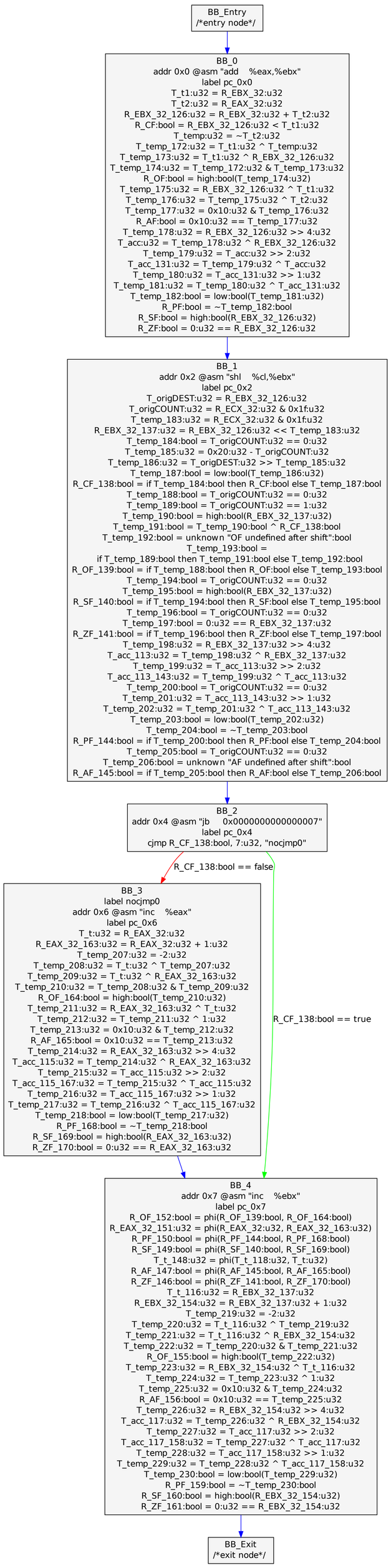
BB_Entry
/*entry node*/
BB_0
addr 0x0 @asm "add %eax,%ebx"
label pc_0x0
T_t1:u32 = R_EBX_32:u32
T_t2:u32 = R_EAX_32:u32
R_EBX_32_126:u32 = R_EBX_32:u32 + T_t2:u32
R_CF:bool = R_EBX_32_126:u32 < T_t1:u32
T_temp:u32 = ~T_t2:u32
T_temp_172:u32 = T_t1:u32 ^ T_temp:u32
T_temp_173:u32 = T_t1:u32 ^ R_EBX_32_126:u32
T_temp_174:u32 = T_temp_172:u32 & T_temp_173:u32
R_OF:bool = high:bool(T_temp_174:u32)
T_temp_175:u32 = R_EBX_32_126:u32 ^ T_t1:u32
T_temp_176:u32 = T_temp_175:u32 ^ T_t2:u32
T_temp_177:u32 = 0x10:u32 & T_temp_176:u32
R_AF:bool = 0x10:u32 == T_temp_177:u32
T_temp_178:u32 = R_EBX_32_126:u32 >> 4:u32
T_acc:u32 = T_temp_178:u32 ^ R_EBX_32_126:u32
T_temp_179:u32 = T_acc:u32 >> 2:u32
T_acc_131:u32 = T_temp_179:u32 ^ T_acc:u32
T_temp_180:u32 = T_acc_131:u32 >> 1:u32
T_temp_181:u32 = T_temp_180:u32 ^ T_acc_131:u32
T_temp_182:bool = low:bool(T_temp_181:u32)
R_PF:bool = ~T_temp_182:bool
R_SF:bool = high:bool(R_EBX_32_126:u32)
R_ZF:bool = 0:u32 == R_EBX_32_126:u32
BB_1
addr 0x2 @asm "shl %cl,%ebx"
label pc_0x2
T_origDEST:u32 = R_EBX_32_126:u32
T_origCOUNT:u32 = R_ECX_32:u32 & 0x1f:u32
T_temp_183:u32 = R_ECX_32:u32 & 0x1f:u32
R_EBX_32_137:u32 = R_EBX_32_126:u32 << T_temp_183:u32
T_temp_184:bool = T_origCOUNT:u32 == 0:u32
T_temp_185:u32 = 0x20:u32 - T_origCOUNT:u32
T_temp_186:u32 = T_origDEST:u32 >> T_temp_185:u32
T_temp_187:bool = low:bool(T_temp_186:u32)
R_CF_138:bool = if T_temp_184:bool then R_CF:bool else T_temp_187:bool
T_temp_188:bool = T_origCOUNT:u32 == 0:u32
T_temp_189:bool = T_origCOUNT:u32 == 1:u32
T_temp_190:bool = high:bool(R_EBX_32_137:u32)
T_temp_191:bool = T_temp_190:bool ^ R_CF_138:bool
T_temp_192:bool = unknown "OF undefined after shift":bool
T_temp_193:bool =
if T_temp_189:bool then T_temp_191:bool else T_temp_192:bool
R_OF_139:bool = if T_temp_188:bool then R_OF:bool else T_temp_193:bool
T_temp_194:bool = T_origCOUNT:u32 == 0:u32
T_temp_195:bool = high:bool(R_EBX_32_137:u32)
R_SF_140:bool = if T_temp_194:bool then R_SF:bool else T_temp_195:bool
T_temp_196:bool = T_origCOUNT:u32 == 0:u32
T_temp_197:bool = 0:u32 == R_EBX_32_137:u32
R_ZF_141:bool = if T_temp_196:bool then R_ZF:bool else T_temp_197:bool
T_temp_198:u32 = R_EBX_32_137:u32 >> 4:u32
T_acc_113:u32 = T_temp_198:u32 ^ R_EBX_32_137:u32
T_temp_199:u32 = T_acc_113:u32 >> 2:u32
T_acc_113_143:u32 = T_temp_199:u32 ^ T_acc_113:u32
T_temp_200:bool = T_origCOUNT:u32 == 0:u32
T_temp_201:u32 = T_acc_113_143:u32 >> 1:u32
T_temp_202:u32 = T_temp_201:u32 ^ T_acc_113_143:u32
T_temp_203:bool = low:bool(T_temp_202:u32)
T_temp_204:bool = ~T_temp_203:bool
R_PF_144:bool = if T_temp_200:bool then R_PF:bool else T_temp_204:bool
T_temp_205:bool = T_origCOUNT:u32 == 0:u32
T_temp_206:bool = unknown "AF undefined after shift":bool
R_AF_145:bool = if T_temp_205:bool then R_AF:bool else T_temp_206:bool
BB_2
addr 0x4 @asm "jb 0x0000000000000007"
label pc_0x4
cjmp R_CF_138:bool, 7:u32, "nocjmp0"
R_CF_138:bool == false
R_CF_138:bool == true
BB_3
label nocjmp0
addr 0x6 @asm "inc %eax"
label pc_0x6
T_t:u32 = R_EAX_32:u32
R_EAX_32_163:u32 = R_EAX_32:u32 + 1:u32
T_temp_207:u32 = -2:u32
T_temp_208:u32 = T_t:u32 ^ T_temp_207:u32
T_temp_209:u32 = T_t:u32 ^ R_EAX_32_163:u32
T_temp_210:u32 = T_temp_208:u32 & T_temp_209:u32
R_OF_164:bool = high:bool(T_temp_210:u32)
T_temp_211:u32 = R_EAX_32_163:u32 ^ T_t:u32
T_temp_212:u32 = T_temp_211:u32 ^ 1:u32
T_temp_213:u32 = 0x10:u32 & T_temp_212:u32
R_AF_165:bool = 0x10:u32 == T_temp_213:u32
T_temp_214:u32 = R_EAX_32_163:u32 >> 4:u32
T_acc_115:u32 = T_temp_214:u32 ^ R_EAX_32_163:u32
T_temp_215:u32 = T_acc_115:u32 >> 2:u32
T_acc_115_167:u32 = T_temp_215:u32 ^ T_acc_115:u32
T_temp_216:u32 = T_acc_115_167:u32 >> 1:u32
T_temp_217:u32 = T_temp_216:u32 ^ T_acc_115_167:u32
T_temp_218:bool = low:bool(T_temp_217:u32)
R_PF_168:bool = ~T_temp_218:bool
R_SF_169:bool = high:bool(R_EAX_32_163:u32)
R_ZF_170:bool = 0:u32 == R_EAX_32_163:u32
BB_4
addr 0x7 @asm "inc %ebx"
label pc_0x7
R_OF_152:bool = phi(R_OF_139:bool, R_OF_164:bool)
R_EAX_32_151:u32 = phi(R_EAX_32:u32, R_EAX_32_163:u32)
R_PF_150:bool = phi(R_PF_144:bool, R_PF_168:bool)
R_SF_149:bool = phi(R_SF_140:bool, R_SF_169:bool)
T_t_148:u32 = phi(T_t_118:u32, T_t:u32)
R_AF_147:bool = phi(R_AF_145:bool, R_AF_165:bool)
R_ZF_146:bool = phi(R_ZF_141:bool, R_ZF_170:bool)
T_t_116:u32 = R_EBX_32_137:u32
R_EBX_32_154:u32 = R_EBX_32_137:u32 + 1:u32
T_temp_219:u32 = -2:u32
T_temp_220:u32 = T_t_116:u32 ^ T_temp_219:u32
T_temp_221:u32 = T_t_116:u32 ^ R_EBX_32_154:u32
T_temp_222:u32 = T_temp_220:u32 & T_temp_221:u32
R_OF_155:bool = high:bool(T_temp_222:u32)
T_temp_223:u32 = R_EBX_32_154:u32 ^ T_t_116:u32
T_temp_224:u32 = T_temp_223:u32 ^ 1:u32
T_temp_225:u32 = 0x10:u32 & T_temp_224:u32
R_AF_156:bool = 0x10:u32 == T_temp_225:u32
T_temp_226:u32 = R_EBX_32_154:u32 >> 4:u32
T_acc_117:u32 = T_temp_226:u32 ^ R_EBX_32_154:u32
T_temp_227:u32 = T_acc_117:u32 >> 2:u32
T_acc_117_158:u32 = T_temp_227:u32 ^ T_acc_117:u32
T_temp_228:u32 = T_acc_117_158:u32 >> 1:u32
T_temp_229:u32 = T_temp_228:u32 ^ T_acc_117_158:u32
T_temp_230:bool = low:bool(T_temp_229:u32)
R_PF_159:bool = ~T_temp_230:bool
R_SF_160:bool = high:bool(R_EBX_32_154:u32)
R_ZF_161:bool = 0:u32 == R_EBX_32_154:u32
BB_Exit
/*exit node*/