Verification Techniques with Linux Examples Project 2, Part II

April 6, 2016

Suresh C. Kothari
Department of Electrical and Computer Engineering

General Comments

- We verify one LOCK instance (one call to LOCK within one function) at a time.
- The BIG MPG is for related LOCK instances, the small MPG is for one particular LOCK instance.
- The Event Flow Graph (EFG) is often adequate for verification. In some instances, especially when a path feasibility check is required, the Control Flow Graph (CFG) is needed for additional information.

Related Locks in Big MPG

The LOCK instances in a BIG MPG are related in different ways. Here is one example of how they may be related.

1. The Big MPG has 4 nodes for functions F, G, LOCK, and UNLOCK.
2. F calls: LOCK, G, UNLOCK.
3. G calls: UNLOCK, LOCK.
4. Let us differentiate the two LOCK instances by denoting them as LOCK(F) and LOCK(G). Similarly denote the two UNLOCK instances by UNLOCK(F) and UNLOCK(G).
5. LOCK(F), LOCK(G), UNLOCK(F), and UNLOCK(G) are for the same object.
6. LOCK(F) and LOCK(G) are related. Note that UNLOCK(G) matches with LOCK(F).
7. For the instance LOCK(F), the EFG for G will show only the UNLOCK(G). The CFG will include also the LOCK(G).

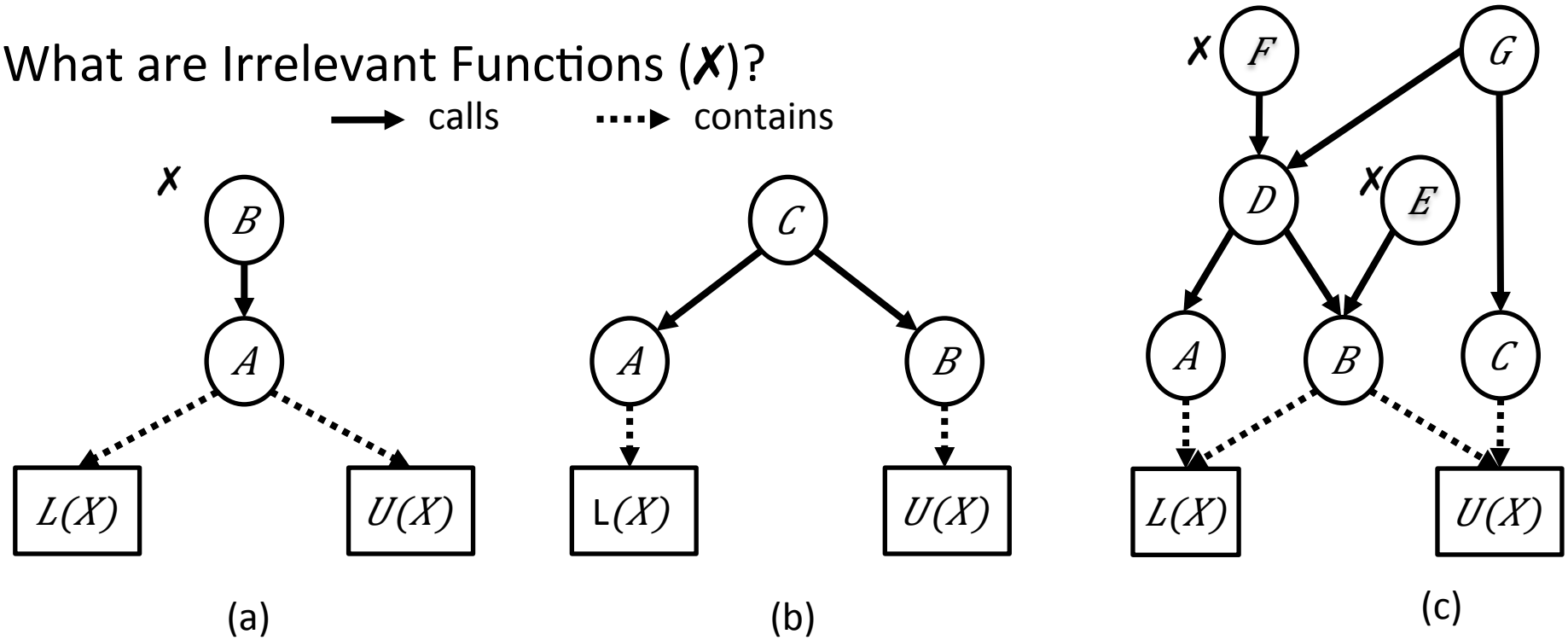
Optimality of MPG

The MPG is designed to be minimal to avoid the effort to through functions not necessary for verification.

1. The pairing may or may not happen at an intermediate node, i.e. a node that is not a root node of MPG.
2. Example: F calls: G1 followed G2. G1 calls LOCK and G2 calls UNLOCK. The MPG for LOCK(G1) includes F, G1, G2, LOCK(G1), LOCK(G2). The pairing does not happen at G1 but it happens at F.
3. If a pairing happens at an intermediate node, then we can use that node for *modular verification* as discussed earlier lecture notes.
4. The extreme case is when the pairing happens only at the root nodes on MPG.
5. The root nodes cannot be redundant for verification, otherwise the MPG is not minimal.

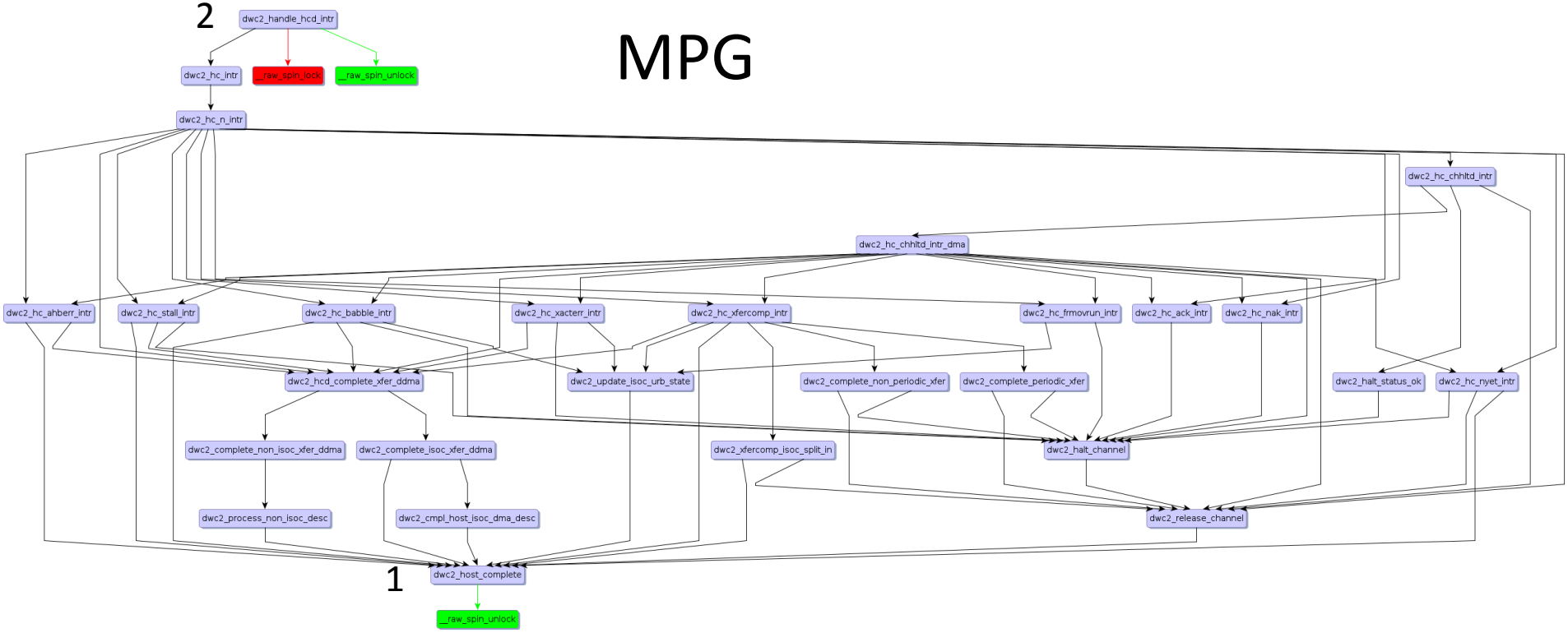
A glimpse of math behind MPG

The minimality of MPG is based on graph theory. Intuitive Idea: (a) Include parents of *unbalanced* nodes so that all nodes in the MPG are balanced. (b) Remove parents if all their children are balanced.

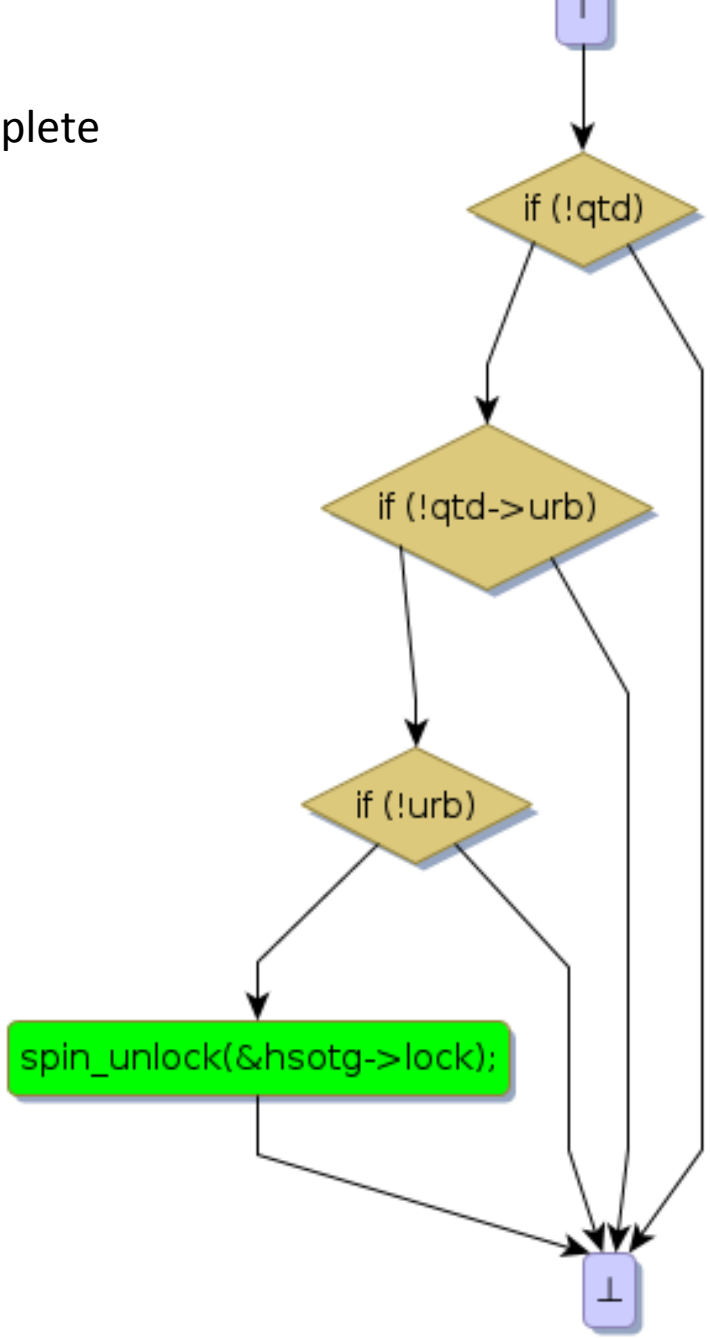


ANALYSIS OF 62332+24

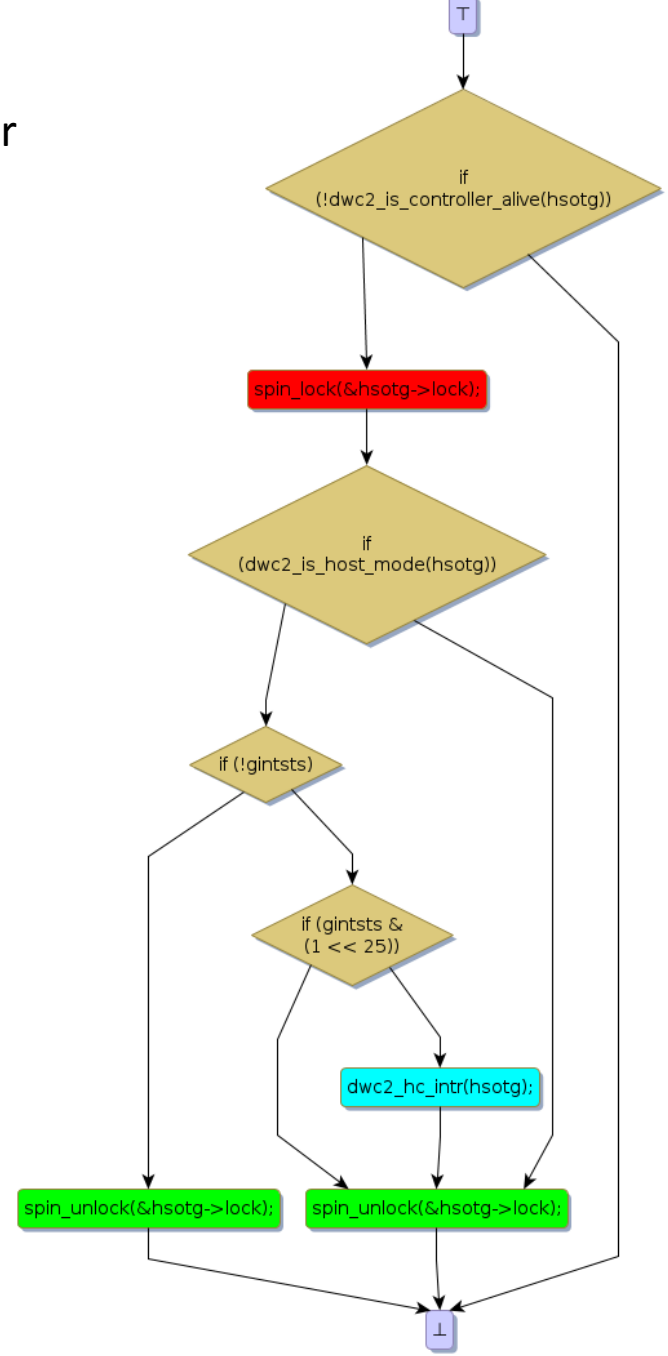
MPG


2
dwc2_handle_hcd_intr
dwc2_hc_intr
_raw_spin_lock
_raw_spin_unlock
dwc2_hc_n_intr
dwc2_hc_chhltd_intr
dwc2_hc_chhltd_intr_dma
dwc2_hc_ahberr_intr
dwc2_hc_stall_intr
dwc2_hc_babble_intr
dwc2_hc_xacterr_intr
dwc2_hc_xfercomp_intr
dwc2_hc_frmovrun_intr
dwc2_hc_ack_intr
dwc2_hc_nak_intr
dwc2_hcd_complete_xfer_ddma
dwc2_update_isoc_urb_state
dwc2_complete_non_periodic_xfer
dwc2_complete_periodic_xfer
dwc2_halt_status_ok
dwc2_hc_nyet_intr
dwc2_complete_non_isoc_xfer_ddma
dwc2_complete_isoc_xfer_ddma
dwc2_xfercomp_isoc_split_in
dwc2_halt_channel
dwc2_process_non_isoc_desc
dwc2_cmpl_host_isoc_dma_desc
dwc2_release_channel
1
dwc2_host_complete
_raw_spin_unlock

1. dwc2_host_complete

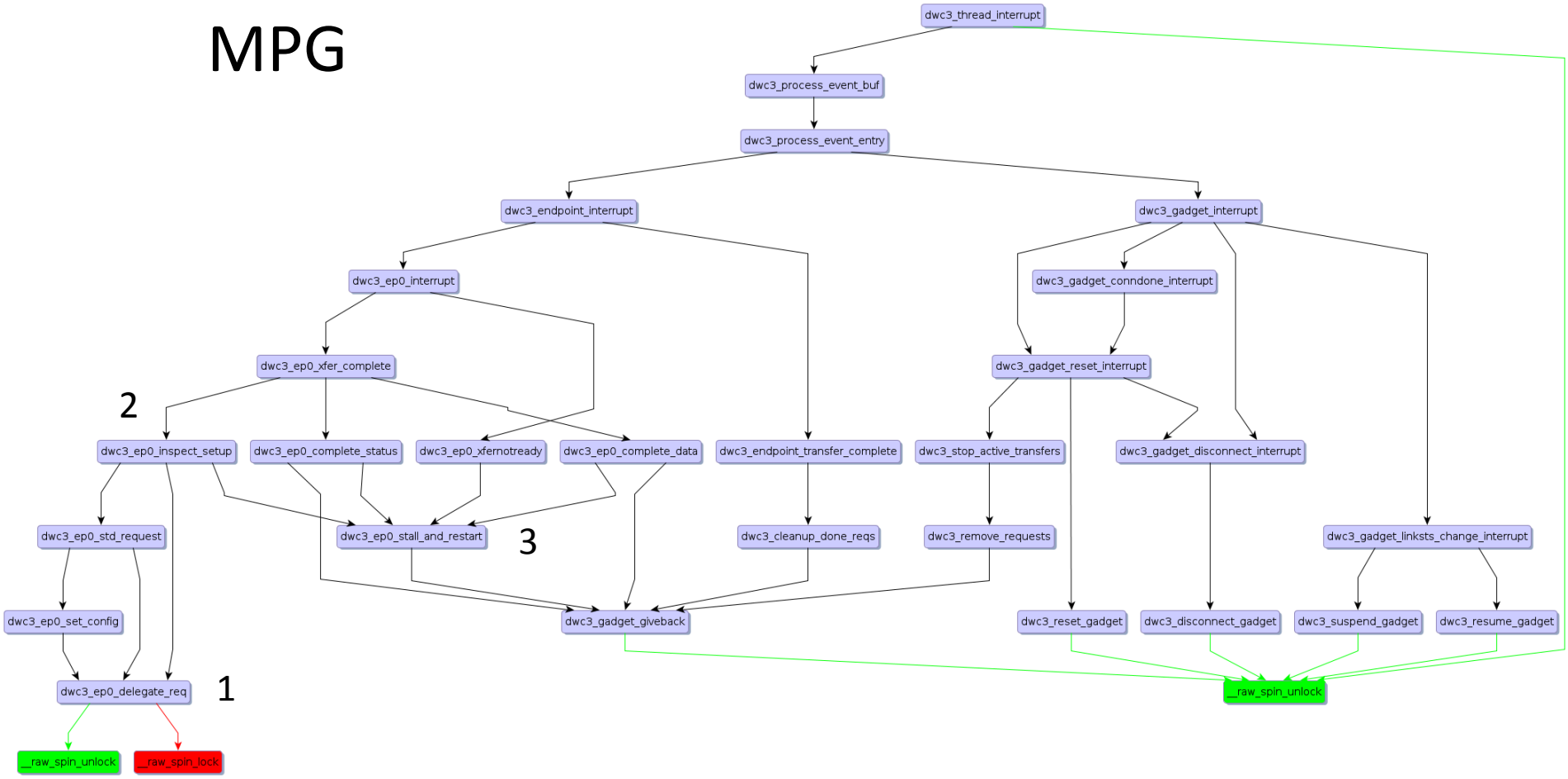


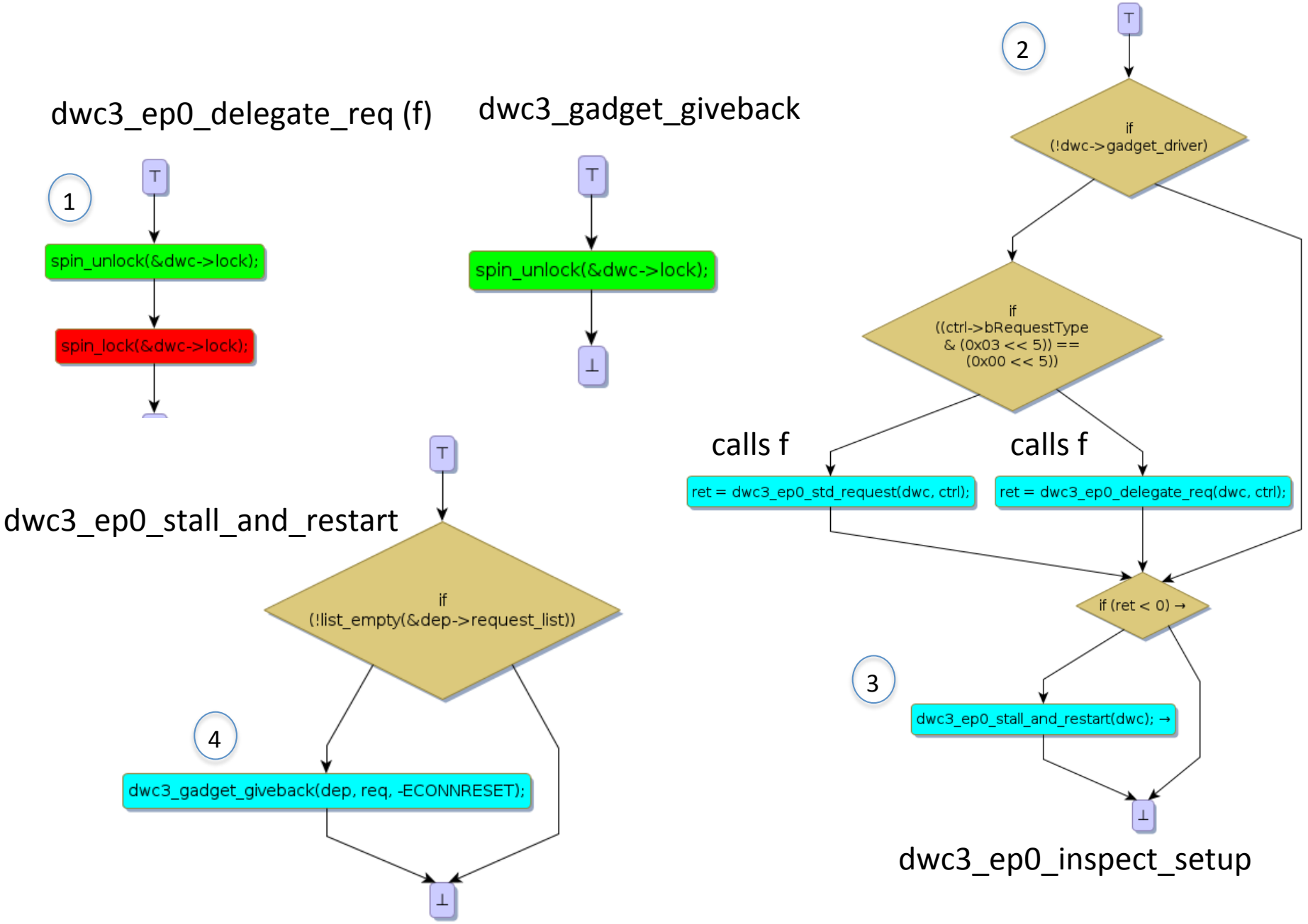
2. dwc2_handle_hcd_intr



ANALYSIS OF 13118+22

MPG


dwc3_thread_interrupt
dwc3_process_event_buf
dwc3_process_event_entry
dwc3_endpoint_interrupt
dwc3_gadget_interrupt
dwc3_ep0_interrupt
dwc3_gadget_conndone_interrupt
dwc3_ep0_xfer_complete
dwc3_gadget_reset_interrupt
2
dwc3_ep0_inspect_setup
dwc3_ep0_complete_status
dwc3_ep0_xfernotready
dwc3_ep0_complete_data
dwc3_endpoint_transfer_complete
dwc3_stop_active_transfers
dwc3_gadget_disconnect_interrupt
dwc3_ep0_std_request
dwc3_ep0_stall_and_restart
3
dwc3_cleanup_done_reqs
dwc3_remove_requests
dwc3_gadget_linksts_change_interrupt
dwc3_ep0_set_config
dwc3_gadget_giveback
dwc3_reset_gadget
dwc3_disconnect_gadget
dwc3_suspend_gadget
dwc3_resume_gadget
dwc3_ep0_delegate_req
1
__raw_spin_unlock
__raw_spin_unlock
__raw_spin_lock


dwc3_ep0_delegate_req (f)
1
spin_unlock(&dwc->lock);
spin_lock(&dwc->lock);
dwc3_gadget_giveback
spin_unlock(&dwc->lock);
dwc3_ep0_stall_and_restart
if
(!list_empty(&dep->request_list))
4
dwc3_gadget_giveback(dep, req, -ECONNRESET);
2
if
(!dwc->gadget_driver)
if
((ctrl->bRequestType
& (0x03 << 5)) ==
(0x00 << 5))
calls f
calls f
ret = dwc3_ep0_std_request(dwc, ctrl);
ret = dwc3_ep0_delegate_req(dwc, ctrl);
if (ret < 0) →
3
dwc3_ep0_stall_and_restart(dwc); →
dwc3_ep0_inspect_setup

ANALYSIS OF 25423

dwc3_suspen EFG