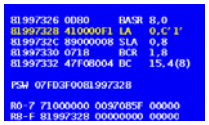
z390 ZSTRMAC Structured Macro Assembler Support



The z390 Portable Mainframe Macro Assembler and Emulator includes the following ZSTRMAC structured programming support tools:

- Structured Programming Extensions (SPE's) for use in coding structured conditional macro code
  - First SHARE presentation on this support scheduled for Tuesday March 3, 2009 in Austin
- Structured Programming Macros (SPM's) for use in coding assembler compatible with HLASM
  - IBM supplies set of SPM macros as part of HLASM Toolkit
  - The latest SHARE presentation on the HLASM SPM macros by Ed Jaffe
  - z390 currently has subset of compatible macros IF, ELSE, ELSEIF, ENDIF, DO, ENDDO
  - z390 macros are updated from original 1978 public domain SPM's in SHARE CBT tape #177
  - z390 v1.4.03b has the latest structured SPM's in the z390\mac directory. Also included are translated versions of all the structured SPM's in the mac\spm directory. The translated version are for use on systems with HLASM or other mainframe assemblers which do not yet support the SPE's. Using the SPM's and SPE's you never have to code another unconditional branch in macro code or assembler!

Structured Programming Extensions (SPE's) for use in coding structured conditional macro code

z390 Macro support has been extended to allow coding mainframe conditional macro code using the following structured programming extensions to the standard HLASM conditional macro language:

- Alternate selection of code blocks
  - AIF (expression) - execute the following block of code if expression is true
  - AELSEIF (expression) - end prior block and execute following block if expression is true
  - AELSE - end prior block and execute following block if prior AIF and AELSEIF false
  - AEND - end last block for AIF at current level of nesting
- Repeat execution of code block
  - AWHILE (expression) - repeat following code block while expression is true (test at beginning)
  - AEXIT AWHILE - exit to end of inner most AWHILE (for exceptions within nested AIF)
  - AEND - end block of code for AWHILE at current level of nesting
  - AUNTIL (expression) - repeat following code block until expression is true (test at end)
  - AEXIT AUNTIL - exit to end of inner most AUNTIL (for exceptions within nested AIF)
  - AEND - end block of code for AUNTIL at current level of nesting
- Perform code block
  - ACALL name - call the named code block and return to next instruction
  - AENTRY name - define start of performed block of code (skip over if entered sequentially)
  - AEXIT AENTRY - exit from AENTRY block of code (for exceptions within nested AIF)
  - AEND - end the current performed code block and exit to next statement after APM
- Perform selection of code blocks based on index from 0 to 255
  - ACASE (expression) - execute selected block based on value of index expression
  - AWHEN values - define end of previous block and start of code block for index values
    - values can be decimal (0-255), character C'?', or hex X'??'

        - one or more values may be specified separated by commas
        - a range of values may be specified as (value1,value2)
        - for example AWHEN (C'0',C'9') defines EBCDIC digits 240-249
    - AEXIT ACASE - exit to end of current ACASE (for exceptions within nested AIF)
    - AELSE - define optional code block if no AWHEN block defined for current index
    - AEND - end code block for ACASE
- Additional extension to indent label field by preceding with colon (:)
- Release Log:
    - v1.4.02e - a translator has been added (linklib\zstrmac.mlc) which will read structured macro assembler source code from SYSUT1 and write standard HLASM macro assembler source code on SYSUT2
    - v1.4.03 the support is included in the z390 macro processor so no separate translation will be required to use the extensions to write structured macro code. The mz390 macro processor will perform the translation automatically during source input if option ZSTRMAC is on
    - v1.4.03a - ASELECT changed to ACASE and APM changed to ACALL, and lower case support
    - v1.4.03b - SPM's upgraded to structured form using SPE's and SELECT and CASENTRY added

The following source programs are included for general use:

1. **linklib\ZSTRMAC.ZSM - structured translator which can be translated to standard HLASM compatible code using itself on z390 v1.4.03+ or using the bootstrap version which is HLASM compatible rt\test\STRMAC1.MLC. Using either method, the generated HLASM compatible translator is here linklib\ZSTRMAC.TXT.**
2. **ZSTRMAC.BAT - command using %1 to set SYSUT1 input and %2 to set SYSUT2 output and translate structured macro assembler to standard HLASM macro assembler.**
3. **ZSTRSPM.BAT - command to translate all the SPM's in z390\mac directory to HLASM compatible code versions in z390\mac\spm directory using calls to the ZSTRMAC.BAT translator command.**

The following regression test programs are included:

1. **rt\test\ZSTRMAC1.MLC - bootstrap version of translator written in standard HLASM.**
2. **rt\test\ZSTRMAC2.ZSM - structured version of the translator which uses all the structures.**
3. **rt\test\TESTSPE1.ZSM - test program for ZSTRMAC1 with all the basic structures.**
4. **rt\test\TESTSPE2.ZSM - test program for ZSTRMAC2 with all the basic structures in lower case**
5. **rt\test\TESTSPE3.ZSM - test of all 256 ACASE values using all forms of AWHEN operands**
6. **rt\test\TESTSPE4.ZSM - test error messages**

The following demo programs using SPE's are include:

1. **demo\DEMOM8Q1.ZSM - solve 8 queens chess problem suing recursive structured macro**

The following z390 utility programs using SPE's are included:

1. **linklib\RTGENDIR.MLC - read Windows directory listing and create list of file names**
2. **linklib\RTGENCMP.MLC - read merged list of files from 2 directories and generate compare commands**
3. **linklib\RTGENDIF.MLC - read difference files and generate erase commands for identical files**

The following z390 system macros have been structured using the SPE's:

1. **EQUREGS - generate EQU symbols if not already generated for GPR and FPR registers**
2. **ZCLOSE - close files (called from CLOSE and other user macros in concatenated directories)**
3. **ZOPEN - open files (called from OPEN and other user macros in concatenated directories)**
4. **(See all the structured SPM macros below)**

Structured Programming Macros (SPM's) for use in coding structured assembler compatible with HLASM

z390 startmg with PTF v1403b includes the following set of Structured Programming Macros (SPM's) in the z390\mac directory:

- **IF, ELSEIF, ELSE, ENDIF - alternate selection**
- **DO, ENDDO - iteration**
- **SELECT, WHEN, OTHRWISE, ENDSEL - selection by value using compares (v1403b first release)**
- **CASENTRY, CASE, ENDCASE - selection by value using branch table (v1403b first release)**
- **PM, PENTRY, PEXIT - perform block of code and return to next instruction**
- **ASMMSP.CPY - copybook for compatibility with HLASM programs that require this copybook**
- **ZSTRGBL.CPY - copybook included in each SPM with shared global variables**
- **ZSTREQU - macro called from ZSTRGBL to define shared EQU's for condition code operands**

These macros have been updated from the original public domain macros in the SHARE CBT tape #177 dated 1978. All these macros have been updated to use SPE structured programming macro extensions to eliminate all macro labels and use of AGO's. Also included are translated versions of all the structured SPM's in the z390\mac\spm directory. The translated version are for use on systems with HLASM or other mainframe assemblers which do not yet support the SPE's.

The following regression test programs are included:

1. **rt\test\TESTSPM1.MLC - test IF, DO, SELECT, and CASENTRY structures (v1403b first release)**

The following z390 utilities are written in structured form using these SPM's:

1. **linklib\REPRO.MLC - VSAM load and unload utility**
2. **linklib\SUPERZAP.MLC - dump and path file utility**
3. **linklib\FPCONMFC.MLC - interface between z390 BFP/DFP/HFP regression tests and the mainframe compatible external format conversion routine linklib\FPCONVRT.MAC contributed by David Bond.**

The ZSTRMAC1.MLC bootstrap conditional macro code program is the last unstructured macro code program I ever intend to write with 169 explicit macro labels. The structured version has no labels, and I would submit that it is significantly easier to read and maintain using ACALL to reduce the logic down into small logical blocks. Any and all feedback is welcome. Send comments and suggestions to Don Higgins.