in arch/arm/include/asm/global_data.h

```
#ifdef CONFIG_ARM64
#define DECLARE_GLOBAL_DATA_PTR         register volatile gd_t *gd asm ("x18")
#else
#define DECLARE_GLOBAL_DATA_PTR         register volatile gd_t *gd asm ("r9")
#endif
```

在ARM(32)中，u-boot运行期间的的r9 register不做他用，总是指向struct global_data。

如果在u-boot任何.c文件中要访问该structure，只要添加如下statment即可。

```
DECLARE_GLOBAL_DATA_PTR;
```

in include/asm-generic/global_data.h

```
typedef struct global_data {
        bd_t *bd;
        unsigned long flags;
        unsigned int baudrate;
        unsigned long cpu_clk;  /* CPU clock in Hz!             */
        unsigned long bus_clk;
        /* We cannot bracket this with CONFIG_PCI due to mpc5xxx */
        unsigned long pci_clk;
        unsigned long mem_clk;
#if defined(CONFIG_LCD) || defined(CONFIG_VIDEO) || 1
        unsigned long fb_base;  /* Base address of framebuffer mem */
#endif
#if defined(CONFIG_POST) || defined(CONFIG_LOGBUFFER)
        unsigned long post_log_word;  /* Record POST activities */
        unsigned long post_log_res; /* success of POST test */
        unsigned long post_init_f_time;  /* When post_init_f started */
#endif
#ifdef CONFIG_BOARD_TYPES
        unsigned long board_type;
#endif
        unsigned long have_console;     /* serial_init() was called */
#ifdef CONFIG_PRE_CONSOLE_BUFFER
        unsigned long precon_buf_idx;   /* Pre-Console buffer index */
#endif
#ifdef CONFIG_MODEM_SUPPORT
        unsigned long do_mdm_init;
        unsigned long be_quiet;
#endif
        unsigned long env_addr; /* Address  of Environment struct */
        unsigned long env_valid;        /* Checksum of Environment valid? */

        unsigned long ram_top;  /* Top address of RAM used by U-Boot */

        unsigned long relocaddr;        /* Start address of U-Boot in RAM */
        phys_size_t ram_size;   /* RAM size */
        unsigned long mon_len;  /* monitor len */
        unsigned long irq_sp;           /* irq stack pointer */
        unsigned long start_addr_sp;    /* start_addr_stackpointer */
        unsigned long reloc_off;
        struct global_data *new_gd;     /* relocated global data */
        const void *fdt_blob;   /* Our device tree, NULL if none */
        void *new_fdt;          /* Relocated FDT */
        unsigned long fdt_size; /* Space reserved for relocated FDT */
        void **jt;              /* jump table */
        char env_buf[32];       /* buffer for getenv() before reloc. */
#ifdef CONFIG_TRACE
        void            *trace_buff;    /* The trace buffer */
#endif
#if defined(CONFIG_SYS_I2C)
        int             cur_i2c_bus;    /* current used i2c bus */
#endif
        unsigned long timebase_h;
        unsigned long timebase_l;
```

```
54.         struct arch_global_data arch;   /* architecture-specific data */
55.  } gd_t;
```

该global structure的初始分配(relocate code以前)是在arch/arm/lib/crt0.S中

```
1.  #if defined(CONFIG_SPL_BUILD) && defined(CONFIG_SPL_STACK)
2.          ldr     sp, =(CONFIG_SPL_STACK)
3.  #else
4.          ldr     sp, =(CONFIG_SYS_INIT_SP_ADDR)
                                                    ①
5.  #endif
6.          bic     sp, sp, #7      /* 8-byte alignment for ABI compliance */
7.          sub     sp, sp, #GD_SIZE        /* allocate one GD above SP */
                    ②
8.          bic     sp, sp, #7      /* 8-byte alignment for ABI compliance */
9.          mov     r9, sp          /* GD is above SP */

10.         mov     r0, #0
11.         bl      board_init_f
```

①

这里sp指向0x08000000 - sizof(gd_t) (取整)

```
1.  #ifndef CONFIG_SYS_TEXT_BASE
2.  #define CONFIG_SYS_TEXT_BASE 0x08000000 /* (CONFIG_SYS_BASE_ADDR + CONFIG_SYS_INI
    T_SP_SIZE + CONFIG_SYS_MALLOC_LEN) */
3.  #endif
4.
5.  #define CONFIG_SYS_BASE_ADDR (CONFIG_SYS_TEXT_BASE - 0x800)
6.  #define CONFIG_SYS_INIT_SP_ADDR (CONFIG_SYS_BASE_ADDR + CONFIG_SYS_INIT_SP_SIZE -
     GENERATED_GBL_DATA_SIZE)
```

②

r9 = 0x08000000 - sizof(gd_t) - sizof(gd_t)

我的感觉CONFIG_SYS_INIT_SP_ADDR 无需再减去GENERATED_GBL_DATA_SIZE。这里多减去GENERATED_GBL_DATA_SIZE也不会错，但好像没有必要。

对gd_t structure中很多成员的初始化，在board_init_f() function中 (in arch/arm/lib/board.c)