#------------------------------------------------------------------------------

# Define uVision editor settings for A166 Assembler files

#

# Copyright : (c) 2012 - 2016 ARM Ltd. and ARM Germany GmbH

# Version : 1.0.0

# Product : uVision

# Author : EG

#

#

# (A) A166 Keyword sections

# 1) A166 Instructions

# 2) A166 Operator

# 3) A166 Dircectives

# 4) A166 Dircectives Operand

# 5) A166 Assembler Statements

# 6) A166 Assembler Macro Processor Statements

# 7) A166 Assembler Macro Processor Operators

# 8) A166 Predefined Macros

# 9) A166 Register

#

# (B) A166 Keyword assignement

#

#------------------------------------------------------------------------------

# (A) A166 Keyword sections

#

# 1) A166 Instructions

a166\_instruction=addb addw addc baddc subb subw subcb subcw mul mulu div divu divl divlu cplb cplw negb negw \

bfldh bfldl bset bclr bmov bmovn and band xor bxor or bor bcmp \

cmpi1 cmpi2 cmpd1 cmpd2 shr shl ror rol ashr movb movw movbz movbs \

push pop scxt andb andw xorb xorw orb orw \

jmpr jmpa jmpi jmps jbjnb jbc jnbs calla calli callr calls pcall trap \

ret rets reti retp srst srvsdt diswdt idle pwrdn einit \

prior nop extp exts extr add cpl neg sub subc cmp mov call jmp cmpb jb jnb

# 2) A166 Operator

# | \

a166\_operator= < << <= == > >= >> ^ \| \

!= & \* + - . / \

and bof byte0 byte1 byte2 byte3 byte4 byte5 byte6 byte7 \

data3 data4 data8 data16 data32 data64 dpp0 dpp1 dpp2 dpp3 eq ge gt high le low lt mod ne not \

or pag pof ptr seg shl short shr sof uge ugt ule ult word0 word2 xor

# 3) A166 Dircectives

a166\_directive=$absolute $case $cond $date $debug $eject $else $elseif $endif $errorprint $expdecnum $extmac $gen $genonly $if $incdir $include \

$list $macro $mod166 $mod167 $modinf $modv2 $mpl $noabsolute $noamake $nocase $nocond $nodebug $noerrorprint $nogen $nolines \

$nolist $nomacro $nomod166 $nompl $nonsegmented $noobject $nopaging $noprint $nosymbols $nosymlist $notype $noxref $object \

$pagelength $pagewidth $paging $print $reguse $reset $restore $save $segmented $set $symbols $symlist \

$tabs $title $type $usedextonly $xref

# 4) A166 Directives Operand

a166\_directive\_operand=$

# 5) A166 Assembler Statements

a166\_statement=assume bit cgroup db dbit dbptr dd defa defb defr df32 df64 dgroup dpptr ds dsb dsd dsptr dsw dw \

else elseif end endif endp ends equ even extern extrn global if label lit name org pecdef proc public \

regbank regdef section set sskdef typedec

# 6) A166 Assembler Macro Processor Statements

a166\_macro=macro exitm endm local rept irp irpc

# 7) A166 Assembler Macro Processor Operators

a166\_macro\_operator=nul & < > % ;; !

# 8) A166 Predefined Macros

a166\_predef\_macro=\_\_date\_\_ \_\_time\_\_

# 9) A166 Register

a166\_register=r0 r1 r2 r3 r4 r5 r6 r7 r8 \

r9 r10 r11 r12 r13 r14 r15 \

rh0 rh1 rh2 rh3 rh4 rh5 rh6 rh7 \

rl0 rl1 rl2 rl3 rl4 rl5 rl6 rl7

# (B) A166 Keyword assignement

#

keywords.$(file.patterns.asm)=$(a166\_instruction) $(a166\_operator)

keywords2.$(file.patterns.asm)=

keywords3.$(file.patterns.asm)=$(a166\_register)

keywords4.$(file.patterns.asm)=$(a166\_directive) $(a166\_directive\_operand) $(a166\_statement)

keywords5.$(file.patterns.asm)=$(a166\_macro) $(a166\_macro\_operator) $(a166\_predef\_macro)

keywords6.$(file.patterns.asm)=

# (C)

#

comment.block.asm=;

command.compile.$(file.patterns.asm)=masm $(FileNameExt)

command.name.0.\*.asm=Link

command.0.\*.asm=link $(FileName)

statement.indent.$(file.patterns.asm)=9 .if