Last update: 8/25/83
99/8 Bugs found after 7/23/83 release

(1) The variables TEMP2 and LNBUF+2 were in same location. Program lockup resulted during READ if (a) a user subprogram was called or (b) GCHAR was called, due to the loss of the pointer to the next DATA statement.
found:7/27/83
fix: move LNBUF to a new location, in PSCAN (grom1) ann FMEX (grom2).
action: Revised grom codes 4435B and 4436B were shipped on 7/28/83. Production of 4435A and 4436A was stopped at that time.

(2) Errors in IF-THEN-ELSE handler.
(a) IF --- THEN ---- ! tail remark with no ELSE gives syntax error.
(b) IF --- THEN line-number :: statement-2 does not execute statement-2 on false condition.

Both in PARSE in 32KB rom.
found: 7/28/83
fix:revised IF-THEN-ELSE code on 7/29/83
action:

(3) A$=RPT$("A",4090) gives spurious warning.
found: 7/28/83
fix: change JLT REPT2 to JLE REPT2 in line 706 of STRING in 32KB rom
action:

(4) MB expander communication.
In RXD routine, the statement INV R8 was too early in code, and could be executed before valid data was accepted, causing a lockup on the final handshake.
found: 7/28/83
fix: move the statement down to after the second validity test. in ROM0.
action:

(5) CALL LOAD does not report i/o errors properly. The op code is correct but the result code is always zero.
found: 7/29/83
fix: in DSRLNK change SRL R1,4 to SRL R1,5 to truly right justify the error code. In LOADER add SLA R0,4 to put code in most significant nibble for error handler. Both in 32KB rom.
action:

(6) The ON WARNING flag is not reset by a RUN.
found: 7/29/83
fix: clear the bit in KILSYM in the 32KB rom
action:

(7) In an error routine, CALL ERR makes RETURN NEXT fail. (Seems to replace pointer to error line with pointer to CALL ERR line.

found: 7/29/83
fix: (problem had nothing to do with ERR. RETURN NEXT
did not map in new line.) Add code to map in line on 8/1/83.
32KB rom
action:

(8) Parameter passing through multiple levels of subprograms
did not work in some array cases.
found: 8/01/83
fix: in SUBPRGX in 32KB rom, on 8/02/03
action:

(9) CALL INIT will lose open files.
found: 8/02/83
fix: the 'fix string pointers' routine stopped too soon.
Just extend to run through I/O table.
At the same time, a garbage collection vulnerability was
discovered and fixed. 8/03/83
action:

(10) VAL can give errors on correct input
found: 8/03/83
fix: 'off-by-one' errors in algorithm corrected.
8/04/83 in 32KB rom
action:

(11) CALL SOUND gives VDP crash if an EOF call is embedded.
found: 8/04/83
fix: move SOUND scratch buffer out of SLPAB (Grom0)
action:

(12) IF cond THEN GOSUB xxxx ELSE stmt1::stmt2
executes subroutine and then stmt2 if cond is true.
(4A XBasic bug)
found: 8/04/83
fix: add 'ELSE' handling code in NUDND routine used
by RETURN. (PARSE in 32KB rom)
action:

(13) BREAK line-number does not always allow a proper continue.
Text pointer was not being saved on breakpoint.
found: 8/04/83
fix: Add statement to save pointer before taking break,
in PARSE in 32KB rom.
action:

(14) CALL LOAD does not close its file on data error.
found: 8/04/83
fix: call to close code before error in ASSMSUP in 32KB rom
action:

(15) RANDOMIZE expression did not allow tail remarks or
continued statement.
found: 8/05/83
fix: change BL @PARSE, DATA 0 to BL @PARSE, DATA TREM$
in PARSE in 32KB rom
action:

(16) COMPATABILITY: 99/4 XBasic only changed y-velocity and y-position on DELSPRITE, so that sprite could be re-displayed with just a CALL LOCATE. 99/8 is initializing all parameters.
found: 8/07/83
fix: remove some code in GSUB (GROMO)
action:

(17) COMPATIBILITY: 99/4 XBasic CALL CHARSET affects only upper case & digits & punctuation (99/4 character set). some programs depend on that.
found: (early, but several programs from user group library were found to fail on 8/07/83)
fix: Change CALL CHARSET IN GSUB (GromO)
action:

(18) Scanning joystick leaves keyboard 1 or 2 selected, so that next INPUT or ACCEPT fails.
found: 8/05/83
fix: add CLR @KEYBD in INITKB in FLMGRSUP (32KB rom)
action:

(19) A pair of statement separators is not accepted on 99/8. It is in 99/4A basic. (e.g. FOR I=1 TO 100 :: :: PRINT I)
foundL 8/07/83
fix: add table entry in statement table in PARSE, and a couple of lines of code to support it (from XBasic)
action:

(20) POSITION returns values in the range 0-255 rather than 1-256.
found: 8/07/83
fix: Change INC @FAC+1 to DINC @FAC in POSIT3 in GSUB in grom0, so that 255->256 instead of zero.
action:

(21) COMPATABILITY: Warning messages look garbled in programs that use lower-case character set for graphics.
found: 8/05/08
fix: (possible) change all error/warngin messages to upper case.
action:

(22) RUN "file" / break / SAVE "file2",MERGE runs the program automatically after saving it.
found: 8/07/83
fix: change B LRMA0 to B LRMA1 at end of save-merge code to clear the GONOGO flag. in FMEX in grom2
action:

(23) GPL packages that try to print '0.' in fixed mode get only '.'. This goes back to a bug in extended basic, for which PRINT USING "###.":0 produces " ." instead of the desired " 0.".
found: 8/10/83
fix: delete two lines in CNSLEA in CNSXB in 32KB rom.
action:

(24) Accept at will sometimes bloop when you type in the last permitted character, instead when you overtype the

last character. VARW was not being cleared on entry.
found: 8/07/83
fix: move DCLR @VARW to label READ01 in PSCAN (grom1)
and
action:

(25) The following sequence causes the system to lock up:
>RUN DSK
*String number mismatch
>RUN "DSK1.FILENAME"
LOCKS UP
found:8/11/83
fix: Don't do a SETREG after a call from gpl
action:

(26) The GPL bit-reversal routine did not work.
found:8/11/83
fix:MOVB from VDP was done to wrong destination. We changed it to the right destination.

(27) If after arriving at a breakpoint one does anything other than a run or a continue, the breakpoint is not cleared.
found: pre-history, but forgotten.
fix: Giving breakpoints it's own variable into which it saves SMTSRT.
action:

(28) CNS does not handle certain underflow/overflow conditions.
I.e.; The number 123 printed into the format .DD yields the string 124
found: 8/16/83
fix: **********NOT FIXED**********
this is the same as EXTENDED BASIC.
action:

(29) ON ERR was not doing a CLSALL.
found:8/17/83
fix: Put in a CLSALL

(30) The LOAD subprogram in basic cannot load MUNCHMAN, INVADERS, and other games. This is because the loader that these programs invoke is AORG'ed at >2000 which is where T.I. Extended BASIC II has one of it's ROMS. We are writing a universal loader which will be able to load these games int a 4A as well as an 8.

(31) Command Module ROM was not mapped back in after an XML2 SPGROM, therefore command modules would not play if the P-system groms were in the system.
found: 8/21/83
fix: move XTAB9 to ROM 0, put SPGROM in XTAB 9 and change MONITOR to XML3 SPGROM.
action: Re-release CD6433B (ROM 0) and CD4434C (GROM 0)

(32) Disk based Speak and Spell will not play with T.I. Extended Basic II. This is because it was designed to run with the Speech Editor which does not care if there is garbage at the end of a string. It merely skips it and goes on looking until it either finds a >60 or runs out of string.
found:8/24/83
fix:

action:

(33) Scott Foresman Addition and Subtraction II will garbage the speech for "seventeen". This is because the speech data will not work on a TMS5200C, the new speech chip for the T.I. 99/8.
found: pre-history
fix:
action: Re-release SF Addition Subtraction II.

(34) OPEN #1:"DSK1.FILENAME",SEQUENTIAL 20 does not allocate 20 records on the disk. This was found to be a HEXBUS Interface compatability with the CC40 feature. The CC40 BASIC & Hexbus Interfaces clear the record number on an open whereas Extended Basic I & II use the record number to tell the dsr how many records to allocate.
found:8/27/83
fix:
action:

(35) We are now told to allow for ROM only applications.
found:8/27/83
fix:Re-insert the code we had deleted after we find the room.(GROM 0 only)
action:

(36) Very large programs [resolve this number] saved to disk on a 99/8 cannot b read by the 99/4A. This is because of a difference in the values of STLN & ENLN. On the TI99/8 these are the lower word of a two word physical pointer. Since the 4A assumes memtop to be FFFE, it ends up thinking that it does not have sufficient memory to load the program and retuns an IO error 20. Large programs written on a 4A can be read on an 8.
fix:*******NOT FIXED******
This can only be fixed by rewriting the OS from the ground up or re-releasing the 4A.
action: