000111 (0519)    [flow    ] goTo : 129                      → fst2zx
L3058   r3 b058   eexx     1011101111 (0751)    [flow    ] goTo : 187                      → eex2
L3059   r3 b059   chs1     1100000111 (0775)    [flow    ] goTo : 193                      → chs2
L3060   r3 b060   clok     0000101110 (0046)    [register] using W : clear B
L3061   r3 b061            1110010000 (0912)    [rom     ] selectRom : 7
L3062   r3 b062   ent1     0100101000 (0296)    [other   ] (stack) M → stack
L3063   r3 b063   ent2     1010011101 (0669)    [flow    ] jumpSub : 167                   → ofl3
L3064   r3 b064            1000000111 (0519)    [flow    ] goTo : 129                      → fst2zx
L3065   r3 b065   sqt0     0110000101 (0389)    [flow    ] jumpSub : 97                    → save
L3066   r3 b066   sqt1z4   1001100100 (0612)    [status  ] clear statusBit : bit 9
L3067   r3 b067   sqt1     0011000111 (0199)    [flow    ] goTo : 49                       → sqt2
L3068   r3 b068   inv1     0110000101 (0389)    [flow    ] jumpSub : 97                    → save
L3069   r3 b069            1011101110 (0750)    [register] using W : clear A
L3070   r3 b070            1111100010 (0994)    [register] using P : increment A
L3071   r3 b071            0100100111 (0295)    [flow    ] (if no carry then) goTo : 73    → div0
L3072   r3 b072   mul0     1000010000 (0528)    [rom     ] selectRom : 4
L3073   r3 b073   div0     1000010000 (0528)    [rom     ] selectRom : 4
L3074   r3 b074   fix2     1001100100 (0612)    [status  ] clear statusBit : bit 9
L3075   r3 b075            0001011111 (0095)    [flow    ] goTo : 23                       → fix3
L3076   r3 b076   cos2     0110010001 (0401)    [flow    ] jumpSub : 100                   → sav9
L3077   r3 b077   cos2z4   1001000100 (0580)    [status  ] set statusBit : bit 9
L3078   r3 b078   trecz4   0101000100 (0324)    [status  ] set statusBit : bit 5
L3079   r3 b079            0011000111 (0199)    [flow    ] goTo : 49                       → sqt2
L3080   r3 b080   frmt     0100001110 (0270)    [register] using W : leftShift A
L3081   r3 b081            1111101110 (1006)    [register] using W : increment A
L3082   r3 b082   fmt1     0100001110 (0270)    [register] using W : leftShift A
L3083   r3 b083            0010101000 (0168)    [other   ] (register) exchange C M
L3084   r3 b084            1110101010 (0938)    [register] using X : exchange A C
L3085   r3 b085            0010101000 (0168)    [other   ] (register) exchange C M
L3086   r3 b086   fstpz4   0111001011 (0459)    [flow    ] goTo : 114                      → fstp
L3087   r3 b087            0000000000 (0000)    [no oper ] nOp
L3088   r3 b088            0000000000 (0000)    [no oper ] nOp
L3089   r3 b089   exp1     1000100100 (0548)    [status  ] clear statusBit : bit 8
```

```
p: L3055:  .........                          :  NO OPERATION
p: L3056:  .11....1.1  -> L3141         SQAR  :  JSB SAVE
p: L3057:  .1..1...11  -> L3110               :  GO TO MUL0
p: L3060:  .........                          :  NO OPERATION
p: L3061:  .....1....  -> L0062   *****  SQT2  :  SELECT ROM 0
p: L3062:  11111.111.                    DIG9  :  A + 1 -> A[W]
p: L3063:  11111.111.                    DIG8  :  A + 1 -> A[W]
p: L3064:  11111.111.                    DIG7  :  A + 1 -> A[W]
p: L3065:  ...1..1.11  -> L3022               :  IF NO CARRY GO TO DIG6
p: L3066:  1....1....  -> L4067   *****  SUBT  :  SELECT ROM 4
p: L3067:  .........                          :  NO OPERATION
p: L3070:  1.1..11..1  -> L3246         CLRX  :  JSB OFL2
p: L3071:  1......111  -> L3201               :  GO TO FST2ZX
p: L3072:  1.111.1111  -> L3273         EEXX  :  GO TO EEX2
p: L3073:  11.....111  -> L3301         CHS1  :  GO TO CHS2
p: L3074:  ....1.111.                    CLOK  :  0 -> B[W]
p: L3075:  111..1....  -> L7076   *****        :  SELECT ROM 7
p: L3076:  .1..1.1...                    ENT1  :  C -> STACK
p: L3077:  1.1..111.1  -> L3247         ENT2  :  JSB OFL3
p: L3100:  1......111  -> L3201               :  GO TO FST2ZX
p: L3101:  .11....1.1  -> L3141         SQT0  :  JSB SAVE
p: L3102:  1..11..1..                    SQT1Z4:  0 -> S9
p: L3103:  ..11...111  -> L3061         SQT1  :  GO TO SQT2
p: L3104:  .11....1.1  -> L3141         INV1  :  JSB SAVE
p: L3105:  1.111.111.                         :  0 -> A[W]
p: L3106:  11111...1.                         :  A + 1 -> A[P]
p: L3107:  .1..1..111  -> L3111               :  IF NO CARRY GO TO DIV0
p: L3110:  1....1....  -> L4111   *****  MUL0  :  SELECT ROM 4
p: L3111:  1....1....  -> L4112   *****  DIV0  :  SELECT ROM 4
p: L3112:  1..11..1..                    FIX2  :  0 -> S9
p: L3113:  ...1.11111  -> L3027               :  GO TO FIX3
p: L3114:  .11..1...1  -> L3144         COS2  :  JSB SAV9
p: L3115:  1..1...1..                    COS2Z4:  1 -> S9
p: L3116:  .1.1...1..                    TRECZ4:  1 -> S5
p: L3117:  ..11...111  -> L3061               :  GO TO SQT2
p: L3120:  .1....111.                    FRMT  :  SHIFT LEFT A[W]
p: L3121:  11111.111.                         :  A + 1 -> A[W]
p: L3122:  .1....111.                    FMT1  :  SHIFT LEFT A[W]
p: L3123:  ..1.1.1...                         :  C EXCHANGE M
p: L3124:  111.1.1.1.                         :  A EXCHANGE C[X]
p: L3125:  ..1.1.1...                         :  C EXCHANGE M
p: L3126:  .111..1.11  -> L3162         FSTPZ4:  GO TO FSTP
p: L3127:  .........                          :  NO OPERATION
p: L3130:  .........                          :  NO OPERATION
p: L3131:  1...1..1..                    EXP1  :  0 -> S8
```

```
L3090   r3 b090   lnn2z4   1001000100 (0580)   [status  ] set statusBit : bit 9
L3091   r3 b091            0110000101 (0389)   [flow    ] jumpSub : 97                    → save
L3092   r3 b092   nty1z4   0010000100 (0132)   [status  ] set statusBit : bit 2
L3093   r3 b093            0000011111 (0031)   [flow    ] goTo : 7                        → lexx
L3094   r3 b094   sci2z4   1000011101 (0541)   [flow    ] jumpSub : 135                   → dsp0z4
L3095   r3 b095            0101000011 (0323)   [flow    ] goTo : 80                       → frmt
L3096   r3 b096   sav1     0011000100 (0196)   [status  ] set statusBit : bit 3
L3097   r3 b097   save     1010100100 (0676)   [status  ] clear statusBit : bit 10
L3098   r3 b098            1100010000 (0784)   [rom     ] selectRom : 6
L3099   r3 b099   sav2     1100010000 (0784)   [rom     ] selectRom : 6
L3100   r3 b100   sav9     0001000100 (0068)   [status  ] set statusBit : bit 1
L3101   r3 b101            0110000111 (0391)   [flow    ] goTo : 97                       → save
L3102   r3 b102   rcl0     1001000100 (0580)   [status  ] set statusBit : bit 9
L3103   r3 b103            0110100111 (0423)   [flow    ] goTo : 105                      → str1
L3104   r3 b104   str0     1001100100 (0612)   [status  ] clear statusBit : bit 9
L3105   r3 b105   str1     0010000100 (0132)   [status  ] set statusBit : bit 2
L3106   r3 b106            1000011101 (0541)   [flow    ] jumpSub : 135                   → dsp0z4
L3107   r3 b107            0111010101 (0469)   [flow    ] jumpSub : 117                   → chk0
L3108   r3 b108            0110001101 (0397)   [flow    ] jumpSub : 99                    → sav2
L3109   r3 b109            1001010100 (0596)   [status  ] if statusBit ≠ 1 : bit 9
L3110   r3 b110            0111000111 (0455)   [flow    ]    (then) goTo : 113            → str2
L3111   r3 b111            1111011001 (0985)   [flow    ] jumpSub : 246                   → fst4
L3112   r3 b112            0111111011 (0507)   [flow    ] goTo : 126                      → fstxzj
L3113   r3 b113   str2     1011110000 (0752)   [other   ] (data) C → data
L3114   r3 b114   fstp     0111010100 (0468)   [status  ] if statusBit ≠ 1 : bit 7
L3115   r3 b115            0011111111 (0255)   [flow    ]    (then) goTo : 63             → ent2
L3116   r3 b116            0111111111 (0511)   [flow    ] goTo : 127                      → fst1zx
L3117   r3 b117   chk0     0000001100 (0012)   [pointer ] value → P : value 0
L3118   r3 b118            1001100010 (0610)   [register] using P : if A ≥ 1
L3119   r3 b119            1111111111 (1023)   [flow    ]    (then) goTo : 255            → retnzx
L3120   r3 b120   fstpz5   0111001011 (0459)   [flow    ] goTo : 114                      → fstp
L3121   r3 b121   asmdz4   1000011101 (0541)   [flow    ] jumpSub : 135                   → dsp0z4
L3122   r3 b122            0111010101 (0469)   [flow    ] jumpSub : 117                   → chk0
L3123   r3 b123            1000010000 (0528)   [rom     ] selectRom : 4
L3124   r3 b124            0000000000 (0000)   [no oper ] nOp
L3125   r3 b125            0000000000 (0000)   [no oper ] nOp
L3126   r3 b126   fstxzj    1110101110 (0942)   [register] using W : exchange A C
L3127   r3 b127   fst1zx    1010011101 (0669)   [flow    ] jumpSub : 167                   → ofl3
L3128   r3 b128   fst1zj    0111000100 (0452)   [status  ] set statusBit : bit 7
L3129   r3 b129   fst2zx    1000010101 (0533)   [flow    ] jumpSub : 133                   → dsp1
L3130   r3 b130            1111010101 (0981)   [flow    ] jumpSub : 245                   → fst3
L3131   r3 b131            1101011011 (0859)   [flow    ] goTo : 214                      → den2
L3132   r3 b132   chs3     0011111110 (0254)   [register] using S : 0 − C − 1 → C
L3133   r3 b133   dsp1     1010100100 (0676)   [status  ] clear statusBit : bit 10
L3134   r3 b134            1000100011 (0547)   [flow    ] goTo : 136                      → dsp7
```

```
p: L3132:  1..1...1..                    LNN2Z4:  1 -> S9
p: L3133:  .11....1.1  -> L3141                :  JSB SAVE
p: L3134:  ..1....1..                    NTY1Z4:  1 -> S2
p: L3135:  .....11111  -> L3007                :  GO TO LEXX
p: L3136:  1....111.1  -> L3207          SCI2Z4:  JSB DSP0Z4
p: L3137:  .1.1....11  -> L3120                :  GO TO FRMT
p: L3140:  ..11...1..                    SAV1  :  1 -> S3
p: L3141:  1.1.1..1..                    SAVE  :  0 -> S10
p: L3142:  11...1....  -> L6143   *****        :  SELECT ROM 6
p: L3143:  11...1....  -> L6144   *****  SAV2  :  SELECT ROM 6
p: L3144:  ...1...1..                    SAV9  :  1 -> S1
p: L3145:  .11....111  -> L3141                :  GO TO SAVE
p: L3146:  1..1...1..                    RCL0  :  1 -> S9
p: L3147:  .11.1..111  -> L3151                :  GO TO STR1
p: L3150:  1..11..1..                    STR0  :  0 -> S9
p: L3151:  ..1....1..                    STR1  :  1 -> S2
p: L3152:  1....111.1  -> L3207                :  JSB DSP0Z4
p: L3153:  .111.1.1.1  -> L3165                :  JSB CHK0
p: L3154:  .11...11.1  -> L3143                :  JSB SAV2
p: L3155:  1..1.1.1..                          :  IF S9 # 1
p: L3156:  .111...111  -> L3161                :       THEN GO TO STR2
p: L3157:  1111.11..1  -> L3366                :  JSB FST4
p: L3160:  .111111.11  -> L3176                :  GO TO FSTXZJ
p: L3161:  1.1111....                    STR2  :  C -> DATA
p: L3162:  .111.1.1..                    FSTP  :  IF S7 # 1
p: L3163:  ..11111111  -> L3077                :       THEN GO TO ENT2
p: L3164:  0.111111111 -> L3177                :  GO TO FST1ZX
p: L3165:  ......11..                    CHK0  :  0 -> P
p: L3166:  1..11...1.                          :  IF A[P] >= 1
p: L3167:  1111111111  -> L3377                :       THEN GO TO RETNZX
p: L3170:  .111..1.11  -> L3162          FSTPZ5:  GO TO FSTP
p: L3171:  1....111.1  -> L3207          ASMDZ4:  JSB DSP0Z4
p: L3172:  .111.1.1.1  -> L3165                :  JSB CHK0
p: L3173:  1....1....  -> L4174   *****        :  SELECT ROM 4
p: L3174:  ..........                          :  NO OPERATION
p: L3175:  ..........                          :  NO OPERATION
p: L3176:  111.1.111.                    FSTXZJ:  A EXCHANGE C[W]
p: L3177:  1.1..111.1  -> L3247          FST1ZX:  JSB OFL3
p: L3200:  .111...1..                    FST1ZJ:  1 -> S7
p: L3201:  1....1.1.1  -> L3205          FST2ZX:  JSB DSP1
p: L3202:  1111.1.1.1  -> L3365                :  JSB FST3
p: L3203:  11.1.11.11  -> L3326                :  GO TO DEN2
p: L3204:  ..1111111.                    CHS3  :  0 − C − 1 -> C[S]
p: L3205:  1.1.1..1..                    DSP1  :  0 -> S10
p: L3206:  1...1...11  -> L3210                :  GO TO DSP7
```

```
L3135   r3 b135   dsp0z4    1011001110 (0718)    [register] using W : rightShift A                                  p: L3207:  1.11..111.                  DSP0Z4:  SHIFT RIGHT A[W]
L3136   r3 b136   dsp7      0110011110 (0414)    [register] using S : C → A                                         p: L3210:  .11..1111.                  DSP7  :  C -> A[S]
L3137   r3 b137             1000100100 (0548)    [status  ] clear statusBit : bit 8                                 p: L3211:  1...1..1..                          :  0 -> S8
L3138   r3 b138             1001010011 (0595)    [flow    ] goTo : 148                          → dsp8               p: L3212:  1..1.1..11  -> L3224                :  GO TO DSP8
L3139   r3 b139   dsp2      0111111010 (0506)    [register] using XS : increment C                                  p: L3213:  .111111.1.                  DSP2  :  C + 1 -> C[XS]
L3140   r3 b140   dsp3      1000000100 (0516)    [status  ] set statusBit : bit 8                                   p: L3214:  1......1..                  DSP3  :  1 -> S8
L3141   r3 b141             0101010100 (0340)    [status  ] if statusBit ≠ 1 : bit 5                                p: L3215:  .1.1.1.1..                          :  IF S5 # 1
L3142   r3 b142             1001001011 (0587)    [flow    ]    (then) goTo : 146                → dsp5               p: L3216:  1..1..1.11  -> L3222                :       THEN GO TO DSP5
L3143   r3 b143             0111101010 (0490)    [register] using X : increment C                                   p: L3217:  .1111.1.1.                          :  C + 1 -> C[X]
L3144   r3 b144             1000101111 (0559)    [flow    ] (if no carry then) goTo : 139       → dsp2               p: L3220:  1...1.1111  -> L3213                :  IF NO CARRY GO TO DSP2
L3145   r3 b145   dsp4      0000101000 (0040)    [other   ] (display) displayToggle                                  p: L3221:  ....1.1...                  DSP4  :  DISPLAY TOGGLE
L3146   r3 b146   dsp5      0000010100 (0020)    [status  ] if statusBit ≠ 1 : bit 0                                 p: L3222:  .....1.1..                  DSP5  :  IF S0 # 1
L3147   r3 b147             1000110011 (0563)    [flow    ]    (then) goTo : 140                → dsp3               p: L3223:  1...11..11  -> L3214                :       THEN GO TO DSP3
L3148   r3 b148   dsp8      0000100100 (0036)    [status  ] clear statusBit : bit 0                                 p: L3224:  ....1..1..                  DSP8  :  0 -> S0
L3149   r3 b149   dsp6      0000011100 (0028)    [pointer ] decrement P                                             p: L3225:  .....111..                  DSP6  :  P - 1 -> P
L3150   r3 b150             1100101100 (0812)    [pointer ] if P ≠ value : value 12                                 p: L3226:  11..1.11..                          :  IF P # 12
L3151   r3 b151             1001010111 (0599)    [flow    ]    (then) goTo : 149                → dsp6               p: L3227:  1..1.1.111  -> L3225                :       THEN GO TO DSP6
L3152   r3 b152             1000101000 (0552)    [other   ] (display) displayOff                                    p: L3230:  1...1.1...                          :  DISPLAY OFF
L3153   r3 b153             1000010100 (0532)    [status  ] if statusBit ≠ 1 : bit 8                                 p: L3231:  1....1.1..                          :  IF S8 # 1
L3154   r3 b154             1001000111 (0583)    [flow    ]    (then) goTo : 145                → dsp4               p: L3232:  1..1...111  -> L3221                :       THEN GO TO DSP4
L3155   r3 b155             0100001110 (0270)    [register] using W : leftShift A                                   p: L3233:  .1....111.                          :  SHIFT LEFT A[W]
L3156   r3 b156             0101100100 (0356)    [status  ] clear statusBit : bit 5                                 p: L3234:  .1.11..1..                          :  0 -> S5
L3157   r3 b157             1010010100 (0660)    [status  ] if statusBit ≠ 1 : bit 10                                p: L3235:  1.1..1.1..                          :  IF S10 # 1
L3158   r3 b158             0001111111 (0127)    [flow    ]    (then) goTo : 31                 → tkra               p: L3236:  ...1111111  -> L3037                :       THEN GO TO TKRA
L3159   r3 b159             1000010000 (0528)    [rom     ] selectRom : 4                                           p: L3237:  1....1....  -> L4240   *****       :  SELECT ROM 4
L3160   r3 b160             0000000000 (0000)    [no oper ] nOp                                                     p: L3240:  ..........                          :  NO OPERATION
L3161   r3 b161   ofl1      0011010010 (0210)    [register] using WP : clear c                                      p: L3241:  ..11.1..1.                  OFL1  :  0 -> C[WP]
L3162   r3 b162             0101110010 (0370)    [register] using WP : decrement C                                  p: L3242:  .1.111..1.                          :  C - 1 -> C[WP]
L3163   r3 b163             0011011010 (0218)    [register] using XS : clear c                                      p: L3243:  ..11.11.1.                          :  0 -> C[XS]
L3164   r3 b164             1110001010 (0906)    [register] using X : A + B → A                                     p: L3244:  111...1.1.                          :  A + B -> A[X]
L3165   r3 b165             1010011111 (0671)    [flow    ] (if no carry then) goTo : 167       → ofl3               p: L3245:  1.1..11111  -> L3247                :  IF NO CARRY GO TO OFL3
L3166   r3 b166   ofl2      0011001110 (0206)    [register] using W : clear c                                       p: L3246:  ..11..111.                  OFL2  :  0 -> C[W]
L3167   r3 b167   ofl3      0000110100 (0052)    [status  ] clear all statusBits                                    p: L3247:  ....11.1..                  OFL3  :  CLEAR STATUS
L3168   r3 b168             0110001110 (0398)    [register] using W : C → A                                         p: L3250:  .11...111.                          :  C -> A[W]
L3169   r3 b169   ofl4      1100001100 (0780)    [pointer ] value → P : value 12                                    p: L3251:  11....11..                  OFL4  :  12 -> P
L3170   r3 b170             0100101010 (0298)    [register] using X : A → B                                         p: L3252:  .1..1.1.1.                          :  A -> B[X]
L3171   r3 b171             0110001010 (0394)    [register] using X : C → A                                         p: L3253:  .11...1.1.                          :  C -> A[X]
L3172   r3 b172             0110111010 (0442)    [register] using XS : if C = 0                                     p: L3254:  .11.111.1.                          :  IF C[XS] = 0
L3173   r3 b173             1011000111 (0711)    [flow    ]    (then) goTo : 177                → ofl5               p: L3255:  1.11...111  -> L3261                :       THEN GO TO OFL5
L3174   r3 b174             0010101010 (0170)    [register] using X : 0 - C → C                                     p: L3256:  ..1.1.1.1.                          :  0 - C -> C[X]
L3175   r3 b175             0101111010 (0378)    [register] using XS : decrement C                                  p: L3257:  .1.1111.1.                          :  C - 1 -> C[XS]
L3176   r3 b176             1010000111 (0647)    [flow    ] (if no carry then) goTo : 161       → ofl1               p: L3260:  1.1....111  -> L3241                :  IF NO CARRY GO TO OFL1
L3177   r3 b177   ofl5      1110101010 (0938)    [register] using X : exchange A C                                  p: L3261:  111.1.1.1.                  OFL5  :  A EXCHANGE C[X]
L3178   r3 b178             0100010100 (0276)    [status  ] if statusBit ≠ 1 : bit 4                                 p: L3262:  .1...1.1..                          :  IF S4 # 1
L3179   r3 b179             0000100111 (0039)    [flow    ]    (then) goTo : 9                  → rnd0               p: L3263:  ....1..111  -> L3011                :       THEN GO TO RND0
```

```
L3180   r3 b180            1100101010 (0810)    [register] using X : exchange A B
L3181   r3 b181            0000101010 (0042)    [register] using X : clear B
L3182   r3 b182            1000010101 (0533)    [flow    ] jumpSub : 133                  → dsp1
L3183   r3 b183            1100101100 (0812)    [pointer ] if P ≠ value : value 12
L3184   r3 b184            1000011111 (0543)    [flow    ]    (then) goTo : 135           → dsp0z4
L3185   r3 b185            0100001010 (0266)    [register] using X : leftShift A
L3186   r3 b186            1011111011 (0763)    [flow    ] goTo : 190                     → eex3
L3187   r3 b187   eex2      0100000100 (0260)    [status  ] set statusBit : bit 4
L3188   r3 b188            1011010100 (0724)    [status  ] if statusBit ≠ 1 : bit 11
L3189   r3 b189            0001110011 (0115)    [flow    ]    (then) goTo : 28            → dig1
L3190   r3 b190   eex3      1011001110 (0718)    [register] using W : rightShift A
L3191   r3 b191            1110110010 (0946)    [register] using WP : exchange A C
L3192   r3 b192            1100011011 (0795)    [flow    ] goTo : 198                     → eex4
L3193   r3 b193   chs2      1011001110 (0718)    [register] using W : rightShift A
L3194   r3 b194            0100010100 (0276)    [status  ] if statusBit ≠ 1 : bit 4
L3195   r3 b195            1000010011 (0531)    [flow    ]    (then) goTo : 132           → chs3
L3196   r3 b196            1110110010 (0946)    [register] using WP : exchange A C
L3197   r3 b197            0011111010 (0250)    [register] using XS : 0 − C − 1 → C
L3198   r3 b198   eex4      0110001110 (0398)    [register] using W : C → A
L3199   r3 b199            0110111010 (0442)    [register] using XS : if C = 0
L3200   r3 b200            1100101111 (0815)    [flow    ]    (then) goTo : 203           → eex5
L3201   r3 b201            0011011010 (0218)    [register] using XS : clear c
L3202   r3 b202            0010101010 (0170)    [register] using X : 0 − C → C
L3203   r3 b203   eex5      1101001100 (0844)    [pointer ] value → P : value 13
L3204   r3 b204   eex6      0100010110 (0278)    [register] using MS : leftShift A
L3205   r3 b205            0101101010 (0362)    [register] using X : decrement C
L3206   r3 b206            1001111110 (0638)    [register] using S : if A ≥ 1
L3207   r3 b207            1110110011 (0947)    [flow    ]    (then) goTo : 236           → eex8
L3208   r3 b208            1001110110 (0630)    [register] using MS : if A ≥ 1
L3209   r3 b209            1100110011 (0819)    [flow    ]    (then) goTo : 204           → eex6
L3210   r3 b210            0011001010 (0202)    [register] using X : clear c
L3211   r3 b211   den1      1000010101 (0533)    [flow    ] jumpSub : 133                  → dsp1
L3212   r3 b212            1011010110 (0726)    [register] using MS : rightShift A
L3213   r3 b213   den7      0110011110 (0414)    [register] using S : C → A
L3214   r3 b214   den2      1100101100 (0812)    [pointer ] if P ≠ value : value 12
L3215   r3 b215            1101111111 (0895)    [flow    ]    (then) goTo : 223           → den4
L3216   r3 b216            0010001110 (0142)    [register] using W : B → C
L3217   r3 b217            0111101110 (0494)    [register] using W : increment C
L3218   r3 b218            0001001100 (0076)    [pointer ] value → P : value 1
L3219   r3 b219   den3      0100010010 (0274)    [register] using WP : leftShift A
L3220   r3 b220            0000111100 (0060)    [pointer ] increment P
L3221   r3 b221            0110100010 (0418)    [register] using P : if C = 0
L3222   r3 b222            1101101111 (0879)    [flow    ]    (then) goTo : 219           → den3
L3223   r3 b223   den4      1110101110 (0942)    [register] using W : exchange A C
L3224   r3 b224            0011101100 (0236)    [pointer ] if P ≠ value : value 3
```

```
p: L3264:  11..1.1.1.                        :  A EXCHANGE B[X]
p: L3265:  ....1.1.1.                        :  0 -> B[X]
p: L3266:  1....1.1.1  -> L3205              :  JSB DSP1
p: L3267:  11..1.11..                        :  IF P # 12
p: L3270:  1....11111  -> L3207              :       THEN GO TO DSP0Z4
p: L3271:  .1....1.1.                        :  SHIFT LEFT A[X]
p: L3272:  1.11111.11  -> L3276              :  GO TO EEX3
p: L3273:  .1.....1..                  EEX2  :  1 -> S4
p: L3274:  1.11.1.1..                        :  IF S11 # 1
p: L3275:  ...111..11  -> L3034              :       THEN GO TO DIG1
p: L3276:  1.11..111.                  EEX3  :  SHIFT RIGHT A[W]
p: L3277:  111.11..1.                        :  A EXCHANGE C[WP]
p: L3300:  11...11.11  -> L3306              :  GO TO EEX4
p: L3301:  1.11..111.                  CHS2  :  SHIFT RIGHT A[W]
p: L3302:  .1...1.1..                        :  IF S4 # 1
p: L3303:  1....1..11  -> L3204              :       THEN GO TO CHS3
p: L3304:  111.11..1.                        :  A EXCHANGE C[WP]
p: L3305:  ..11111.1.                        :  0 - C - 1 -> C[XS]
p: L3306:  .11...111.                  EEX4  :  C -> A[W]
p: L3307:  .11.111.1.                        :  IF C[XS] = 0
p: L3310:  11..1.1111  -> L3313              :       THEN GO TO EEX5
p: L3311:  ..11.11.1.                        :  0 -> C[XS]
p: L3312:  ..1.1.1.1.                        :  0 - C -> C[X]
p: L3313:  11.1..11..                  EEX5  :  13 -> P
p: L3314:  .1...1.11.                  EEX6  :  SHIFT LEFT A[MS]
p: L3315:  .1.11.1.1.                        :  C - 1 -> C[X]
p: L3316:  1..111111.                        :  IF A[S] >= 1
p: L3317:  111.11..11  -> L3354              :       THEN GO TO EEX8
p: L3320:  1..111.11.                        :  IF A[MS] >= 1
p: L3321:  11..11..11  -> L3314              :       THEN GO TO EEX6
p: L3322:  ..11..1.1.                        :  0 -> C[X]
p: L3323:  1....1.1.1  -> L3205        DEN1  :  JSB DSP1
p: L3324:  1.11.1.11.                        :  SHIFT RIGHT A[MS]
p: L3325:  .11..1111.                  DEN7  :  C -> A[S]
p: L3326:  11..1.11..                  DEN2  :  IF P # 12
p: L3327:  11.1111111  -> L3337              :       THEN GO TO DEN4
p: L3330:  ..1...111.                        :  B -> C[W]
p: L3331:  .1111.111.                        :  C + 1 -> C[W]
p: L3332:  ...1..11..                        :  1 -> P
p: L3333:  .1...1..1.                  DEN3  :  SHIFT LEFT A[WP]
p: L3334:  ....1111..                        :  P + 1 -> P
p: L3335:  .11.1...1.                        :  IF C[P] = 0
p: L3336:  11.11.1111  -> L3333              :       THEN GO TO DEN3
p: L3337:  111.1.111.                  DEN4  :  A EXCHANGE C[W]
p: L3340:  ..111.11..                        :  IF P # 3
```

```
L3225   r3 b225          1110010111 (0919)   [flow    ]    (then) goTo : 229              → den5
L3226   r3 b226          0011001010 (0202)   [register] using X : clear c
L3227   r3 b227          0110000100 (0388)   [status  ] set statusBit : bit 6
L3228   r3 b228          1100011011 (0795)   [flow    ] goTo : 198                      → eex4
L3229   r3 b229   den5    0110010100 (0404)   [status  ] if statusBit ≠ 1 : bit 6
L3230   r3 b230          1110100011 (0931)   [flow    ]    (then) goTo : 232             → den6
L3231   r3 b231          0000011100 (0028)   [pointer ] decrement P
L3232   r3 b232   den6    1010010010 (0658)   [register] using WP : rightShift B
L3233   r3 b233          1100011001 (0793)   [flow    ] jumpSub : 198                   → eex4
L3234   r3 b234   eex7    0000011100 (0028)   [pointer ] decrement P
L3235   r3 b235          0111101010 (0490)   [register] using X : increment C
L3236   r3 b236   eex8    0000000010 (0002)   [register] using P : if B = 0
L3237   r3 b237          1110101011 (0939)   [flow    ]    (then) goTo : 234             → eex7
L3238   r3 b238          1011000100 (0708)   [status  ] set statusBit : bit 11
L3239   r3 b239          1011010110 (0726)   [register] using MS : rightShift A
L3240   r3 b240          1110100110 (0934)   [register] using M : exchange A C
L3241   r3 b241          0100010100 (0276)   [status  ] if statusBit ≠ 1 : bit 4
L3242   r3 b242          1101001111 (0847)   [flow    ]    (then) goTo : 211             → den1
L3243   r3 b243   eex9    1010100101 (0677)   [flow    ] jumpSub : 169                   → ofl4
L3244   r3 b244          1000000111 (0519)   [flow    ] goTo : 129                      → fst2zx
L3245   r3 b245   fst3    1011110110 (0758)   [register] using MS : clear A
L3246   r3 b246   fst4    0111010100 (0468)   [status  ] if statusBit ≠ 1 : bit 7
L3247   r3 b247          1111100111 (0999)   [flow    ]    (then) goTo : 249             → fst5
L3248   r3 b248          0100101000 (0296)   [other   ] (stack) M → stack
L3249   r3 b249   fst5    0111000100 (0452)   [status  ] set statusBit : bit 7
L3250   r3 b250          0011001110 (0206)   [register] using W : clear c
L3251   r3 b251          0101101110 (0366)   [register] using W : decrement C
L3252   r3 b252          0010111110 (0190)   [register] using S : 0 − C → C
L3253   r3 b253          0111111110 (0510)   [register] using S : increment C
L3254   r3 b254          1000101110 (0558)   [register] using W : exchange B C
L3255   r3 b255   retnzx   0000110000 (0048)   [other   ] (flow) return
```

```
p: L3341:  111..1.111  -> L3345              :       THEN GO TO DEN5
p: L3342:  ..11..1.1.                        :  0 -> C[X]
p: L3343:  .11....1..                        :  1 -> S6
p: L3344:  11...11.11  -> L3306              :  GO TO EEX4
p: L3345:  .11..1.1..                DEN5    :  IF S6 # 1
p: L3346:  111.1...11  -> L3350              :       THEN GO TO DEN6
p: L3347:  .....111..                        :  P − 1 -> P
p: L3350:  1.1..1..1.                DEN6    :  SHIFT RIGHT B[WP]
p: L3351:  11...11..1  -> L3306              :  JSB EEX4
p: L3352:  .....111..                EEX7    :  P − 1 -> P
p: L3353:  .1111.1.1.                        :  C + 1 -> C[X]
p: L3354:  ........1.                EEX8    :  IF B[P] = 0
p: L3355:  111.1.1.11  -> L3352              :       THEN GO TO EEX7
p: L3356:  1.11...1..                        :  1 -> S11
p: L3357:  1.11.1.11.                        :  SHIFT RIGHT A[MS]
p: L3360:  111.1..11.                        :  A EXCHANGE C[M]
p: L3361:  .1...1.1..                        :  IF S4 # 1
p: L3362:  11.1..1111  -> L3323              :       THEN GO TO DEN1
p: L3363:  1.1.1..1.1  -> L3251      EEX9    :  JSB OFL4
p: L3364:  1......111  -> L3201              :  GO TO FST2ZX
p: L3365:  1.1111.11.                FST3    :  0 -> A[MS]
p: L3366:  .111.1.1..                FST4    :  IF S7 # 1
p: L3367:  11111..111  -> L3371              :       THEN GO TO FST5
p: L3370:  .1..1.1...                        :  C -> STACK
p: L3371:  .111...1..                FST5    :  1 -> S7
p: L3372:  ..11..111.                        :  0 -> C[W]
p: L3373:  .1.11.111.                        :  C − 1 -> C[W]
p: L3374:  ..1.11111.                        :  0 − C -> C[S]
p: L3375:  .11111111.                        :  C + 1 -> C[S]
p: L3376:  1...1.111.                        :  B EXCHANGE C[W]
p: L3377:  ....11....                RETNZX  :  RETURN
```

```
L4000   r4 b000   prfx       0000000000 (0000)     [no oper ] nOp                                  p: L4000:  ..........                  PRFX  :  NO OPERATION
L4001   r4 b001   prfxz3     1000010011 (0531)     [flow    ] goTo : 132                → pfx1     p: L4001:  1....1..11  -> L4204       PRFXZ3:  GO TO PFX1
L4002   r4 b002   sci1       0101110011 (0371)     [flow    ] goTo : 92                 → sci2     p: L4002:  .1.111..11  -> L4134       SCI1  :  GO TO SCI2
L4003   r4 b003   tenx       1100100011 (0803)     [flow    ] goTo : 200                → tnx2     p: L4003:  11..1...11  -> L4310       TENX  :  GO TO TNX2
L4004   r4 b004   logg       0101100011 (0355)     [flow    ] goTo : 88                 → log2     p: L4004:  .1.11...11  -> L4130       LOGG  :  GO TO LOG2
L4005   r4 b005   tnx3       1100010000 (0784)     [rom     ] selectRom : 6                        p: L4005:  11...1....  -> L6006  *****  TNX3  :  SELECT ROM 6
L4006   r4 b006   xtoy       0110000101 (0389)     [flow    ] jumpSub : 97              → save     p: L4006:  .11....1.1  -> L4141       XTOY  :  JSB SAVE
L4007   r4 b007              0101101011 (0363)     [flow    ] goTo : 90                 → xty1     p: L4007:  .1.11.1.11  -> L4132              :  GO TO XTY1
L4008   r4 b008   dpct       1001100011 (0611)     [flow    ] goTo : 152                → dpc1     p: L4010:  1..11...11  -> L4230       DPCT  :  GO TO DPC1
L4009   r4 b009   percz3     0100001011 (0267)     [flow    ] goTo : 66                 → pct1     p: L4011:  .1....1.11  -> L4102       PERCZ3:  GO TO PCT1
L4010   r4 b010   dmst       1000100100 (0548)     [status  ] clear statusBit : bit 8              p: L4012:  1...1..1..                  DMST  :  0 -> S8
L4011   r4 b011   tdms       0001000011 (0067)     [flow    ] goTo : 16                 → tdm1     p: L4013:  ...1....11  -> L4020       TDMS  :  GO TO TDM1
L4012   r4 b012   stdd       0010000011 (0131)     [flow    ] goTo : 32                 → std1     p: L4014:  ..1.....11  -> L4040       STDD  :  GO TO STD1
L4013   r4 b013   dmsd       0000010000 (0016)     [rom     ] selectRom : 0                        p: L4015:  .....1....  -> L0016  *****  DMSD  :  SELECT ROM 0
L4014   r4 b014   fact       0110000101 (0389)     [flow    ] jumpSub : 97              → save     p: L4016:  .11....1.1  -> L4141       FACT  :  JSB SAVE
L4015   r4 b015              1111111111 (1023)     [flow    ] goTo : 255                → fac2     p: L4017:  1111111111  -> L4377              :  GO TO FAC2
L4016   r4 b016   tdm1       0110000101 (0389)     [flow    ] jumpSub : 97              → save     p: L4020:  .11....1.1  -> L4141       TDM1  :  JSB SAVE
L4017   r4 b017              0000110111 (0055)     [flow    ] goTo : 13                 → dmsd     p: L4021:  ....11.111  -> L4015              :  GO TO DMSD
L4018   r4 b018   dig6       1000001011 (0523)     [flow    ] goTo : 130                → dsp0     p: L4022:  1.....1.11  -> L4202       DIG6  :  GO TO DSP0
L4019   r4 b019   dig5       1000001011 (0523)     [flow    ] goTo : 130                → dsp0     p: L4023:  1.....1.11  -> L4202       DIG5  :  GO TO DSP0
L4020   r4 b020   dig4       1000001011 (0523)     [flow    ] goTo : 130                → dsp0     p: L4024:  1.....1.11  -> L4202       DIG4  :  GO TO DSP0
L4021   r4 b021   sig2       1010010000 (0656)     [rom     ] selectRom : 5                        p: L4025:  1.1..1....  -> L5026  *****  SIG2  :  SELECT ROM 5
L4022   r4 b022   addd       0000000000 (0000)     [no oper ] nOp                                  p: L4026:  ..........                  ADDD  :  NO OPERATION
L4023   r4 b023   adddz3     0110100011 (0419)     [flow    ] goTo : 104                → amd1     p: L4027:  .11.1...11  -> L4150       ADDDZ3:  GO TO AMD1
L4024   r4 b024   tpolz0     1011010111 (0727)     [flow    ] goTo : 181                → tpl3     p: L4030:  1.11.1.111  -> L4265       TPOLZ0:  GO TO TPL3
L4025   r4 b025              0000000000 (0000)     [no oper ] nOp                                  p: L4031:  ..........                  :  NO OPERATION
L4026   r4 b026   dig3       1000001011 (0523)     [flow    ] goTo : 130                → dsp0     p: L4032:  1.....1.11  -> L4202       DIG3  :  GO TO DSP0
L4027   r4 b027   dig2       1000001011 (0523)     [flow    ] goTo : 130                → dsp0     p: L4033:  1.....1.11  -> L4202       DIG2  :  GO TO DSP0
L4028   r4 b028   dig1       1000001011 (0523)     [flow    ] goTo : 130                → dsp0     p: L4034:  1.....1.11  -> L4202       DIG1  :  GO TO DSP0
L4029   r4 b029              0000000000 (0000)     [no oper ] nOp                                  p: L4035:  ..........                  :  NO OPERATION
L4030   r4 b030   mult       0000000000 (0000)     [no oper ] nOp                                  p: L4036:  ..........                  MULT  :  NO OPERATION
L4031   r4 b031   multz3     0110111011 (0443)     [flow    ] goTo : 110                → amd4     p: L4037:  .11.111.11  -> L4156       MULTZ3:  GO TO AMD4
L4032   r4 b032   std1       0110000001 (0385)     [flow    ] jumpSub : 96              → sav1     p: L4040:  .11......1  -> L4140       STD1  :  JSB SAV1
L4033   r4 b033              1010010000 (0656)     [rom     ] selectRom : 5                        p: L4041:  1.1..1....  -> L5042  *****       :  SELECT ROM 5
L4034   r4 b034   sigmz3     1101000011 (0835)     [flow    ] goTo : 208                → sgma     p: L4042:  11.1....11  -> L4320       SIGMZ3:  GO TO SGMA
L4035   r4 b035   dspt       0111001011 (0459)     [flow    ] goTo : 114                → piii     p: L4043:  .111..1.11  -> L4162       DSPT  :  GO TO PIII
L4036   r4 b036   dig0       0111000111 (0455)     [flow    ] goTo : 113                → cons     p: L4044:  .111...111  -> L4161       DIG0  :  GO TO CONS
L4037   r4 b037              0000000000 (0000)     [no oper ] nOp                                  p: L4045:  ..........                  :  NO OPERATION
L4038   r4 b038   divdz3     0111001111 (0463)     [flow    ] goTo : 115                → amd5     p: L4046:  .111..1111  -> L4163       DIVDZ3:  GO TO AMD5
L4039   r4 b039   atn2       0101000100 (0324)     [status  ] set statusBit : bit 5                p: L4047:  .1.1...1..                  ATN2  :  1 -> S5
L4040   r4 b040   atan       0101111001 (0377)     [flow    ] jumpSub : 94              → sav9     p: L4050:  .1.1111..1  -> L4136       ATAN  :  JSB SAV9
L4041   r4 b041              0100000111 (0263)     [flow    ] goTo : 65                 → sqt1     p: L4051:  .1.....111  -> L4101              :  GO TO SQT1
L4042   r4 b042   acos       0100101111 (0303)     [flow    ] goTo : 75                 → acs1     p: L4052:  .1..1.1111  -> L4113       ACOS  :  GO TO ACS1
L4043   r4 b043   asin       0010011111 (0159)     [flow    ] goTo : 39                 → atn2     p: L4053:  ..1..11111  -> L4047       ASIN  :  GO TO ATN2
L4044   r4 b044   trec       1001001111 (0591)     [flow    ] goTo : 147                → trc1     p: L4054:  1..1..1111  -> L4223       TREC  :  GO TO TRC1
```

```
L4045   r4 b045   tpolz3     1010100011 (0675)     [flow    ] goTo : 168                          → tpl1
L4046   r4 b046   sqrt       0110000101 (0389)     [flow    ] jumpSub : 97                        → save
L4047   r4 b047              0100000111 (0263)     [flow    ] goTo : 65                           → sqt1
L4048   r4 b048              0000000000 (0000)     [no oper ] nOp
L4049   r4 b049              0000000000 (0000)     [no oper ] nOp
L4050   r4 b050   dig9       1111101110 (1006)     [register] using W : increment A
L4051   r4 b051   dig8       1111101110 (1006)     [register] using W : increment A
L4052   r4 b052   dig7       0111000011 (0451)     [flow    ] (if no carry then) goTo : 112       → con1
L4053   r4 b053   tpl6       1010010000 (0656)     [rom     ] selectRom : 5
L4054   r4 b054   subt       0000000000 (0000)     [no oper ] nOp
L4055   r4 b055   subtz3     0110101111 (0431)     [flow    ] goTo : 107                          → amd2
L4056   r4 b056   cler       0110000101 (0389)     [flow    ] jumpSub : 97                        → save
L4057   r4 b057              1111010111 (0983)     [flow    ] goTo : 245                          → clr2
L4058   r4 b058   grad       1111101110 (1006)     [register] using W : increment A
L4059   r4 b059   radn       0011111111 (0255)     [flow    ] (if no carry then) goTo : 63        → mode
L4060   r4 b060   clok       0000000000 (0000)     [no oper ] nOp
L4061   r4 b061              0000000000 (0000)     [no oper ] nOp
L4062   r4 b062   degr       1101101110 (0878)     [register] using W : decrement A
L4063   r4 b063   mode       0000001100 (0012)     [pointer ] value → P : value 0
L4064   r4 b064              0100111011 (0315)     [flow    ] goTo : 78                           → shft
L4065   r4 b065   sqt1       0110010000 (0400)     [rom     ] selectRom : 3
L4066   r4 b066   pct1       0110000001 (0385)     [flow    ] jumpSub : 96                        → sav1
L4067   r4 b067              1000100100 (0548)     [status  ] clear statusBit : bit 8
L4068   r4 b068   c100       1100101000 (0808)     [other   ] (stack) rotateDown
L4069   r4 b069              0100101000 (0296)     [other   ] (stack) M → stack
L4070   r4 b070              0101101010 (0362)     [register] using X : decrement C
L4071   r4 b071              0101101010 (0362)     [register] using X : decrement C
L4072   r4 b072              1000010100 (0532)     [status  ] if statusBit ≠ 1 : bit 8
L4073   r4 b073   mul0z3     1010001111 (0655)     [flow    ]    (then) goTo : 163                → mul0
L4074   r4 b074   div0z3     1010000111 (0647)     [flow    ] goTo : 161                          → div0
L4075   r4 b075   acs1       0101111001 (0377)     [flow    ] jumpSub : 94                        → sav9
L4076   r4 b076              0110010000 (0400)     [rom     ] selectRom : 3
L4077   r4 b077   trc2       0110010000 (0400)     [rom     ] selectRom : 3
L4078   r4 b078   shft       0100001110 (0270)     [register] using W : leftShift A
L4079   r4 b079              0000111100 (0060)     [pointer ] increment P
L4080   r4 b080              1101101100 (0876)     [pointer ] if P ≠ value : value 13
L4081   r4 b081              0100111011 (0315)     [flow    ]    (then) goTo : 78                 → shft
L4082   r4 b082   mrg0       0010101000 (0168)     [other   ] (register) exchange C M
L4083   r4 b083              1110100010 (0930)     [register] using P : exchange A C
L4084   r4 b084   mreg       0010101000 (0168)     [other   ] (register) exchange C M
L4085   r4 b085              0110010000 (0400)     [rom     ] selectRom : 3
L4086   r4 b086              0000000000 (0000)     [no oper ] nOp
L4087   r4 b087              0000000000 (0000)     [no oper ] nOp
L4088   r4 b088   log2       0101000100 (0324)     [status  ] set statusBit : bit 5
L4089   r4 b089              0110010000 (0400)     [rom     ] selectRom : 3
```

```
p: L4055:  1.1.1...11  -> L4250          TPOLZ3:  GO TO TPL1
p: L4056:  .11....1.1  -> L4141          SQRT  :  JSB SAVE
p: L4057:  .1.....111  -> L4101                :  GO TO SQT1
p: L4060:  ..........                          :  NO OPERATION
p: L4061:  ..........                          :  NO OPERATION
p: L4062:  11111.111.                  DIG9  :  A + 1 -> A[W]
p: L4063:  11111.111.                  DIG8  :  A + 1 -> A[W]
p: L4064:  .111....11  -> L4160          DIG7  :  IF NO CARRY GO TO CON1
p: L4065:  1.1..1....  -> L5066   *****  TPL6  :  SELECT ROM 5
p: L4066:  ..........                    SUBT  :  NO OPERATION
p: L4067:  .11.1.1111  -> L4153          SUBTZ3:  GO TO AMD2
p: L4070:  .11....1.1  -> L4141          CLER  :  JSB SAVE
p: L4071:  1111.1.111  -> L4365                :  GO TO CLR2
p: L4072:  11111.111.                    GRAD  :  A + 1 -> A[W]
p: L4073:  ..11111111  -> L4077          RADN  :  IF NO CARRY GO TO MODE
p: L4074:  ..........                    CLOK  :  NO OPERATION
p: L4075:  ..........                          :  NO OPERATION
p: L4076:  11.11.111.                    DEGR  :  A - 1 -> A[W]
p: L4077:  ......11..                    MODE  :  0 -> P
p: L4100:  .1..111.11  -> L4116                :  GO TO SHFT
p: L4101:  .11..1....  -> L3102   *****  SQT1  :  SELECT ROM 3
p: L4102:  .11......1  -> L4140          PCT1  :  JSB SAV1
p: L4103:  1...1..1..                          :  0 -> S8
p: L4104:  11..1.1...                    C100  :  DOWN ROTATE
p: L4105:  .1..1.1...                          :  C -> STACK
p: L4106:  .1.11.1.1.                          :  C - 1 -> C[X]
p: L4107:  .1.11.1.1.                          :  C - 1 -> C[X]
p: L4110:  1....1.1..                          :  IF S8 # 1
p: L4111:  1.1...1111  -> L4243          MUL0Z3:       THEN GO TO MUL0
p: L4112:  1.1....111  -> L4241          DIV0Z3:  GO TO DIV0
p: L4113:  .1.1111..1  -> L4136          ACS1  :  JSB SAV9
p: L4114:  .11..1....  -> L3115   *****        :  SELECT ROM 3
p: L4115:  .11..1....  -> L3116   *****  TRC2  :  SELECT ROM 3
p: L4116:  .1....111.                    SHFT  :  SHIFT LEFT A[W]
p: L4117:  ....1111..                          :  P + 1 -> P
p: L4120:  11.11.11..                          :  IF P # 13
p: L4121:  .1..111.11  -> L4116                :       THEN GO TO SHFT
p: L4122:  ..1.1.1...                    MRG0  :  C EXCHANGE M
p: L4123:  111.1...1.                          :  A EXCHANGE C[P]
p: L4124:  ..1.1.1...                    MREG  :  C EXCHANGE M
p: L4125:  .11..1....  -> L3126   *****        :  SELECT ROM 3
p: L4126:  ..........                          :  NO OPERATION
p: L4127:  ..........                          :  NO OPERATION
p: L4130:  .1.1...1..                    LOG2  :  1 -> S5
p: L4131:  .11..1....  -> L3132   *****        :  SELECT ROM 3
```

```
L4090   r4 b090   xty1       1000011101 (0541)     [flow    ] jumpSub : 135                    → exch
L4091   r4 b091              0110010000 (0400)     [rom     ] selectRom : 3
L4092   r4 b092   sci2       1010100100 (0676)     [status  ] clear statusBit : bit 10
L4093   r4 b093              0110010000 (0400)     [rom     ] selectRom : 3
L4094   r4 b094   sav9       0001000100 (0068)     [status  ] set statusBit : bit 1
L4095   r4 b095              0110000111 (0391)     [flow    ] goTo : 97                        → save
L4096   r4 b096   sav1       0011000100 (0196)     [status  ] set statusBit : bit 3
L4097   r4 b097   save       1010000100 (0644)     [status  ] set statusBit : bit 10
L4098   r4 b098              1100010000 (0784)     [rom     ] selectRom : 6
L4099   r4 b099   savx       1100010000 (0784)     [rom     ] selectRom : 6
L4100   r4 b100   adr9       1100010000 (0784)     [rom     ] selectRom : 6
L4101   r4 b101   sav2       1010000100 (0644)     [status  ] set statusBit : bit 10
L4102   r4 b102              0011000100 (0196)     [status  ] set statusBit : bit 3
L4103   r4 b103              0110001111 (0399)     [flow    ] goTo : 99                        → savx
L4104   r4 b104   amd1       0110000100 (0388)     [status  ] set statusBit : bit 6
L4105   r4 b105              0100000100 (0260)     [status  ] set statusBit : bit 4
L4106   r4 b106              0111010111 (0471)     [flow    ] goTo : 117                       → amd7
L4107   r4 b107   amd2       0110100100 (0420)     [status  ] clear statusBit : bit 6
L4108   r4 b108   amd3       0100000100 (0260)     [status  ] set statusBit : bit 4
L4109   r4 b109              0111010111 (0471)     [flow    ] goTo : 117                       → amd7
L4110   r4 b110   amd4       0110000100 (0388)     [status  ] set statusBit : bit 6
L4111   r4 b111              0111010011 (0467)     [flow    ] goTo : 116                       → amd6
L4112   r4 b112   con1       1111101110 (1006)     [register] using W : increment A
L4113   r4 b113   cons       1100010000 (0784)     [rom     ] selectRom : 6
L4114   r4 b114   piii       1100010000 (0784)     [rom     ] selectRom : 6
L4115   r4 b115   amd5       0110100100 (0420)     [status  ] clear statusBit : bit 6
L4116   r4 b116   amd6       0100100100 (0292)     [status  ] clear statusBit : bit 4
L4117   r4 b117   amd7       0010010100 (0148)     [status  ] if statusBit ≠ 1 : bit 2
L4118   r4 b118              0111111111 (0511)     [flow    ]    (then) goTo : 127             → amd8
L4119   r4 b119              1010100100 (0676)     [status  ] clear statusBit : bit 10
L4120   r4 b120              0110010000 (0400)     [rom     ] selectRom : 3
L4121   r4 b121   am13       1010100100 (0676)     [status  ] clear statusBit : bit 10
L4122   r4 b122              1010010000 (0656)     [rom     ] selectRom : 5
L4123   r4 b123              0000000000 (0000)     [no oper ] nOp
L4124   r4 b124   amd9z3     1000101111 (0559)     [flow    ] goTo : 139                       → amd9
L4125   r4 b125              0000000000 (0000)     [no oper ] nOp
L4126   r4 b126   fst1       0110010000 (0400)     [rom     ] selectRom : 3
L4127   r4 b127   amd8       0110000101 (0389)     [flow    ] jumpSub : 97                     → save
L4128   r4 b128              0110101000 (0424)     [other   ] (stack) stack → A
L4129   r4 b129              0111100111 (0487)     [flow    ] goTo : 121                       → am13
L4130   r4 b130   dsp0       1010100100 (0676)     [status  ] clear statusBit : bit 10
L4131   r4 b131              1000010111 (0535)     [flow    ] goTo : 133                       → dspx
L4132   r4 b132   pfx1       1010000100 (0644)     [status  ] set statusBit : bit 10
L4133   r4 b133   dspx       1001100100 (0612)     [status  ] clear statusBit : bit 9
L4134   r4 b134              0110010000 (0400)     [rom     ] selectRom : 3

p: L4132:  1....111.1  -> L4207          XTY1  :  JSB EXCH
p: L4133:  .11..1....  -> L3134   *****        :  SELECT ROM 3
p: L4134:  1.1.1..1..                    SCI2  :  0 -> S10
p: L4135:  .11..1....  -> L3136   *****        :  SELECT ROM 3
p: L4136:  ...1...1..                    SAV9  :  1 -> S1
p: L4137:  .11....111  -> L4141                :  GO TO SAVE
p: L4140:  ..11...1..                    SAV1  :  1 -> S3
p: L4141:  1.1....1..                    SAVE  :  1 -> S10
p: L4142:  11...1....  -> L6143   *****        :  SELECT ROM 6
p: L4143:  11...1....  -> L6144   *****  SAVX  :  SELECT ROM 6
p: L4144:  11...1....  -> L6145   *****  ADR9  :  SELECT ROM 6
p: L4145:  1.1....1..                    SAV2  :  1 -> S10
p: L4146:  ..11...1..                          :  1 -> S3
p: L4147:  .11...1111  -> L4143                :  GO TO SAVX
p: L4150:  .11....1..                    AMD1  :  1 -> S6
p: L4151:  .1.....1..                          :  1 -> S4
p: L4152:  .111.1.111  -> L4165                :  GO TO AMD7
p: L4153:  .11.1..1..                    AMD2  :  0 -> S6
p: L4154:  .1.....1..                    AMD3  :  1 -> S4
p: L4155:  .111.1.111  -> L4165                :  GO TO AMD7
p: L4156:  .11....1..                    AMD4  :  1 -> S6
p: L4157:  .111.1..11  -> L4164                :  GO TO AMD6
p: L4160:  11111.111.                    CON1  :  A + 1 -> A[W]
p: L4161:  11...1....  -> L6162   *****  CONS  :  SELECT ROM 6
p: L4162:  11...1....  -> L6163   *****  PIII  :  SELECT ROM 6
p: L4163:  .11.1..1..                    AMD5  :  0 -> S6
p: L4164:  .1..1..1..                    AMD6  :  0 -> S4
p: L4165:  ..1..1.1..                    AMD7  :  IF S2 # 1
p: L4166:  0.111111111 -> L4177                :       THEN GO TO AMD8
p: L4167:  1.1.1..1..                          :  0 -> S10
p: L4170:  .11..1....  -> L3171   *****        :  SELECT ROM 3
p: L4171:  1.1.1..1..                    AM13  :  0 -> S10
p: L4172:  1.1..1....  -> L5173   *****        :  SELECT ROM 5
p: L4173:  ..........                          :  NO OPERATION
p: L4174:  1...1.1111  -> L4213          AMD9Z3:  GO TO AMD9
p: L4175:  ..........                          :  NO OPERATION
p: L4176:  .11..1....  -> L3177   *****  FST1  :  SELECT ROM 3
p: L4177:  .11....1.1  -> L4141          AMD8  :  JSB SAVE
p: L4200:  .11.1.1...                          :  STACK -> A
p: L4201:  .1111..111  -> L4171                :  GO TO AM13
p: L4202:  1.1.1..1..                    DSP0  :  0 -> S10
p: L4203:  1....1.111  -> L4205                :  GO TO DSPX
p: L4204:  1.1....1..                    PFX1  :  1 -> S10
p: L4205:  1..11..1..                    DSPX  :  0 -> S9
p: L4206:  .11..1....  -> L3207   *****        :  SELECT ROM 3
```

```
L4135  r4 b135  exch   0110101000 (0424)  [other   ] (stack) stack → A                                  p: L4207:  .11.1.1...                  EXCH  :  STACK -> A
L4136  r4 b136  exc1   0100101000 (0296)  [other   ] (stack) M → stack                                  p: L4210:  .1..1.1...                  EXC1  :  C -> STACK
L4137  r4 b137         1110101110 (0942)  [register] using W : exchange A C                            p: L4211:  111.1.111.                        :  A EXCHANGE C[W]
L4138  r4 b138         0000110000 (0048)  [other   ] (flow) return                                      p: L4212:  ....11....                        :  RETURN
L4139  r4 b139  amd9   1001010100 (0596)  [status  ] if statusBit ≠ 1 : bit 9                          p: L4213:  1..1.1.1..                  AMD9  :  IF S9 # 1
L4140  r4 b140         1001000011 (0579)  [flow    ]    (then) goTo : 144              → am12           p: L4214:  1..1....11  -> L4220               :       THEN GO TO AM12
L4141  r4 b141  am10   0110000001 (0385)  [flow    ] jumpSub : 96                 → sav1           p: L4215:  .11......1  -> L4140       AM10  :  JSB SAV1
L4142  r4 b142  am11   1110101110 (0942)  [register] using W : exchange A C                            p: L4216:  111.1.111.                  AM11  :  A EXCHANGE C[W]
L4143  r4 b143         0111100111 (0487)  [flow    ] goTo : 121                   → am13           p: L4217:  .1111..111  -> L4171               :  GO TO AM13
L4144  r4 b144  am12   0110010101 (0405)  [flow    ] jumpSub : 101                → sav2           p: L4220:  .11..1.1.1  -> L4145       AM12  :  JSB SAV2
L4145  r4 b145         1011110000 (0752)  [other   ] (data) C → data                                    p: L4221:  1.1111....                        :  C -> DATA
L4146  r4 b146         1000111111 (0575)  [flow    ] goTo : 143                           N         p: L4222:  1...111.11  -> L4216               :  GO TO AM11
L4147  r4 b147  trc1   0101111001 (0377)  [flow    ] jumpSub : 94                 → sav9           p: L4223:  .1.1111..1  -> L4136       TRC1  :  JSB SAV9
L4148  r4 b148         0110101000 (0424)  [other   ] (stack) stack → A                                  p: L4224:  .11.1.1...                        :  STACK -> A
L4149  r4 b149         0011000100 (0196)  [status  ] set statusBit : bit 3                             p: L4225:  ..11...1..                        :  1 -> S3
L4150  r4 b150         1010100100 (0676)  [status  ] clear statusBit : bit 10                          p: L4226:  1.1.1..1..                        :  0 -> S10
L4151  r4 b151         0100110111 (0311)  [flow    ] goTo : 77                    → trc2           p: L4227:  .1..11.111  -> L4115               :  GO TO TRC2
L4152  r4 b152  dpc1   0110000001 (0385)  [flow    ] jumpSub : 96                 → sav1           p: L4230:  .11......1  -> L4140       DPC1  :  JSB SAV1
L4153  r4 b153         1100101000 (0808)  [other   ] (stack) rotateDown                               p: L4231:  11..1.1...                        :  DOWN ROTATE
L4154  r4 b154         0100101000 (0296)  [other   ] (stack) M → stack                                  p: L4232:  .1..1.1...                        :  C -> STACK
L4155  r4 b155         1010011001 (0665)  [flow    ] jumpSub : 166                → sub1           p: L4233:  1.1..11..1  -> L4246               :  JSB SUB1
L4156  r4 b156         0100010011 (0275)  [flow    ] goTo : 68                    → c100           p: L4234:  .1...1..11  -> L4104               :  GO TO C100
L4157  r4 b157         0000000000 (0000)  [no oper ] nOp                                            p: L4235:  ..........                        :  NO OPERATION
L4158  r4 b158         0000000000 (0000)  [no oper ] nOp                                            p: L4236:  ..........                        :  NO OPERATION
L4159  r4 b159  rcxy   1010010000 (0656)  [rom     ] selectRom : 5                                  p: L4237:  1.1..1....  -> L5240  *****  RCXY  :  SELECT ROM 5
L4160  r4 b160  tkraz3 0011010000 (0208)  [rom     ] keys → romAddress : 1                          p: L4240:  ..11.1....                  TKRAZ3:  KEYS -> ROM ADDRESS
L4161  r4 b161  div0   0011100100 (0228)  [status  ] clear statusBit : bit 3                          p: L4241:  ..111..1..                  DIV0  :  0 -> S3
L4162  r4 b162         1010010111 (0663)  [flow    ] goTo : 165                   → div1           p: L4242:  1.1..1.111  -> L4245               :  GO TO DIV1
L4163  r4 b163  mul0   0011100100 (0228)  [status  ] clear statusBit : bit 3                          p: L4243:  ..111..1..                  MUL0  :  0 -> S3
L4164  r4 b164  mul1   0010010000 (0144)  [rom     ] selectRom : 1                                  p: L4244:  ..1..1....  -> L1245  *****  MUL1  :  SELECT ROM 1
L4165  r4 b165  div1   0010010000 (0144)  [rom     ] selectRom : 1                                  p: L4245:  ..1..1....  -> L1246  *****  DIV1  :  SELECT ROM 1
L4166  r4 b166  sub1   0011111110 (0254)  [register] using S : 0 − C − 1 → C                          p: L4246:  ..1111111.                  SUB1  :  0 - C - 1 -> C[S]
L4167  r4 b167  add1   0000010000 (0016)  [rom     ] selectRom : 0                                  p: L4247:  .....1....  -> L0250  *****  ADD1  :  SELECT ROM 0
L4168  r4 b168  tpl1   0101111001 (0377)  [flow    ] jumpSub : 94                 → sav9           p: L4250:  .1.1111..1  -> L4136       TPL1  :  JSB SAV9
L4169  r4 b169         0011000100 (0196)  [status  ] set statusBit : bit 3                             p: L4251:  ..11...1..                        :  1 -> S3
L4170  r4 b170         0100100100 (0292)  [status  ] clear statusBit : bit 4                          p: L4252:  .1..1..1..                        :  0 -> S4
L4171  r4 b171         0110111110 (0446)  [register] using S : if C = 0                               p: L4253:  .11.11111.                        :  IF C[S] = 0
L4172  r4 b172         1010111011 (0699)  [flow    ]    (then) goTo : 174              → tpl2           p: L4254:  1.1.111.11  -> L4256               :       THEN GO TO TPL2
L4173  r4 b173         0100000100 (0260)  [status  ] set statusBit : bit 4                             p: L4255:  .1.....1..                        :  1 -> S4
L4174  r4 b174  tpl2   1100101000 (0808)  [other   ] (stack) rotateDown                               p: L4256:  11..1.1...                  TPL2  :  DOWN ROTATE
L4175  r4 b175         1000100001 (0545)  [flow    ] jumpSub : 136                → exc1           p: L4257:  1...1....1  -> L4210               :  JSB EXC1
L4176  r4 b176         1001100110 (0614)  [register] using M : if A ≥ 1                              p: L4260:  1..11..11.                        :  IF A[M] >= 1
L4177  r4 b177         0011010111 (0215)  [flow    ]    (then) goTo : 53               → tpl6           p: L4261:  ..11.1.111  -> L4065               :       THEN GO TO TPL6
L4178  r4 b178         0011010010 (0210)  [register] using WP : clear c                               p: L4262:  ..11.1..1.                        :  0 -> C[WP]
L4179  r4 b179         0111100010 (0482)  [register] using P : increment C                            p: L4263:  .1111...1.                        :  C + 1 -> C[P]
```

```
L4180   r4 b180             0011010101 (0213)   [flow    ] (if no carry then) jumpSub : 53    → tpl6
L4181   r4 b181   tpl3      0001100100 (0100)   [status  ] clear statusBit : bit 1
L4182   r4 b182             1000011101 (0541)   [flow    ] jumpSub : 135                      → exch
L4183   r4 b183             0110001110 (0398)   [register] using W : C → A
L4184   r4 b184             1010010001 (0657)   [flow    ] jumpSub : 164                      → mul1
L4185   r4 b185             1011110000 (0752)   [other   ] (data) C → data
L4186   r4 b186             1100101101 (0813)   [flow    ] jumpSub : 203                      → rest
L4187   r4 b187             0110010001 (0401)   [flow    ] jumpSub : 100                      → adr9
L4188   r4 b188             1010010001 (0657)   [flow    ] jumpSub : 164                      → mul1
L4189   r4 b189             1011111000 (0760)   [other   ] (data) data → C
L4190   r4 b190             1010011101 (0669)   [flow    ] jumpSub : 167                      → add1
L4191   r4 b191             1101111010 (0890)   [register] using XS : decrement A
L4192   r4 b192             1101111010 (0890)   [register] using XS : decrement A
L4193   r4 b193             1101111010 (0890)   [register] using XS : decrement A
L4194   r4 b194             1100010111 (0791)   [flow    ] (if no carry then) goTo : 197      → tpl5
L4195   r4 b195             0101111010 (0378)   [register] using XS : decrement C
L4196   r4 b196             1111000101 (0965)   [flow    ] (if no carry then) jumpSub : 241   → reg9
L4197   r4 b197   tpl5      0110001110 (0398)   [register] using W : C → A
L4198   r4 b198             0100000101 (0261)   [flow    ] jumpSub : 65                       → sqt1
L4199   r4 b199             1111000111 (0967)   [flow    ] goTo : 241                         → reg9
L4200   r4 b200   tnx2      0110000101 (0389)   [flow    ] jumpSub : 97                       → save
L4201   r4 b201             0010000100 (0132)   [status  ] set statusBit : bit 2
L4202   r4 b202             0000010111 (0023)   [flow    ] goTo : 5                           → tnx3
L4203   r4 b203   rest      0011001110 (0206)   [register] using W : clear c
L4204   r4 b204             1001110000 (0624)   [other   ] (data) C → dataAddress
L4205   r4 b205             0000000000 (0000)   [no oper ] nOp
L4206   r4 b206             1011111000 (0760)   [other   ] (data) data → C
L4207   r4 b207             0000110000 (0048)   [other   ] (flow) return
L4208   r4 b208   sgma      1001010100 (0596)   [status  ] if statusBit ≠ 1 : bit 9
L4209   r4 b209             1101001111 (0847)   [flow    ]    (then) goTo : 211               → sig1
L4210   r4 b210             1001111111 (0639)   [flow    ] goTo : 159                         → rcxy
L4211   r4 b211   sig1      0110000001 (0385)   [flow    ] jumpSub : 96                       → sav1
L4212   r4 b212             0001010111 (0087)   [flow    ] goTo : 21                          → sig2
L4213   r4 b213             0000000000 (0000)   [no oper ] nOp
L4214   r4 b214             0000000000 (0000)   [no oper ] nOp
L4215   r4 b215             0000000000 (0000)   [no oper ] nOp
L4216   r4 b216             0000000000 (0000)   [no oper ] nOp
L4217   r4 b217             0000000000 (0000)   [no oper ] nOp
L4218   r4 b218             0000000000 (0000)   [no oper ] nOp
L4219   r4 b219             0000000000 (0000)   [no oper ] nOp
L4220   r4 b220             0000000000 (0000)   [no oper ] nOp
L4221   r4 b221             0000000000 (0000)   [no oper ] nOp
L4222   r4 b222             0000000000 (0000)   [no oper ] nOp
L4223   r4 b223             0000000000 (0000)   [no oper ] nOp
L4224   r4 b224             0000000000 (0000)   [no oper ] nOp
```

```
p: L4264:  ..11.1.1.1  -> L4065              :  JSB TPL6
p: L4265:  ...11..1..                 TPL3   :  0 -> S1
p: L4266:  1....111.1  -> L4207              :  JSB EXCH
p: L4267:  .11...111.                        :  C -> A[W]
p: L4270:  1.1..1...1  -> L4244              :  JSB MUL1
p: L4271:  1.1111....                        :  C -> DATA
p: L4272:  11..1.11.1  -> L4313              :  JSB REST
p: L4273:  .11..1...1  -> L4144              :  JSB ADR9
p: L4274:  1.1..1...1  -> L4244              :  JSB MUL1
p: L4275:  1.11111...                        :  DATA -> C
p: L4276:  1.1..111.1  -> L4247              :  JSB ADD1
p: L4277:  11.1111.1.                        :  A - 1 -> A[XS]
p: L4300:  11.1111.1.                        :  A - 1 -> A[XS]
p: L4301:  11.1111.1.                        :  A - 1 -> A[XS]
p: L4302:  11...1.111  -> L4305              :  IF NO CARRY GO TO TPL5
p: L4303:  .1.1111.1.                        :  C - 1 -> C[XS]
p: L4304:  1111...1.1  -> L4361              :  JSB REG9
p: L4305:  .11...111.                 TPL5   :  C -> A[W]
p: L4306:  .1.....1.1  -> L4101              :  JSB SQT1
p: L4307:  1111...111  -> L4361              :  GO TO REG9
p: L4310:  .11....1.1  -> L4141       TNX2   :  JSB SAVE
p: L4311:  ..1....1..                        :  1 -> S2
p: L4312:  .....1.111  -> L4005              :  GO TO TNX3
p: L4313:  ..11..111.                 REST   :  0 -> C[W]
p: L4314:  1..111....                        :  C -> DATA ADDRESS
p: L4315:  ..........                        :  NO OPERATION
p: L4316:  1.11111...                        :  DATA -> C
p: L4317:  ....11....                        :  RETURN
p: L4320:  1..1.1.1..                 SGMA   :  IF S9 # 1
p: L4321:  11.1..1111  -> L4323              :       THEN GO TO SIG1
p: L4322:  1..1111111  -> L4237              :  GO TO RCXY
p: L4323:  .11......1  -> L4140       SIG1   :  JSB SAV1
p: L4324:  ...1.1.111  -> L4025              :  GO TO SIG2
p: L4325:  ..........                        :  NO OPERATION
p: L4326:  ..........                        :  NO OPERATION
p: L4327:  ..........                        :  NO OPERATION
p: L4330:  ..........                        :  NO OPERATION
p: L4331:  ..........                        :  NO OPERATION
p: L4332:  ..........                        :  NO OPERATION
p: L4333:  ..........                        :  NO OPERATION
p: L4334:  ..........                        :  NO OPERATION
p: L4335:  ..........                        :  NO OPERATION
p: L4336:  ..........                        :  NO OPERATION
p: L4337:  ..........                        :  NO OPERATION
p: L4340:  ..........                        :  NO OPERATION
```

```
L4225   r4 b225             0000000000 (0000)     [no oper ] nOp                                p: L4341:  .........                        :  NO OPERATION
L4226   r4 b226             0000000000 (0000)     [no oper ] nOp                                p: L4342:  .........                        :  NO OPERATION
L4227   r4 b227             0000000000 (0000)     [no oper ] nOp                                p: L4343:  .........                        :  NO OPERATION
L4228   r4 b228             0000000000 (0000)     [no oper ] nOp                                p: L4344:  .........                        :  NO OPERATION
L4229   r4 b229             0000000000 (0000)     [no oper ] nOp                                p: L4345:  .........                        :  NO OPERATION
L4230   r4 b230             0000000000 (0000)     [no oper ] nOp                                p: L4346:  .........                        :  NO OPERATION
L4231   r4 b231             0000000000 (0000)     [no oper ] nOp                                p: L4347:  .........                        :  NO OPERATION
L4232   r4 b232             0000000000 (0000)     [no oper ] nOp                                p: L4350:  .........                        :  NO OPERATION
L4233   r4 b233             0000000000 (0000)     [no oper ] nOp                                p: L4351:  .........                        :  NO OPERATION
L4234   r4 b234             0000000000 (0000)     [no oper ] nOp                                p: L4352:  .........                        :  NO OPERATION
L4235   r4 b235             0000000000 (0000)     [no oper ] nOp                                p: L4353:  .........                        :  NO OPERATION
L4236   r4 b236             0000000000 (0000)     [no oper ] nOp                                p: L4354:  .........                        :  NO OPERATION
L4237   r4 b237             0000000000 (0000)     [no oper ] nOp                                p: L4355:  .........                        :  NO OPERATION
L4238   r4 b238             0000000000 (0000)     [no oper ] nOp                                p: L4356:  .........                        :  NO OPERATION
L4239   r4 b239             0000000000 (0000)     [no oper ] nOp                                p: L4357:  .........                        :  NO OPERATION
L4240   r4 b240             0000000000 (0000)     [no oper ] nOp                                p: L4360:  .........                        :  NO OPERATION
L4241   r4 b241   reg9       0000010000 (0016)     [rom     ] selectRom : 0                      p: L4361:  .....1....  -> L0362   *****  REG9  :  SELECT ROM 0
L4242   r4 b242             0000000000 (0000)     [no oper ] nOp                                p: L4362:  .........                        :  NO OPERATION
L4243   r4 b243             0000000000 (0000)     [no oper ] nOp                                p: L4363:  .........                        :  NO OPERATION
L4244   r4 b244             0000000000 (0000)     [no oper ] nOp                                p: L4364:  .........                        :  NO OPERATION
L4245   r4 b245   clr2       1010010000 (0656)     [rom     ] selectRom : 5                      p: L4365:  1.1..1....  -> L5366   *****  CLR2  :  SELECT ROM 5
L4246   r4 b246             0000000000 (0000)     [no oper ] nOp                                p: L4366:  .........                        :  NO OPERATION
L4247   r4 b247             0000000000 (0000)     [no oper ] nOp                                p: L4367:  .........                        :  NO OPERATION
L4248   r4 b248             0000000000 (0000)     [no oper ] nOp                                p: L4370:  .........                        :  NO OPERATION
L4249   r4 b249             0000000000 (0000)     [no oper ] nOp                                p: L4371:  .........                        :  NO OPERATION
L4250   r4 b250             0000000000 (0000)     [no oper ] nOp                                p: L4372:  .........                        :  NO OPERATION
L4251   r4 b251             0000000000 (0000)     [no oper ] nOp                                p: L4373:  .........                        :  NO OPERATION
L4252   r4 b252             0000000000 (0000)     [no oper ] nOp                                p: L4374:  .........                        :  NO OPERATION
L4253   r4 b253             0000000000 (0000)     [no oper ] nOp                                p: L4375:  .........                        :  NO OPERATION
L4254   r4 b254   retnzx     0000110000 (0048)     [other   ] (flow) return                      p: L4376:  ....11....                 RETNZX:  RETURN
L4255   r4 b255   fac2       1100010000 (0784)     [rom     ] selectRom : 6                      p: L4377:  11...1....  -> L6000   *****  FAC2  :  SELECT ROM 6
```

```
L5000   r5 b000   err2       1100010000 (0784)    [rom     ] selectRom : 6                                  p: L5000:  11...1....  -> L6001   *****  ERR2  :  SELECT ROM 6
L5001   r5 b001   adr5       0111100010 (0482)    [register] using P : increment C                         p: L5001:  .1111...1.                      ADR5  :  C + 1 -> C[P]
L5002   r5 b002   adr6       0111100010 (0482)    [register] using P : increment C                         p: L5002:  .1111...1.                      ADR6  :  C + 1 -> C[P]
L5003   r5 b003   adr7       0111100010 (0482)    [register] using P : increment C                         p: L5003:  .1111...1.                      ADR7  :  C + 1 -> C[P]
L5004   r5 b004   adr8       0111100010 (0482)    [register] using P : increment C                         p: L5004:  .1111...1.                      ADR8  :  C + 1 -> C[P]
L5005   r5 b005   adr9       0111100010 (0482)    [register] using P : increment C                         p: L5005:  .1111...1.                      ADR9  :  C + 1 -> C[P]
L5006   r5 b006   adr0       0101001110 (0334)    [register] using W : A - C → C                           p: L5006:  .1.1..111.                      ADR0  :  A - C -> C[W]
L5007   r5 b007              1001110000 (0624)    [other   ] (data) C → dataAddress                        p: L5007:  1..111....                             :  C -> DATA ADDRESS
L5008   r5 b008              0000000000 (0000)    [no oper ] nOp                                           p: L5010:  ..........                             :  NO OPERATION
L5009   r5 b009              1011111000 (0760)    [other   ] (data) data → C                               p: L5011:  1.11111...                             :  DATA -> C
L5010   r5 b010              0100010100 (0276)    [status  ] if statusBit ≠ 1 : bit 4                      p: L5012:  .1...1.1..                             :  IF S4 # 1
L5011   r5 b011              1111111111 (1023)    [flow    ]    (then) goTo : 255              → retnzx    p: L5013:  1111111111  -> L5377               :       THEN GO TO RETNZX
L5012   r5 b012              1110101110 (0942)    [register] using W : exchange A C                        p: L5014:  111.1.111.                             :  A EXCHANGE C[W]
L5013   r5 b013              1000010100 (0532)    [status  ] if statusBit ≠ 1 : bit 8                      p: L5015:  1....1.1..                             :  IF S8 # 1
L5014   r5 b014              1010011111 (0671)    [flow    ]    (then) goTo : 167              → add1      p: L5016:  1.1..11111  -> L5247               :       THEN GO TO ADD1
L5015   r5 b015              1010011011 (0667)    [flow    ] goTo : 166                        → sub1      p: L5017:  1.1..11.11  -> L5246               :  GO TO SUB1
L5016   r5 b016   fst2       0110010000 (0400)    [rom     ] selectRom : 3                                  p: L5020:  .11..1....  -> L3021   *****  FST2  :  SELECT ROM 3
L5017   r5 b017              0000000000 (0000)    [no oper ] nOp                                           p: L5021:  ..........                             :  NO OPERATION
L5018   r5 b018              0000000000 (0000)    [no oper ] nOp                                           p: L5022:  ..........                             :  NO OPERATION
L5019   r5 b019              0000000000 (0000)    [no oper ] nOp                                           p: L5023:  ..........                             :  NO OPERATION
L5020   r5 b020              0000000000 (0000)    [no oper ] nOp                                           p: L5024:  ..........                             :  NO OPERATION
L5021   r5 b021   pwo2z0     1110010011 (0915)    [flow    ] goTo : 228                        → pwo2      p: L5025:  111..1..11  -> L5344         PWO2Z0:  GO TO PWO2
L5022   r5 b022   sgmaz4     0100000100 (0260)    [status  ] set statusBit : bit 4                         p: L5026:  .1.....1..                      SGMAZ4:  1 -> S4
L5023   r5 b023              1010100100 (0676)    [status  ] clear statusBit : bit 10                      p: L5027:  1.1.1..1..                             :  0 -> S10
L5024   r5 b024              1010010001 (0657)    [flow    ] jumpSub : 164                     → mul1      p: L5030:  1.1..1...1  -> L5244               :  JSB MUL1
L5025   r5 b025              0000001001 (0009)    [flow    ] jumpSub : 2                       → adr6      p: L5031:  ......1..1  -> L5002               :  JSB ADR6
L5026   r5 b026              1011101101 (0749)    [flow    ] jumpSub : 187                     → stor      p: L5032:  1.111.11.1  -> L5273               :  JSB STOR
L5027   r5 b027              0010110101 (0181)    [flow    ] jumpSub : 45                      → rest      p: L5033:  ..1.11.1.1  -> L5055               :  JSB REST
L5028   r5 b028              0110001110 (0398)    [register] using W : C → A                               p: L5034:  .11...111.                             :  C -> A[W]
L5029   r5 b029              0000001101 (0013)    [flow    ] jumpSub : 3                       → adr7      p: L5035:  ......11.1  -> L5003               :  JSB ADR7
L5030   r5 b030              0010001111 (0143)    [flow    ] goTo : 35                         → sig1      p: L5036:  ..1...1111  -> L5043               :  GO TO SIG1
L5031   r5 b031              0000000000 (0000)    [no oper ] nOp                                           p: L5037:  ..........                             :  NO OPERATION
L5032   r5 b032              0000000000 (0000)    [no oper ] nOp                                           p: L5040:  ..........                             :  NO OPERATION
L5033   r5 b033              0000000000 (0000)    [no oper ] nOp                                           p: L5041:  ..........                             :  NO OPERATION
L5034   r5 b034   stddz4     0100001011 (0267)    [flow    ] goTo : 66                         → stdd      p: L5042:  .1....1.11  -> L5102         STDDZ4:  GO TO STDD
L5035   r5 b035   sig1       1011101101 (0749)    [flow    ] jumpSub : 187                     → stor      p: L5043:  1.111.11.1  -> L5273         SIG1  :  JSB STOR
L5036   r5 b036              0011001001 (0201)    [flow    ] jumpSub : 50                      → yget      p: L5044:  ..11..1..1  -> L5062               :  JSB YGET
L5037   r5 b037              0000010001 (0017)    [flow    ] jumpSub : 4                       → adr8      p: L5045:  .....1...1  -> L5004               :  JSB ADR8
L5038   r5 b038              1011101101 (0749)    [flow    ] jumpSub : 187                     → stor      p: L5046:  1.111.11.1  -> L5273               :  JSB STOR
L5039   r5 b039              0011001110 (0206)    [register] using W : clear c                             p: L5047:  ..11..111.                             :  0 -> C[W]
L5040   r5 b040              0111100010 (0482)    [register] using P : increment C                         p: L5050:  .1111...1.                             :  C + 1 -> C[P]
L5041   r5 b041              0110001110 (0398)    [register] using W : C → A                               p: L5051:  .11...111.                             :  C -> A[W]
L5042   r5 b042              0000000101 (0005)    [flow    ] jumpSub : 1                       → adr5      p: L5052:  .......1.1  -> L5001               :  JSB ADR5
L5043   r5 b043              1011101101 (0749)    [flow    ] jumpSub : 187                     → stor      p: L5053:  1.111.11.1  -> L5273               :  JSB STOR
L5044   r5 b044              0001000011 (0067)    [flow    ] goTo : 16                         → fst2      p: L5054:  ...1....11  -> L5020               :  GO TO FST2
```

```
L5045   r5 b045   rest       0011001110 (0206)     [register] using W : clear c                                  p: L5055:  ..11..111.                  REST  :  0 -> C[W]
L5046   r5 b046              1001110000 (0624)     [other   ] (data) C → dataAddress                             p: L5056:  1..111....                        :  C -> DATA ADDRESS
L5047   r5 b047              0000000000 (0000)     [no oper ] nOp                                                p: L5057:  ..........                        :  NO OPERATION
L5048   r5 b048              1011111000 (0760)     [other   ] (data) data → C                                    p: L5060:  1.11111...                        :  DATA -> C
L5049   r5 b049              0000110000 (0048)     [other   ] (flow) return                                      p: L5061:  ....11....                        :  RETURN
L5050   r5 b050   yget       1100101000 (0808)     [other   ] (stack) rotateDown                                 p: L5062:  11..1.1...                  YGET  :  DOWN ROTATE
L5051   r5 b051              0100101000 (0296)     [other   ] (stack) M → stack                                  p: L5063:  .1..1.1...                        :  C -> STACK
L5052   r5 b052              0110001110 (0398)     [register] using W : C → A                                    p: L5064:  .11...111.                        :  C -> A[W]
L5053   r5 b053              0000110000 (0048)     [other   ] (flow) return                                      p: L5065:  ....11....                        :  RETURN
L5054   r5 b054   tploz4     1000100100 (0548)     [status  ] clear statusBit : bit 8                            p: L5066:  1...1..1..                  TPLOZ4:  0 -> S8
L5055   r5 b055              1001100100 (0612)     [status  ] clear statusBit : bit 9                            p: L5067:  1..11..1..                        :  0 -> S9
L5056   r5 b056   tpl0zj     1010010101 (0661)     [flow    ] jumpSub : 165                  → div1               p: L5070:  1.1..1.1.1  -> L5245         TPL0ZJ:  JSB DIV1
L5057   r5 b057              0000000000 (0000)     [no oper ] nOp                                                p: L5071:  ..........                        :  NO OPERATION
L5058   r5 b058              0000000000 (0000)     [no oper ] nOp                                                p: L5072:  ..........                        :  NO OPERATION
L5059   r5 b059              0000000000 (0000)     [no oper ] nOp                                                p: L5073:  ..........                        :  NO OPERATION
L5060   r5 b060              0000000000 (0000)     [no oper ] nOp                                                p: L5074:  ..........                        :  NO OPERATION
L5061   r5 b061              0000000000 (0000)     [no oper ] nOp                                                p: L5075:  ..........                        :  NO OPERATION
L5062   r5 b062              0000000000 (0000)     [no oper ] nOp                                                p: L5076:  ..........                        :  NO OPERATION
L5063   r5 b063              0000000000 (0000)     [no oper ] nOp                                                p: L5077:  ..........                        :  NO OPERATION
L5064   r5 b064              0000000000 (0000)     [no oper ] nOp                                                p: L5100:  ..........                        :  NO OPERATION
L5065   r5 b065   sqt1       0110010000 (0400)     [rom     ] selectRom : 3                                      p: L5101:  .11..1....  -> L3102   *****  SQT1  :  SELECT ROM 3
L5066   r5 b066   stdd       1010100100 (0676)     [status  ] clear statusBit : bit 10                           p: L5102:  1.1.1..1..                  STDD  :  0 -> S10
L5067   r5 b067              0100100100 (0292)     [status  ] clear statusBit : bit 4                            p: L5103:  .1..1..1..                        :  0 -> S4
L5068   r5 b068              0000001101 (0013)     [flow    ] jumpSub : 3                    → adr7               p: L5104:  ......11.1  -> L5003              :  JSB ADR7
L5069   r5 b069              0110001110 (0398)     [register] using W : C → A                                    p: L5105:  .11...111.                        :  C -> A[W]
L5070   r5 b070              1010010001 (0657)     [flow    ] jumpSub : 164                  → mul1               p: L5106:  1.1..1...1  -> L5244              :  JSB MUL1
L5071   r5 b071              0000000101 (0005)     [flow    ] jumpSub : 1                    → adr5               p: L5107:  .......1.1  -> L5001              :  JSB ADR5
L5072   r5 b072              0001111110 (0126)     [register] using S : if C ≥ 1                                 p: L5110:  ...111111.                        :  IF C[S] >= 1
L5073   r5 b073              0000000011 (0003)     [flow    ]    (then) goTo : 0             → err2               p: L5111:  ........11  -> L5000              :        THEN GO TO ERR2
L5074   r5 b074              1010010101 (0661)     [flow    ] jumpSub : 165                  → div1               p: L5112:  1.1..1.1.1  -> L5245              :  JSB DIV1
L5075   r5 b075              0000001001 (0009)     [flow    ] jumpSub : 2                    → adr6               p: L5113:  ......1..1  -> L5002              :  JSB ADR6
L5076   r5 b076              1110101110 (0942)     [register] using W : exchange A C                             p: L5114:  111.1.111.                        :  A EXCHANGE C[W]
L5077   r5 b077              1010011001 (0665)     [flow    ] jumpSub : 166                  → sub1               p: L5115:  1.1..11..1  -> L5246              :  JSB SUB1
L5078   r5 b078              0100101000 (0296)     [other   ] (stack) M → stack                                  p: L5116:  .1..1.1...                        :  C -> STACK
L5079   r5 b079              0000000101 (0005)     [flow    ] jumpSub : 1                    → adr5               p: L5117:  .......1.1  -> L5001              :  JSB ADR5
L5080   r5 b080              1110101110 (0942)     [register] using W : exchange A C                             p: L5120:  111.1.111.                        :  A EXCHANGE C[W]
L5081   r5 b081              0011001110 (0206)     [register] using W : clear c                                  p: L5121:  ..11..111.                        :  0 -> C[W]
L5082   r5 b082              0111100010 (0482)     [register] using P : increment C                              p: L5122:  .1111...1.                        :  C + 1 -> C[P]
L5083   r5 b083              1010011001 (0665)     [flow    ] (if no carry then) jumpSub : 166  → sub1           p: L5123:  1.1..11..1  -> L5246              :  JSB SUB1
L5084   r5 b084              0110101000 (0424)     [other   ] (stack) stack → A                                  p: L5124:  .11.1.1...                        :  STACK -> A
L5085   r5 b085              1010010101 (0661)     [flow    ] jumpSub : 165                  → div1               p: L5125:  1.1..1.1.1  -> L5245              :  JSB DIV1
L5086   r5 b086              0100000101 (0261)     [flow    ] jumpSub : 65                   → sqt1               p: L5126:  .1.....1.1  -> L5101              :  JSB SQT1
L5087   r5 b087              0100101000 (0296)     [other   ] (stack) M → stack                                  p: L5127:  .1..1.1...                        :  C -> STACK
L5088   r5 b088              0000001101 (0013)     [flow    ] jumpSub : 3                    → adr7               p: L5130:  ......11.1  -> L5003              :  JSB ADR7
L5089   r5 b089              0110001110 (0398)     [register] using W : C → A                                    p: L5131:  .11...111.                        :  C -> A[W]
```

```
L5090   r5 b090             0000000101 (0005)    [flow    ] jumpSub : 1                      → adr5
L5091   r5 b091             1010000101 (0645)    [flow    ] jumpSub : 161                    → div0
L5092   r5 b092   rcxy       0100100100 (0292)    [status  ] clear statusBit : bit 4
L5093   r5 b093             0111010100 (0468)    [status  ] if statusBit ≠ 1 : bit 7
L5094   r5 b094             0110000011 (0387)    [flow    ]    (then) goTo : 96              → rxy1
L5095   r5 b095             0100101000 (0296)    [other   ] (stack) M → stack
L5096   r5 b096   rxy1       0110001110 (0398)    [register] using W : C → A
L5097   r5 b097             0000010001 (0017)    [flow    ] jumpSub : 4                      → adr8
L5098   r5 b098             0100101000 (0296)    [other   ] (stack) M → stack
L5099   r5 b099             0110001110 (0398)    [register] using W : C → A
L5100   r5 b100             0000001101 (0013)    [flow    ] jumpSub : 3                      → adr7
L5101   r5 b101             0111111011 (0507)    [flow    ] goTo : 126                       → fst1
L5102   r5 b102             0000000000 (0000)    [no oper ] nOp
L5103   r5 b103             0000000000 (0000)    [no oper ] nOp
L5104   r5 b104             0000000000 (0000)    [no oper ] nOp
L5105   r5 b105             0000000000 (0000)    [no oper ] nOp
L5106   r5 b106             0000000000 (0000)    [no oper ] nOp
L5107   r5 b107             0000000000 (0000)    [no oper ] nOp
L5108   r5 b108             0000000000 (0000)    [no oper ] nOp
L5109   r5 b109             0000000000 (0000)    [no oper ] nOp
L5110   r5 b110             0000000000 (0000)    [no oper ] nOp
L5111   r5 b111             0000000000 (0000)    [no oper ] nOp
L5112   r5 b112             0000000000 (0000)    [no oper ] nOp
L5113   r5 b113             0000000000 (0000)    [no oper ] nOp
L5114   r5 b114             0000000000 (0000)    [no oper ] nOp
L5115   r5 b115             0000000000 (0000)    [no oper ] nOp
L5116   r5 b116             0000000000 (0000)    [no oper ] nOp
L5117   r5 b117   am10       1011101101 (0749)    [flow    ] jumpSub : 187                    → stor
L5118   r5 b118             1110101110 (0942)    [register] using W : exchange A C
L5119   r5 b119             0110010000 (0400)    [rom     ] selectRom : 3
L5120   r5 b120             0000000000 (0000)    [no oper ] nOp
L5121   r5 b121             0000000000 (0000)    [no oper ] nOp
L5122   r5 b122             0000000000 (0000)    [no oper ] nOp
L5123   r5 b123   amd9z4     1101110001 (0881)    [flow    ] jumpSub : 220                    → dcod          N
L5124   r5 b124             1001010100 (0596)    [status  ] if statusBit ≠ 1 : bit 9
L5125   r5 b125             0111010111 (0471)    [flow    ]    (then) goTo : 117             → am10
L5126   r5 b126   fst1       0110010000 (0400)    [rom     ] selectRom : 3
L5127   r5 b127             0000000000 (0000)    [no oper ] nOp
L5128   r5 b128             0000000000 (0000)    [no oper ] nOp
L5129   r5 b129             0000000000 (0000)    [no oper ] nOp
L5130   r5 b130             0000000000 (0000)    [no oper ] nOp
L5131   r5 b131             0000000000 (0000)    [no oper ] nOp
L5132   r5 b132             0000000000 (0000)    [no oper ] nOp
L5133   r5 b133             0000000000 (0000)    [no oper ] nOp
L5134   r5 b134             0000000000 (0000)    [no oper ] nOp
```

```
p: L5132:  .......1.1  -> L5001              :  JSB ADR5
p: L5133:  1.1....1.1  -> L5241              :  JSB DIV0
p: L5134:  .1..1..1..                  RCXY  :  0 -> S4
p: L5135:  .111.1.1..                        :  IF S7 # 1
p: L5136:  .11.....11  -> L5140              :       THEN GO TO RXY1
p: L5137:  .1..1.1...                        :  C -> STACK
p: L5140:  .11...111.                  RXY1  :  C -> A[W]
p: L5141:  .....1...1  -> L5004              :  JSB ADR8
p: L5142:  .1..1.1...                        :  C -> STACK
p: L5143:  .11...111.                        :  C -> A[W]
p: L5144:  ......11.1  -> L5003              :  JSB ADR7
p: L5145:  .111111.11  -> L5176              :  GO TO FST1
p: L5146:  ..........                        :  NO OPERATION
p: L5147:  ..........                        :  NO OPERATION
p: L5150:  ..........                        :  NO OPERATION
p: L5151:  ..........                        :  NO OPERATION
p: L5152:  ..........                        :  NO OPERATION
p: L5153:  ..........                        :  NO OPERATION
p: L5154:  ..........                        :  NO OPERATION
p: L5155:  ..........                        :  NO OPERATION
p: L5156:  ..........                        :  NO OPERATION
p: L5157:  ..........                        :  NO OPERATION
p: L5160:  ..........                        :  NO OPERATION
p: L5161:  ..........                        :  NO OPERATION
p: L5162:  ..........                        :  NO OPERATION
p: L5163:  ..........                        :  NO OPERATION
p: L5164:  ..........                        :  NO OPERATION
p: L5165:  1.111.11.1  -> L5273        AM10  :  JSB STOR
p: L5166:  111.1.111.                        :  A EXCHANGE C[W]
p: L5167:  .11..1....  -> L3170   *****       :  SELECT ROM 3
p: L5170:  ..........                        :  NO OPERATION
p: L5171:  ..........                        :  NO OPERATION
p: L5172:  ..........                        :  NO OPERATION
p: L5173:  11.111...1  -> L5334        AMD9Z4:  JSB DCOD
p: L5174:  1..1.1.1..                        :  IF S9 # 1
p: L5175:  .111.1.111  -> L5165              :       THEN GO TO AM10
p: L5176:  .11..1....  -> L3177   *****  FST1  :  SELECT ROM 3
p: L5177:  ..........                        :  NO OPERATION
p: L5200:  ..........                        :  NO OPERATION
p: L5201:  ..........                        :  NO OPERATION
p: L5202:  ..........                        :  NO OPERATION
p: L5203:  ..........                        :  NO OPERATION
p: L5204:  ..........                        :  NO OPERATION
p: L5205:  ..........                        :  NO OPERATION
p: L5206:  ..........                        :  NO OPERATION
```

```
L5135   r5 b135              0000000000 (0000)     [no oper ] nOp                                                  p: L5207:  .........                        :  NO OPERATION
L5136   r5 b136              0000000000 (0000)     [no oper ] nOp                                                  p: L5210:  .........                        :  NO OPERATION
L5137   r5 b137              0000000000 (0000)     [no oper ] nOp                                                  p: L5211:  .........                        :  NO OPERATION
L5138   r5 b138              0000000000 (0000)     [no oper ] nOp                                                  p: L5212:  .........                        :  NO OPERATION
L5139   r5 b139              0000000000 (0000)     [no oper ] nOp                                                  p: L5213:  .........                        :  NO OPERATION
L5140   r5 b140              0000000000 (0000)     [no oper ] nOp                                                  p: L5214:  .........                        :  NO OPERATION
L5141   r5 b141              0000000000 (0000)     [no oper ] nOp                                                  p: L5215:  .........                        :  NO OPERATION
L5142   r5 b142              0000000000 (0000)     [no oper ] nOp                                                  p: L5216:  .........                        :  NO OPERATION
L5143   r5 b143              0000000000 (0000)     [no oper ] nOp                                                  p: L5217:  .........                        :  NO OPERATION
L5144   r5 b144              0000000000 (0000)     [no oper ] nOp                                                  p: L5220:  .........                        :  NO OPERATION
L5145   r5 b145              0000000000 (0000)     [no oper ] nOp                                                  p: L5221:  .........                        :  NO OPERATION
L5146   r5 b146              0000000000 (0000)     [no oper ] nOp                                                  p: L5222:  .........                        :  NO OPERATION
L5147   r5 b147              0000000000 (0000)     [no oper ] nOp                                                  p: L5223:  .........                        :  NO OPERATION
L5148   r5 b148              0000000000 (0000)     [no oper ] nOp                                                  p: L5224:  .........                        :  NO OPERATION
L5149   r5 b149              0000000000 (0000)     [no oper ] nOp                                                  p: L5225:  .........                        :  NO OPERATION
L5150   r5 b150              0000000000 (0000)     [no oper ] nOp                                                  p: L5226:  .........                        :  NO OPERATION
L5151   r5 b151              0000000000 (0000)     [no oper ] nOp                                                  p: L5227:  .........                        :  NO OPERATION
L5152   r5 b152              0000000000 (0000)     [no oper ] nOp                                                  p: L5230:  .........                        :  NO OPERATION
L5153   r5 b153              0000000000 (0000)     [no oper ] nOp                                                  p: L5231:  .........                        :  NO OPERATION
L5154   r5 b154              0000000000 (0000)     [no oper ] nOp                                                  p: L5232:  .........                        :  NO OPERATION
L5155   r5 b155              0000000000 (0000)     [no oper ] nOp                                                  p: L5233:  .........                        :  NO OPERATION
L5156   r5 b156              0000000000 (0000)     [no oper ] nOp                                                  p: L5234:  .........                        :  NO OPERATION
L5157   r5 b157              0000000000 (0000)     [no oper ] nOp                                                  p: L5235:  .........                        :  NO OPERATION
L5158   r5 b158              0000000000 (0000)     [no oper ] nOp                                                  p: L5236:  .........                        :  NO OPERATION
L5159   r5 b159              0000000000 (0000)     [no oper ] nOp                                                  p: L5237:  .........                        :  NO OPERATION
L5160   r5 b160   rcxyz4     0101110011 (0371)     [flow    ] goTo : 92                          → rcxy             p: L5240:  .1.111..11  -> L5134          RCXYZ4:  GO TO RCXY
L5161   r5 b161   div0       0011100100 (0228)     [status  ] clear statusBit : bit 3                              p: L5241:  ..111..1..                      DIV0  :  0 -> S3
L5162   r5 b162              1010010111 (0663)     [flow    ] goTo : 165                         → div1             p: L5242:  1.1..1.111  -> L5245               :  GO TO DIV1
L5163   r5 b163   mul0       0011100100 (0228)     [status  ] clear statusBit : bit 3                              p: L5243:  ..111..1..                      MUL0  :  0 -> S3
L5164   r5 b164   mul1       0010010000 (0144)     [rom     ] selectRom : 1                                        p: L5244:  ..1..1....  -> L1245   *****  MUL1  :  SELECT ROM 1
L5165   r5 b165   div1       0010010000 (0144)     [rom     ] selectRom : 1                                        p: L5245:  ..1..1....  -> L1246   *****  DIV1  :  SELECT ROM 1
L5166   r5 b166   sub1       0011111110 (0254)     [register] using S : 0 - C - 1 → C                              p: L5246:  ..1111111.                      SUB1  :  0 - C - 1 -> C[S]
L5167   r5 b167   add1       0000010000 (0016)     [rom     ] selectRom : 0                                        p: L5247:  .....1....  -> L0250   *****  ADD1  :  SELECT ROM 0
L5168   r5 b168              0000000000 (0000)     [no oper ] nOp                                                  p: L5250:  .........                        :  NO OPERATION
L5169   r5 b169              0000000000 (0000)     [no oper ] nOp                                                  p: L5251:  .........                        :  NO OPERATION
L5170   r5 b170              0000000000 (0000)     [no oper ] nOp                                                  p: L5252:  .........                        :  NO OPERATION
L5171   r5 b171              0000000000 (0000)     [no oper ] nOp                                                  p: L5253:  .........                        :  NO OPERATION
L5172   r5 b172              0000000000 (0000)     [no oper ] nOp                                                  p: L5254:  .........                        :  NO OPERATION
L5173   r5 b173              0000000000 (0000)     [no oper ] nOp                                                  p: L5255:  .........                        :  NO OPERATION
L5174   r5 b174              0000000000 (0000)     [no oper ] nOp                                                  p: L5256:  .........                        :  NO OPERATION
L5175   r5 b175              0000000000 (0000)     [no oper ] nOp                                                  p: L5257:  .........                        :  NO OPERATION
L5176   r5 b176   dvof       0011010010 (0210)     [register] using WP : clear c                                   p: L5260:  ..11.1..1.                      DVOF  :  0 -> C[WP]
L5177   r5 b177              0101110010 (0370)     [register] using WP : decrement C                               p: L5261:  .1.111..1.                      :  C - 1 -> C[WP]
L5178   r5 b178              0011011010 (0218)     [register] using XS : clear c                                   p: L5262:  ..11.11.1.                      :  0 -> C[XS]
L5179   r5 b179              0100010000 (0272)     [rom     ] selectRom : 2                                        p: L5263:  .1...1....  -> L2264   *****  :  SELECT ROM 2
```

```
L5180   r5 b180   dvofz2   1011000011 (0707)   [flow    ] goTo : 176                       → dvof     p: L5264:  1.11....11  -> L5260          DVOFZ2:  GO TO DVOF
L5181   r5 b181   ofl1     0011010010 (0210)   [register] using WP : clear c                           p: L5265:  ..11.1..1.                    OFL1  :  0 -> C[WP]
L5182   r5 b182            0101110010 (0370)   [register] using WP : decrement C                       p: L5266:  .1.111..1.                          :  C - 1 -> C[WP]
L5183   r5 b183            0011011010 (0218)   [register] using XS : clear c                            p: L5267:  ..11.11.1.                          :  0 -> C[XS]
L5184   r5 b184            1110001010 (0906)   [register] using X : A + B → A                           p: L5270:  111...1.1.                          :  A + B -> A[X]
L5185   r5 b185            1011101111 (0751)   [flow    ] (if no carry then) goTo : 187    → stor     p: L5271:  1.111.1111  -> L5273            :  IF NO CARRY GO TO STOR
L5186   r5 b186            0011001110 (0206)   [register] using W : clear c                           p: L5272:  ..11..111.                          :  0 -> C[W]
L5187   r5 b187   stor     0110001110 (0398)   [register] using W : C → A                             p: L5273:  .11...111.                    STOR  :  C -> A[W]
L5188   r5 b188   ofl4     1100001100 (0780)   [pointer ] value → P : value 12                         p: L5274:  11....11..                    OFL4  :  12 -> P
L5189   r5 b189            0100101010 (0298)   [register] using X : A → B                             p: L5275:  .1..1.1.1.                          :  A -> B[X]
L5190   r5 b190            0110001010 (0394)   [register] using X : C → A                             p: L5276:  .11...1.1.                          :  C -> A[X]
L5191   r5 b191            0110111010 (0442)   [register] using XS : if C = 0                         p: L5277:  .11.111.1.                          :  IF C[XS] = 0
L5192   r5 b192            1100010011 (0787)   [flow    ]    (then) goTo : 196          → ofl5     p: L5300:  11...1..11  -> L5304            :       THEN GO TO OFL5
L5193   r5 b193            0010101010 (0170)   [register] using X : 0 - C → C                         p: L5301:  ..1.1.1.1.                          :  0 - C -> C[X]
L5194   r5 b194            0101111010 (0378)   [register] using XS : decrement C                       p: L5302:  .1.1111.1.                          :  C - 1 -> C[XS]
L5195   r5 b195            1011010111 (0727)   [flow    ] (if no carry then) goTo : 181    → ofl1     p: L5303:  1.11.1.111  -> L5265            :  IF NO CARRY GO TO OFL1
L5196   r5 b196   ofl5     1110110110 (0950)   [register] using MS : exchange A C                      p: L5304:  111.11.11.                    OFL5  :  A EXCHANGE C[MS]
L5197   r5 b197            1011111000 (0760)   [other   ] (data) data → C                              p: L5305:  1.11111...                          :  DATA -> C
L5198   r5 b198            1110101110 (0942)   [register] using W : exchange A C                       p: L5306:  111.1.111.                          :  A EXCHANGE C[W]
L5199   r5 b199            1011110000 (0752)   [other   ] (data) C → data                              p: L5307:  1.1111....                          :  C -> DATA
L5200   r5 b200            0000110000 (0048)   [other   ] (flow) return                                p: L5310:  ....11....                          :  RETURN
L5201   r5 b201            0000000000 (0000)   [no oper ] nOp                                          p: L5311:  ..........                          :  NO OPERATION
L5202   r5 b202            0000000000 (0000)   [no oper ] nOp                                          p: L5312:  ..........                          :  NO OPERATION
L5203   r5 b203            0000000000 (0000)   [no oper ] nOp                                          p: L5313:  ..........                          :  NO OPERATION
L5204   r5 b204            0000000000 (0000)   [no oper ] nOp                                          p: L5314:  ..........                          :  NO OPERATION
L5205   r5 b205            0000000000 (0000)   [no oper ] nOp                                          p: L5315:  ..........                          :  NO OPERATION
L5206   r5 b206            0000000000 (0000)   [no oper ] nOp                                          p: L5316:  ..........                          :  NO OPERATION
L5207   r5 b207            0000000000 (0000)   [no oper ] nOp                                          p: L5317:  ..........                          :  NO OPERATION
L5208   r5 b208            0000000000 (0000)   [no oper ] nOp                                          p: L5320:  ..........                          :  NO OPERATION
L5209   r5 b209            0000000000 (0000)   [no oper ] nOp                                          p: L5321:  ..........                          :  NO OPERATION
L5210   r5 b210            0000000000 (0000)   [no oper ] nOp                                          p: L5322:  ..........                          :  NO OPERATION
L5211   r5 b211            0000000000 (0000)   [no oper ] nOp                                          p: L5323:  ..........                          :  NO OPERATION
L5212   r5 b212            0000000000 (0000)   [no oper ] nOp                                          p: L5324:  ..........                          :  NO OPERATION
L5213   r5 b213            0000000000 (0000)   [no oper ] nOp                                          p: L5325:  ..........                          :  NO OPERATION
L5214   r5 b214            0000000000 (0000)   [no oper ] nOp                                          p: L5326:  ..........                          :  NO OPERATION
L5215   r5 b215            0000000000 (0000)   [no oper ] nOp                                          p: L5327:  ..........                          :  NO OPERATION
L5216   r5 b216            0000000000 (0000)   [no oper ] nOp                                          p: L5330:  ..........                          :  NO OPERATION
L5217   r5 b217            0000000000 (0000)   [no oper ] nOp                                          p: L5331:  ..........                          :  NO OPERATION
L5218   r5 b218            0000000000 (0000)   [no oper ] nOp                                          p: L5332:  ..........                          :  NO OPERATION
L5219   r5 b219            0000000000 (0000)   [no oper ] nOp                                          p: L5333:  ..........                          :  NO OPERATION
L5220   r5 b220   dcod     0100010100 (0276)   [status  ] if statusBit ≠ 1 : bit 4                     p: L5334:  .1...1.1..                    DCOD  :  IF S4 # 1
L5221   r5 b221            1110000111 (0903)   [flow    ]    (then) goTo : 225          → dcd1     p: L5335:  111....111  -> L5341            :       THEN GO TO DCD1
L5222   r5 b222            0110010100 (0404)   [status  ] if statusBit ≠ 1 : bit 6                     p: L5336:  .11..1.1..                          :  IF S6 # 1
L5223   r5 b223            1010011011 (0667)   [flow    ]    (then) goTo : 166          → sub1     p: L5337:  1.1..11.11  -> L5246            :       THEN GO TO SUB1
L5224   r5 b224            1010011111 (0671)   [flow    ] goTo : 167                       → add1     p: L5340:  1.1..11111  -> L5247            :  GO TO ADD1
```

```
L5225   r5 b225   dcd1       0110010100 (0404)     [status  ] if statusBit ≠ 1 : bit 6
L5226   r5 b226              1010010111 (0663)     [flow    ]    (then) goTo : 165              → div1
L5227   r5 b227              1010010011 (0659)     [flow    ] goTo : 164                        → mul1
L5228   r5 b228   pwo2       0011001110 (0206)     [register] using W : clear c
L5229   r5 b229              0101111110 (0382)     [register] using S : decrement C
L5230   r5 b230              0010001100 (0140)     [pointer ] value → P : value 2
L5231   r5 b231              0010011000 (0152)     [constant] loadConstant : 2
L5232   r5 b232              0010101000 (0168)     [other   ] (register) exchange C M
L5233   r5 b233              0011001110 (0206)     [register] using W : clear c
L5234   r5 b234   clr2       1011101110 (0750)     [register] using W : clear A
L5235   r5 b235              1100001100 (0780)     [pointer ] value → P : value 12
L5236   r5 b236   clr3       0101100010 (0354)     [register] using P : decrement C
L5237   r5 b237              1001110000 (0624)     [other   ] (data) C → dataAddress
L5238   r5 b238              1110101110 (0942)     [register] using W : exchange A C
L5239   r5 b239              0100101000 (0296)     [other   ] (stack) M → stack
L5240   r5 b240              1011110000 (0752)     [other   ] (data) C → data
L5241   r5 b241              1110101110 (0942)     [register] using W : exchange A C
L5242   r5 b242              0111100010 (0482)     [register] using P : increment C
L5243   r5 b243              0111100010 (0482)     [register] using P : increment C
L5244   r5 b244              1110110011 (0947)     [flow    ] (if no carry then) goTo : 236    → clr3
L5245   r5 b245              0111111011 (0507)     [flow    ] goTo : 126                        → fst1
L5246   r5 b246   clr1z4     0011001110 (0206)     [register] using W : clear c
L5247   r5 b247              0110011000 (0408)     [constant] loadConstant : 6
L5248   r5 b248              1110101011 (0939)     [flow    ] goTo : 234                        → clr2
L5249   r5 b249              0000000000 (0000)     [no oper ] nOp
L5250   r5 b250              0000000000 (0000)     [no oper ] nOp
L5251   r5 b251              0000000000 (0000)     [no oper ] nOp
L5252   r5 b252              0000000000 (0000)     [no oper ] nOp
L5253   r5 b253              0000000000 (0000)     [no oper ] nOp
L5254   r5 b254              0000000000 (0000)     [no oper ] nOp
L5255   r5 b255   retnzx      0000110000 (0048)     [other   ] (flow) return
```

```
p: L5341:  .11..1.1..                    DCD1  :  IF S6 # 1
p: L5342:  1.1..1.111  -> L5245                :        THEN GO TO DIV1
p: L5343:  1.1..1..11  -> L5244                :  GO TO MUL1
p: L5344:  ..11..111.                    PWO2  :  0 -> C[W]
p: L5345:  .1.111111.                          :  C - 1 -> C[S]
p: L5346:  ..1...11..                          :  2 -> P
p: L5347:  ..1..11...                          :  LOAD CONSTANT 2
p: L5350:  ..1.1.1...                          :  C EXCHANGE M
p: L5351:  ..11..111.                          :  0 -> C[W]
p: L5352:  1.111.111.                    CLR2  :  0 -> A[W]
p: L5353:  11....11..                          :  12 -> P
p: L5354:  .1.11...1.                    CLR3  :  C - 1 -> C[P]
p: L5355:  1..111....                          :  C -> DATA ADDRESS
p: L5356:  111.1.111.                          :  A EXCHANGE C[W]
p: L5357:  .1..1.1...                          :  C -> STACK
p: L5360:  1.1111....                          :  C -> DATA
p: L5361:  111.1.111.                          :  A EXCHANGE C[W]
p: L5362:  .1111...1.                          :  C + 1 -> C[P]
p: L5363:  .1111...1.                          :  C + 1 -> C[P]
p: L5364:  111.11..11  -> L5354                :  IF NO CARRY GO TO CLR3
p: L5365:  .111111.11  -> L5176                :  GO TO FST1
p: L5366:  ..11..111.                    CLR1Z4:  0 -> C[W]
p: L5367:  .11..11...                          :  LOAD CONSTANT 6
p: L5370:  111.1.1.11  -> L5352                :  GO TO CLR2
p: L5371:  ..........                          :  NO OPERATION
p: L5372:  ..........                          :  NO OPERATION
p: L5373:  ..........                          :  NO OPERATION
p: L5374:  ..........                          :  NO OPERATION
p: L5375:  ..........                          :  NO OPERATION
p: L5376:  ..........                          :  NO OPERATION
p: L5377:  ....11....                    RETNZX:  RETURN
```

```
L6000   r6 b000   factz4     1011110011 (0755)     [flow    ] goTo : 188                          → fact
L6001   r6 b001   err2z1     0000011111 (0031)     [flow    ] goTo : 7                            → errr
L6002   r6 b002   tdmsz0     0000110100 (0052)     [status  ] clear all statusBits
L6003   r6 b003              0111111111 (0511)     [flow    ] goTo : 127                          → tdmszj
L6004   r6 b004   oflw       0111111010 (0506)     [register] using XS : increment C
L6005   r6 b005              0111111001 (0505)     [flow    ] (if no carry then) jumpSub : 126    → fst1
L6006   r6 b006   tenxzj     1111010001 (0977)     [flow    ] jumpSub : 244                       → tnx3
L6007   r6 b007   errr       0011001110 (0206)     [register] using W : clear c
L6008   r6 b008              0000110100 (0052)     [status  ] clear all statusBits
L6009   r6 b009              0101000100 (0324)     [status  ] set statusBit : bit 5
L6010   r6 b010   rnd0z3     0110001110 (0398)     [register] using W : C → A
L6011   r6 b011              1010101000 (0680)     [other   ] (register) M → C
L6012   r6 b012              1110101110 (0942)     [register] using W : exchange A C
L6013   r6 b013              0001100111 (0103)     [flow    ] goTo : 25                           → rndx
L6014   r6 b014   rnd3       1011010110 (0726)     [register] using MS : rightShift A
L6015   r6 b015              1111101010 (1002)     [register] using X : increment A
L6016   r6 b016              0000111011 (0059)     [flow    ] (if no carry then) goTo : 14        → rnd3
L6017   r6 b017   rnd4       1101001100 (0844)     [pointer ] value → P : value 13
L6018   r6 b018   rnd5       1100111010 (0826)     [register] using XS : exchange A B
L6019   r6 b019              0100111010 (0314)     [register] using XS : A → B
L6020   r6 b020   rnd6       0000011100 (0028)     [pointer ] decrement P
L6021   r6 b021              0010101100 (0172)     [pointer ] if P ≠ value : value 2
L6022   r6 b022              0010110111 (0183)     [flow    ]    (then) goTo : 45                 → rnd7
L6023   r6 b023   rnof       1011101110 (0750)     [register] using W : clear A
L6024   r6 b024              1101101010 (0874)     [register] using X : decrement A
L6025   r6 b025   rndx       0000101110 (0046)     [register] using W : clear B
L6026   r6 b026              1000000100 (0516)     [status  ] set statusBit : bit 8
L6027   r6 b027              0001001100 (0076)     [pointer ] value → P : value 1
L6028   r6 b028              0100111010 (0314)     [register] using XS : A → B
L6029   r6 b029              0110000110 (0390)     [register] using M : C → A
L6030   r6 b030              0100010110 (0278)     [register] using MS : leftShift A
L6031   r6 b031              1001100010 (0610)     [register] using P : if A ≥ 1
L6032   r6 b032              0001000111 (0071)     [flow    ]    (then) goTo : 17                 → rnd4
L6033   r6 b033              1000100100 (0548)     [status  ] clear statusBit : bit 8
L6034   r6 b034              1110001100 (0908)     [pointer ] value → P : value 14
L6035   r6 b035              0110001010 (0394)     [register] using X : C → A
L6036   r6 b036              0001111010 (0122)     [register] using XS : if C ≥ 1
L6037   r6 b037              0000111011 (0059)     [flow    ]    (then) goTo : 14                 → rnd3
L6038   r6 b038   rnd1       0000011100 (0028)     [pointer ] decrement P
L6039   r6 b039              0010101100 (0172)     [pointer ] if P ≠ value : value 2
L6040   r6 b040              0010101011 (0171)     [flow    ]    (then) goTo : 42                 → rnd2
L6041   r6 b041              0001011111 (0095)     [flow    ] goTo : 23                           → rnof
L6042   r6 b042   rnd2       1101101010 (0874)     [register] using X : decrement A
L6043   r6 b043              0010011011 (0155)     [flow    ] (if no carry then) goTo : 38        → rnd1
L6044   r6 b044              0001001011 (0075)     [flow    ] goTo : 18                           → rnd5
```

```
p: L6000:  1.1111..11  -> L6274
p: L6001:  .....11111  -> L6007
p: L6002:  ....11.1..
p: L6003:  0.111111111 -> L6177
p: L6004:  .111111.1.
p: L6005:  .111111..1  -> L6176
p: L6006:  1111.1...1  -> L6364
p: L6007:  ..11..111.
p: L6010:  ....11.1..
p: L6011:  .1.1...1..
p: L6012:  .11...111.
p: L6013:  1.1.1.1...
p: L6014:  111.1.111.
p: L6015:  ...11..111  -> L6031
p: L6016:  1.11.1.11.
p: L6017:  11111.1.1.
p: L6020:  ....111.11  -> L6016
p: L6021:  11.1..11..
p: L6022:  11..111.1.
p: L6023:  .1..111.1.
p: L6024:  .....111..
p: L6025:  ..1.1.11..
p: L6026:  ..1.11.111  -> L6055
p: L6027:  1.111.111.
p: L6030:  11.11.1.1.
p: L6031:  ....1.111.
p: L6032:  1......1..
p: L6033:  ...1..11..
p: L6034:  .1..111.1.
p: L6035:  .11....11.
p: L6036:  .1...1.11.
p: L6037:  1..11...1.
p: L6040:  ...1...111  -> L6021
p: L6041:  1...1..1..
p: L6042:  111...11..
p: L6043:  .11...1.1.
p: L6044:  ...1111.1.
p: L6045:  ....111.11  -> L6016
p: L6046:  .....111..
p: L6047:  ..1.1.11..
p: L6050:  ..1.1.1.11  -> L6052
p: L6051:  ...1.11111  -> L6027
p: L6052:  11.11.1.1.
p: L6053:  ..1..11.11  -> L6046
p: L6054:  ...1..1.11  -> L6022
```

```
FACTZ4:  GO TO FACT
ERR2Z1:  GO TO ERRR
TDMSZ0:  CLEAR STATUS
      :  GO TO TDMSZJ
OFLW  :  C + 1 -> C[XS]
      :  JSB FST1
TENXZJ:  JSB TNX3
ERRR  :  0 -> C[W]
      :  CLEAR STATUS
      :  1 -> S5
RND0Z3:  C -> A[W]
      :  M -> C
      :  A EXCHANGE C[W]
      :  GO TO RNDX
RND3  :  SHIFT RIGHT A[MS]
      :  A + 1 -> A[X]
      :  IF NO CARRY GO TO RND3
RND4  :  13 -> P
RND5  :  A EXCHANGE B[XS]
      :  A -> B[XS]
RND6  :  P - 1 -> P
      :  IF P # 2
      :      THEN GO TO RND7
RNOF  :  0 -> A[W]
      :  A - 1 -> A[X]
RNDX  :  0 -> B[W]
      :  1 -> S8
      :  1 -> P
      :  A -> B[XS]
      :  C -> A[M]
      :  SHIFT LEFT A[MS]
      :  IF A[P] >= 1
      :      THEN GO TO RND4
      :  0 -> S8
      :  14 -> P
      :  C -> A[X]
      :  IF C[XS] >= 1
      :      THEN GO TO RND3
RND1  :  P - 1 -> P
      :  IF P # 2
      :      THEN GO TO RND2
      :  GO TO RNOF
RND2  :  A - 1 -> A[X]
      :  IF NO CARRY GO TO RND1
      :  GO TO RND5
```

```
L6045   r6 b045   rnd7       1101111010 (0890)     [register] using XS : decrement A                          p: L6055:  11.1111.1.                 RND7  :  A − 1 -> A[XS]
L6046   r6 b046              0001010011 (0083)     [flow    ] (if no carry then) goTo : 20        → rnd6      p: L6056:  ...1.1..11  -> L6024                :  IF NO CARRY GO TO RND6
L6047   r6 b047              0100100010 (0290)     [register] using P : A → B                                 p: L6057:  .1..1...1.                 :  A -> B[P]
L6048   r6 b048              0000011100 (0028)     [pointer ] decrement P                                     p: L6060:  .....111..                 :  P − 1 -> P
L6049   r6 b049              1011110010 (0754)     [register] using WP : clear A                              p: L6061:  1.1111..1.                 :  0 -> A[WP]
L6050   r6 b050              0110001010 (0394)     [register] using X : C → A                                 p: L6062:  .11...1.1.                 :  C -> A[X]
L6051   r6 b051              1110010110 (0918)     [register] using MS : A + B → A                            p: L6063:  111..1.11.                 :  A + B -> A[MS]
L6052   r6 b052              0011101011 (0235)     [flow    ] (if no carry then) goTo : 58        → rnd8      p: L6064:  ..111.1.11  -> L6072                :  IF NO CARRY GO TO RND8
L6053   r6 b053              1011010110 (0726)     [register] using MS : rightShift A                         p: L6065:  1.11.1.11.                 :  SHIFT RIGHT A[MS]
L6054   r6 b054              1111111110 (1022)     [register] using S : increment A                           p: L6066:  111111111                  :  A + 1 -> A[S]
L6055   r6 b055              1111101010 (1002)     [register] using X : increment A                           p: L6067:  11111.1.1.                 :  A + 1 -> A[X]
L6056   r6 b056              1000010100 (0532)     [status  ] if statusBit ≠ 1 : bit 8                        p: L6070:  1....1.1..                 :  IF S8 # 1
L6057   r6 b057              0011101111 (0239)     [flow    ]    (then) goTo : 59                 → rnd9      p: L6071:  ..111.1111  -> L6073                :       THEN GO TO RND9
L6058   r6 b058   rnd8       0000111100 (0060)     [pointer ] increment P                                     p: L6072:  ....1111..                 RND8  :  P + 1 -> P
L6059   r6 b059   rnd9       1011010110 (0726)     [register] using MS : rightShift A                         p: L6073:  1.11.1.11.                 RND9  :  SHIFT RIGHT A[MS]
L6060   r6 b060              0000110110 (0054)     [register] using MS : clear B                              p: L6074:  ....11.11.                 :  0 -> B[MS]
L6061   r6 b061              1101111010 (0890)     [register] using XS : decrement A                          p: L6075:  11.1111.1.                 :  A − 1 -> A[XS]
L6062   r6 b062              1001111010 (0634)     [register] using XS : if A ≥ 1                             p: L6076:  1..1111.1.                 :  IF A[XS] >= 1
L6063   r6 b063              0100000111 (0263)     [flow    ]    (then) goTo : 65                 → rn10      p: L6077:  .1.....111  -> L6101                :       THEN GO TO RN10
L6064   r6 b064              0001011111 (0095)     [flow    ] goTo : 23                           → rnof      p: L6100:  ...1.11111  -> L6027                :  GO TO RNOF
L6065   r6 b065   rn10       1111110010 (1018)     [register] using XS : increment A                          p: L6101:  1111111.1.                 RN10  :  A + 1 -> A[XS]
L6066   r6 b066              1100101110 (0814)     [register] using W : exchange A B                          p: L6102:  11..1.111.                 :  A EXCHANGE B[W]
L6067   r6 b067              1111100010 (0994)     [register] using P : increment A                           p: L6103:  11111...1.                 :  A + 1 -> A[P]
L6068   r6 b068              1111100010 (0994)     [register] using P : increment A                           p: L6104:  11111...1.                 :  A + 1 -> A[P]
L6069   r6 b069   rn11       0100010110 (0278)     [register] using MS : leftShift A                          p: L6105:  .1...1.11.                 RN11  :  SHIFT LEFT A[MS]
L6070   r6 b070              1101111010 (0890)     [register] using XS : decrement A                          p: L6106:  11.1111.1.                 :  A − 1 -> A[XS]
L6071   r6 b071              0100010111 (0279)     [flow    ] (if no carry then) goTo : 69        → rn11      p: L6107:  .1...1.111  -> L6105                :  IF NO CARRY GO TO RN11
L6072   r6 b072              1011110010 (0754)     [register] using WP : clear A                              p: L6110:  1.1111..1.                 :  0 -> A[WP]
L6073   r6 b073              1101110010 (0882)     [register] using WP : decrement A                          p: L6111:  11.111..1.                 :  A − 1 -> A[WP]
L6074   r6 b074              1011010110 (0726)     [register] using MS : rightShift A                         p: L6112:  1.11.1.11.                 :  SHIFT RIGHT A[MS]
L6075   r6 b075              1100101110 (0814)     [register] using W : exchange A B                          p: L6113:  11..1.111.                 :  A EXCHANGE B[W]
L6076   r6 b076              1000010100 (0532)     [status  ] if statusBit ≠ 1 : bit 8                        p: L6114:  1....1.1..                 :  IF S8 # 1
L6077   r6 b077              0101010111 (0343)     [flow    ]    (then) goTo : 85                 → rnrt      p: L6115:  .1.1.1.111  -> L6125                :       THEN GO TO RNRT
L6078   r6 b078              1110101010 (0938)     [register] using X : exchange A C                          p: L6116:  111.1.1.1.                 :  A EXCHANGE C[X]
L6079   r6 b079              0000101010 (0042)     [register] using X : clear B                               p: L6117:  ....1.1.1.                 :  0 -> B[X]
L6080   r6 b080              0110111010 (0442)     [register] using XS : if C = 0                             p: L6120:  .11.111.1.                 :  IF C[XS] = 0
L6081   r6 b081              0101010011 (0339)     [flow    ]    (then) goTo : 84                 → rtrn      p: L6121:  .1.1.1..11  -> L6124                :       THEN GO TO RTRN
L6082   r6 b082              0010101010 (0170)     [register] using X : 0 − C → C                             p: L6122:  ..1.1.1.1.                 :  0 − C -> C[X]
L6083   r6 b083              0101111010 (0378)     [register] using XS : decrement C                          p: L6123:  .1.1111.1.                 :  C − 1 -> C[XS]
L6084   r6 b084   rtrn       1110101010 (0938)     [register] using X : exchange A C                          p: L6124:  111.1.1.1.                 RTRN  :  A EXCHANGE C[X]
L6085   r6 b085   rnrt       0101010100 (0340)     [status  ] if statusBit ≠ 1 : bit 5                        p: L6125:  .1.1.1.1..                 RNRT  :  IF S5 # 1
L6086   r6 b086              1111111011 (1019)     [flow    ]    (then) goTo : 254                → ret3      p: L6126:  1111111.11  -> L6376                :       THEN GO TO RET3
L6087   r6 b087              1000000011 (0515)     [flow    ] goTo : 128                          → fst2      p: L6127:  1.......11  -> L6200                :  GO TO FST2
L6088   r6 b088              0000000000 (0000)     [no oper ] nOp                                             p: L6130:  ..........                 :  NO OPERATION
L6089   r6 b089              0000000000 (0000)     [no oper ] nOp                                             p: L6131:  ..........                 :  NO OPERATION
```

```
L6090   r6 b090             0000000000 (0000)    [no oper ] nOp                                    p: L6132:  .........                      :  NO OPERATION
L6091   r6 b091             0000000000 (0000)    [no oper ] nOp                                    p: L6133:  .........                      :  NO OPERATION
L6092   r6 b092             0000000000 (0000)    [no oper ] nOp                                    p: L6134:  .........                      :  NO OPERATION
L6093   r6 b093             0000000000 (0000)    [no oper ] nOp                                    p: L6135:  .........                      :  NO OPERATION
L6094   r6 b094             0000000000 (0000)    [no oper ] nOp                                    p: L6136:  .........                      :  NO OPERATION
L6095   r6 b095             0000000000 (0000)    [no oper ] nOp                                    p: L6137:  .........                      :  NO OPERATION
L6096   r6 b096             0000000000 (0000)    [no oper ] nOp                                    p: L6140:  .........                      :  NO OPERATION
L6097   r6 b097             0000000000 (0000)    [no oper ] nOp                                    p: L6141:  .........                      :  NO OPERATION
L6098   r6 b098             0000000000 (0000)    [no oper ] nOp                                    p: L6142:  .........                      :  NO OPERATION
L6099   r6 b099  savezx     1011010011 (0723)    [flow    ] goTo : 180                  → save     p: L6143:  1.11.1..11  -> L6264          SAVEZX:  GO TO SAVE
L6100   r6 b100  sav2zx     1010010011 (0659)    [flow    ] goTo : 164                  → sav2     p: L6144:  1.1..1..11  -> L6244          SAV2ZX:  GO TO SAV2
L6101   r6 b101  adr9z4     0110001110 (0398)    [register] using W : C → A                       p: L6145:  .11...111.                    ADR9Z4:  C -> A[W]
L6102   r6 b102             0011001110 (0206)    [register] using W : clear c                     p: L6146:  ..11..111.                         :  0 -> C[W]
L6103   r6 b103             0101100010 (0354)    [register] using P : decrement C                 p: L6147:  .1.11...1.                         :  C - 1 -> C[P]
L6104   r6 b104             1001110000 (0624)    [other   ] (data) C → dataAddress                p: L6150:  1..111....                         :  C -> DATA ADDRESS
L6105   r6 b105             1110101110 (0942)    [register] using W : exchange A C                p: L6151:  111.1.111.                         :  A EXCHANGE C[W]
L6106   r6 b106             0110001110 (0398)    [register] using W : C → A                       p: L6152:  .11...111.                         :  C -> A[W]
L6107   r6 b107             1111101111 (1007)    [flow    ] goTo : 251                  → svrt     p: L6153:  11111.1111  -> L6373               :  GO TO SVRT
L6108   r6 b108             0000000000 (0000)    [no oper ] nOp                                    p: L6154:  .........                      :  NO OPERATION
L6109   r6 b109             0000000000 (0000)    [no oper ] nOp                                    p: L6155:  .........                      :  NO OPERATION
L6110   r6 b110             0000000000 (0000)    [no oper ] nOp                                    p: L6156:  .........                      :  NO OPERATION
L6111   r6 b111             0000000000 (0000)    [no oper ] nOp                                    p: L6157:  .........                      :  NO OPERATION
L6112   r6 b112             0000000000 (0000)    [no oper ] nOp                                    p: L6160:  .........                      :  NO OPERATION
L6113   r6 b113             0000000000 (0000)    [no oper ] nOp                                    p: L6161:  .........                      :  NO OPERATION
L6114   r6 b114  consz4     1000000111 (0519)    [flow    ] goTo : 129                  → cons     p: L6162:  1......111  -> L6201          CONSZ4:  GO TO CONS
L6115   r6 b115  piiiz4     0111011001 (0473)    [flow    ] jumpSub : 118               → push     p: L6163:  .111.11..1  -> L6166          PIIIZ4:  JSB PUSH
L6116   r6 b116             0000110100 (0052)    [status  ] clear all statusBits                   p: L6164:  ....11.1..                         :  CLEAR STATUS
L6117   r6 b117             0010010000 (0144)    [rom     ] selectRom : 1                          p: L6165:  ..1..1....  -> L1166   *****      :  SELECT ROM 1
L6118   r6 b118  push       0111010100 (0468)    [status  ] if statusBit ≠ 1 : bit 7               p: L6166:  .111.1.1..                    PUSH  :  IF S7 # 1
L6119   r6 b119             0111100111 (0487)    [flow    ]    (then) goTo : 121        → pret     p: L6167:  .1111..111  -> L6171               :       THEN GO TO PRET
L6120   r6 b120             0100101000 (0296)    [other   ] (stack) M → stack                     p: L6170:  .1..1.1...                         :  C -> STACK
L6121   r6 b121  pret       0011001110 (0206)    [register] using W : clear c                     p: L6171:  ..11..111.                    PRET  :  0 -> C[W]
L6122   r6 b122             0000110000 (0048)    [other   ] (flow) return                         p: L6172:  ....11....                         :  RETURN
L6123   r6 b123             0000000000 (0000)    [no oper ] nOp                                    p: L6173:  .........                      :  NO OPERATION
L6124   r6 b124  lstxzj     1010100100 (0676)    [status  ] clear statusBit : bit 10               p: L6174:  1.1.1..1..                    LSTXZJ:  0 -> S10
L6125   r6 b125             0110010001 (0401)    [flow    ] jumpSub : 100               → sav2zx   p: L6175:  .11..1...1  -> L6144               :  JSB SAV2ZX
L6126   r6 b126  fst1       0110010000 (0400)    [rom     ] selectRom : 3                          p: L6176:  .11..1....  -> L3177   *****  FST1  :  SELECT ROM 3
L6127   r6 b127  tdmszj     0001100101 (0101)    [flow    ] jumpSub : 25                → rndx     p: L6177:  ...11..1.1  -> L6031          TDMSZJ:  JSB RNDX
L6128   r6 b128  fst2       0110010000 (0400)    [rom     ] selectRom : 3                          p: L6200:  .11..1....  -> L3201   *****  FST2  :  SELECT ROM 3
L6129   r6 b129  cons       0100001110 (0270)    [register] using W : leftShift A                 p: L6201:  .1....111.                    CONS  :  SHIFT LEFT A[W]
L6130   r6 b130             0100001110 (0270)    [register] using W : leftShift A                 p: L6202:  .1....111.                         :  SHIFT LEFT A[W]
L6131   r6 b131             0111011001 (0473)    [flow    ] jumpSub : 118               → push     p: L6203:  .111.11..1  -> L6166               :  JSB PUSH
L6132   r6 b132             1101111010 (0890)    [register] using XS : decrement A                p: L6204:  11.1111.1.                         :  A - 1 -> A[XS]
L6133   r6 b133             1000011111 (0543)    [flow    ] (if no carry then) goTo : 135  → con7   p: L6205:  1....11111  -> L6207               :  IF NO CARRY GO TO CON7
L6134   r6 b134             0111110011 (0499)    [flow    ] goTo : 124                  → lstxzj   p: L6206:  .11111..11  -> L6174               :  GO TO LSTXZJ
```

```
L6135   r6 b135   con7       1101111010 (0890)     [register] using XS : decrement A                                  p: L6207:  11.1111.1.                  CON7  :  A - 1 -> A[XS]
L6136   r6 b136              1000110111 (0567)     [flow    ] (if no carry then) goTo : 141       → con8              p: L6210:  1...11.111  -> L6215                :  IF NO CARRY GO TO CON8
L6137   r6 b137              0010011000 (0152)     [constant] loadConstant : 2                                        p: L6211:  ..1..11...                  :  LOAD CONSTANT 2
L6138   r6 b138              0101011000 (0344)     [constant] loadConstant : 5                                        p: L6212:  .1.1.11...                  :  LOAD CONSTANT 5
L6139   r6 b139              0100011000 (0280)     [constant] loadConstant : 4                                        p: L6213:  .1...11...                  :  LOAD CONSTANT 4
L6140   r6 b140              0111111011 (0507)     [flow    ] goTo : 126                          → fst1              p: L6214:  .111111.11  -> L6176         :  GO TO FST1
L6141   r6 b141   con8       1101111010 (0890)     [register] using XS : decrement A                                  p: L6215:  11.1111.1.                  CON8  :  A - 1 -> A[XS]
L6142   r6 b142              1001100111 (0615)     [flow    ] (if no carry then) goTo : 153       → con9              p: L6216:  1..11..111  -> L6231          :  IF NO CARRY GO TO CON9
L6143   r6 b143              0100011000 (0280)     [constant] loadConstant : 4                                        p: L6217:  .1...11...                  :  LOAD CONSTANT 4
L6144   r6 b144              0101011000 (0344)     [constant] loadConstant : 5                                        p: L6220:  .1.1.11...                  :  LOAD CONSTANT 5
L6145   r6 b145              0011011000 (0216)     [constant] loadConstant : 3                                        p: L6221:  ..11.11...                  :  LOAD CONSTANT 3
L6146   r6 b146              0101011000 (0344)     [constant] loadConstant : 5                                        p: L6222:  .1.1.11...                  :  LOAD CONSTANT 5
L6147   r6 b147              1001011000 (0600)     [constant] loadConstant : 9                                        p: L6223:  1..1.11...                  :  LOAD CONSTANT 9
L6148   r6 b148              0010011000 (0152)     [constant] loadConstant : 2                                        p: L6224:  ..1..11...                  :  LOAD CONSTANT 2
L6149   r6 b149              0011011000 (0216)     [constant] loadConstant : 3                                        p: L6225:  ..11.11...                  :  LOAD CONSTANT 3
L6150   r6 b150              0111011000 (0472)     [constant] loadConstant : 7                                        p: L6226:  .111.11...                  :  LOAD CONSTANT 7
L6151   r6 b151              0101101010 (0362)     [register] using X : decrement C                                   p: L6227:  .1.11.1.1.                  :  C - 1 -> C[X]
L6152   r6 b152              0111111001 (0505)     [flow    ] (if no carry then) jumpSub : 126    → fst1              p: L6230:  .111111..1  -> L6176          :  JSB FST1
L6153   r6 b153   con9       0011011000 (0216)     [constant] loadConstant : 3                                        p: L6231:  ..11.11...                  CON9  :  LOAD CONSTANT 3
L6154   r6 b154              0111011000 (0472)     [constant] loadConstant : 7                                        p: L6232:  .111.11...                  :  LOAD CONSTANT 7
L6155   r6 b155              1000011000 (0536)     [constant] loadConstant : 8                                        p: L6233:  1....11...                  :  LOAD CONSTANT 8
L6156   r6 b156              0101011000 (0344)     [constant] loadConstant : 5                                        p: L6234:  .1.1.11...                  :  LOAD CONSTANT 5
L6157   r6 b157              0100011000 (0280)     [constant] loadConstant : 4                                        p: L6235:  .1...11...                  :  LOAD CONSTANT 4
L6158   r6 b158              0001011000 (0088)     [constant] loadConstant : 1                                        p: L6236:  ...1.11...                  :  LOAD CONSTANT 1
L6159   r6 b159              0001011000 (0088)     [constant] loadConstant : 1                                        p: L6237:  ...1.11...                  :  LOAD CONSTANT 1
L6160   r6 b160              0111011000 (0472)     [constant] loadConstant : 7                                        p: L6240:  .111.11...                  :  LOAD CONSTANT 7
L6161   r6 b161              1000011000 (0536)     [constant] loadConstant : 8                                        p: L6241:  1....11...                  :  LOAD CONSTANT 8
L6162   r6 b162              0100011000 (0280)     [constant] loadConstant : 4                                        p: L6242:  .1...11...                  :  LOAD CONSTANT 4
L6163   r6 b163              0111111011 (0507)     [flow    ] goTo : 126                          → fst1              p: L6243:  .111111.11  -> L6176          :  GO TO FST1
L6164   r6 b164   sav2       0000001100 (0012)     [pointer ] value → P : value 0                                     p: L6244:  ......11..                  SAV2  :  0 -> P
L6165   r6 b165   sav1       0100001110 (0270)     [register] using W : leftShift A                                   p: L6245:  .1....111.                  SAV1  :  SHIFT LEFT A[W]
L6166   r6 b166              0000111100 (0060)     [pointer ] increment P                                             p: L6246:  ....1111..                  :  P + 1 -> P
L6167   r6 b167              1100101100 (0812)     [pointer ] if P ≠ value : value 12                                 p: L6247:  11..1.11..                  :  IF P # 12
L6168   r6 b168              1010010111 (0663)     [flow    ]    (then) goTo : 165                → sav1              p: L6250:  1.1..1.111  -> L6245          :       THEN GO TO SAV1
L6169   r6 b169              1011111110 (0766)     [register] using S : clear A                                       p: L6251:  1.1111111.                  :  0 -> A[S]
L6170   r6 b170              1110101110 (0942)     [register] using W : exchange A C                                  p: L6252:  111.1.111.                  :  A EXCHANGE C[W]
L6171   r6 b171              1001110000 (0624)     [other   ] (data) C → dataAddress                                  p: L6253:  1..111....                  :  C -> DATA ADDRESS
L6172   r6 b172              0010100100 (0164)     [status  ] clear statusBit : bit 2                                 p: L6254:  ..1.1..1..                  :  0 -> S2
L6173   r6 b173              1011111000 (0760)     [other   ] (data) data → C                                         p: L6255:  1.11111...                  :  DATA -> C
L6174   r6 b174              1110101110 (0942)     [register] using W : exchange A C                                  p: L6256:  111.1.111.                  :  A EXCHANGE C[W]
L6175   r6 b175              1011100100 (0740)     [status  ] clear statusBit : bit 11                                p: L6257:  1.111..1..                  :  0 -> S11
L6176   r6 b176              0000101110 (0046)     [register] using W : clear B                                       p: L6260:  ....1.111.                  :  0 -> B[W]
L6177   r6 b177              0001010100 (0084)     [status  ] if statusBit ≠ 1 : bit 1                                p: L6261:  ...1.1.1..                  :  IF S1 # 1
L6178   r6 b178              1111101111 (1007)     [flow    ]    (then) goTo : 251                → svrt              p: L6262:  11111.1111  -> L6373           :       THEN GO TO SVRT
L6179   r6 b179              0110010111 (0407)     [flow    ] goTo : 101                          → adr9z4            p: L6263:  .11..1.111  -> L6145          :  GO TO ADR9Z4
```

```
L6180   r6 b180   save       0100101110 (0302)     [register] using W : A → B
L6181   r6 b181              1110101110 (0942)     [register] using W : exchange A C
L6182   r6 b182              0011001110 (0206)     [register] using W : clear c
L6183   r6 b183              1001110000 (0624)     [other   ] (data) C → dataAddress
L6184   r6 b184              0010001110 (0142)     [register] using W : B → C
L6185   r6 b185              1110101110 (0942)     [register] using W : exchange A C
L6186   r6 b186              1011110000 (0752)     [other   ] (data) C → data
L6187   r6 b187              0110010011 (0403)     [flow    ] goTo : 100                          → sav2zx
L6188   r6 b188   fact       0001111110 (0126)     [register] using S : if C ≥ 1
L6189   r6 b189              0000011111 (0031)     [flow    ]    (then) goTo : 7                  → errr
L6190   r6 b190              0001111010 (0122)     [register] using XS : if C ≥ 1
L6191   r6 b191              0000011111 (0031)     [flow    ]    (then) goTo : 7                  → errr
L6192   r6 b192   fac0       0001101010 (0106)     [register] using X : if C ≥ 1
L6193   r6 b193              1100010011 (0787)     [flow    ]    (then) goTo : 196                → fac1
L6194   r6 b194              0000011100 (0028)     [pointer ] decrement P
L6195   r6 b195              1100101111 (0815)     [flow    ] goTo : 203                          → fact1
L6196   r6 b196   fac1       0000011100 (0028)     [pointer ] decrement P
L6197   r6 b197              0011101100 (0236)     [pointer ] if P ≠ value : value 3
L6198   r6 b198              1100100011 (0803)     [flow    ]    (then) goTo : 200                → fac2
L6199   r6 b199              0000010011 (0019)     [flow    ] goTo : 4                            → oflw
L6200   r6 b200   fac2       0101101010 (0362)     [register] using X : decrement C
L6201   r6 b201              1100000001 (0769)     [flow    ] (if no carry then) jumpSub : 192    → fac0
L6202   r6 b202   nrm20      0100010000 (0272)     [rom     ] selectRom : 2
L6203   r6 b203   fact1      0001110010 (0114)     [register] using WP : if C ≥ 1
L6204   r6 b204              0000011111 (0031)     [flow    ]    (then) goTo : 7                  → errr
L6205   r6 b205              1110101010 (0938)     [register] using X : exchange A C
L6206   r6 b206              1011001100 (0716)     [pointer ] value → P : value 11
L6207   r6 b207              0110101010 (0426)     [register] using X : if C = 0
L6208   r6 b208              1101010111 (0855)     [flow    ]    (then) goTo : 213                → fact2
L6209   r6 b209              0101101010 (0362)     [register] using X : decrement C
L6210   r6 b210              0001101010 (0106)     [register] using X : if C ≥ 1
L6211   r6 b211              0000010011 (0019)     [flow    ]    (then) goTo : 4                  → oflw
L6212   r6 b212              0100001110 (0270)     [register] using W : leftShift A
L6213   r6 b213   fact2      1110101110 (0942)     [register] using W : exchange A C
L6214   r6 b214              1011101110 (0750)     [register] using W : clear A
L6215   r6 b215              1111100010 (0994)     [register] using P : increment A
L6216   r6 b216              0010101110 (0174)     [register] using W : 0 − C → C
L6217   r6 b217              1100101011 (0811)     [flow    ] (if no carry then) goTo : 202       → nrm20
L6218   r6 b218              1110101110 (0942)     [register] using W : exchange A C
L6219   r6 b219              1001001110 (0590)     [register] using W : rightShift C
L6220   r6 b220              0111111110 (0510)     [register] using S : increment C
L6221   r6 b221   fact3      1100001100 (0780)     [pointer ] value → P : value 12
L6222   r6 b222              0100110110 (0310)     [register] using MS : A → B
L6223   r6 b223   fact4      1111001110 (0974)     [register] using W : A + C → A
L6224   r6 b224              1101111111 (0895)     [flow    ] (if no carry then) goTo : 223       → fact4

p: L6264:  .1..1.111.                    SAVE  :  A -> B[W]
p: L6265:  111.1.111.                          :  A EXCHANGE C[W]
p: L6266:  ..11..111.                          :  0 -> C[W]
p: L6267:  1..111....                          :  C -> DATA ADDRESS
p: L6270:  ..1...111.                          :  B -> C[W]
p: L6271:  111.1.111.                          :  A EXCHANGE C[W]
p: L6272:  1.1111....                          :  C -> DATA
p: L6273:  .11..1..11  -> L6144                :  GO TO SAV2ZX
p: L6274:  ...111111.                    FACT  :  IF C[S] >= 1
p: L6275:  .....11111  -> L6007                :       THEN GO TO ERRR
p: L6276:  ...1111.1.                          :  IF C[XS] >= 1
p: L6277:  .....11111  -> L6007                :       THEN GO TO ERRR
p: L6300:  ...11.1.1.                    FAC0  :  IF C[X] >= 1
p: L6301:  11...1..11  -> L6304                :       THEN GO TO FAC1
p: L6302:  .....111..                          :  P − 1 -> P
p: L6303:  11..1.1111  -> L6313                :  GO TO FACT1
p: L6304:  .....111..                    FAC1  :  P − 1 -> P
p: L6305:  ..111.11..                          :  IF P # 3
p: L6306:  11..1...11  -> L6310                :       THEN GO TO FAC2
p: L6307:  .....1..11  -> L6004                :  GO TO OFLW
p: L6310:  .1.11.1.1.                    FAC2  :  C − 1 -> C[X]
p: L6311:  11.......1  -> L6300                :  JSB FAC0
p: L6312:  .1...1....  -> L2313   *****  NRM20 :  SELECT ROM 2
p: L6313:  ...111..1.                    FACT1 :  IF C[WP] >= 1
p: L6314:  .....11111  -> L6007                :       THEN GO TO ERRR
p: L6315:  111.1.1.1.                          :  A EXCHANGE C[X]
p: L6316:  1.11..11..                          :  11 -> P
p: L6317:  .11.1.1.1.                          :  IF C[X] = 0
p: L6320:  11.1.1.111  -> L6325                :       THEN GO TO FACT2
p: L6321:  .1.11.1.1.                          :  C − 1 -> C[X]
p: L6322:  ...11.1.1.                          :  IF C[X] >= 1
p: L6323:  .....1..11  -> L6004                :       THEN GO TO OFLW
p: L6324:  .1....111.                          :  SHIFT LEFT A[W]
p: L6325:  111.1.111.                    FACT2 :  A EXCHANGE C[W]
p: L6326:  1.111.111.                          :  0 -> A[W]
p: L6327:  11111...1.                          :  A + 1 -> A[P]
p: L6330:  ..1.1.111.                          :  0 − C -> C[W]
p: L6331:  11..1.1.11  -> L6312                :  IF NO CARRY GO TO NRM20
p: L6332:  111.1.111.                          :  A EXCHANGE C[W]
p: L6333:  1..1..111.                          :  SHIFT RIGHT C[W]
p: L6334:  .11111111.                          :  C + 1 -> C[S]
p: L6335:  11....11..                    FACT3 :  12 -> P
p: L6336:  .1..11.11.                          :  A -> B[MS]
p: L6337:  1111..111.                    FACT4 :  A + C -> A[W]
p: L6340:  11.1111111  -> L6337                :  IF NO CARRY GO TO FACT4
```

```
L6225   r6 b225             1101001110 (0846)    [register] using W : A − C → A
L6226   r6 b226             0100001110 (0270)    [register] using W : leftShift A
L6227   r6 b227   fact5      1111001110 (0974)    [register] using W : A + C → A
L6228   r6 b228             1110001111 (0911)    [flow    ] (if no carry then) goTo : 227       → fact5
L6229   r6 b229             1111111110 (1022)    [register] using S : increment A
L6230   r6 b230             1100101110 (0814)    [register] using W : exchange A B
L6231   r6 b231             1111011001 (0985)    [flow    ] jumpSub : 246                      → shft
L6232   r6 b232             1011001100 (0716)    [pointer ] value → P : value 11
L6233   r6 b233             1111011001 (0985)    [flow    ] jumpSub : 246                      → shft
L6234   r6 b234             0010001110 (0142)    [register] using W : B → C
L6235   r6 b235             0000110010 (0050)    [register] using WP : clear B
L6236   r6 b236             1010001110 (0654)    [register] using W : rightShift B
L6237   r6 b237             1100101110 (0814)    [register] using W : exchange A B
L6238   r6 b238             1110010110 (0918)    [register] using MS : A + B → A
L6239   r6 b239             1101110111 (0887)    [flow    ] (if no carry then) goTo : 221       → fact3
L6240   r6 b240             1110101110 (0942)    [register] using W : exchange A C
L6241   r6 b241             1000101010 (0554)    [register] using X : exchange B C
L6242   r6 b242   fact6      0111101010 (0490)    [register] using X : increment C
L6243   r6 b243   fact7      1100101001 (0809)    [flow    ] (if no carry then) jumpSub : 202    → nrm20
L6244   r6 b244   tnx3       1000100100 (0548)    [status  ] clear statusBit : bit 8
L6245   r6 b245             0100010000 (0272)    [rom     ] selectRom : 2
L6246   r6 b246   shft       0000000010 (0002)    [register] using P : if B = 0
L6247   r6 b247             1111101011 (1003)    [flow    ]    (then) goTo : 250                → shfr
L6248   r6 b248             1010010010 (0658)    [register] using WP : rightShift B
L6249   r6 b249             1111101010 (1002)    [register] using X : increment A
L6250   r6 b250   shfr       0000110000 (0048)    [other   ] (if no carry then) (flow) return
L6251   r6 b251   svrt       1010010100 (0660)    [status  ] if statusBit ≠ 1 : bit 10
L6252   r6 b252             1111111011 (1019)    [flow    ]    (then) goTo : 254                → ret3
L6253   r6 b253   ret4       1000010000 (0528)    [rom     ] selectRom : 4
L6254   r6 b254   ret3       0110010000 (0400)    [rom     ] selectRom : 3
L6255   r6 b255             0000000000 (0000)    [no oper ] nOp
```

```
p: L6341:  11.1..111.                        :  A − C -> A[W]
p: L6342:  .1....111.                        :  SHIFT LEFT A[W]
p: L6343:  1111..111.                  FACT5 :  A + C -> A[W]
p: L6344:  111...1111  -> L6343              :  IF NO CARRY GO TO FACT5
p: L6345:  111111111                         :  A + 1 -> A[S]
p: L6346:  11..1.111.                        :  A EXCHANGE B[W]
p: L6347:  1111.11..1  -> L6366              :  JSB SHFT
p: L6350:  1.11..11..                        :  11 -> P
p: L6351:  1111.11..1  -> L6366              :  JSB SHFT
p: L6352:  ..1...111.                        :  B -> C[W]
p: L6353:  ....11..1.                        :  0 -> B[WP]
p: L6354:  1.1...111.                        :  SHIFT RIGHT B[W]
p: L6355:  11..1.111.                        :  A EXCHANGE B[W]
p: L6356:  111..1.11.                        :  A + B -> A[MS]
p: L6357:  11.111.111  -> L6335              :  IF NO CARRY GO TO FACT3
p: L6360:  111.1.111.                        :  A EXCHANGE C[W]
p: L6361:  1...1.1.1.                        :  B EXCHANGE C[X]
p: L6362:  .1111.1.1.                  FACT6 :  C + 1 -> C[X]
p: L6363:  11..1.1..1  -> L6312        FACT7 :  JSB NRM20
p: L6364:  1...1..1..                  TNX3  :  0 -> S8
p: L6365:  .1...1....  -> L2366  *****       :  SELECT ROM 2
p: L6366:  ........1.                  SHFT  :  IF B[P] = 0
p: L6367:  11111.1.11  -> L6372              :       THEN GO TO SHFR
p: L6370:  1.1..1..1.                        :  SHIFT RIGHT B[WP]
p: L6371:  11111.1.1.                        :  A + 1 -> A[X]
p: L6372:  ....11....                  SHFR  :  RETURN
p: L6373:  1.1..1.1..                  SVRT  :  IF S10 # 1
p: L6374:  1111111.11  -> L6376              :       THEN GO TO RET3
p: L6375:  1....1....  -> L4376  *****  RET4  :  SELECT ROM 4
p: L6376:  .11..1....  -> L3377  *****  RET3  :  SELECT ROM 3
p: L6377:  ..........                        :  NO OPERATION
```

```
L7000   r7 b000   prfx       0000010111 (0023)     [flow    ] goTo : 5                          → wt22
L7001   r7 b001   wt30       1010001011 (0651)     [flow    ] goTo : 162                        → wt29
L7002   r7 b002   fix1       0000010111 (0023)     [flow    ] goTo : 5                          → wt22
L7003   r7 b003   expn       0000010111 (0023)     [flow    ] goTo : 5                          → wt22
L7004   r7 b004   lnnn       0000010111 (0023)     [flow    ] goTo : 5                          → wt22
L7005   r7 b005   wt22       0000110111 (0055)     [flow    ] goTo : 13                         → wt21
L7006   r7 b006   invx       0000010111 (0023)     [flow    ] goTo : 5                          → wt22
L7007   r7 b007   wt18       0001000111 (0071)     [flow    ] goTo : 17                         → wt17
L7008   r7 b008   perc       0000010111 (0023)     [flow    ] goTo : 5                          → wt22
L7009   r7 b009   wt09       1010011111 (0671)     [flow    ] goTo : 167                        → wt08
L7010   r7 b010   rcal       0000010111 (0023)     [flow    ] goTo : 5                          → wt22
L7011   r7 b011   stor       0000010111 (0023)     [flow    ] goTo : 5                          → wt22
L7012   r7 b012   rold       0000010111 (0023)     [flow    ] goTo : 5                          → wt22
L7013   r7 b013   wt21       1010010011 (0659)     [flow    ] goTo : 164                        → wt20
L7014   r7 b014   exc1       0000010111 (0023)     [flow    ] goTo : 5                          → wt22
L7015   r7 b015   reg9       1001011000 (0600)     [constant] loadConstant : 9
L7016   r7 b016              1001001111 (0591)     [flow    ] goTo : 147                        → rega
L7017   r7 b017   wt17       1010010111 (0663)     [flow    ] goTo : 165                        → wt16
L7018   r7 b018   dig6       0011000011 (0195)     [flow    ] goTo : 48                         → reg6
L7019   r7 b019   dig5       0001111111 (0127)     [flow    ] goTo : 31                         → reg5
L7020   r7 b020   dig4       0100011000 (0280)     [constant] loadConstant : 4
L7021   r7 b021              0111110111 (0503)     [flow    ] goTo : 125                        → reg1
L7022   r7 b022   addd       0000010111 (0023)     [flow    ] goTo : 5                          → wt22
L7023   r7 b023   reg8       1000011000 (0536)     [constant] loadConstant : 8
L7024   r7 b024              1001001111 (0591)     [flow    ] goTo : 147                        → rega
L7025   r7 b025   wt06       1010100011 (0675)     [flow    ] goTo : 168                        → wt05
L7026   r7 b026   dig3       0111101011 (0491)     [flow    ] goTo : 122                        → reg3
L7027   r7 b027   dig2       0111110011 (0499)     [flow    ] goTo : 124                        → reg2
L7028   r7 b028   dig1       0001011000 (0088)     [constant] loadConstant : 1
L7029   r7 b029              0111110111 (0503)     [flow    ] goTo : 125                        → reg1
L7030   r7 b030   mult       0000010111 (0023)     [flow    ] goTo : 5                          → wt22
L7031   r7 b031   reg5       0101011000 (0344)     [constant] loadConstant : 5
L7032   r7 b032              1001001111 (0591)     [flow    ] goTo : 147                        → rega
L7033   r7 b033   wt11       0010011111 (0159)     [flow    ] goTo : 39                         → wt10
L7034   r7 b034   sigp       1001000011 (0579)     [flow    ] goTo : 144                        → chs3
L7035   r7 b035   dcpt       1101000111 (0839)     [flow    ] goTo : 209                        → retn
L7036   r7 b036   dig0       0000011000 (0024)     [constant] loadConstant : 0
L7037   r7 b037              0111110111 (0503)     [flow    ] goTo : 125                        → reg1
L7038   r7 b038   divd       0000010111 (0023)     [flow    ] goTo : 5                          → wt22
L7039   r7 b039   wt10       0000100111 (0039)     [flow    ] goTo : 9                          → wt09
L7040   r7 b040   tang       0000010111 (0023)     [flow    ] goTo : 5                          → wt22
L7041   r7 b041   dl06       0010110111 (0183)     [flow    ] goTo : 45                         → dl05
L7042   r7 b042   coss       0000010111 (0023)     [flow    ] goTo : 5                          → wt22
L7043   r7 b043   sinn       0000010111 (0023)     [flow    ] goTo : 5                          → wt22
L7044   r7 b044   tpol       0000010111 (0023)     [flow    ] goTo : 5                          → wt22
```

```
p: L7000:  .....1.111  -> L7005          PRFX  :  GO TO WT22
p: L7001:  1.1...1.11  -> L7242          WT30  :  GO TO WT29
p: L7002:  .....1.111  -> L7005          FIX1  :  GO TO WT22
p: L7003:  .....1.111  -> L7005          EXPN  :  GO TO WT22
p: L7004:  .....1.111  -> L7005          LNNN  :  GO TO WT22
p: L7005:  ....11.111  -> L7015          WT22  :  GO TO WT21
p: L7006:  .....1.111  -> L7005          INVX  :  GO TO WT22
p: L7007:  ...1...111  -> L7021          WT18  :  GO TO WT17
p: L7010:  .....1.111  -> L7005          PERC  :  GO TO WT22
p: L7011:  1.1..11111  -> L7247          WT09  :  GO TO WT08
p: L7012:  .....1.111  -> L7005          RCAL  :  GO TO WT22
p: L7013:  .....1.111  -> L7005          STOR  :  GO TO WT22
p: L7014:  .....1.111  -> L7005          ROLD  :  GO TO WT22
p: L7015:  1.1..1..11  -> L7244          WT21  :  GO TO WT20
p: L7016:  .....1.111  -> L7005          EXC1  :  GO TO WT22
p: L7017:  1..1.11...                    REG9  :  LOAD CONSTANT 9
p: L7020:  1..1..1111  -> L7223                :  GO TO REGA
p: L7021:  1.1..1.111  -> L7245          WT17  :  GO TO WT16
p: L7022:  ..11....11  -> L7060          DIG6  :  GO TO REG6
p: L7023:  ...1111111  -> L7037          DIG5  :  GO TO REG5
p: L7024:  .1...11...                    DIG4  :  LOAD CONSTANT 4
p: L7025:  .11111.111  -> L7175                :  GO TO REG1
p: L7026:  .....1.111  -> L7005          ADDD  :  GO TO WT22
p: L7027:  1....11...                    REG8  :  LOAD CONSTANT 8
p: L7030:  1..1..1111  -> L7223                :  GO TO REGA
p: L7031:  1.1.1...11  -> L7250          WT06  :  GO TO WT05
p: L7032:  .1111.1.11  -> L7172          DIG3  :  GO TO REG3
p: L7033:  .11111..11  -> L7174          DIG2  :  GO TO REG2
p: L7034:  ...1.11...                    DIG1  :  LOAD CONSTANT 1
p: L7035:  .11111.111  -> L7175                :  GO TO REG1
p: L7036:  .....1.111  -> L7005          MULT  :  GO TO WT22
p: L7037:  .1.1.11...                    REG5  :  LOAD CONSTANT 5
p: L7040:  1..1..1111  -> L7223                :  GO TO REGA
p: L7041:  ..1..11111  -> L7047          WT11  :  GO TO WT10
p: L7042:  1..1....11  -> L7220          SIGP  :  GO TO CHS3
p: L7043:  11.1...111  -> L7321          DCPT  :  GO TO RETN
p: L7044:  .....11...                    DIG0  :  LOAD CONSTANT 0
p: L7045:  .11111.111  -> L7175                :  GO TO REG1
p: L7046:  .....1.111  -> L7005          DIVD  :  GO TO WT22
p: L7047:  ....1..111  -> L7011          WT10  :  GO TO WT09
p: L7050:  .....1.111  -> L7005          TANG  :  GO TO WT22
p: L7051:  ..1.11.111  -> L7055          DL06  :  GO TO DL05
p: L7052:  .....1.111  -> L7005          COSS  :  GO TO WT22
p: L7053:  .....1.111  -> L7005          SINN  :  GO TO WT22
p: L7054:  .....1.111  -> L7005          TPOL  :  GO TO WT22
```

```
L7045   r7 b045   dl05       0010111111 (0191)     [flow    ] goTo : 47                          → dl04
L7046   r7 b046   sqar       0000010111 (0023)     [flow    ] goTo : 5                           → wt22
L7047   r7 b047   dl04       0011011111 (0223)     [flow    ] goTo : 55                          → dl03
L7048   r7 b048   reg6       0110011000 (0408)     [constant] loadConstant : 6
L7049   r7 b049              1001001111 (0591)     [flow    ] goTo : 147                         → rega
L7050   r7 b050   dig9       0000111111 (0063)     [flow    ] goTo : 15                          → reg9
L7051   r7 b051   dig8       0001011111 (0095)     [flow    ] goTo : 23                          → reg8
L7052   r7 b052   dig7       0111011000 (0472)     [constant] loadConstant : 7
L7053   r7 b053              0111110111 (0503)     [flow    ] goTo : 125                         → reg1
L7054   r7 b054   subt       0000010111 (0023)     [flow    ] goTo : 5                           → wt22
L7055   r7 b055   dl03       1111110111 (1015)     [flow    ] goTo : 253                         → rtrn
L7056   r7 b056   clrx       1011101110 (0750)     [register] using W : clear A
L7057   r7 b057              0000110101 (0053)     [flow    ] jumpSub : 13                       → wt21
L7058   r7 b058   eexx       0111010011 (0467)     [flow    ] goTo : 116                         → eex1
L7059   r7 b059   chs1       1000111011 (0571)     [flow    ] goTo : 142                         → chs2
L7060   r7 b060   clok       0111001011 (0459)     [flow    ] goTo : 114                         → chs4
L7061   r7 b061   ent1       0110010000 (0400)     [rom     ] selectRom : 3
L7062   r7 b062   ent1z3     1001010100 (0596)     [status  ] if statusBit ≠ 1 : bit 9
L7063   r7 b063              0011110111 (0247)     [flow    ]    (then) goTo : 61                → ent1
L7064   r7 b064   init       0000110100 (0052)     [status  ] clear all statusBits
L7065   r7 b065              0001111110 (0126)     [register] using S : if C ≥ 1
L7066   r7 b066              1111111011 (1019)     [flow    ]    (then) goTo : 254               → err2
L7067   r7 b067              0101101010 (0362)     [register] using X : decrement C
L7068   r7 b068              0101101010 (0362)     [register] using X : decrement C
L7069   r7 b069              0110111010 (0442)     [register] using XS : if C = 0
L7070   r7 b070              1111111011 (1019)     [flow    ]    (then) goTo : 254               → err2
L7071   r7 b071              0100100111 (0295)     [flow    ] goTo : 73                          → int2
L7072   r7 b072   int1       1001000110 (0582)     [register] using M : rightShift C
L7073   r7 b073   int2       0111101010 (0490)     [register] using X : increment C
L7074   r7 b074              0100100011 (0291)     [flow    ] (if no carry then) goTo : 72       → int1
L7075   r7 b075   int3       1110101110 (0942)     [register] using W : exchange A C
L7076   r7 b076              1010101000 (0680)     [other   ] (register) M → C
L7077   r7 b077              0001011000 (0088)     [constant] loadConstant : 1
L7078   r7 b078              0011011000 (0216)     [constant] loadConstant : 3
L7079   r7 b079              0001000110 (0070)     [register] using M : if A ≥ C
L7080   r7 b080              1111111011 (1019)     [flow    ]    (then) goTo : 254               → err2
L7081   r7 b081              0110011000 (0408)     [constant] loadConstant : 6
L7082   r7 b082              1010001100 (0652)     [pointer ] value → P : value 10
L7083   r7 b083              0001000010 (0066)     [register] using P : if A ≥ C
L7084   r7 b084              1111111011 (1019)     [flow    ]    (then) goTo : 254               → err2
L7085   r7 b085              1000001100 (0524)     [pointer ] value → P : value 8
L7086   r7 b086              1011010010 (0722)     [register] using WP : rightShift A
L7087   r7 b087              0111001100 (0460)     [pointer ] value → P : value 7
L7088   r7 b088              0110011000 (0408)     [constant] loadConstant : 6
L7089   r7 b089              0111001100 (0460)     [pointer ] value → P : value 7
```

```
p: L7055:  ..1.111111  -> L7057          DL05  :  GO TO DL04
p: L7056:  .....1.111  -> L7005          SQAR  :  GO TO WT22
p: L7057:  ..11.11111  -> L7067          DL04  :  GO TO DL03
p: L7060:  .11..11...                    REG6  :  LOAD CONSTANT 6
p: L7061:  1..1..1111  -> L7223                :  GO TO REGA
p: L7062:  ....111111  -> L7017          DIG9  :  GO TO REG9
p: L7063:  ...1.11111  -> L7027          DIG8  :  GO TO REG8
p: L7064:  .111.11...                    DIG7  :  LOAD CONSTANT 7
p: L7065:  .11111.111  -> L7175                :  GO TO REG1
p: L7066:  .....1.111  -> L7005          SUBT  :  GO TO WT22
p: L7067:  111111.111  -> L7375          DL03  :  GO TO RTRN
p: L7070:  1.111.111.                    CLRX  :  0 -> A[W]
p: L7071:  ....11.1.1  -> L7015                :  JSB WT21
p: L7072:  .111.1..11  -> L7164          EEXX  :  GO TO EEX1
p: L7073:  1...111.11  -> L7216          CHS1  :  GO TO CHS2
p: L7074:  .111..1.11  -> L7162          CLOK  :  GO TO CHS4
p: L7075:  .11..1....  -> L3076  *****   ENT1  :  SELECT ROM 3
p: L7076:  1..1.1.1..                    ENT1Z3:  IF S9 # 1
p: L7077:  ..1111.111  -> L7075                :       THEN GO TO ENT1
p: L7100:  ....11.1..                    INIT  :  CLEAR STATUS
p: L7101:  ...111111.                          :  IF C[S] >= 1
p: L7102:  1111111.11  -> L7376                :       THEN GO TO ERR2
p: L7103:  .1.11.1.1.                          :  C - 1 -> C[X]
p: L7104:  .1.11.1.1.                          :  C - 1 -> C[X]
p: L7105:  .11.111.1.                          :  IF C[XS] = 0
p: L7106:  1111111.11  -> L7376                :       THEN GO TO ERR2
p: L7107:  .1..1..111  -> L7111                :  GO TO INT2
p: L7110:  1..1...11.                    INT1  :  SHIFT RIGHT C[M]
p: L7111:  .1111.1.1.                    INT2  :  C + 1 -> C[X]
p: L7112:  .1..1...11  -> L7110                :  IF NO CARRY GO TO INT1
p: L7113:  111.1.111.                    INT3  :  A EXCHANGE C[W]
p: L7114:  1.1.1.1...                          :  M -> C
p: L7115:  ...1.11...                          :  LOAD CONSTANT 1
p: L7116:  ..11.11...                          :  LOAD CONSTANT 3
p: L7117:  ...1...11.                          :  IF A >= C[M]
p: L7120:  1111111.11  -> L7376                :       THEN GO TO ERR2
p: L7121:  .11..11...                          :  LOAD CONSTANT 6
p: L7122:  1.1...11..                          :  10 -> P
p: L7123:  ...1....1.                          :  IF A >= C[P]
p: L7124:  1111111.11  -> L7376                :       THEN GO TO ERR2
p: L7125:  1.....11..                          :  8 -> P
p: L7126:  1.11.1..1.                          :  SHIFT RIGHT A[WP]
p: L7127:  .111..11..                          :  7 -> P
p: L7130:  .11..11...                          :  LOAD CONSTANT 6
p: L7131:  .111..11..                          :  7 -> P
```

```
L7090   r7 b090             0001000010 (0066)    [register] using P : if A ≥ C                          p: L7132:  ...1....1.                        :  IF A >= C[P]
L7091   r7 b091             1111111011 (1019)    [flow    ]    (then) goTo : 254              → err2     p: L7133:  1111111.11  -> L7376             :       THEN GO TO ERR2
L7092   r7 b092             0010101000 (0168)    [other   ] (register) exchange C M                      p: L7134:  ..1.1.1...                        :  C EXCHANGE M
L7093   r7 b093             0101001100 (0332)    [pointer ] value → P : value 5                         p: L7135:  .1.1..11..                        :  5 -> P
L7094   r7 b094             1011010010 (0722)    [register] using WP : rightShift A                     p: L7136:  1.11.1..1.                        :  SHIFT RIGHT A[WP]
L7095   r7 b095             1011010010 (0722)    [register] using WP : rightShift A                     p: L7137:  1.11.1..1.                        :  SHIFT RIGHT A[WP]
L7096   r7 b096             1011010010 (0722)    [register] using WP : rightShift A                     p: L7140:  1.11.1..1.                        :  SHIFT RIGHT A[WP]
L7097   r7 b097             1011010010 (0722)    [register] using WP : rightShift A                     p: L7141:  1.11.1..1.                        :  SHIFT RIGHT A[WP]
L7098   r7 b098             1100101110 (0814)    [register] using W : exchange A B                      p: L7142:  11..1.111.                        :  A EXCHANGE B[W]
L7099   r7 b099             1101110010 (0882)    [register] using WP : decrement A                      p: L7143:  11.111..1.                        :  A - 1 -> A[WP]
L7100   r7 b100             1011101010 (0746)    [register] using X : clear A                           p: L7144:  1.111.1.1.                        :  0 -> A[X]
L7101   r7 b101             1000001100 (0524)    [pointer ] value → P : value 8                         p: L7145:  1.....11..                        :  8 -> P
L7102   r7 b102             1101100010 (0866)    [register] using P : decrement A                       p: L7146:  11.11...1.                        :  A - 1 -> A[P]
L7103   r7 b103             1011001100 (0716)    [pointer ] value → P : value 11                        p: L7147:  1.11..11..                        :  11 -> P
L7104   r7 b104             1111100010 (0994)    [register] using P : increment A                       p: L7150:  11111...1.                        :  A + 1 -> A[P]
L7105   r7 b105             1111100010 (0994)    [register] using P : increment A                       p: L7151:  11111...1.                        :  A + 1 -> A[P]
L7106   r7 b106             1101111110 (0894)    [register] using S : decrement A                       p: L7152:  11.111111.                        :  A - 1 -> A[S]
L7107   r7 b107             1100101110 (0814)    [register] using W : exchange A B                      p: L7153:  11..1.111.                        :  A EXCHANGE B[W]
L7108   r7 b108   key2      0000100100 (0036)    [status  ] clear statusBit : bit 0                     p: L7154:  ....1..1..                  KEY2  :  0 -> S0
L7109   r7 b109             0010100101 (0165)    [flow    ] jumpSub : 41                 → dl06     p: L7155:  ..1.1..1.1  -> L7051              :  JSB DL06
L7110   r7 b110             0010100101 (0165)    [flow    ] jumpSub : 41                 → dl06     p: L7156:  ..1.1..1.1  -> L7051              :  JSB DL06
L7111   r7 b111             0000010100 (0020)    [status  ] if statusBit ≠ 1 : bit 0                    p: L7157:  .....1.1..                        :  IF S0 # 1
L7112   r7 b112             1000110011 (0563)    [flow    ]    (then) goTo : 140              → key3     p: L7160:  1...11..11  -> L7214             :       THEN GO TO KEY3
L7113   r7 b113             1010011011 (0667)    [flow    ] goTo : 166                   → wt12     p: L7161:  1.1..11.11  -> L7246              :  GO TO WT12
L7114   r7 b114   chs4      1010000100 (0644)    [status  ] set statusBit : bit 10                      p: L7162:  1.1....1..                  CHS4  :  1 -> S10
L7115   r7 b115             1010010011 (0659)    [flow    ] goTo : 164                   → wt20     p: L7163:  1.1..1..11  -> L7244               :  GO TO WT20
L7116   r7 b116   eex1      0000001010 (0010)    [register] using X : if B = 0                          p: L7164:  ......1.1.                  EEX1  :  IF B[X] = 0
L7117   r7 b117             1000100011 (0547)    [flow    ]    (then) goTo : 136              → eex2     p: L7165:  1...1...11  -> L7210             :       THEN GO TO EEX2
L7118   r7 b118             0000101010 (0042)    [register] using X : clear B                           p: L7166:  ....1.1.1.                        :  0 -> B[X]
L7119   r7 b119             0000011111 (0031)    [flow    ] goTo : 7                     → wt18     p: L7167:  .....11111  -> L7007               :  GO TO WT18
L7120   r7 b120             0000000000 (0000)    [no oper ] nOp                                         p: L7170:  ..........                        :  NO OPERATION
L7121   r7 b121             0000000000 (0000)    [no oper ] nOp                                         p: L7171:  ..........                        :  NO OPERATION
L7122   r7 b122   reg3      0011011000 (0216)    [constant] loadConstant : 3                            p: L7172:  ..11.11...                  REG3  :  LOAD CONSTANT 3
L7123   r7 b123             1001001111 (0591)    [flow    ] goTo : 147                   → rega     p: L7173:  1..1..1111  -> L7223               :  GO TO REGA
L7124   r7 b124   reg2      0010011000 (0152)    [constant] loadConstant : 2                            p: L7174:  ..1..11...                  REG2  :  LOAD CONSTANT 2
L7125   r7 b125   reg1      1001001111 (0591)    [flow    ] goTo : 147                   → rega     p: L7175:  1..1..1111  -> L7223               :  GO TO REGA
L7126   r7 b126   fst1      0110010000 (0400)    [rom     ] selectRom : 3                               p: L7176:  .11..1....  -> L3177   *****  FST1  :  SELECT ROM 3
L7127   r7 b127   key1      0000010100 (0020)    [status  ] if statusBit ≠ 1 : bit 0                    p: L7177:  .....1.1..                  KEY1  :  IF S0 # 1
L7128   r7 b128             0000000111 (0007)    [flow    ]    (then) goTo : 1                → wt30     p: L7200:  .......111  -> L7001             :       THEN GO TO WT30
L7129   r7 b129             1000010100 (0532)    [status  ] if statusBit ≠ 1 : bit 8                    p: L7201:  1....1.1..                        :  IF S8 # 1
L7130   r7 b130             0110110011 (0435)    [flow    ]    (then) goTo : 108              → key2     p: L7202:  .11.11..11  -> L7154               :       THEN GO TO KEY2
L7131   r7 b131             1000101000 (0552)    [other   ] (display) displayOff                        p: L7203:  1...1.1...                        :  DISPLAY OFF
L7132   r7 b132             1000100100 (0548)    [status  ] clear statusBit : bit 8                     p: L7204:  1...1..1..                        :  0 -> S8
L7133   r7 b133             1100001100 (0780)    [pointer ] value → P : value 12                        p: L7205:  11....11..                        :  12 -> P
L7134   r7 b134             0011001110 (0206)    [register] using W : clear c                           p: L7206:  ..11..111.                        :  0 -> C[W]
```

```
L7135   r7 b135             0011010000 (0208)   [rom     ] keys → romAddress : 1
L7136   r7 b136   eex2      1100101010 (0810)   [register] using X : exchange A B
L7137   r7 b137             1101101010 (0874)   [register] using X : decrement A
L7138   r7 b138             1100101010 (0810)   [register] using X : exchange A B
L7139   r7 b139             1010010111 (0663)   [flow    ] goTo : 165                       → wt16
L7140   r7 b140   key3      1000000100 (0516)   [status  ] set statusBit : bit 8
L7141   r7 b141             0010000111 (0135)   [flow    ] goTo : 33                        → wt11
L7142   r7 b142   chs2      1010010100 (0660)   [status  ] if statusBit ≠ 1 : bit 10
L7143   r7 b143             0111001011 (0459)   [flow    ]    (then) goTo : 114             → chs4
L7144   r7 b144   chs3      1010100100 (0676)   [status  ] clear statusBit : bit 10
L7145   r7 b145             1001110000 (0624)   [other   ] (data) C → dataAddress
L7146   r7 b146             1001011011 (0603)   [flow    ] goTo : 150                       → chs5
L7147   r7 b147   rega      1001110000 (0624)   [other   ] (data) C → dataAddress
L7148   r7 b148             1010010100 (0660)   [status  ] if statusBit ≠ 1 : bit 10
L7149   r7 b149             1001101111 (0623)   [flow    ]    (then) goTo : 155             → regb
L7150   r7 b150   chs5      1110101110 (0942)   [register] using W : exchange A C
L7151   r7 b151             0110001110 (0398)   [register] using W : C → A
L7152   r7 b152             1001000110 (0582)   [register] using M : rightShift C
L7153   r7 b153             1011110000 (0752)   [other   ] (data) C → data
L7154   r7 b154             1010011011 (0667)   [flow    ] goTo : 166                       → wt12
L7155   r7 b155   regb      1011111000 (0760)   [other   ] (data) data → C
L7156   r7 b156             1100001100 (0780)   [pointer ] value → P : value 12
L7157   r7 b157             0110100010 (0418)   [register] using P : if C = 0
L7158   r7 b158             1010000011 (0643)   [flow    ]    (then) goTo : 160             → regc
L7159   r7 b159             0011001110 (0206)   [register] using W : clear c
L7160   r7 b160   regc      1110101110 (0942)   [register] using W : exchange A C
L7161   r7 b161             0100010110 (0278)   [register] using MS : leftShift A
L7162   r7 b162   wt29      0010110101 (0181)   [flow    ] jumpSub : 45                     → dl05
L7163   r7 b163   wt24      0010111101 (0189)   [flow    ] jumpSub : 47                     → dl04
L7164   r7 b164   wt20      0010111101 (0189)   [flow    ] jumpSub : 47                     → dl04
L7165   r7 b165   wt16      0010111101 (0189)   [flow    ] jumpSub : 47                     → dl04
L7166   r7 b166   wt12      0010111101 (0189)   [flow    ] jumpSub : 47                     → dl04
L7167   r7 b167   wt08      0011011101 (0221)   [flow    ] jumpSub : 55                     → dl03
L7168   r7 b168   wt05      1010101000 (0680)   [other   ] (register) M → C
L7169   r7 b169             1000101000 (0552)   [other   ] (display) displayOff
L7170   r7 b170             0000101000 (0040)   [other   ] (display) displayToggle
L7171   r7 b171             1010010100 (0660)   [status  ] if statusBit ≠ 1 : bit 10
L7172   r7 b172             0111111111 (0511)   [flow    ]    (then) goTo : 127             → key1
L7173   r7 b173   tim0      0001001100 (0076)   [pointer ] value → P : value 1                                  N
L7174   r7 b174             1111110010 (1010)   [register] using WP : increment A
L7175   r7 b175             0111111111 (0511)   [flow    ] (if no carry then) goTo : 127    → key1
L7176   r7 b176             0110001100 (0396)   [pointer ] value → P : value 6                                  N
L7177   r7 b177             1111100010 (0994)   [register] using P : increment A
L7178   r7 b178             1010001011 (0651)   [flow    ] (if no carry then) goTo : 162    → wt29
L7179   r7 b179             0111001100 (0460)   [pointer ] value → P : value 7
```

```
p: L7207:  ..11.1....                    :  KEYS -> ROM ADDRESS
p: L7210:  11..1.1.1.                EEX2  :  A EXCHANGE B[X]
p: L7211:  11.11.1.1.                      :  A - 1 -> A[X]
p: L7212:  11..1.1.1.                      :  A EXCHANGE B[X]
p: L7213:  1.1..1.111  -> L7245            :  GO TO WT16
p: L7214:  1......1..                KEY3  :  1 -> S8
p: L7215:  ..1....111  -> L7041            :  GO TO WT11
p: L7216:  1.1..1.1..                CHS2  :  IF S10 # 1
p: L7217:  .111..1.11  -> L7162            :       THEN GO TO CHS4
p: L7220:  1.1.1..1..                CHS3  :  0 -> S10
p: L7221:  1..111....                      :  C -> DATA ADDRESS
p: L7222:  1..1.11.11  -> L7226            :  GO TO CHS5
p: L7223:  1..111....                REGA  :  C -> DATA ADDRESS
p: L7224:  1.1..1.1..                      :  IF S10 # 1
p: L7225:  1..11.1111  -> L7233            :       THEN GO TO REGB
p: L7226:  111.1.111.                CHS5  :  A EXCHANGE C[W]
p: L7227:  .11...111.                      :  C -> A[W]
p: L7230:  1..1...11.                      :  SHIFT RIGHT C[M]
p: L7231:  1.1111....                      :  C -> DATA
p: L7232:  1.1..11.11  -> L7246            :  GO TO WT12
p: L7233:  1.11111...                REGB  :  DATA -> C
p: L7234:  11....11..                      :  12 -> P
p: L7235:  .11.1...1.                      :  IF C[P] = 0
p: L7236:  1.1.....11  -> L7240            :       THEN GO TO REGC
p: L7237:  ..11..111.                      :  0 -> C[W]
p: L7240:  111.1.111.                REGC  :  A EXCHANGE C[W]
p: L7241:  .1...1.11.                      :  SHIFT LEFT A[MS]
p: L7242:  ..1.11.1.1  -> L7055      WT29  :  JSB DL05
p: L7243:  ..1.1111.1  -> L7057      WT24  :  JSB DL04
p: L7244:  ..1.1111.1  -> L7057      WT20  :  JSB DL04
p: L7245:  ..1.1111.1  -> L7057      WT16  :  JSB DL04
p: L7246:  ..1.1111.1  -> L7057      WT12  :  JSB DL04
p: L7247:  ..11.111.1  -> L7067      WT08  :  JSB DL03
p: L7250:  1.1.1.1...                WT05  :  M -> C
p: L7251:  1...1.1...                      :  DISPLAY OFF
p: L7252:  ....1.1...                      :  DISPLAY TOGGLE
p: L7253:  1.1..1.1..                      :  IF S10 # 1
p: L7254:  0.111111111 -> L7177            :       THEN GO TO KEY1
p: L7255:  ...1..11..                TIM0  :  1 -> P
p: L7256:  111111..1.                      :  A + 1 -> A[WP]
p: L7257:  0.111111111 -> L7177            :  IF NO CARRY GO TO KEY1
p: L7260:  .11...11..                      :  6 -> P
p: L7261:  11111...1.                      :  A + 1 -> A[P]
p: L7262:  1.1...1.11  -> L7242            :  IF NO CARRY GO TO WT29
p: L7263:  .111..11..                      :  7 -> P
```

```
L7180   r7 b180          1111100010 (0994)    [register] using P : increment A
L7181   r7 b181          0001000010 (0066)    [register] using P : if A ≥ C
L7182   r7 b182          1011100011 (0739)    [flow    ]    (then) goTo : 184              → tim1
L7183   r7 b183          1010001111 (0655)    [flow    ] goTo : 163                        → wt24
L7184   r7 b184   tim1   1011110010 (0754)    [register] using WP : clear A
L7185   r7 b185          1001001100 (0588)    [pointer ] value → P : value 9                                 N
L7186   r7 b186          1111100010 (0994)    [register] using P : increment A
L7187   r7 b187          0000110111 (0055)    [flow    ] (if no carry then) goTo : 13      → wt21
L7188   r7 b188          1010001100 (0652)    [pointer ] value → P : value 10
L7189   r7 b189          1111100010 (0994)    [register] using P : increment A
L7190   r7 b190          0001000010 (0066)    [register] using P : if A ≥ C
L7191   r7 b191          1100000111 (0775)    [flow    ]    (then) goTo : 193              → tim2
L7192   r7 b192          1010010111 (0663)    [flow    ] goTo : 165                        → wt16
L7193   r7 b193   tim2   0011010010 (0210)    [register] using WP : clear c
L7194   r7 b194          1011110010 (0754)    [register] using WP : clear A
L7195   r7 b195          1011001100 (0716)    [pointer ] value → P : value 11                                N
L7196   r7 b196          1111100010 (0994)    [register] using P : increment A
L7197   r7 b197          1100101011 (0811)    [flow    ] (if no carry then) goTo : 202     → tim3
L7198   r7 b198          1011101110 (0750)    [register] using W : clear A
L7199   r7 b199          1100001100 (0780)    [pointer ] value → P : value 12
L7200   r7 b200          1111100010 (0994)    [register] using P : increment A
L7201   r7 b201          1010011111 (0671)    [flow    ] (if no carry then) goTo : 167     → wt08
L7202   r7 b202   tim3   0001000110 (0070)    [register] using M : if A ≥ C
L7203   r7 b203          1100110111 (0823)    [flow    ]    (then) goTo : 205              → tim4
L7204   r7 b204          0000100111 (0039)    [flow    ] goTo : 9                          → wt09
L7205   r7 b205   tim4   1011101110 (0750)    [register] using W : clear A
L7206   r7 b206          1011001100 (0716)    [pointer ] value → P : value 11
L7207   r7 b207          1111100010 (0994)    [register] using P : increment A
L7208   r7 b208          0001100111 (0103)    [flow    ] (if no carry then) goTo : 25      → wt06
L7209   r7 b209   retn   1010101000 (0680)    [other   ] (register) M → C
L7210   r7 b210          0011010010 (0210)    [register] using WP : clear c
L7211   r7 b211          0010001100 (0140)    [pointer ] value → P : value 2
L7212   r7 b212          0000001010 (0010)    [register] using X : if B = 0
L7213   r7 b213          1101100011 (0867)    [flow    ]    (then) goTo : 216              → ret1
L7214   r7 b214          0100011000 (0280)    [constant] loadConstant : 4
L7215   r7 b215          1101100111 (0871)    [flow    ] goTo : 217                        → ret2
L7216   r7 b216   ret1   0110011000 (0408)    [constant] loadConstant : 6
L7217   r7 b217   ret2   0010101000 (0168)    [other   ] (register) exchange C M
L7218   r7 b218          0011001110 (0206)    [register] using W : clear c
L7219   r7 b219          1100001100 (0780)    [pointer ] value → P : value 12
L7220   r7 b220          0000101110 (0046)    [register] using W : clear B
L7221   r7 b221   ret3   1001110000 (0624)    [other   ] (data) C → dataAddress
L7222   r7 b222          1000101110 (0558)    [register] using W : exchange B C
L7223   r7 b223          1011111000 (0760)    [other   ] (data) data → C
L7224   r7 b224          0001100010 (0098)    [register] using P : if C ≥ 1
```

```
p: L7264:  11111...1.                        :  A + 1 -> A[P]
p: L7265:  ...1....1.                        :  IF A >= C[P]
p: L7266:  1.111...11  -> L7270              :       THEN GO TO TIM1
p: L7267:  1.1...1111  -> L7243              :  GO TO WT24
p: L7270:  1.1111..1.                  TIM1  :  0 -> A[WP]
p: L7271:  1..1..11..                        :  9 -> P
p: L7272:  11111...1.                        :  A + 1 -> A[P]
p: L7273:  ....11.111  -> L7015              :  IF NO CARRY GO TO WT21
p: L7274:  1.1...11..                        :  10 -> P
p: L7275:  11111...1.                        :  A + 1 -> A[P]
p: L7276:  ...1....1.                        :  IF A >= C[P]
p: L7277:  11.....111  -> L7301              :       THEN GO TO TIM2
p: L7300:  1.1..1.111  -> L7245              :  GO TO WT16
p: L7301:  ..11.1..1.                  TIM2  :  0 -> C[WP]
p: L7302:  1.1111..1.                        :  0 -> A[WP]
p: L7303:  1.11..11..                        :  11 -> P
p: L7304:  11111...1.                        :  A + 1 -> A[P]
p: L7305:  11..1.1.11  -> L7312              :  IF NO CARRY GO TO TIM3
p: L7306:  1.111.111.                        :  0 -> A[W]
p: L7307:  11....11..                        :  12 -> P
p: L7310:  11111...1.                        :  A + 1 -> A[P]
p: L7311:  1.1..11111  -> L7247              :  IF NO CARRY GO TO WT08
p: L7312:  ...1...11.                  TIM3  :  IF A >= C[M]
p: L7313:  11..11.111  -> L7315              :       THEN GO TO TIM4
p: L7314:  ....1..111  -> L7011              :  GO TO WT09
p: L7315:  1.111.111.                  TIM4  :  0 -> A[W]
p: L7316:  1.11..11..                        :  11 -> P
p: L7317:  11111...1.                        :  A + 1 -> A[P]
p: L7320:  ...11..111  -> L7031              :  IF NO CARRY GO TO WT06
p: L7321:  1.1.1.1...                  RETN  :  M -> C
p: L7322:  ..11.1..1.                        :  0 -> C[WP]
p: L7323:  ..1...11..                        :  2 -> P
p: L7324:  ......1.1.                        :  IF B[X] = 0
p: L7325:  11.11...11  -> L7330              :       THEN GO TO RET1
p: L7326:  .1...11...                        :  LOAD CONSTANT 4
p: L7327:  11.11..111  -> L7331              :  GO TO RET2
p: L7330:  .11..11...                  RET1  :  LOAD CONSTANT 6
p: L7331:  ..1.1.1...                  RET2  :  C EXCHANGE M
p: L7332:  ..11..111.                        :  0 -> C[W]
p: L7333:  11....11..                        :  12 -> P
p: L7334:  ....1.111.                        :  0 -> B[W]
p: L7335:  1..111....                  RET3  :  C -> DATA ADDRESS
p: L7336:  1...1.111.                        :  B EXCHANGE C[W]
p: L7337:  1.11111...                        :  DATA -> C
p: L7340:  ...11...1.                        :  IF C[P] >= 1
```

```
L7225   r7 b225             1110010011 (0915)     [flow    ]    (then) goTo : 228              → ret4
L7226   r7 b226             1110101101 (0941)     [flow    ] jumpSub : 235                     → fixx
L7227   r7 b227             1011110000 (0752)     [other   ] (data) C → data
L7228   r7 b228   ret4       1000101110 (0558)     [register] using W : exchange B C
L7229   r7 b229             0111100010 (0482)     [register] using P : increment C
L7230   r7 b230             1101110111 (0887)     [flow    ] (if no carry then) goTo : 221     → ret3
L7231   r7 b231             1011000110 (0710)     [register] using M : rightShift A
L7232   r7 b232             1110101110 (0942)     [register] using W : exchange A C
L7233   r7 b233             1110101101 (0941)     [flow    ] jumpSub : 235                     → fixx
L7234   r7 b234             0111111011 (0507)     [flow    ] goTo : 126                        → fst1
L7235   r7 b235   fixx       0110110010 (0434)     [register] using WP : if C = 0
L7236   r7 b236             1111110111 (1015)     [flow    ]    (then) goTo : 253              → rtrn
L7237   r7 b237             1110101110 (0942)     [register] using W : exchange A C
L7238   r7 b238             0100001110 (0270)     [register] using W : leftShift A
L7239   r7 b239             1000001100 (0524)     [pointer ] value → P : value 8
L7240   r7 b240             0100010010 (0274)     [register] using WP : leftShift A
L7241   r7 b241             0110001100 (0396)     [pointer ] value → P : value 6
L7242   r7 b242             0100010010 (0274)     [register] using WP : leftShift A
L7243   r7 b243             0100010010 (0274)     [register] using WP : leftShift A
L7244   r7 b244             0100010010 (0274)     [register] using WP : leftShift A
L7245   r7 b245             1100001100 (0780)     [pointer ] value → P : value 12
L7246   r7 b246             1111101010 (1002)     [register] using X : increment A
L7247   r7 b247   fix2       1001100010 (0610)     [register] using P : if A ≥ 1
L7248   r7 b248             1111110011 (1011)     [flow    ]    (then) goTo : 252              → fxrt
L7249   r7 b249             1101101010 (0874)     [register] using X : decrement A
L7250   r7 b250             0100000110 (0262)     [register] using M : leftShift A
L7251   r7 b251             1111011111 (0991)     [flow    ] goTo : 247                        → fix2
L7252   r7 b252   fxrt       1110101110 (0942)     [register] using W : exchange A C
L7253   r7 b253   rtrn       0000110000 (0048)     [other   ] (flow) return
L7254   r7 b254   err2       0011001110 (0206)     [register] using W : clear c
L7255   r7 b255             0100101101 (0301)     [flow    ] jumpSub : 75                      → int3
```

```
p: L7341:  111..1..11  -> L7344              :       THEN GO TO RET4
p: L7342:  111.1.11.1  -> L7353              :  JSB FIXX
p: L7343:  1.1111....                        :  C -> DATA
p: L7344:  1...1.111.                  RET4  :  B EXCHANGE C[W]
p: L7345:  .1111...1.                        :  C + 1 -> C[P]
p: L7346:  11.111.111  -> L7335              :  IF NO CARRY GO TO RET3
p: L7347:  1.11...11.                        :  SHIFT RIGHT A[M]
p: L7350:  111.1.111.                        :  A EXCHANGE C[W]
p: L7351:  111.1.11.1  -> L7353              :  JSB FIXX
p: L7352:  .111111.11  -> L7176              :  GO TO FST1
p: L7353:  .11.11..1.                  FIXX  :  IF C[WP] = 0
p: L7354:  111111.111  -> L7375              :       THEN GO TO RTRN
p: L7355:  111.1.111.                        :  A EXCHANGE C[W]
p: L7356:  .1....111.                        :  SHIFT LEFT A[W]
p: L7357:  1.....11..                        :  8 -> P
p: L7360:  .1...1..1.                        :  SHIFT LEFT A[WP]
p: L7361:  .11...11..                        :  6 -> P
p: L7362:  .1...1..1.                        :  SHIFT LEFT A[WP]
p: L7363:  .1...1..1.                        :  SHIFT LEFT A[WP]
p: L7364:  .1...1..1.                        :  SHIFT LEFT A[WP]
p: L7365:  11....11..                        :  12 -> P
p: L7366:  11111.1.1.                        :  A + 1 -> A[X]
p: L7367:  1..11...1.                  FIX2  :  IF A[P] >= 1
p: L7370:  111111..11  -> L7374              :       THEN GO TO FXRT
p: L7371:  11.11.1.1.                        :  A - 1 -> A[X]
p: L7372:  .1.....11.                        :  SHIFT LEFT A[M]
p: L7373:  1111.11111  -> L7367              :  GO TO FIX2
p: L7374:  111.1.111.                  FXRT  :  A EXCHANGE C[W]
p: L7375:  ....11....                  RTRN  :  RETURN
p: L7376:  ..11..111.                  ERR2  :  0 -> C[W]
p: L7377:  .1..1.11.1  -> L7113              :  JSB INT3
```