GPMC (General Purpose Memory Controller)

GPMC is an unified memory controller dedicated to interfacing external memory devices like

- Asynchronous SRAM like memories and application specific integrated circuit devices.
- Asynchronous, synchronous, and page mode burst NOR flash devices NAND flash
- Pseudo-SRAM devices

GPMC is found on Texas Instruments SoC's (OMAP based) IP details: https://www.ti.com/lit/pdf/spruh73 section 7.1

GPMC generic timing calculation:

GPMC has certain timings that has to be programmed for proper functioning of the peripheral, while peripheral has another set of timings. To have peripheral work with gpmc, peripheral timings has to be translated to the form gpmc can understand. The way it has to be translated depends on the connected peripheral. Also there is a dependency for certain gpmc timings on gpmc clock frequency. Hence a generic timing routine was developed to achieve above requirements.

Generic routine provides a generic method to calculate gpmc timings from gpmc peripheral timings. struct gpmc_device_timings fields has to be updated with timings from the datasheet of the peripheral that is connected to gpmc. A few of the peripheral timings can be fed either in time or in cycles, provision to handle this scenario has been provided (refer struct gpmc_device_timings definition). It may so happen that timing as specified by peripheral datasheet is not present in timing structure, in this scenario, try to correlate peripheral timing to the one available. If that doesn't work, try to add a new field as required by peripheral, educate generic timing routine to handle it, make sure that it does not break any of the existing. Then there may be cases where peripheral datasheet doesn't mention certain fields of struct gpmc_device_timings, zero those entries.

Generic timing routine has been verified to work properly on multiple onenand's and tusb6010 peripherals.

A word of caution: generic timing routine has been developed based on understanding of gpmc timings, peripheral timings, available custom timing routines, a kind of reverse engineering without most of the datasheets & hardware (to be exact none of those supported in mainline having custom timing routine) and by simulation.

gpmc timing dependency on peripheral timings:

[<gpmc_timing>: <peripheral timing1>, <peripheral timing2> ...]

1. common

cs_on:
    t_ceasu
adv_on:
    t_avdasu, t_ceavd

2. sync common

sync_clk:
    clk
page_burst_access:
    t_bacc
clk_activation:
    t_ces, t_avds

3. read async muxed

adv_rd_off:
    t_avdp_r
oe_on:
    t_oeasu, t_aavdh
access:
    t_iaa, t_oe, t_ce, t_aa
rd_cycle:
    t_rd_cycle, t_cez_r, t_oez

4. read async non-muxed

adv_rd_off:
    t_avdp_r
oe_on:
    t_oeasu
access:
    t_iaa, t_oe, t_ce, t_aa
rd_cycle:

t_rd_cycle, t_cez_r, t_oez

5. read sync muxed

adv_rd_off:
t_avdp_r, t_avdh
oe_on:
t_oeasu, t_ach, cyc_aavdh_oe
access:
t_iaa, cyc_iaa, cyc_oe
rd_cycle:
t_cez_r, t_oez, t_ce_rdyz

6. read sync non-muxed

adv_rd_off:
t_avdp_r
oe_on:
t_oeasu
access:
t_iaa, cyc_iaa, cyc_oe
rd_cycle:
t_cez_r, t_oez, t_ce_rdyz

7. write async muxed

adv_wr_off:
t_avdp_w
we_on, wr_data_mux_bus:
t_weasu, t_aavdh, cyc_aavhd_we
we_off:
t_wpl
cs_wr_off:
t_wph
wr_cycle:
t_cez_w, t_wr_cycle

8. write async non-muxed

adv_wr_off:
t_avdp_w
we_on, wr_data_mux_bus:
t_weasu
we_off:
t_wpl
cs_wr_off:
t_wph
wr_cycle:
t_cez_w, t_wr_cycle

9. write sync muxed

adv_wr_off:
t_avdp_w, t_avdh
we_on, wr_data_mux_bus:
t_weasu, t_rdyo, t_aavdh, cyc_aavhd_we
we_off:
t_wpl, cyc_wpl
cs_wr_off:
t_wph
wr_cycle:
t_cez_w, t_ce_rdyz

10. write sync non-muxed

adv_wr_off:
t_avdp_w
we_on, wr_data_mux_bus:
t_weasu, t_rdyo
we_off:
t_wpl, cyc_wpl
cs_wr_off:

t_wph

wr_cycle:

t_cez_w, t_ce_rdyz

Note:

Many of gpmc timings are dependent on other gpmc timings (a few gpmc timings purely dependent on other gpmc timings, a reason that some of the gpmc timings are missing above), and it will result in indirect dependency of peripheral timings to gpmc timings other than mentioned above, refer timing routine for more details. To know what these peripheral timings correspond to, please see explanations in struct gpmc_device_timings definition. And for gpmc timings refer IP details (link above).