struct eth_q_rx_desc {
    uint32 status; /* Desc status word */
    uint16 buf1size; /* Size of buffer 1 */
    uint16 buf2size; /* Size of buffer 2 */
    uint32 buffer1; /* Address of buffer 1 */
    uint32 buffer2; /* Address of buffer 2 */
};
#define rdctl1 buf1size /* Buffer 1 size field has control bits too */
#define rdctl2 buf2size /* Buffer 2 size field has control bits too */

#define ETH_QUARK_RDST_OWN 0x80000000 /* Descrp. owned by DMA */
#define ETH_QUARK_RDST_ES 0x00008000 /* Error Summary */
#define ETH_QUARK_RDST_FS 0x00000200 /* First Segment */
#define ETH_QUARK_RDST_LS 0x00000100 /* Last segment */
#define ETH_QUARK_RDST_FTETH 0x00000020 /* Frame Type = Ethernet */
#define ETH_QUARK_RDCTL1_DIC 0x8000 /* Dis. Int on Complet. */
#define ETH_QUARK_RDCTL1_RER 0x8000 /* Recv End of Ring */

#define ETH_QUARK_RX_RING_SIZE 32
#define ETH_QUARK_TX_RING_SIZE 16

#define ETH_QUARK_INIT_DELAY 500000 /* Delay in micro secs */
#define ETH_QUARK_MAX_RETRIES 3 /* Max retries for init */

16.6 Rings And Buffers In Memory

From a device’s perspective, an input or output ring consists of a linked list of descriptors in memory. We said that each descriptor on a ring contains a status word that specifies whether the associated buffer is empty or full. The descriptor also contains a pointer to a buffer in memory and a pointer to the next descriptor on the list. Figure 16.1 illustrates the conceptual organization of transmit and receive rings, and shows that each descriptor contains a pointer to a buffer as well as a pointer to the next ring.