DWC3 driver

TODO

Please pick something while reading :)

- Convert interrupt handler to per-ep-thread-irq

  As it turns out some DWC3-commands ~1ms to complete. Currently we spin until the command completes which is bad.

  Implementation idea:

  - dwc core implements a demultiplexing irq chip for interrupts per endpoint. The interrupt numbers are allocated during probe and belong to the device. If MSI provides per-endpoint interrupt this dummy interrupt chip can be replaced with "real" interrupts.
  - interrupts are requested / allocated on usb_ep_enable() and removed on usb_ep_disable(). Worst case are 32 interrupts, the lower limit is two for ep0/1.
  - dwc3_send_gadget_ep_cmd() will sleep in wait_for_completion_timeout() until the command completes.
  - the interrupt handler is split into the following pieces:
    - primary handler of the device goes through every event and calls generic_handle_irq() for event it. On return from generic_handle_irq() in acknowledges the event counter so interrupt goes away (eventually).
    - threaded handler of the device none
    - primary handler of the EP-interrupt reads the event and tries to process it. Everything that requires sleeping is handed over to the Thread. The event is saved in an per-endpoint data-structure. We probably have to pay attention not to process events once we handed something to thread so we don't process event X prio Y where X > Y.
    - threaded handler of the EP-interrupt handles the remaining EP work which might sleep such as waiting for command completion.

  Latency:

  > There should be no increase in latency since the interrupt-thread has a high priority and will be run before an average task in user land (except the user changed priorities).