

new/usr/src/uts/common/sys/fct.h

12606 Thu Apr 28 06:28:10 2011

new/usr/src/uts/common/sys/fct.h

*** NO COMMENTS ***

_____unchanged_portion_omitted_____

```
316 /*
317  * port topology
318  */
319 #define PORT_TOPOLOGY_UNKNOWN 0
320 #define PORT_TOPOLOGY_PT_TO_PT 1
321 #define PORT_TOPOLOGY_PRIVATE_LOOP 2
322 #define PORT_TOPOLOGY_PUBLIC_LOOP 2
323 #define PORT_TOPOLOGY_FABRIC_PT_TO_PT 5
324 #define PORT_TOPOLOGY_FABRIC_BIT 4

326 #define PORT_FLOGI_DONE(li) (((li)->port_fca_flogi_done) || \
327                               ((li)->port_fct_flogi_done))

329 /*
330  * port speed
331  */
332 #define PORT_SPEED_UNKNOWN 0
333 #define PORT_SPEED_1G 1
334 #define PORT_SPEED_2G 2
335 #define PORT_SPEED_4G 4
336 #define PORT_SPEED_8G 8
337 #define PORT_SPEED_10G 16

339 /*
340  * Abort commands
341  */
342 #define FCT_TERMINATE_CMD 1

344 /*
345  * FCT port states.
346  */
347 #define FCT_STATE_OFFLINE 0
348 #define FCT_STATE_ONLINING 1
349 #define FCT_STATE_ONLINE 2
350 #define FCT_STATE_OFFLINING 3

352 /*
353  * fct ctl commands. These should not conflict with stmf ctl commands
354  */
355 #define FCT_CMD_PORT_ONLINE (STMF_LPORT_CTL_CMDS | 0x01)
356 #define FCT_CMD_PORT_ONLINE_COMPLETE (STMF_LPORT_CTL_CMDS | 0x02)
357 #define FCT_CMD_PORT_OFFLINE (STMF_LPORT_CTL_CMDS | 0x03)
358 #define FCT_CMD_PORT_OFFLINE_COMPLETE (STMF_LPORT_CTL_CMDS | 0x04)
359 #define FCT_ACK_PORT_ONLINE_COMPLETE (STMF_LPORT_CTL_CMDS | 0x05)
360 #define FCT_ACK_PORT_OFFLINE_COMPLETE (STMF_LPORT_CTL_CMDS | 0x06)
361 #define FCT_CMD_FORCE_LIP (STMF_LPORT_CTL_CMDS | 0x07)

363 /*
364  * IO flags for cmd flow.
365  */
366 #define FCT_IOF_FCA_DONE 0x10000
367 #define FCT_IOF_FORCE_FCA_DONE 0x20000

369 /*
370  * Fill CTIU preamble
371  */
372 #ifdef lint
373 #define FCT_FILL_CTIU_PREAMPLE(x_payload, x_ctop) _NOTE(EMPTY)
374 #else
```

1

new/usr/src/uts/common/sys/fct.h

```
375 #define FCT_FILL_CTIU_PREAMPLE(x_payload, x_ctop) \
376     do { \
377         x_payload[0] = 0x01; \
377         x_payload[0] = 0x02; \
378         x_payload[4] = 0xFC; \
379         x_payload[5] = 0x02; \
380         x_payload[8] = 0xFF & (x_ctop >> 8); \
381         x_payload[9] = 0xFF & (x_ctop); \
382     } while (0)
383 #endif

385 uint64_t fct_netbuf_to_value(uint8_t *buf, uint8_t nbytes);
386 void fct_value_to_netbuf(uint64_t value, uint8_t *buf, uint8_t nbytes);
387 void *fct_alloc(fct_struct_id_t struct_id, int additional_size, int flags);
388 void fct_free(void *ptr);
389 fct_cmd_t *fct_scsi_task_alloc(struct fct_local_port *port,
390                               uint16_t rp_handle, uint32_t rportid, uint8_t *lun,
391                               uint16_t cdb_length, uint16_t task_ext);
392 fct_status_t fct_register_local_port(fct_local_port_t *port);
393 fct_status_t fct_deregister_local_port(fct_local_port_t *port);
394 void fct_handle_event(fct_local_port_t *port, int event_id,
395                      uint32_t event_flags, caddr_t arg);
396 void fct_post_rcvd_cmd(fct_cmd_t *cmd, stmf_data_buf_t *dbuf);
397 void fct_queue_cmd_for_termination(fct_cmd_t *cmd, fct_status_t s);
398 void fct_queue_scsi_task_for_termination(fct_cmd_t *cmd, fct_status_t s);
399 fct_cmd_t *fct_handle_to_cmd(fct_local_port_t *port, uint32_t fct_handle);
400 void fct_ctl(struct stmf_local_port *lport, int cmd, void *arg);
401 void fct_cmd_fca_aborted(fct_cmd_t *cmd, fct_status_t s, uint32_t ioflags);
402 uint16_t fct_get_rp_handle(fct_local_port_t *port, uint32_t rportid);
403 void fct_send_response_done(fct_cmd_t *cmd, fct_status_t s, uint32_t ioflags);
404 void fct_send_cmd_done(fct_cmd_t *cmd, fct_status_t s, uint32_t ioflags);
405 void fct_scsi_data_xfer_done(fct_cmd_t *cmd, stmf_data_buf_t *dbuf,
406                             uint32_t ioflags);
407 fct_status_t fct_port_initialize(fct_local_port_t *port, uint32_t rflags,
408                                 char *additional_info);
409 fct_status_t fct_port_shutdown(fct_local_port_t *port, uint32_t rflags,
410                               char *additional_info);
411 fct_status_t fct_handle_rcvd_flogi(fct_local_port_t *port,
412                                   fct_flogi_xchg_t *fx);
413 void fct_log_local_port_event(fct_local_port_t *port, char *subclass);
414 void fct_log_remote_port_event(fct_local_port_t *port, char *subclass,
415                                uint8_t *rp_pwn, uint32_t rp_id);
416 void fct_wmn_to_str(char *to_ptr, const uint8_t *from_ptr);

418 #ifdef __cplusplus
419 }
```

_____unchanged_portion_omitted_____

2