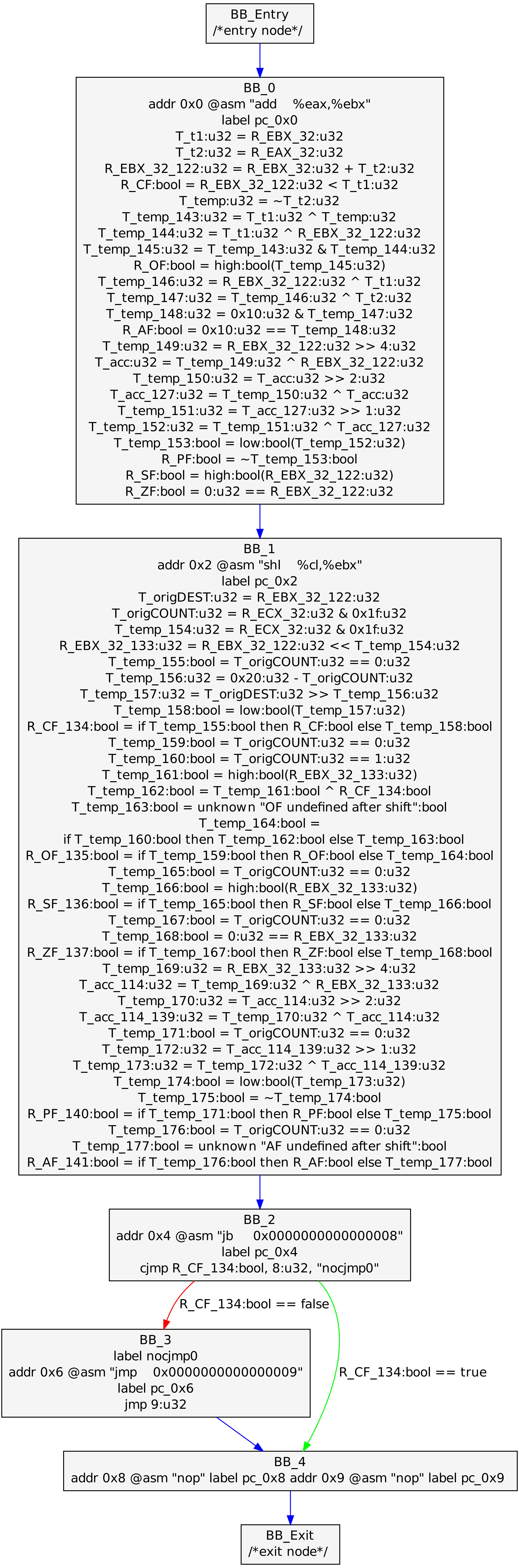

```
BB_Entry
/*entry node*/
```

```
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_122:u32 = R_EBX_32:u32 + T_t2:u32
R_CF:bool = R_EBX_32_122:u32 < T_t1:u32
T_temp:u32 = ~T_t2:u32
T_temp_143:u32 = T_t1:u32 ^ T_temp:u32
T_temp_144:u32 = T_t1:u32 ^ R_EBX_32_122:u32
T_temp_145:u32 = T_temp_143:u32 & T_temp_144:u32
R_OF:bool = high:bool(T_temp_145:u32)
T_temp_146:u32 = R_EBX_32_122:u32 ^ T_t1:u32
T_temp_147:u32 = T_temp_146:u32 ^ T_t2:u32
T_temp_148:u32 = 0x10:u32 & T_temp_147:u32
R_AF:bool = 0x10:u32 == T_temp_148:u32
T_temp_149:u32 = R_EBX_32_122:u32 >> 4:u32
T_acc:u32 = T_temp_149:u32 ^ R_EBX_32_122:u32
T_temp_150:u32 = T_acc:u32 >> 2:u32
T_acc_127:u32 = T_temp_150:u32 ^ T_acc:u32
T_temp_151:u32 = T_acc_127:u32 >> 1:u32
T_temp_152:u32 = T_temp_151:u32 ^ T_acc_127:u32
T_temp_153:bool = low:bool(T_temp_152:u32)
R_PF:bool = ~T_temp_153:bool
R_SF:bool = high:bool(R_EBX_32_122:u32)
R_ZF:bool = 0:u32 == R_EBX_32_122:u32
```

```
BB_1
addr 0x2 @asm "shl    %cl,%ebx"
label pc_0x2
T_origDEST:u32 = R_EBX_32_122:u32
T_origCOUNT:u32 = R_ECX_32:u32 & 0x1f:u32
T_temp_154:u32 = R_ECX_32:u32 & 0x1f:u32
R_EBX_32_133:u32 = R_EBX_32_122:u32 << T_temp_154:u32
T_temp_155:bool = T_origCOUNT:u32 == 0:u32
T_temp_156:u32 = 0x20:u32 - T_origCOUNT:u32
T_temp_157:u32 = T_origDEST:u32 >> T_temp_156:u32
T_temp_158:bool = low:bool(T_temp_157:u32)
R_CF_134:bool = if T_temp_155:bool then R_CF:bool else T_temp_158:bool
T_temp_159:bool = T_origCOUNT:u32 == 0:u32
T_temp_160:bool = T_origCOUNT:u32 == 1:u32
T_temp_161:bool = high:bool(R_EBX_32_133:u32)
T_temp_162:bool = T_temp_161:bool ^ R_CF_134:bool
T_temp_163:bool = unknown "OF undefined after shift":bool
T_temp_164:bool =
if T_temp_160:bool then T_temp_162:bool else T_temp_163:bool
R_OF_135:bool = if T_temp_159:bool then R_OF:bool else T_temp_164:bool
T_temp_165:bool = T_origCOUNT:u32 == 0:u32
T_temp_166:bool = high:bool(R_EBX_32_133:u32)
R_SF_136:bool = if T_temp_165:bool then R_SF:bool else T_temp_166:bool
T_temp_167:bool = T_origCOUNT:u32 == 0:u32
T_temp_168:bool = 0:u32 == R_EBX_32_133:u32
R_ZF_137:bool = if T_temp_167:bool then R_ZF:bool else T_temp_168:bool
T_temp_169:u32 = R_EBX_32_133:u32 >> 4:u32
T_acc_114:u32 = T_temp_169:u32 ^ R_EBX_32_133:u32
T_temp_170:u32 = T_acc_114:u32 >> 2:u32
T_acc_114_139:u32 = T_temp_170:u32 ^ T_acc_114:u32
T_temp_171:bool = T_origCOUNT:u32 == 0:u32
T_temp_172:u32 = T_acc_114_139:u32 >> 1:u32
T_temp_173:u32 = T_temp_172:u32 ^ T_acc_114_139:u32
T_temp_174:bool = low:bool(T_temp_173:u32)
T_temp_175:bool = ~T_temp_174:bool
R_PF_140:bool = if T_temp_171:bool then R_PF:bool else T_temp_175:bool
T_temp_176:bool = T_origCOUNT:u32 == 0:u32
T_temp_177:bool = unknown "AF undefined after shift":bool
R_AF_141:bool = if T_temp_176:bool then R_AF:bool else T_temp_177:bool
```

```
BB_2
addr 0x4 @asm "jb     0x0000000000000008"
label pc_0x4
cjmp R_CF_134:bool, 8:u32, "nocjmp0"
```

```
BB_3
label nocjmp0
addr 0x6 @asm "jmp    0x0000000000000009"
label pc_0x6
jmp 9:u32
```

```
BB_4
addr 0x8 @asm "nop" label pc_0x8 addr 0x9 @asm "nop" label pc_0x9
```

```
BB_Exit
/*exit node*/
```