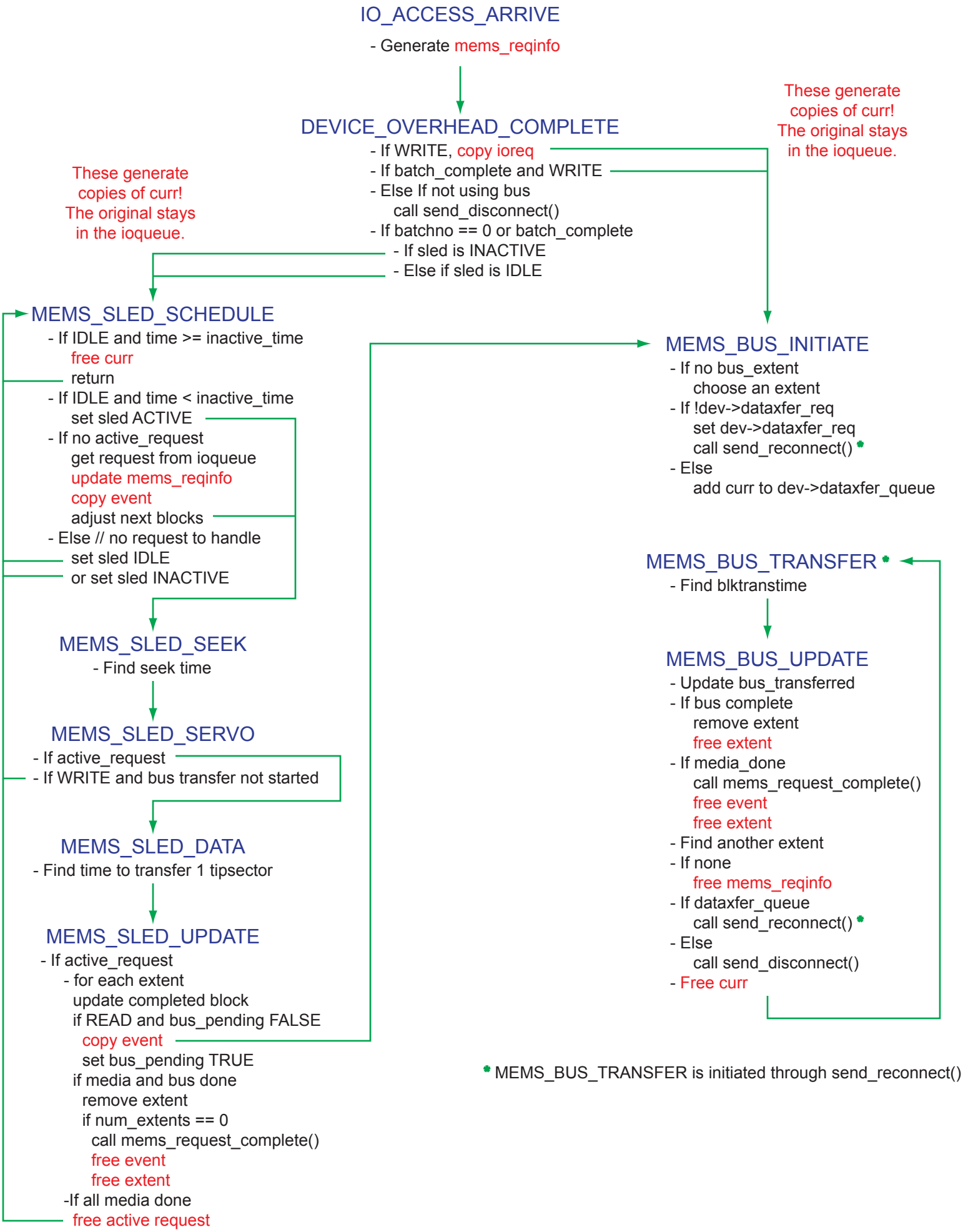
IO_ACCESS_ARRIVE
- Generate mems_reqinfo
DEVICE_OVERHEAD_COMPLETE
- If WRITE, copy ioreq
- If batch_complete and WRITE
- Else If not using bus
call send_disconnect()
- If batchno == 0 or batch_complete
- If sled is INACTIVE
- Else if sled is IDLE
These generate copies of curr! The original stays in the ioqueue.
These generate copies of curr! The original stays in the ioqueue.
MEMS_SLED_SCHEDULE
- If IDLE and time >= inactive_time
free curr
return
- If IDLE and time < inactive_time
set sled ACTIVE
- If no active_request
get request from ioqueue
update mems_reqinfo
copy event
adjust next blocks
- Else // no request to handle
set sled IDLE
or set sled INACTIVE
MEMS_SLED_SEEK
- Find seek time
MEMS_SLED_SERVO
- If active_request
- If WRITE and bus transfer not started
MEMS_SLED_DATA
- Find time to transfer 1 tipsector
MEMS_SLED_UPDATE
- If active_request
- for each extent
update completed block
if READ and bus_pending FALSE
copy event
set bus_pending TRUE
if media and bus done
remove extent
if num_extents == 0
call mems_request_complete()
free event
free extent
-If all media done
free active request
MEMS_BUS_INITIATE
- If no bus_extent
choose an extent
- If !dev->dataxfer_req
set dev->dataxfer_req
call send_reconnect() *
- Else
add curr to dev->dataxfer_queue
MEMS_BUS_TRANSFER *
- Find blktranstime
MEMS_BUS_UPDATE
- Update bus_transferred
- If bus complete
remove extent
free extent
- If media_done
call mems_request_complete()
free event
free extent
- Find another extent
- If none
free mems_reqinfo
- If dataxfer_queue
call send_reconnect() *
- Else
call send_disconnect()
- Free curr
* MEMS_BUS_TRANSFER is initiated through send_reconnect()