DDR configuration and initilaization for Synopsis UMCTL-based platforms

Technical Note

TNxxxx
Rev. 15 (master@02a6757)
Mon Jan 10 12:46:03 2022 +0100

Confidential

TERMS and ABBREVIATIONS

```
PUB                PHY utility block
SoC                System on a Chip
```

Introduction

Purpose of this document is to specify and document the design and requirements of the DDR customization for Synopsis UMCTL-based SoC’s. This currently implies Fireant, Maserati and Laguna.

Configuring DDR for these platforms is a delicate process including a lot of configuration parameters. It is furthermore complicated by the needs for customer customization, sourcing changes and system tuning.

As we wish to provide a better method for enabling customers, many different boards and current/future SoC’s, this document outlines a possible solution for this.

Primary target audience is:

- **SW Engineers:** This is the group of people who will do the software implementation of the proposed solution. Implementation and specification must be in alignment.
- **SW-Application:** Parts of this implementation will be directly exposed to customers, mainly the configuration tool and associated documentation. This group must be consulted to ensure catering for customer needs and being able to support the system solution as a whole.
- **Chip design:** This group will be consulted to ensure all relevant parameters are exposed to customization, and that the driver follow the necessary hardware initialization steps.

Requirements

The system must support the following high-level requirements:

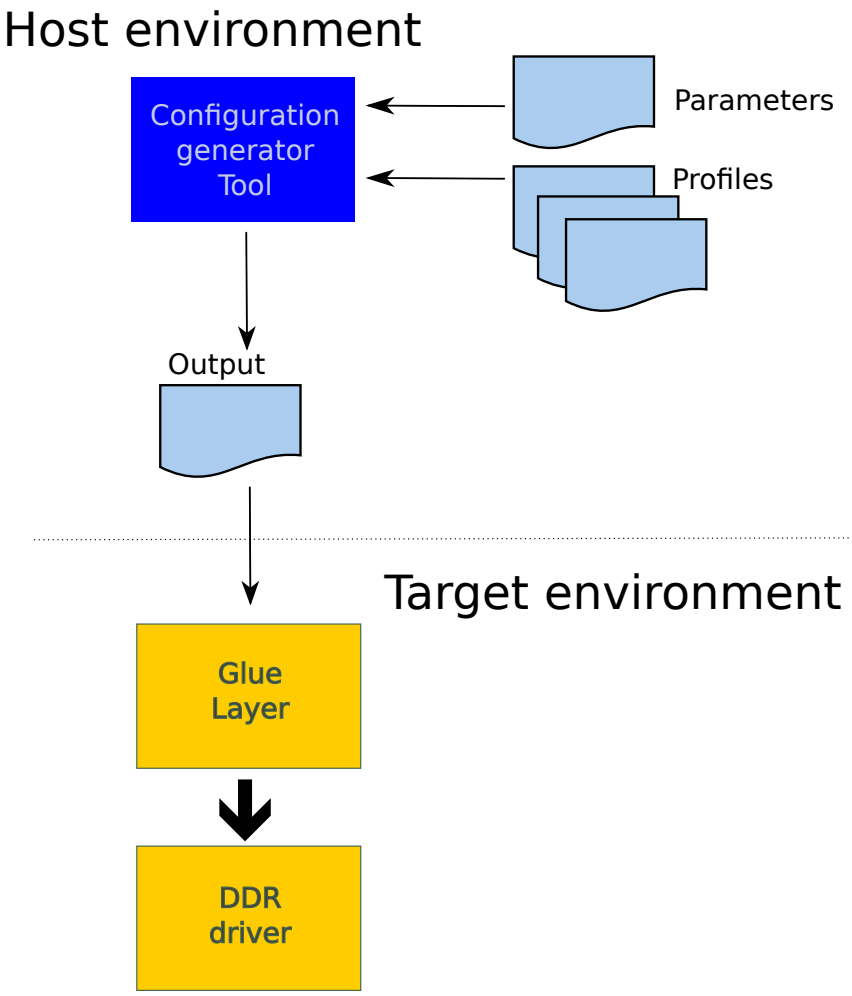
- The system must be able to support the current set of SoC’s and boards using the UMCTL controller. This imply all Fireant boards, Maserati boards and Laguna boards.
- The system must be able to support new SoC’s and their boards without imposing changes to exposed parameters.
- A set of high-level DDR configuration parameters must be exposed for customization of the 'common-case' changes a customer is expected to make.
- It should be possible to extend the configuration genrator tool with profiles/ parameter-sets to support new DDR chip types as required by sourcing needs.
- It should be possible to customize DDR parameters for a board without necessarily recompiling the firmware for the target system.
- The run-time parts of the system initialization should be as compact as possible, and require no floating-point support.

System design

To implement a system which feature the listed requirements, the following solution is proposed:

- A host configuration generator tool is created. This tool will take the actual values of the exposed parameters, the applicable DDR parameter sets and will produce a 'ready-to-use' DDR controller output configuration 'snippet'. All necessary calculations will be done in the configuration tool.
- The host configuration generator tool will be able to produce the output configuration snippets in different formats to support different platforms - binary, "C", YAML and device-tree (DT) format.
- The host configuration generator tool will operate on input data specified in YAML format, and DDR profiles/parameter-sets will be in the same format.
- A system-agnostic driver will be created. The driver will accept the DDR configuration 'snippet' in **binary** form, along with platform-defined DDR register access functions. Using this, the driver will be able to perform the full DDR initialization according to the UMCTL programmers guide.
- A system-specific 'glue' component will be created to extract the DDR configuration 'snippet' and convert it to binary form. It is the responsibility of this component to call the base driver, alongside with the register access functions. This component could optionally provide built-in fallback configuration profiles if initialization with the dynamically provided data fails.

The system architecture is outlined in the following diagram:



Configuration parameters

User-level configuration parameters

At the top level, the following user-level parameters are identified.

- Title: *text* (This text is accompanying the configuration for identification purposes)
- DDR type: `DDR3` , `DDR4`
- DDR speed: *integer* (KHz)
- DDR geometry:

  - Column bits
  - Row bits
  - Bank bits
  - Bank group bits
  - Active ranks 1 or 2
  - DQ bits: *x8/x16*
  - DQ bits used: *x16/x32/x40*
  - Density: *4G/8G*
- 2T mode (???): *enabled/disabled*
- ECC mode: *enabled/disabled*

NOTE Some platforms may not support all parameters.

DDR chip parameters

All DDR chip parameters are derived by the user-level parameters, primarily the DDR mode and speed (grade).

Output DDR configuration

The DDR output configuration data is as follows.

NOTE In the following section, the *stm32mp1* column is extracted from drivers/st/ddr/stm3<2mp1_ddr.c (https://bitbucket.microchip.com/projects/UNGE/repos/sw-arm-trusted-firmware/browse/drivers/st/ddr/stm32mp1_ddr.c?at=refs%2Fheads%2Flaguna-v1) and the associated header file include/drivers/st/stm32mp1_ddr.h (https://bitbucket.microchip.com/projects/UNGE/repos/sw-arm-trusted-firmware/browse/include/drivers/st/stm32mp1_ddr.h?at=refs%2Fheads%2Flaguna-v1). The driver is going to be used as a reference to implement the Microchip equivalent, and it is expected to change the register set as required/desired.

Info

- Title: *text* (from input)
- Speed
- Memory size

Main control registers

register	group	ddr tcl	fa ddr3	fa ddr4	stm32mp1	comments
crcparctl1	umctl2_regs	yes	-	yes	-	write_crc, ca_parity_en
dbictl	umctl2_regs	yes	-	yes	-	dbi_en
dfimisc	umctl2_regs	yes	yes	yes	-	dbi_en
dfitmg0	umctl2_regs	yes	yes	yes	yes	
dfitmg1	umctl2_regs	yes	yes	yes	yes	
dfiupd0	umctl2_regs	yes	yes	yes	yes	
dfiupd1	umctl2_regs	yes	yes	yes	yes	
ecccfg0	umctl2_regs	yes	yes	yes	-	ecc_mode
init0	umctl2_regs	yes	yes	yes	-	pre_cke / post_cke
init1	umctl2_regs	yes	yes	yes	-	DRAM_RSTN_X1024
init3	umctl2_regs	yes	yes	yes	-	params_reg_ddrc_mr / ddrc_emr
init4	umctl2_regs	yes	yes	yes	-	params_reg_ddrc_emr3 / ddrc_emr2
init5	umctl2_regs	yes	yes	yes	-	params_tZQinitc
init6	umctl2_regs	yes	-	yes	-	params_reg_ddrc_mr5
init7	umctl2_regs	yes	-	yes	-	params_reg_ddrc_mr6
mstr	umctl2_regs	yes	yes	yes	yes	
pccfg	umctl2_mp	yes	-	-	yes	Only used for x16 width configurations
pwrctl	umctl2_regs	yes	yes	yes	yes	
rfshctl0	umctl2_regs	yes	yes	yes	yes	
rfshctl3	umctl2_regs	yes	yes	yes	yes	

Timing configuration registers

register	group	ddr tcl	fa ddr3	fa ddr4	stm32mp1	comments
dramtmg0	umctl2_regs	yes	yes	yes	yes	
dramtmg1	umctl2_regs	yes	yes	yes	yes	
dramtmg12	umctl2_regs	yes	-	-	-	Only used with PDA_EN, do need this feature?
dramtmg2	umctl2_regs	yes	yes	yes	yes	
dramtmg3	umctl2_regs	yes	yes	yes	yes	
dramtmg4	umctl2_regs	yes	yes	yes	yes	
dramtmg5	umctl2_regs	yes	yes	yes	yes	
dramtmg8	umctl2_regs	yes	yes	yes	yes	
dramtmg9	umctl2_regs	yes	-	yes	-	Should this be configured even for DDR3?
odtcfg	umctl2_regs	yes	yes	yes	yes	
rfshtmg	umctl2_regs	yes	yes	yes	yes	

Address map configuration registers

register	group	ddr tcl	fa ddr3	fa ddr4	stm32mp1	comments
addrmap0	umctl2_regs	yes	yes	yes	-	
addrmap1	umctl2_regs	yes	yes	yes	yes	
addrmap2	umctl2_regs	yes	yes	yes	yes	
addrmap3	umctl2_regs	yes	yes	yes	yes	
addrmap4	umctl2_regs	yes	yes	yes	yes	
addrmap5	umctl2_regs	yes	yes	yes	yes	

register	group	ddr tcl	fa ddr3	fa ddr4	stm32mp1	comments
addrmap6	umctl2_regs	yes	yes	yes	yes	
addrmap7	umctl2_regs	yes	yes	yes	-	
addrmap8	umctl2_regs	yes	yes	yes	-	

DDR PHY registers

register	group	ddr tcl	fa ddr3	fa ddr4	stm32mp1	comments
dcr	dwc_ddrphy_pub	yes	yes	yes	yes	
dsgcr	dwc_ddrphy_pub	yes	yes	yes	yes	
dtcr0	dwc_ddrphy_pub	yes	yes	yes	-	depends on #lanes (dq_bits_used)
dtcr1	dwc_ddrphy_pub	yes	yes	yes	-	depends params_active_ranks
dxccr	dwc_ddrphy_pub	yes	yes	yes	yes	
pgcr2	dwc_ddrphy_pub	yes	yes	yes	-	params_tRASc_max
schcr1	dwc_ddrphy_pub	yes	-	-	-	params_active_ranks > 1
zq0pr	dwc_ddrphy_pub	-	yes	yes	-	ddrconf→ca_ln_drv : ddrconf→zqdiv
zq1pr	dwc_ddrphy_pub	-	yes	yes	-	ddrconf→zqdiv
zq2pr	dwc_ddrphy_pub	-	yes	yes	-	ddrconf→zqdiv
zqcr	dwc_ddrphy_pub	yes	yes	yes	-	asym_drv_pd/pu and params_tCK_min

DDR PHY timing registers

register	group	ddr tcl	fa ddr3	fa ddr4	stm32mp1	comments
dtpr0	dwc_ddrphy_pub	yes	yes	yes	yes	
dtpr1	dwc_ddrphy_pub	yes	yes	yes	yes	

register	group	ddr tcl	fa ddr3	fa ddr4	stm32mp1	comments
dtpr2	dwc_ddrphy_pub	yes	yes	yes	yes	
dtpr3	dwc_ddrphy_pub	yes	yes	yes	-	depends params_tDLLKc
dtpr4	dwc_ddrphy_pub	yes	yes	yes	-	depends params_tXPc / params_tXPDLLc
dtpr5	dwc_ddrphy_pub	yes	yes	yes	-	depends params_tWTRc / params_tRCDc / params_tRCc
mr0	dwc_ddrphy_pub	yes	yes	yes	yes	params_reg_ddrc_mr
mr1	dwc_ddrphy_pub	yes	yes	yes	yes	ddrc_emr
mr2	dwc_ddrphy_pub	yes	yes	yes	yes	ddrc_emr2
mr3	dwc_ddrphy_pub	yes	yes	yes	yes	params_reg_ddrc_emr3
mr4	dwc_ddrphy_pub	yes	-	yes	-	params_reg_ddrc_mr4
mr5	dwc_ddrphy_pub	yes	-	yes	-	dbi_en / params_dm_en / params_reg_ddrc_mr5
mr6	dwc_ddrphy_pub	yes	-	yes	-	params_reg_ddrc_mr6
ptr0	dwc_ddrphy_pub	yes	yes	yes	yes	
ptr1	dwc_ddrphy_pub	yes	yes	yes	yes	
ptr2	dwc_ddrphy_pub	-	-	-	yes	Should this be used?
ptr3	dwc_ddrphy_pub	yes	yes	yes	-	params_tdinit0 / params_tXS_tRFCc
ptr4	dwc_ddrphy_pub	yes	yes	yes	-	params_tdinit2 / params_tZQinitc

Sparx5 DDR registers not depending on configuration

register	group	ddr tcl	fa ddr3	fa ddr4	comments
bistar0	dwc_ddrphy_pub	-	-	yes	VREF training
bistar1	dwc_ddrphy_pub	-	-	yes	VREF training

register	group	ddr tcl	fa ddr3	fa ddr4	comments
bistar3	dwc_ddrphy_pub	-	-	yes	VREF training
bistudpr	dwc_ddrphy_pub	-	-	yes	VREF training
dx0bdlr0	dwc_ddrphy_pub	-	yes	yes	Used for data training (fixed value)
dx0bdlr1	dwc_ddrphy_pub	-	yes	yes	Used for data training (fixed value)
dx0bdlr2	dwc_ddrphy_pub	-	yes	yes	Used for data training (fixed value)
dx0gcr5	dwc_ddrphy_pub	-	-	yes	Fixed value: Used for VDDQ
dx0gtr0	dwc_ddrphy_pub	yes	yes	yes	Fixed value: DGSL = 2 used, should it ever changed?
dx1bdlr0	dwc_ddrphy_pub	-	yes	yes	Used for data training (fixed value)
dx1bdlr1	dwc_ddrphy_pub	-	yes	yes	Used for data training (fixed value)
dx1bdlr2	dwc_ddrphy_pub	-	yes	yes	Used for data training (fixed value)
dx1gcr5	dwc_ddrphy_pub	-	-	yes	Fixed value: Used for VDDQ
dx1gtr0	dwc_ddrphy_pub	yes	yes	yes	Fixed value: DGSL = 2 used, should it ever changed?
dx2bdlr0	dwc_ddrphy_pub	-	yes	yes	Used for data training (fixed value)
dx2bdlr1	dwc_ddrphy_pub	-	yes	yes	Used for data training (fixed value)

register	group	ddr tcl	fa ddr3	fa ddr4	comments
dx2bdlr2	dwc_ddrphy_pub	-	yes	yes	Used for data training (fixed value)
dx2gcr0	dwc_ddrphy_pub	yes	-	-	Only used for x16 width configurations
dx2gcr5	dwc_ddrphy_pub	-	-	yes	Fixed value: Used for VDDQ
dx2gtr0	dwc_ddrphy_pub	yes	yes	yes	Fixed value: DGSL = 2 used, should it ever changed?
dx3bdlr0	dwc_ddrphy_pub	-	yes	yes	Used for data training (fixed value)
dx3bdlr1	dwc_ddrphy_pub	-	yes	yes	Used for data training (fixed value)
dx3bdlr2	dwc_ddrphy_pub	-	yes	yes	Used for data training (fixed value)
dx3gcr0	dwc_ddrphy_pub	yes	-	-	Only used for x16 width configurations
dx3gcr5	dwc_ddrphy_pub	-	-	yes	Fixed value: Used for VDDQ
dx3gtr0	dwc_ddrphy_pub	yes	yes	yes	Fixed value: DGSL = 2 used, should it ever changed?
dx4bdlr0	dwc_ddrphy_pub	-	yes	yes	Used for data training (fixed value)
dx4bdlr1	dwc_ddrphy_pub	-	yes	yes	Used for data training (fixed value)
dx4bdlr2	dwc_ddrphy_pub	-	yes	yes	Used for data training (fixed value)

register	group	ddr tcl	fa ddr3	fa ddr4	comments
dx4gcr0	dwc_ddrphy_pub	yes	-	-	Only used for x16 width configurations
dx4gcr5	dwc_ddrphy_pub	-	-	yes	Fixed value: Used for VDDQ
dx4gtr0	dwc_ddrphy_pub	yes	yes	yes	Fixed value: DGSL = 2 used, should it ever changed?
eccpoisonaddr0	umctl2_regs	yes	-	-	*Ignore*: Only used for ECC injection
eccpoisonaddr1	umctl2_regs	yes	-	-	*Ignore*: Only used for ECC injection
iovcr0	dwc_ddrphy_pub	yes	yes	yes	Fixed settings
iovcr1	dwc_ddrphy_pub	yes	yes	yes	Fixed settings
pctrl_0	umctl2_mp	-	yes	yes	Control only (ECC scrubbing)
pgcr0	dwc_ddrphy_pub	yes	yes	yes	Control only
pgcr1	dwc_ddrphy_pub	yes	yes	yes	Fixed settings
pgcr3	dwc_ddrphy_pub	yes	yes	yes	Fixed settings, but **used** to hold params params_rd_dbi_en / params_wr_dbi_en (TCL script)
pgcr7	dwc_ddrphy_pub	yes	yes	yes	Fixed settings
pir	dwc_ddrphy_pub	yes	yes	yes	Fixed settings / depending on DDR3/DDR4
rankctl	umctl2_regs	yes	yes	yes	Fixed settings
rankidr	dwc_ddrphy_pub	yes	yes	yes	Index register for `DX*GTR0` etc
rfshctl1	umctl2_regs	yes	yes	yes	Fixed settings

register	group	ddr tcl	fa ddr3	fa ddr4	comments
sbrctl	umctl2_mp	-	yes	yes	Used for ECC scrubbing init
sbrwdata0	umctl2_mp	-	yes	yes	Used for ECC scrubbing init
swctl	umctl2_regs	yes	yes	yes	Control only
vtcr0	dwc_ddrphy_pub	-	-	yes	Fixed settings
vtcr1	dwc_ddrphy_pub	-	-	yes	Fixed settings