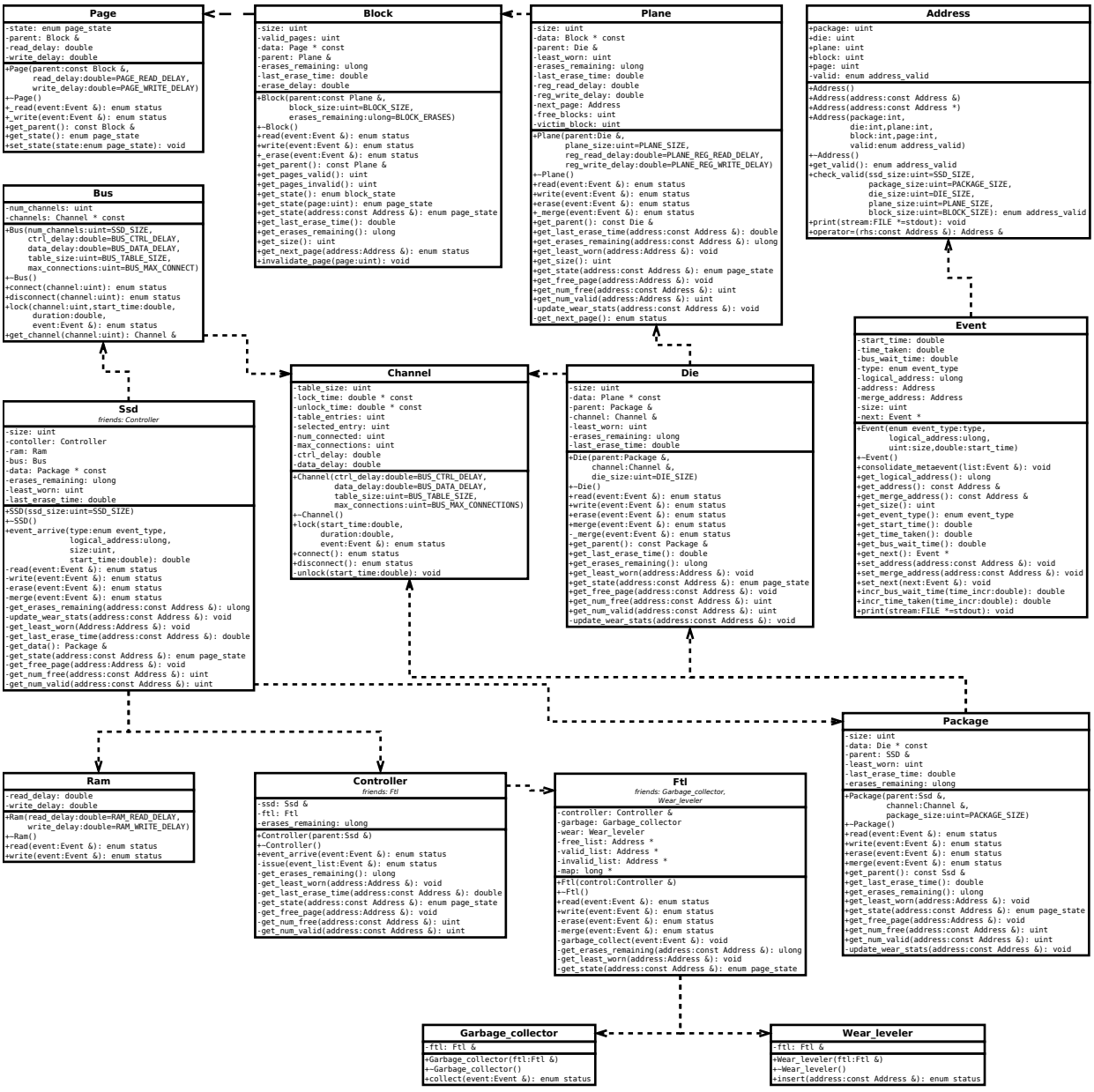
Page
-state: enum page_state
-parent: Block &
-read_delay: double
-write_delay: double
+Page(parent:const Block &,
read_delay:double=PAGE_READ_DELAY,
write_delay:double=PAGE_WRITE_DELAY)
+~Page()
+_read(event:Event &): enum status
+_write(event:Event &): enum status
+get_parent(): const Block &
+get_state(): enum page_state
+set_state(state:enum page_state): void
Block
-size: uint
-valid_pages: uint
-data: Page * const
-parent: Plane &
-erases_remaining: ulong
-last_erase_time: double
-erase_delay: double
+Block(parent:const Plane &,
block_size:uint=BLOCK_SIZE,
erases_remaining:ulong=BLOCK_ERASES)
+~Block()
+read(event:Event &): enum status
+write(event:Event &): enum status
+_erase(event:Event &): enum status
+get_parent(): const Plane &
+get_pages_valid(): uint
+get_pages_invalid(): uint
+get_state(): enum block_state
+get_state(page:uint): enum page_state
+get_state(address:const Address &): enum page_state
+get_last_erase_time(): double
+get_erases_remaining(): ulong
+get_size(): uint
+get_next_page(address:Address &): enum status
+invalidate_page(page:uint): void
Plane
-size: uint
-data: Block * const
-parent: Die &
-least_worn: uint
-erases_remaining: ulong
-last_erase_time: double
-reg_read_delay: double
-reg_write_delay: double
-next_page: Address
-free_blocks: uint
-victim_block: uint
+Plane(parent:Die &,
plane_size:uint=PLANE_SIZE,
reg_read_delay:double=PLANE_REG_READ_DELAY,
reg_write_delay:double=PLANE_REG_WRITE_DELAY)
+~Plane()
+read(event:Event &): enum status
+write(event:Event &): enum status
+erase(event:Event &): enum status
+_merge(event:Event &): enum status
+get_parent(): const Die &
+get_last_erase_time(address:const Address &): double
+get_erases_remaining(address:const Address &): ulong
+get_least_worn(address:Address &): void
+get_size(): uint
+get_state(address:const Address &): enum page_state
+get_free_page(address:Address &): void
+get_num_free(address:const Address &): uint
+get_num_valid(address:Address &): uint
-update_wear_stats(address:const Address &): void
-get_next_page(): enum status
Address
+package: uint
+die: uint
+plane: uint
+block: uint
+page: uint
-valid: enum address_valid
+Address()
+Address(address:const Address &)
+Address(address:const Address *)
+Address(package:int,
die:int,plane:int,
block:int,page:int,
valid:enum address_valid)
+~Address()
+get_valid(): enum address_valid
+check_valid(ssd_size:uint=SSD_SIZE,
package_size:uint=PACKAGE_SIZE,
die_size:uint=DIE_SIZE,
plane_size:uint=PLANE_SIZE,
block_size:uint=BLOCK_SIZE): enum address_valid
+print(stream:FILE *=stdout): void
+operator=(rhs:const Address &): Address &
Bus
-num_channels: uint
-channels: Channel * const
+Bus(num_channels:uint=SSD_SIZE,
ctrl_delay:double=BUS_CTRL_DELAY,
data_delay:double=BUS_DATA_DELAY,
table_size:uint=BUS_TABLE_SIZE,
max_connections:uint=BUS_MAX_CONNECT)
+~Bus()
+connect(channel:uint): enum status
+disconnect(channel:uint): enum status
+lock(channel:uint,start_time:double,
duration:double,
event:Event &): enum status
+get_channel(channel:uint): Channel &
Event
-start_time: double
-time_taken: double
-bus_wait_time: double
-type: enum event_type
-logical_address: ulong
-address: Address
-merge_address: Address
-size: uint
-next: Event *
+Event(enum event_type:type,
logical_address:ulong,
uint:size,double:start_time)
+~Event()
+consolidate_metaevent(list:Event &): void
+get_logical_address(): ulong
+get_address(): const Address &
+get_merge_address(): const Address &
+get_size(): uint
+get_event_type(): enum event_type
+get_start_time(): double
+get_time_taken(): double
+get_bus_wait_time(): double
+get_next(): Event *
+set_address(address:const Address &): void
+set_merge_address(address:const Address &): void
+set_next(next:Event &): void
+incr_bus_wait_time(time_incr:double): double
+incr_time_taken(time_incr:double): double
+print(stream:FILE *=stdout): void
Channel
-table_size: uint
-lock_time: double * const
-unlock_time: double * const
-table_entries: uint
-selected_entry: uint
-num_connected: uint
-max_connections: uint
-ctrl_delay: double
-data_delay: double
+Channel(ctrl_delay:double=BUS_CTRL_DELAY,
data_delay:double=BUS_DATA_DELAY,
table_size:uint=BUS_TABLE_SIZE,
max_connections:uint=BUS_MAX_CONNECTIONS)
+~Channel()
+lock(start_time:double,
duration:double,
event:Event &): enum status
+connect(): enum status
+disconnect(): enum status
-unlock(start_time:double): void
Die
-size: uint
-data: Plane * const
-parent: Package &
-channel: Channel &
-least_worn: uint
-erases_remaining: ulong
-last_erase_time: double
+Die(parent:Package &,
channel:Channel &,
die_size:uint=DIE_SIZE)
+~Die()
+read(event:Event &): enum status
+write(event:Event &): enum status
+erase(event:Event &): enum status
+merge(event:Event &): enum status
-_merge(event:Event &): enum status
+get_parent(): const Package &
+get_last_erase_time(): double
+get_erases_remaining(): ulong
+get_least_worn(address:Address &): void
+get_state(address:const Address &): enum page_state
+get_free_page(address:const Address &): void
+get_num_free(address:const Address &): uint
+get_num_valid(address:const Address &): uint
-update_wear_stats(address:const Address &): void
Ssd
friends: Controller
-size: uint
-contoller: Controller
-ram: Ram
-bus: Bus
-data: Package * const
-erases_remaining: ulong
-least_worn: uint
-last_erase_time: double
+SSD(ssd_size:uint=SSD_SIZE)
+~SSD()
+event_arrive(type:enum event_type,
logical_address:ulong,
size:uint,
start_time:double): double
-read(event:Event &): enum status
-write(event:Event &): enum status
-erase(event:Event &): enum status
-merge(event:Event &): enum status
-get_erases_remaining(address:const Address &): ulong
-update_wear_stats(address:const Address &): void
-get_least_worn(Address:Address &): void
-get_last_erase_time(address:const Address &): double
-get_data(): Package &
-get_state(address:const Address &): enum page_state
-get_free_page(address:Address &): void
-get_num_free(address:const Address &): uint
-get_num_valid(address:const Address &): uint
Package
-size: uint
-data: Die * const
-parent: SSD &
-least_worn: uint
-last_erase_time: double
-erases_remaining: ulong
+Package(parent:Ssd &,
channel:Channel &,
package_size:uint=PACKAGE_SIZE)
+~Package()
+read(event:Event &): enum status
+write(event:Event &): enum status
+erase(event:Event &): enum status
+merge(event:Event &): enum status
+get_parent(): const Ssd &
+get_last_erase_time(): double
+get_erases_remaining(): ulong
+get_least_worn(address:Address &): void
+get_state(address:const Address &): enum page_state
+get_free_page(address:Address &): void
+get_num_free(address:const Address &): uint
+get_num_valid(address:const Address &): uint
-update_wear_stats(address:const Address &): void
Ram
-read_delay: double
-write_delay: double
+Ram(read_delay:double=RAM_READ_DELAY,
write_delay:double=RAM_WRITE_DELAY)
+~Ram()
+read(event:Event &): enum status
+write(event:Event &): enum status
Controller
friends: Ftl
-ssd: Ssd &
-ftl: Ftl
-erases_remaining: ulong
+Controller(parent:Ssd &)
+~Controller()
+event_arrive(event:Event &): enum status
-issue(event_list:Event &): enum status
-get_erases_remaining(): ulong
-get_least_worn(address:Address &): void
-get_last_erase_time(address:const Address &): double
-get_state(address:const Address &): enum page_state
-get_free_page(address:Address &): void
-get_num_free(address:const Address &): uint
-get_num_valid(address:const Address &): uint
Ftl
friends: Garbage_collector,
Wear_leveler
-controller: Controller &
-garbage: Garbage_collector
-wear: Wear_leveler
-free_list: Address *
-valid_list: Address *
-invalid_list: Address *
-map: long *
+Ftl(control:Controller &)
+~Ftl()
+read(event:Event &): enum status
+write(event:Event &): enum status
-erase(event:Event &): enum status
-merge(event:Event &): enum status
-garbage_collect(event:Event &): void
-get_erases_remaining(address:const Address &): ulong
-get_least_worn(address:Address &): void
-get_state(address:const Address &): enum page_state
Garbage_collector
-ftl: Ftl &
+Garbage_collector(ftl:Ftl &)
+~Garbage_collector()
+collect(event:Event &): enum status
Wear_leveler
-ftl: Ftl &
+Wear_leveler(ftl:Ftl &)
+~Wear_leveler()
+insert(address:const Address &): enum status