```
T t:u32 = T t 118:u32 \otimes str "MoveFromPhi"
                                                                   R EAX 32:u32 = R EAX 32 11:u32 @str "MoveFromPhi"
                                                                                                         BB 0
                                                                                   addr 0x0 @asm "add %eax,%ebx"
                                                                                                    label pc_0x0
                                                                                         T_t1:u32 = R_EBX_32:u32
                                                                                      T t2:u32 = R EAX 32 11:u32
                                                                         R_{EBX_32_126:u32} = R_{EBX_32:u32} + T_{t2:u32}
                                                                             R_{CF:bool} = R_{EBX_32_126:u32} < T_{t1:u32}
                                                R OF:bool = high:bool((T_t1:u32 ^ -T_t2:u32) & (T_t1:u32 ^ R_EBX_32_126:u32))
                                              R_AF:bool = 0x10:u32 == (0x10:u32 & (R_EBX_32_126:u32 ^ T_t1:u32 ^ T_t2:u32))
                                                             T_acc:u32 = R_EBX_32_126:u32 >> 4:u32 ^ R_EBX_32_126:u32
                                                                        T_{acc}_{131:u32} = T_{acc:u32} >> 2:u32 ^ T_{acc:u32}
                                                            R_{PF:bool} = \sim low:bool(T_{acc_131:u32} >> 1:u32 ^ T_{acc_131:u32})
                                                                              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_{orig}DEST:u32 = R_{EBX_32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32_126:u32
                                                                          T_{orig}COUNT:u32 = R_{ECX_32:u32 \& 0x1f:u32}
                                                      R_{EBX_32_137:u32} = R_{EBX_32_126:u32} << (R_{ECX_32:u32 \& 0x1f:u32})
                                                                                                 R_CF_138:bool =
                                                                         if T_origCOUNT:u32 == 0:u32 then R_CF:bool else
                                                                  low:bool(T_origDEST:u32 >> 0x20:u32 - T_origCOUNT:u32)
                                                                                                 R_OF_139:bool =
                                                                         if T_origCOUNT:u32 == 0:u32 then R_OF:bool else
                                                                                    if T_{orig}COUNT:u32 == 1:u32 then
                                                                        high:bool(R_EBX_32_137:u32) ^ R_CF_138:bool else
                                                                                 unknown "OF undefined after shift":bool
                                                                                                 R_SF_140:bool =
                                                   if T_{orig}COUNT:u32 == 0:u32 then R_{SF:bool} else high:bool(R_{EBX_32_137:u32})
                                                                                                 R_ZF_141:bool =
                                                   if T_{orig}COUNT:u32 == 0:u32 then R_{Z}F:bool else 0:u32 == R_{E}BX_{32}_{137}:u32
                                                         T acc 113:u32 = R EBX 32 137:u32 >> 4:u32 ^ R EBX 32 137:u32
                                                             T_{acc}_{113}_{143:u32} = T_{acc}_{113:u32} >> 2:u32 ^ T_{acc}_{113:u32}
                                                                                                 R_PF_144:bool =
                                                                         if T origCOUNT:u32 == 0:u32 then R_PF:bool else
                                                               \simlow:bool(T_acc_113_143:u32 >> 1:u32 ^ T_acc_113_143:u32)
                                                                                                R_AF_145:bool =
                                                                         if T origCOUNT:u32 == 0:u32 then R AF:bool else
                                                                                 unknown "AF undefined after shift":bool
                                                                      R ZF 146:bool = R ZF 141:bool @str "MoveFromPhi"
                                                                      R_AF_147:bool = R_AF_145:bool @str "MoveFromPhi"
                                                                      R_SF_149:bool = R_SF_140:bool @str "MoveFromPhi"
                                                                      R_PF_150:bool = R_PF_144:bool @str "MoveFromPhi"
                                                                      R_OF_152:bool = R_OF_139:bool @str "MoveFromPhi"
                                                                                                         BB 2
                                                                                                         0x00000000000000007"
                                                                           addr 0x4 @asm "jb
                                                                                                    label pc 0x4
                                                                                 cjmp R CF 138:bool, 7:u32, "nocjmp0"
                                                                                          \hat{R} CF 138:bool == false
                                                        BB 3
                                                  label nocjmp0
                                      addr 0x6 @asm "inc %eax"
                                                   label pc 0x6
                                  T_t_{162:u32} = R_EAX_{32}_{11:u32}
                       R_EAX_32_163:u32 = R_EAX_32_11:u32 + 1:u32
                                               R OF 164:bool =
      high:bool((T_t_{162:u32} - -2:u32) & (T_t_{162:u32} - R_EAX_{32}_{163:u32}))
                                               R_AF_165:bool =
      0x10:u32 == (0x10:u32 & (R_EAX_32_163:u32 ^ T_t_162:u32 ^ 1:u32))
        T_acc_115:u32 = R_EAX_32_163:u32 >> 4:u32 ^ R_EAX_32_163:u32
           T_acc_115_167:u32 = T_acc_115:u32 >> 2:u32 ^ T_acc_115:u32
                                                                                                                                      R CF 138:bool == true
R PF 168:bool = \sim low:bool(T acc 115 167:u32 >> 1:u32 ^ T acc 115 167:u32)
                         R SF 169:bool = high:bool(R EAX 32 163:u32)
                         R \ ZF \ 170:bool = 0:u32 == R \ EAX \ 32 \ 163:u32
                     R ZF 146:bool = R ZF 170:bool @str "MoveFromPhi"
                     R AF 147:bool = R AF 165:bool @str "MoveFromPhi"
                           T t:u32 = T t 162:u32 \otimes str "MoveFromPhi"
                     R SF 149:bool = R SF 169:bool @str "MoveFromPhi"
                     R PF 150:bool = R PF 168:bool @str "MoveFromPhi"
                 R EAX 32:u32 = R EAX 32 163:u32 @str "MoveFromPhi"
                    R OF 152:bool = R OF 164:bool @str "MoveFromPhi"
                                                                                                          BB 4
                                                                                        addr 0x7 @asm "inc %ebx"
                                                                                                    label pc 0x7
                                                                                   T t 116:u32 = R EBX 32 137:u32
                                                                        R_{EBX_32_154:u32} = R_{EBX_32_137:u32} + 1:u32
                                                                                                 R OF 155:bool =
                                                         high:bool((T_t_116:u32 \ ^-2:u32) \ \& \ (T_t_116:u32 \ ^-R_EBX_32_154:u32))
                                                                                                 R AF 156:bool =
                                                         0x10:u32 == (0x10:u32 & (R_EBX_32_154:u32 ^ T_t_116:u32 ^ 1:u32))
                                                         T acc 117:u32 = R EBX 32 154:u32 >> 4:u32 ^ R EBX 32 154:u32
                                                             T acc 117 158:u32 = T acc 117:u32 >> 2:u32 ^T acc 117:u32
                                                  R PF 159:bool = \simlow:bool(T acc 117 158:u32 >> 1:u32 ^ T acc 117 158:u32)
                                                                           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*/

BB_Entry /*entry node*/