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

# Define uVision editor settings for ARM assembler files

#

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

# Version : 1.0.0

# Product : uVision

# Author : EG

#

#

# 1) Settings for Assembler files (IA32 MASM and NASM)

# 2) Key words

# 3) FPU instructions

# 4) EXT instruction

# 5) MASM directives

# 6) NASM directives

#

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

# Define settings for Assembler files (IA32 MASM and NASM)

# Masm files

#file.patterns.asm=\*.asm;\*.s;\*.A51;\*.A166;\*.A251

#filter.asm=Assembler (asm)|$(file.patterns.asm)|

lexer.$(file.patterns.asm)=asm

#configclass.asm.title=Asm Editor files

# note that key words must be in lower case !

cpu\_instruction=fabsd fabss faddd fadds fcmpd fcmped fcmpes fcmpezd fcmpezs fcmps fcmpzd fcmpzs fcpyd fcpys fcvtds fcvtsd fdivd fdivs fldd fldmd fldmdbd \

fldmdbs fldmdbx fldmead fldmeas fldmeax fldmfdd fldmfds fldmfdx fldmiad fldmias fldmiax fldms fldmx flds fmacd fmacs fmdhr fmdlr fmdrr fmovd \

fmovs fmrdh fmrdl fmrrd fmrrs fmrs fmrx fmscd fmscs fmsr fmsrr fmstat fmuld fmuls fmxr fnegd fnegs fnmacd fnmacs fnmscd fnmscs fnmuld \

fnmuls fsitod fsitos fsqrtd fsqrts fstd fstmd fstmdbd fstmdbs fstmdbx fstmead fstmeas fstmeax fstmfdd fstmfds fstmfdx fstmiad fstmias \

fstmiax fstms fstmx fsts fsubd fsubs ftosid ftosis ftosizd ftosizs ftouid ftouis ftouizd ftouizs fuitod fuitos fconst fconsts fconstd \

fshtos fshtod fsltos fsltod fuhtos fuhtod fultos fultod ftoshs ftoshd ftosls ftosld ftouhs ftouhd ftouls ftould vaba vabaq vqsub vqsubq \

vhsub vhsubq vqadd vqaddq vhadd vhaddq vrhadd vrhaddq vrshl vrshlq vqrshl vqrshlq vabd vabdq vmax vmaxq vmin vminq vpmax vpmin vacge vacgeq \

vaclt vacltq vacgt vacgtq vacle vacleq vrecps vrecpsq vrsqrts vrsqrtsq vtst vtstq vpadd vadd vaddq vsub vsubq veor veorq vbif vbifq vbit \

vbitq vbsl vbslq vdiv vfnma vfnms vnmla vnmls vnmul vceq vceqq vcge vcgeq vclt vcltq vcgt vcgtq vcle vcleq vsli vsliq vsri vsriq vsra \

vsraq vrsra vrsraq vqshlu vqshluq vshl vshlq vqshl vqshlq vshr vshrq vrshr vrshrq vshll vshrn vrshrn vqshrn vqrshrn vqshrun vqrshrun vcvt \

vcvtq vcvtr vcvtb vcvtt vmul vmulq vmla vmlaq vmls vmlsq vfma vfmaq vfms vfmsq vqdmulh vqdmulhq vqrdmulh vqrdmulhqvbicvbicq vand vandq vorr \

vorrq vorn vornq vmov vmovq vmvn vmvnq vabdl vabal vaddl vsubl vaddw vsubw vaddhn vraddhn vsubhn vrsubhn vmlal vmlsl vmull vqdmull vqdmlal \

vqdmlsl vext vextq vpaddl vpaddlq vpadal vpadalq vmovn vqmovn vclz vclzq vcnt vcntq vrev16 vrev16q vrev32 vrev32q vrev64 vrev64q vswp \

vswpq vtrn vtrnq vuzp vuzpq vzip vzipq vqabs vqabsq vqneg vqnegq vcls vclsq vabs vabsq vneg vnegq vrecpe vrecpeq vrsqrte vrsqrteq vqmovun \

vsqrt vmovl vtbl vtbx vld1 vld2 vld3 vld4 vst1 vst2 vst3 vst4 vldr vmov2 vstr vldm vldmia vldmib vldmdb vldmda vldmea vldmfd vstm vstmia \

vstmib vstmdb vstmda vstmea vstmfd vpop vpush vdup vdupq vcmp vcmpe adc add adds addw adr adrl and asr b bal beq bne bcs bhs bcc blo bmi bpl \

bvs bvc bhi bls bge blt bgt ble bic bkpt bl blx bx bxj cbz cbnz cdp cdp2 chka clz clrex cmn cmp cps cpsid cpsie cpy dci dcial dcieq dcine \

dcics dcihs dcicc dcilo dcimi dcipl dcivs dcivc dcihi dcils dcige dcilt dcigt dcile dbg dmb dsb enterx eor hb hbl hblp hbp isb it itt ite \

ittt itet itte itee itttt itett ittet iteet ittte itete ittee iteee ldc ldcl ldc2 ldc2l ldm ldmda ldmdb ldmea ldmed ldmfa ldmfd ldmia ldmib ldr \

ldrb ldrbt ldrex ldrexb ldrexh ldrexd ldrbex ldrhex ldrdex ldrd ldrh ldrht ldrsb ldrsbt ldrsh ldrsht ldrt lds lea leavex lsl lsr mcr mcr2 \

mcrr mcrr2 mla mlas mls mov movs movw movt mov32 mrc mrc2 mrrc mrrc2 mrs vmrs msr vmsr mul muls mvn nop yield wfe wfi sev orr orn pkhbt pkhtb pld \

pldw pli pop push qadd qadd16 qadd8 qaddsubx qasx qdadd qdsub qsax qsub qsub16 qsub8 qsubaddx rev rev16 revsh rfe rfeda rfefa rfedb rfeea rfeia \

rfefd rfeib rfeed ror rrx rsb rsc sadd16 sadd16to32 sadd8 sadd8to16 sadd8to32 saddsubx sasx sbc sel setend sext16 sext8 shadd16 shadd8 shaddsubx shasx \

shsub16 shsub8 shsubaddx shsax smc smi smlabb smlabt smlad smladx smlal smlals smlalbb smlalbt smlald smlaldx smlaltb smlaltt smlatb smlatt \

smlawb smlawt smlsd smlsdx smlsld smlsldx smmla smmlar smmls smmlsr smmul smmulr smuad smuadx smulbb smulbt smull smulls smultb smultt smulwb smulwt \

smusd smusdx srs srsda srsed srsdb srsfd srsia srsea srsib srsfa ssat ssat16 ssax ssub16 ssub8 ssubaddx stc stc2 stcl stc2l stm stmda stmdb \

stmea stmed stmfa stmfd stmia stmib str strb strbt strd strex strexb strexd strexh strbex strdex strhex strh strht strt sub subs subw sunpk16to32 sunpk8to16 \

sunpk8to32 swi svc swp swpb sxtab sxtab16 sxtah sxtb sxtb16 sxth tbb tbh teq tst uadd16 uadd16to32 uadd8 uadd8to16 uadd8to32 uaddsubx uasx \

uext16 uext8 uhadd16 uhadd8 uhaddsubx uhasx uhsax uhsub16 uhsub8 uhsubaddx umaal umlal umlals umull umulls uqadd16 uqadd8 uqaddsubx uqasx uqsax \

uqsub16 uqsub8 uqsubaddx usad8 usada8 usat usat16 usax usub16 usub8 usubaddx uunpk16to32 uunpk8to16 uunpk8to32 uxtab uxtab16 uxtah uxtb uxtb16 uxth \

sbfx ubfx bfi bfc rbit neg udiv sdiv und sys hvc eret moveq movne subeq subne addeq addne lsls adcs asrs bics eors lsls lsrs mvns orns orrs rors rsbs rrxs sbcs

cpu\_instruction2=

# fpu instructions

fpu\_instruction=

# these are MMX, SSE, SSE2 instructions

ext\_instruction=

register=r0 r1 r2 r3 r4 r5 r6 r7 r8 r9 r10 r11 r12 r13 r14 r15 s0 s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 s13 s14 s15 s16 s17 s18 \

s19 s20 s21 s22 s23 s24 s25 s26 s27 s28 s29 s30 s31 lr pc sp

register\_64=

# masm directives

directive=equ export global exportas extern field frame address pop push register restore save state remember unwind on off function proc \

gbla gbll gbls get include if else endif elif incbin include info keep lcla lcll lcls endp import area dcd space thumb preserve8 \

stack noinit readwrite readonly code align heap end

directive\_operand=

# nasm directives, mostly complete, does not parse properly

# the following: %!<env>, %%, %+, %+n %-n, %{n}

directive\_nasm=

directive\_operand\_nasm=

keywords.$(file.patterns.asm)=$(cpu\_instruction) $(cpu\_instruction2)

keywords2.$(file.patterns.asm)=$(fpu\_instruction)

keywords3.$(file.patterns.asm)=$(register) $(register\_64)

keywords4.$(file.patterns.asm)=$(directive) $(directive\_nasm)

keywords5.$(file.patterns.asm)=$(directive\_operand) $(directive\_operand\_nasm)

keywords6.$(file.patterns.asm)=$(ext\_instruction)

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