```
gen_helper_cpsr_read(tmp);

DEF_HELPER_0(cpsr_read, i32)

#define DEF_HELPER_0(name, ret) \
    DEF_HELPER_FLAGS_0(name, 0, ret)

#define DEF_HELPER_FLAGS_0(name, flags, ret) \
static inline void glue(gen_helper_, name)(dh_retvar_decl0(ret)) \
{ \
  int sizemask; \
  sizemask = dh_is_64bit(ret); \
  tcg_gen_helperN(HELPER(name), flags, sizemask, dh_retvar(ret), 0, NULL); \
}
```

```
static inline void tcg_gen_helperN(void *func, int flags, int sizemask,
                                   TCGArg ret, int nargs, TCGArg *args)
{
    TCGv_ptr fn;
    fn = tcg_const_ptr((tcg_target_long)func); //указатель на функцию
    tcg_gen_callN(&tcg_ctx, fn, flags, sizemask, ret,
                  nargs, args);
    tcg_temp_free_ptr(fn);
}
```

```
void tcg_gen_callN(TCGContext *s, TCGv_ptr func, unsigned int flags,
                   int sizemask, TCGArg ret, int nargs, TCGArg *args)
{
    *gen_opc_ptr++ = INDEX_op_call; //имя функции
    nparam = gen_opparam_ptr++;
    *gen_opparam_ptr++ = ret; //возвращаемое значение
    *gen_opparam_ptr++ = GET_TCGV_PTR(func); //указатель на функцию
    *gen_opparam_ptr++ = flags; //флаги
    *nparam = (nb_rets << 16) | (real_args + 1);
    /* total parameters, needed to go backward in the instruction stream */
    *gen_opparam_ptr++ = 1 + nb_rets + real_args + 3;

}
```

Tcg.c:

```
int tcg_gen_code_common(TCGContext *s, uint8_t *gen_code_buf, long search_pc) {
        tcg_dump_ops(s, logfile); /*dump OP = intermediate view */
        tcg_reg_alloc_start(s); //установка глобальных регистров
        s->code_ptr = gen_code_buf; //указатель кода
        for(;;) {
                opc = gen_opc_buf[op_index]; //операция
                def = &tcg_op_defs[opc]; //определение операции
                switch(opc) {
                        case INDEX_op_call:
                                dead_args = s->op_dead_args[op_index];
                                args += tcg_reg_alloc_call(s, def, opc, args, dead_args);
                                goto next;
                …..
                }
                next: op_index++;
        }
}
```