AURIX 2G Startup

Sherry Li
IFCN ATV SMD GC SAE MC

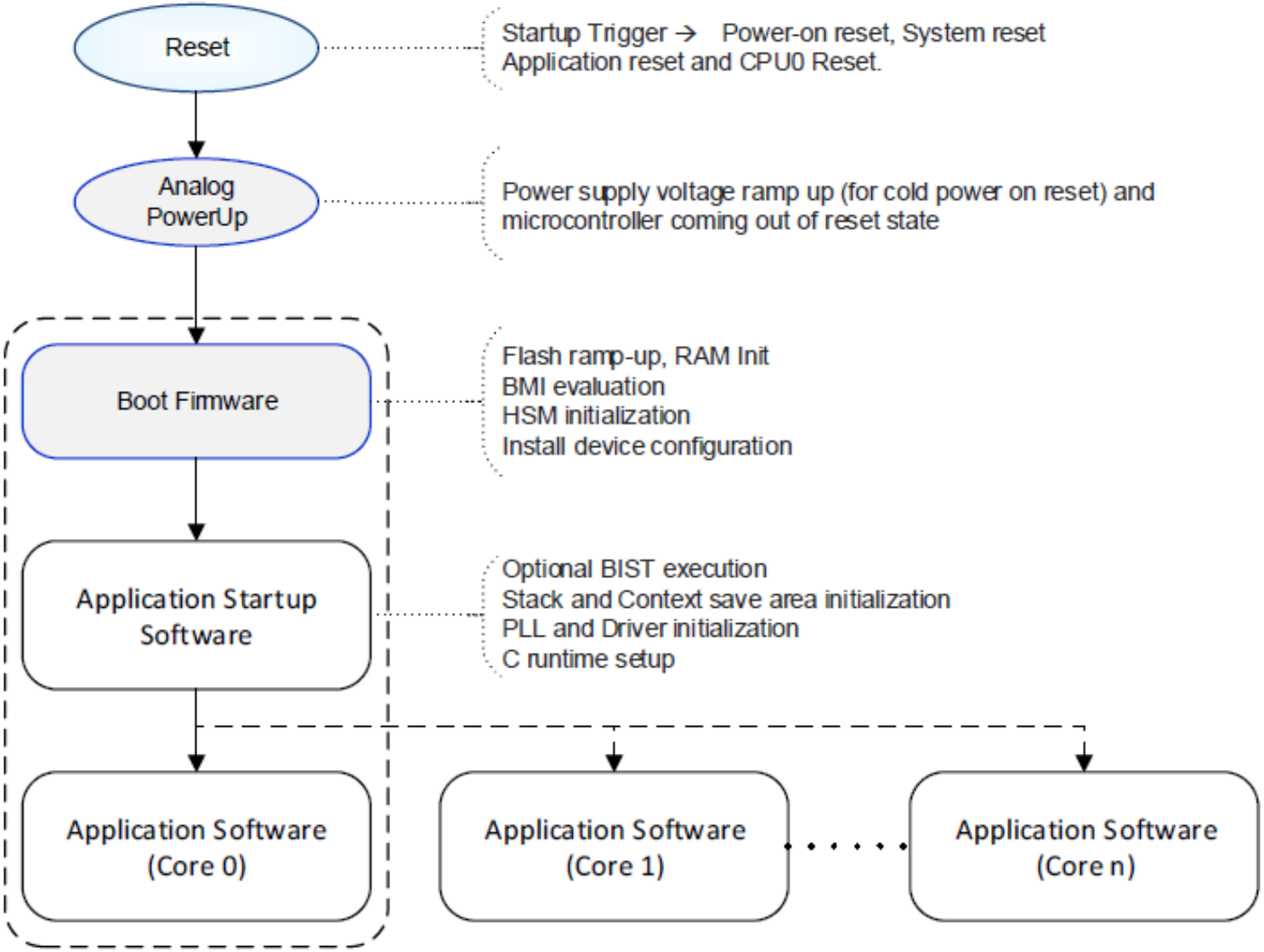
Agenda

1. Power Startup
2. Boot Options and Boot Mode Headers
3. Startup Flow Complete Picture

Agenda

1. Power Startup
2. Boot Options and Boot Mode Headers
3. Startup Flow Complete Picture

TC3xx Startup Overview



TC3xx
Single Supply Mode Start-up

1. External supply ramps up

   (dV_{EXT}/dt =120V/ms - 0.5V/min)

   Robust against residual supply-ramp

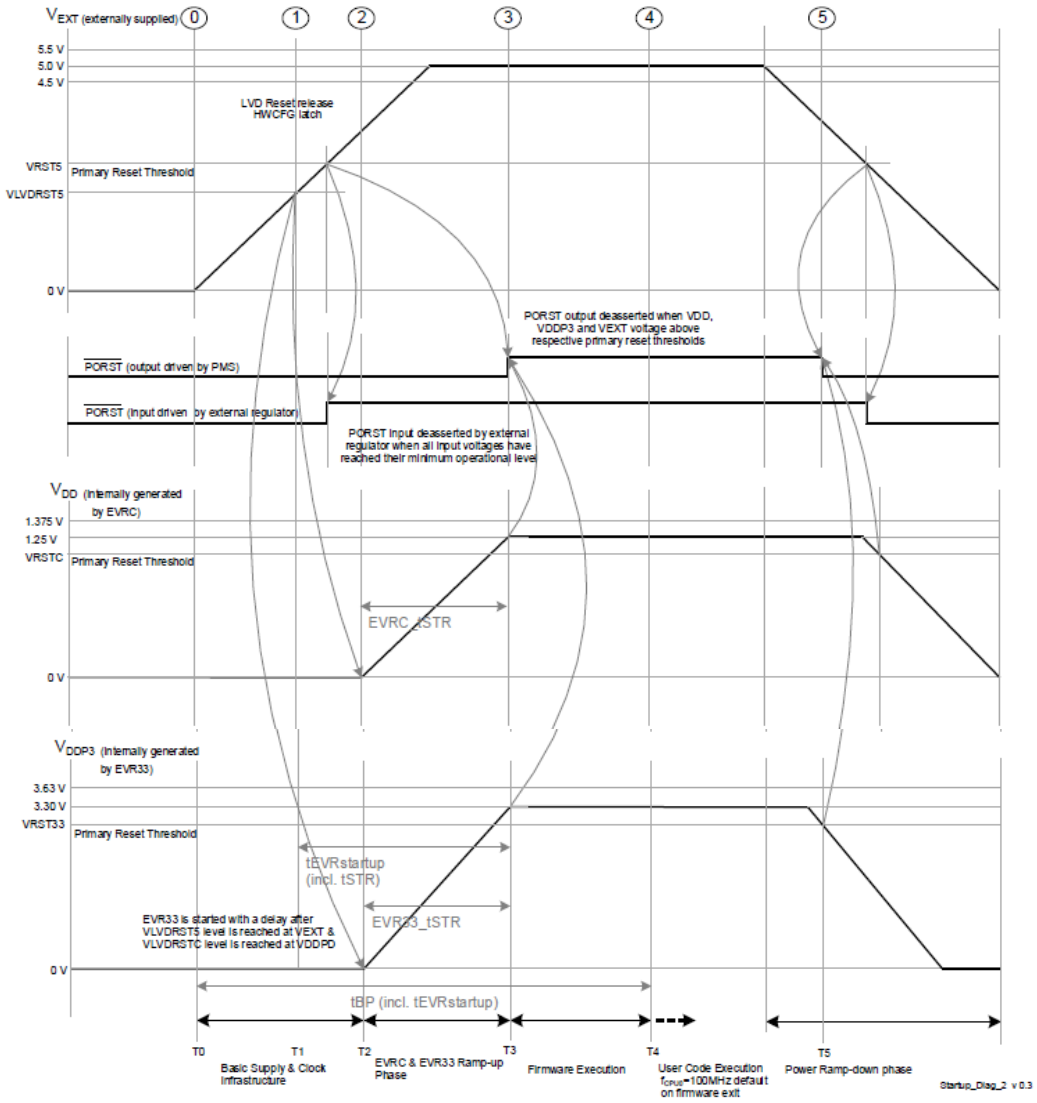
   No power sequencing

@ VEVRSB = VLVDRST5

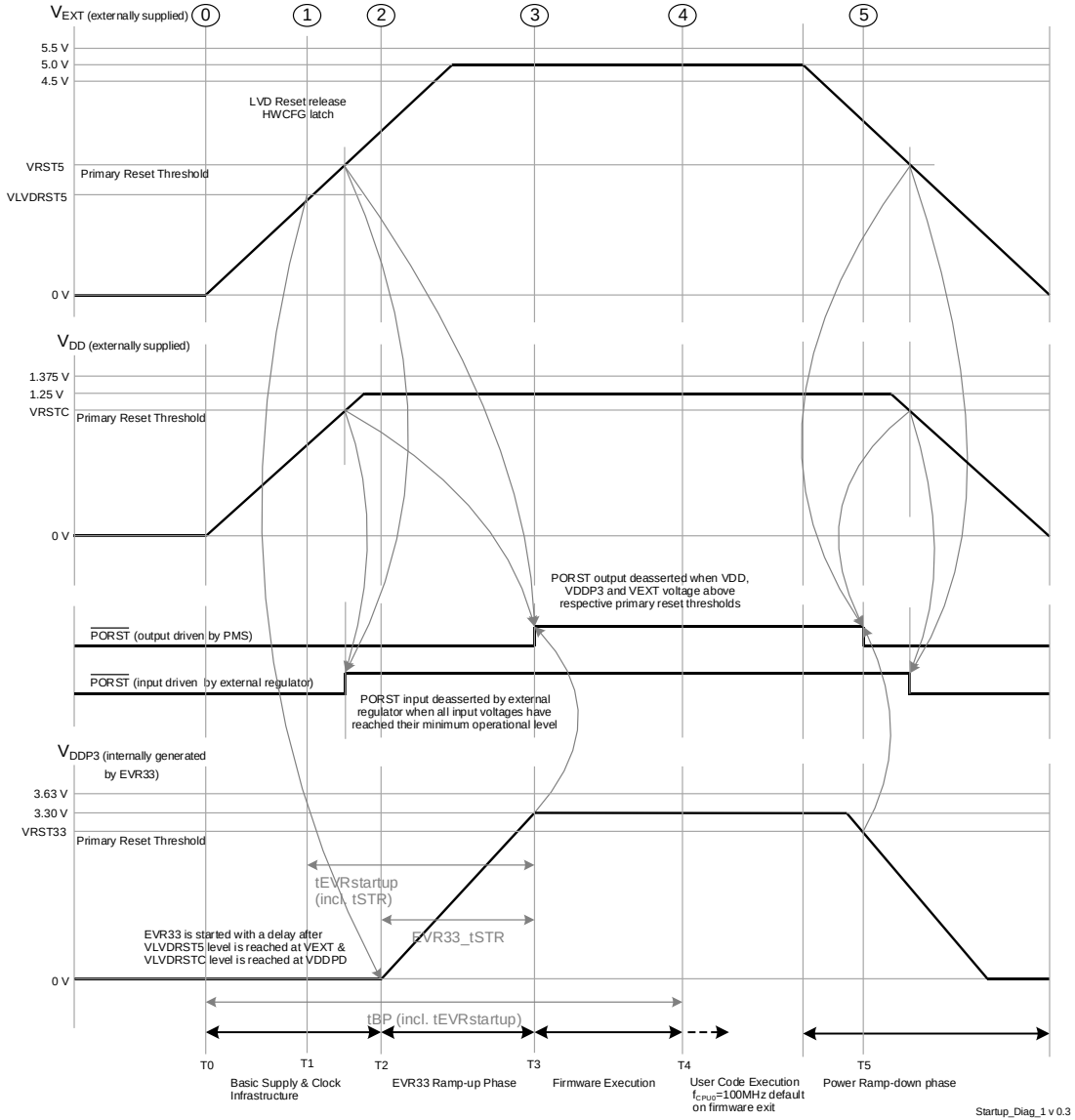
   - Basic Infrastructure ramp-up
   - HWCFG pin evaluation

2. EVRC & EVR33 soft ramp-up
   - Gradual Capacitor charging
   - PORST release when all supplies in operational range
   - Start-up Safety tests (TC39xB)

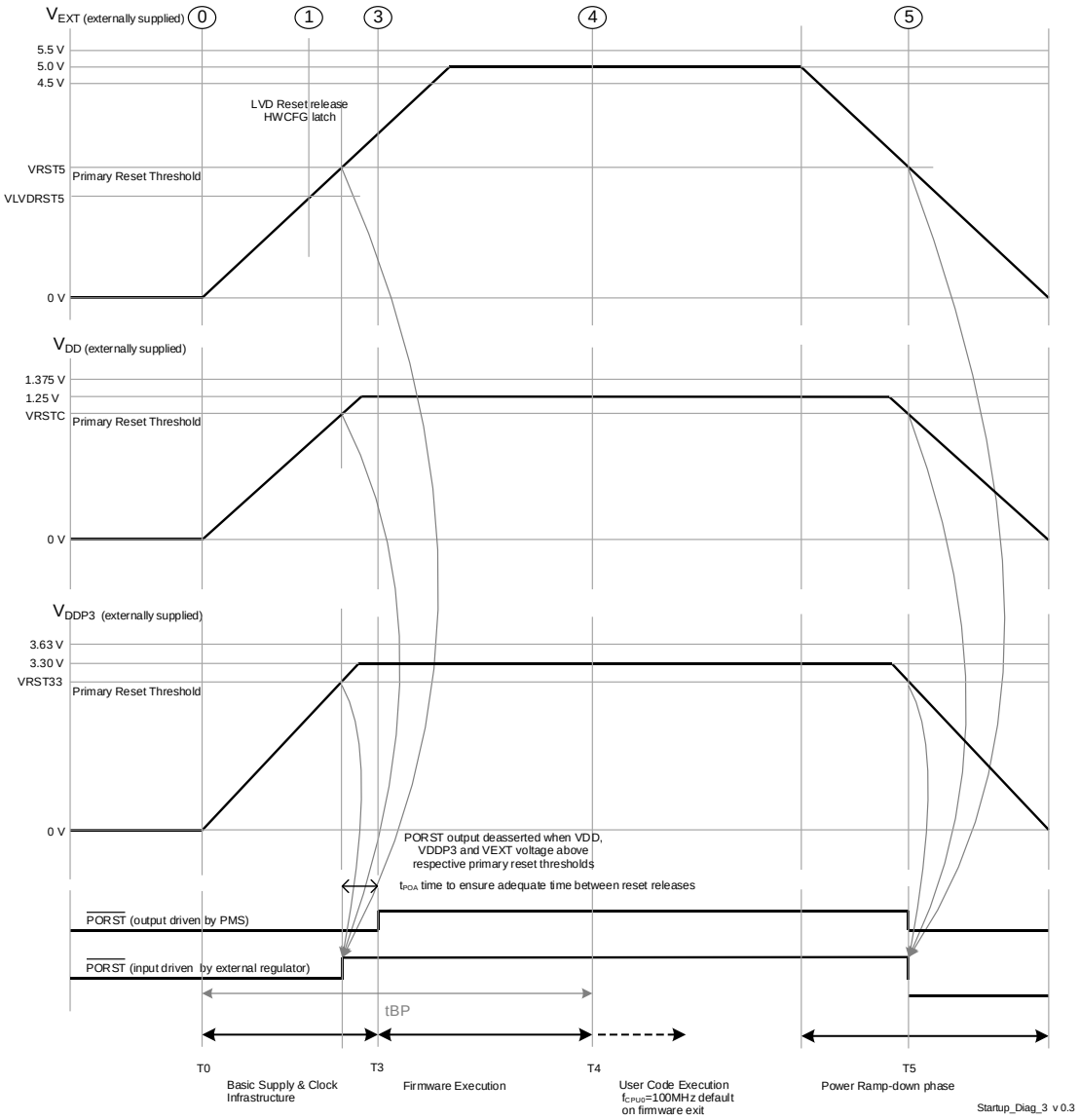
3. Firmware
   - RAM initialization (optional)
   - Startup boot modes….



TC3xx
External Supply Mode Start-up



TC3xx
External Supply Mode Start-up



HWCFG PINs for EVR Startup

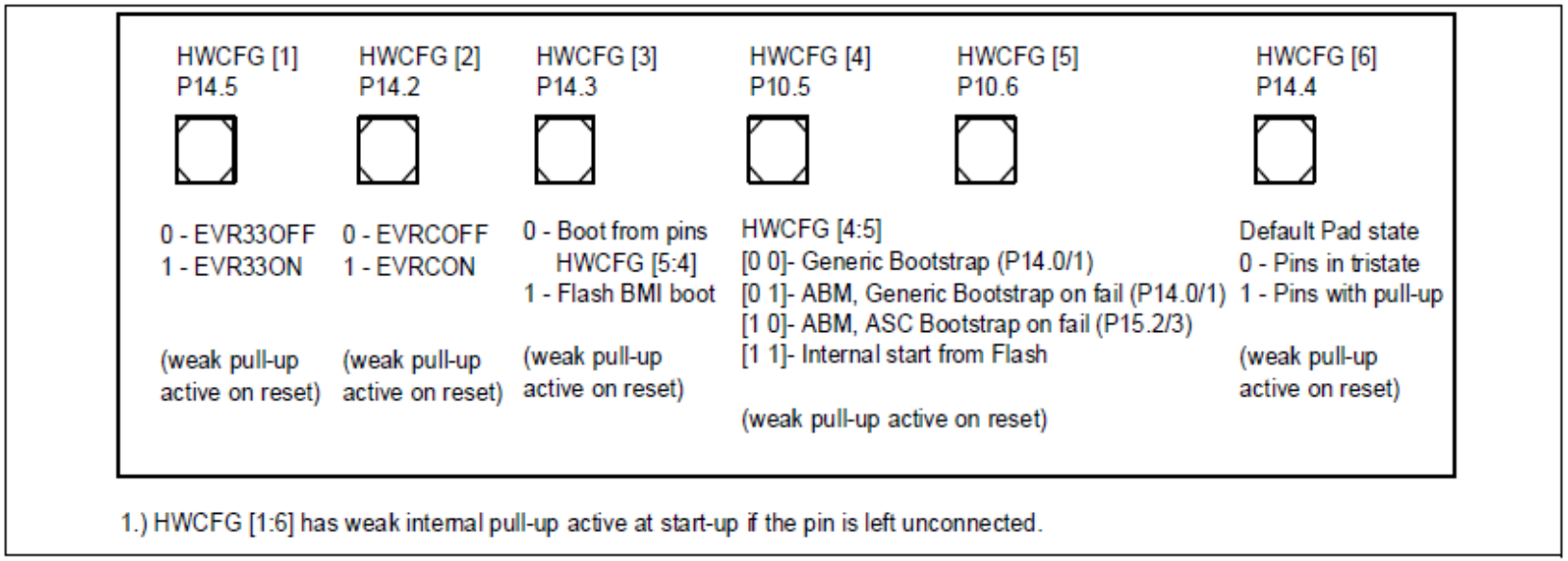


Figure 173 Hardware Configuration (HWCFG) pins

- HWCFG signals are filtered through a spike / glitch filter and are monitored for a constant level over a 28us - 115us nominal debouncing period.
- HWCFG[6] pin is latched during early VEXT supply ramp-up (VEXT < VDDPPA) to decide and set the default reset state of port pins as early as possible.
- HWCFG [1,2,3,6] pins are only latched by the PMS on every initial supply ramp-up
- All HWCFG pins are latched on internal reset release additionally (between 100us – 180us after warm reset assertion) and the status is stored redundantly in STSTAT register by SCU.

Agenda

1. Power Startup
2. Boot Options and Boot Mode Headers
3. Startup Flow Complete Picture

TC3xx Boot Options

- AURIX2G controllers provide following boot modes
  - Internal Start
    - This is usual simple startup mode, where the execution is started from internal flash
    - Start Address is defined by Boot Mode Header (BMHDx.STAD)
  - Bootloader Mode
    - Bootstrap loaders run to download the code to CPU0 PSPR
      - Generic Bootstrap Loader (CAN/ASC) or
      - ASC Bootstrap Loader
  - Alternate Boot Modes (ABM)
    - Application defines the ABM header with Start Address and Memory Area for which CRC is to be checked
    - Execution is started only if required checks are through
    - In case of checks not through, bootstrap loader modes entered

Bootstrap Loader Mode	Channel/node	RxD Line	TxD Line
ASC Bootstrap Loader mode	ASCLIN0	P15.3	P15.2
Generic Bootstrap Loader mode - ASC protocol	ASCLIN0	P14.1	P14.0
Generic Bootstrap Loader mode - CAN protocol	MCMCAN0 module CAN1 node	P14.1	P14.0

TC3xx Boot Mode Selection

- Boot mode is selected depends on the evaluation result of the Boot Mode Headers (BMHDx, x=0...3) after power on.
- If valid BMHD is found,
  - the selected mode is taken either directly from the value configured in BMHD, or according to the state at HWCFG [5:3] pins if pin-configuration is enabled in BMHD.
- If no valid BMHD is found,
  - If HSM boot and Boot Mode Lock is disabled
    - If Halt After Reset request has been received (eg. Debugger)
      - Boot mode is Internal Start from FLASH. (Boot_Addr = PFlash0_Begin)
    - If no Halt After Reset request has been received
      - Boot mode is Generic Bootstrap Loader mode. (Boot_Addr = CPU0_PSPR_begin)
  - If HSM boot or Boot Mode Lock is enabled
    - No Boot Mode is configured and SSW will be in endless loop.
    - If debug interface is not locked by HSM, debugger can still connect to the device to reprogram

BMI Configuration in TC3xx

- The BMI configuration shall not be stored in PFLASH to ensure the boot configuration is not lost every time when PFLASH is erased.
- In AURIX™ TC3xx Platform, four sets of Boot Mode Headers (BMHDx, x=0, 1, 2, 3) are defined in the User Configuration Blocks (UCB) of Data Flash (DFLASH).
- Each set contains an original and copy in UCB_BMHDx_ORIG and UCB_BMHDx_COPY respectively.

Address Range	Size	Unit	Access Type	
			Read	Write
$AF40\ 0000_H$ - $AF40\ 01FF_H$	512 Byte	UCB00 (UCB_BMHD0_ORIG)	Access	SRIBE
$AF40\ 0200_H$ - $AF40\ 03FF_H$	512 Byte	UCB01 (UCB_BMHD1_ORIG)	Access	SRIBE
$AF40\ 0400_H$ - $AF40\ 05FF_H$	512 Byte	UCB02 (UCB_BMHD2_ORIG)	Access	SRIBE
$AF40\ 0600_H$ - $AF40\ 07FF_H$	512 Byte	UCB03 (UCB_BMHD3_ORIG)	Access	SRIBE
$AF40\ 1000_H$ - $AF40\ 11FF_H$	512 Byte	UCB08 (UCB_BMHD0_COPY)	Access	SRIBE
$AF40\ 1200_H$ - $AF40\ 13FF_H$	512 Byte	UCB09 (UCB_BMHD1_COPY)	Access	SRIBE
$AF40\ 1400_H$ - $AF40\ 15FF_H$	512 Byte	UCB10 (UCB_BMHD2_COPY)	Access	SRIBE
$AF40\ 1600_H$ - $AF40\ 17FF_H$	512 Byte	UCB11 (UCB_BMHD3_COPY)	Access	SRIBE

BMHD in DFLASH0_UCB (in TC3xx Memory Map Seg. A)

Seg.	Address Range	Memory Type	
10	0xA0000000 – 0xA02FFFFF	Program Flash 0 (PF0, 3MB),	
	0xA0300000 – 0xA05FFFFF	Program Flash 1 (PF1, 3MB),	
	0xA0600000 – 0xA08FFFFF	Program Flash 2 (PF2, 3MB),	
	0xA0900000 – 0xA0BFFFFF	Program Flash 3 (PF3, 3MB),	
	0xA0C00000 – 0xA0EFFFFF	Program Flash 4 (PF4, 3MB),	
	0xA0F00000 – 0xA0FFFFFF	Program Flash 5 (PF5, 1MB),	
	0xA1000000 – 0xA11FFFFF	Reserved (for PFLASH, 2MB),	
	0xA2000000 – 0xAFFFFFFF	Erase Counter 0 (EC0, 16KB), PFI User Register 0 (PFI0, 256KB), Erase Counter 1 (EC0, 16KB), PFI User Register 1 (PFI0, 256KB), Erase Counter 2 (EC0, 16KB), PFI User Register 2 (PFI0, 256KB), Erase Counter 3 (EC0, 16KB), PFI User Register 3 (PFI0, 256KB), Erase Counter 4 (EC0, 16KB), PFI User Register 4 (PFI0, 256KB), Erase Counter 5 (EC0, 16KB), PFI User Register 5 (PFI0, 256KB),	
	0xAF000000 – 0xAF0FFFFF	Data Flash 0 EEPROM (DF0, 1MB),	
	0xAF400000 – 0xAF405FFF	**Data Flash 0 UCB (DF0, 24KB),**	
	0xAF800000 – 0xAF80FFFF	Data Flash 0 CFS (DF0, 64KB),	
	0xAFC00000 – 0xAFC3FFFF	Data Flash 1 EEPROM (DF1, 128KB),	
	0xAFE00000 – 0xAFFFFFFF	EBU, OLDA, BROM	

UCB_BMHD0_ORIG/COPY
(TC39x B-Step)

Table 205 Register Overview - UCB00 (ascending Offset Address)

Short Name	Long Name	Offset Address
BMI_BMHDID	UCB_BMHD0_ORIG_DATA - Boot Mode Index (BMI) and Boot Mode Header ID (CODE) = B359H	0000_H
STAD	UCB_BMHD0_ORIG_DATA - ABHMDx start address (in case BMI.HWCFG = ABM = 110B) or User Code start address (in case BMI.HWCFG = Flash start = 111B)	0004_H
CRCBMHD	UCB_BMHD0_ORIG_DATA - Check Result for the BMI Header (offset 000H - 007H)	0008_H
CRCBMHD_N	UCB_BMHD0_ORIG_DATA - Inverted Check Result for the BMI Header (offset 000H - 007H)	$000C_H$
PW0	UCB_BMHD0_ORIG_PW - 256-bit password protection, PW0 (least significant)	0100_H
PW1	UCB_BMHD0_ORIG_PW - 256-bit password protection, PW1	0104_H
PW2	UCB_BMHD0_ORIG_PW - 256-bit password protection, PW2	0108_H
PW3	UCB_BMHD0_ORIG_PW - 256-bit password protection, PW3	$010C_H$
PW4	UCB_BMHD0_ORIG_PW - 256-bit password protection, PW4	0110_H
PW5	UCB_BMHD0_ORIG_PW - 256-bit password protection, PW5	0114_H
PW6	UCB_BMHD0_ORIG_PW - 256-bit password protection, PW6	0118_H
PW7	UCB_BMHD0_ORIG_PW - 256-bit password protection, PW7	$011C_H$
CONFIRMATION	UCB_BMHD0_ORIG_CODE - 32-bit CODE	$01F0_H$

UCB_BMHDx_ORIG/COPY (x=1,2,3)
(TC39x B-Step)

Table 206 Register Overview - UCB01 (ascending Offset Address)

Short Name	Long Name	Offset Address
BMI_BMHDID	UCB_BMHD1_ORIG_DATA - Boot Mode Index (BMI) and Boot Mode Header ID (CODE) = B359H	0200_H
STAD	UCB_BMHD1_ORIG_DATA - ABHMDx start address (in case BMI.HWCFG = ABM = 110B) or User Code start address (in case BMI.HWCFG = Flash start = 111B)	0204_H
CRCBMHD	UCB_BMHD1_ORIG_DATA - Check Result for the BMI Header (offset 000H - 007H)	0208_H
CRCBMHD_N	UCB_BMHD1_ORIG_DATA - Inverted Check Result for the BMI Header (offset 000H - 007H)	$020C_H$
CONFIRMATION	UCB_BMHD1_ORIG_CODE - 32-bit CODE	$03F0_H$

BMI – Boot Mode Index

Bit Field	Bit Field Name	Description
bit [0]	PINDIS	**Mode selection by configuration pins**: **0B** Mode selection by HWCFG pins is enabled **1B** Mode selection by HWCFG pins is disabled
bits [3:1]	HWCFG	**Start-up mode selection**: **111B** Internal start from Flash **110B** Alternate Boot Mode (ABM) **100B** Generic Bootstrap Loader Mode (ASC/CAN BSL) **011B** ASC Bootstrap Loader Mode (ASC BSL) Else invalid
bit [4]	LSENA0	**Lockstep monitoring control by SSW for CPU0**: **0B** Lockstep monitoring for CPU0 is disabled **1B** Lockstep monitoring for CPU0 is enabled
bit [5]	LSENA1	**Lockstep monitoring control by SSW for CPU1**: 1) **0B** Lockstep monitoring for CPU1 is disabled **1B** Lockstep monitoring for CPU1 is enabled

1) Only if the respective CPUx Lockstep functionality is available on the product, otherwise the bit is Reserved, must be configured to 0 in UCB_BMHDx

BMI – Boot Mode Index

Bit Field	Bit Field Name	Description
bit [6]	LSENA2	**Lockstep monitoring control by SSW for CPU2**: 1) 0B Lockstep monitoring for CPU2 is disabled 1B Lockstep monitoring for CPU2 is enabled
bit [7]	LSENA3	**Lockstep monitoring control by SSW for CPU3**: 1) **0B** Lockstep monitoring for CPU3 is disabled **1B** Lockstep monitoring for CPU3 is enabled
bit [8]	LBISTENA	**LBIST execution start by SSW**: **0B** LBIST execution start by SSW is disabled **1B** LBIST execution start by SSW upon cold power-on is enabled
bits [11:9]	CHSWENA	**Checker Software (CHSW) execution after SSW**: 2) **101B** CHSW execution after SSW is disabled **else** CHSW execution after SSW is enabled
bits [15:12]	reserved	**Reserved** for future extensions, **must be configured to 0** in UCB_BMHDx

1) Only if the respective CPUx Lockstep functionality is available on the product, otherwise the bit is Reserved, must be configured to 0 in UCB_BMHDx
2) This bitfield is not evaluated during the SSW flow but by the Checker Software

UCB Confirmation Code

Table 183 UCB States

State	Value	Description
UNLOCKED	$4321\ 1234_H$	**Delivery State** The UCB confirmation code is programmed with the UNLOCKED value.
CONFIRMED	$57B5\ 327F_H$	**Operational State** The UCB confirmation code is programmed with the CONFIRMED value. *Note: The UNLOCKED value can be over programmed with the CONFIRMED value.*
ERASED	$0000\ 0000_H$	**Erased State** Behavior as for the ERRORED state.
ERRORED	Others	**Errored State** The UCB confirmation code stored is not the CONFIRMED or UNLOCKED value.

- Since Erased state (0x0) is considered ERRORED state, transition from UNLOCKED state (0x43211234) to CONFIRMED state (0x57B5327F) can be done without erasing the UCB. This over-programming is allowed but the 4 bytes following the confirmation code must be kept 0x00000000 in the unlocked state and in the over-programmed data.
- The data stored in the ORIG and COPY of a UCB pair should be identical. If need to change these two UCB data,
  - Firstly, erase COPY UCB under ORIG and COPY UCB all CONFIRMED state, then program COPY UCB with CONFIRMED state.
  - Secondly, Erase ORIG UCB, then program ORIG UCB with CONFIRMED state.

Boot Mode Header Installation

- Boot Mode Header Installation is dependent on the confirmation states of UCB_BMHDx_ORIG and UCB_BMHDx_COPY.
- If the confirmation code of both ORIG and COPY is **ERRORED**, SSW does not evaluate the UCB.
- Confirmation State Indication is in the DMU register HF_CONFIRM0.
  - The confirmation state of UCB_BMHDx_ORIG (x=0-3) is indicated by DMU_HF_CONFIRM0.PROINBMHDxO.
  - The confirmation state of UCB_BMHDx_COPY (x=0-3) is indicated by DMU_HF_CONFIRM0.PROINBMHDxC.

Table 184 Boot Mode Header 0 Installation

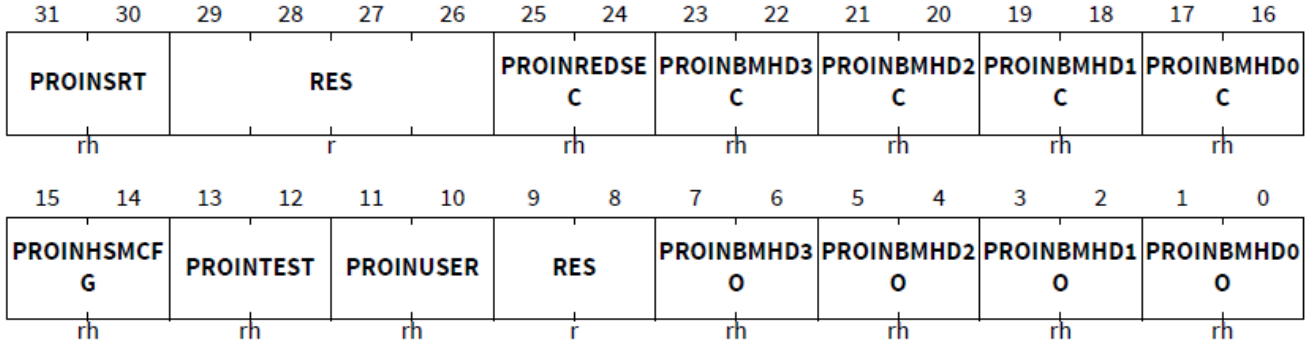
UCB_BMHD0_ORIG Confirmation State	UCB_BMHD0_COPY Confirmation State	Boot Mode Header Installation
UNREAD	Don't Care	No evaluation.
UNLOCKED	Don't Care	SSW evaluates UCB_BMHD0_ORIG. Password installed from UCB_BMHD0_ORIG.
CONFIRMED	Don't Care	SSW evaluates UCB_BMHD0_ORIG. Password installed from UCB_BMHD0_ORIG.
ERRORED	**UNLOCKED**	SSW evaluates UCB_BMHD0_COPY. Password installed from UCB_BMHD0_COPY.
ERRORED	**CONFIRMED**	SSW evaluates UCB_BMHD0_COPY. Password installed from UCB_BMHD0_COPY.
ERRORED	**ERRORED**	No evaluation. No Password installed.

Table 185 Boot Mode Header x Installation(x= 1 - 3)

UCB_BMHDx_ORIG Confirmation State	UCB_BMHDx_COPY Confirmation State	Boot Mode Header Installation
UNREAD	Don't Care	No evaluation.
UNLOCKED	Don't Care	SSW evaluates UCB_BMHDx_ORIG.
CONFIRMED	Don't Care	SSW evaluates UCB_BMHDx_ORIG.
ERRORED	**UNLOCKED**	SSW evaluates UCB_BMHDx_COPY.
ERRORED	**CONFIRMED**	SSW evaluates UCB_BMHDx_COPY.
ERRORED	**ERRORED**	No evaluation.

Confirmation State Indication in FLASH Confirm Status Register 0

HF_CONFIRM0
Flash Confirm Status Register 0 **(0000020_H)** **Application Reset Value: $0000\ 0000_H$**



Field	Bits	Type	Description
PROINBMHD0O	1:0	rh	**UCB_BMHD0_ORIG Confirmation** This bit reflects the confirmed state of UCB_BMHD0_ORIG. 00_B UCB_BMHD0_ORIG state is UNREAD. 01_B UCB_BMHD0_ORIG state is UNLOCKED. 10_B UCB_BMHD0_ORIG state is CONFIRMED. 11_B UCB_BMHD0_ORIG state is ERRORED.
PROINBMHD1O	3:2	rh	**UCB_BMHD1_ORIG Confirmation** This bit reflects the confirmed state of UCB_BMHD1_ORIG. 00_B UCB_BMHD1_ORIG state is UNREAD. 01_B UCB_BMHD1_ORIG state is UNLOCKED. 10_B UCB_BMHD1_ORIG state is CONFIRMED. 11_B UCB_BMHD1_ORIG state is ERRORED.
PROINBMHD2O	5:4	rh	**UCB_BMHD2_ORIG Confirmation** This bit reflects the confirmed state of UCB_BMHD2_ORIG. 00_B UCB_BMHD2_ORIG state is UNREAD. 01_B UCB_BMHD2_ORIG state is UNLOCKED. 10_B UCB_BMHD2_ORIG state is CONFIRMED. 11_B UCB_BMHD2_ORIG state is ERRORED.
PROINBMHD3O	7:6	rh	**UCB_BMHD3_ORIG Confirmation** This bit reflects the confirmed state of UCB_BMHD3_ORIG. 00_B UCB_BMHD3_ORIG state is UNREAD. 01_B UCB_BMHD3_ORIG state is UNLOCKED. 10_B UCB_BMHD3_ORIG state is CONFIRMED. 11_B UCB_BMHD3_ORIG state is ERRORED.

Field	Bits	Type	Description
PROINBMHD0C	17:16	rh	**UCB_BMHD0_COPY Confirmation** This bit reflects the confirmed state of UCB_BMHD0_COPY. 00_B UCB_BMHD0_COPY state is UNREAD. 01_B UCB_BMHD0_COPY state is UNLOCKED. 10_B UCB_BMHD0_COPY state is CONFIRMED. 11_B UCB_BMHD0_COPY state is ERRORED.
PROINBMHD1C	19:18	rh	**UCB_BMHD1_COPY Confirmation** This bit reflects the confirmed state of UCB_BMHD1_COPY. 00_B UCB_BMHD1_COPY state is UNREAD. 01_B UCB_BMHD1_COPY state is UNLOCKED. 10_B UCB_BMHD1_COPY state is CONFIRMED. 11_B UCB_BMHD1_COPY state is ERRORED.
PROINBMHD2C	21:20	rh	**UCB_BMHD2_COPY Confirmation** This bit reflects the confirmed state of UCB_BMHD2_COPY. 00_B UCB_BMHD2_COPY state is UNREAD. 01_B UCB_BMHD2_COPY state is UNLOCKED. 10_B UCB_BMHD2_COPY state is CONFIRMED. 11_B UCB_BMHD2_COPY state is ERRORED.
PROINBMHD3C	23:22	rh	**UCB_BMHD3_COPY Confirmation** This bit reflects the confirmed state of UCB_BMHD3_COPY. 00_B UCB_BMHD3_COPY state is UNREAD. 01_B UCB_BMHD3_COPY state is UNLOCKED. 10_B UCB_BMHD3_COPY state is CONFIRMED. 11_B UCB_BMHD3_COPY state is ERRORED.

BMI and ABM Headers Configurable Addresses

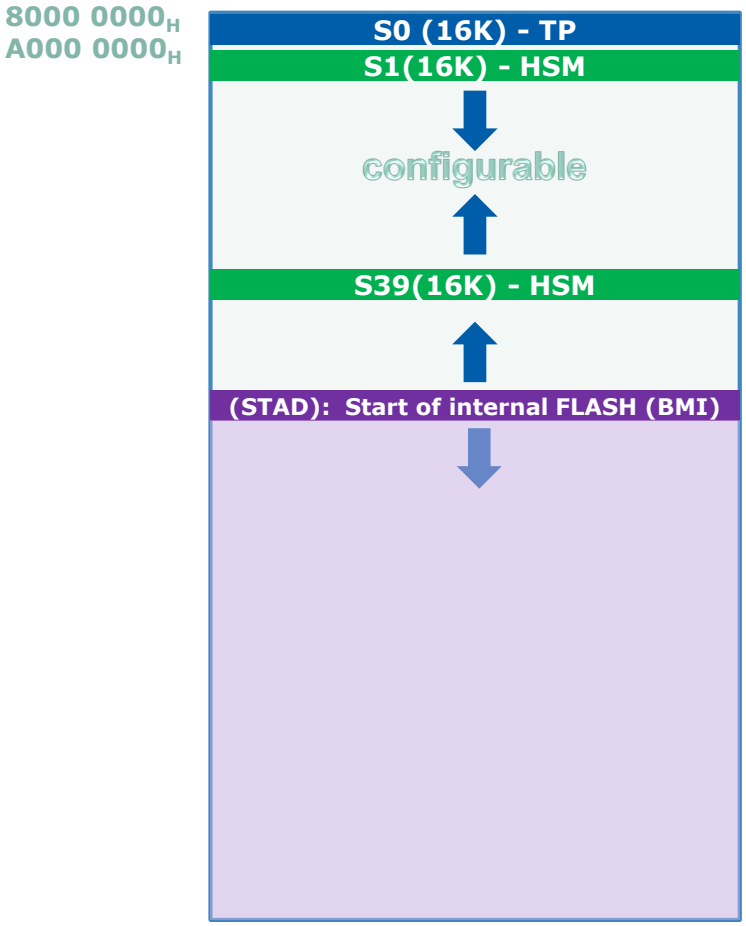
- User start addresses shall be configurable in both **ABM** and **Internal Start from Flash** modes.
- In case of Internal Start from Flash, it is proposed to have configurable start address in BMI.STAD.
- In case of **A**lternate **B**oot **M**ode , Start address is configured in ABM.STADBM like before. It is proposed to have configurable ABM start address stored in BMI.STAD
- STAD must be a valid word-aligned address inside PFLASH.
- This allows flexibility for locating of headers in Flash to avoid non used areas in Flash owing to fixed address. This enables contiguous memory blocks and better management of logistics/ variant management data.

UCB00	Address Range : $AF40\ 0000_H$ - $AF40\ 01FF_H$		Size = 512 Bytes
Offset Addr.	Size (Byte)	BMHDn (Bitfields)	BMI Header Description
000_H	2	BMI	Boot Mode Index (BMI)
			PINDIS[3] : Mode selection by configuration pins: 0 Mode selection by HWCFG pins is enabled 1 Mode selection by HWCFG pins is disabled
			HWCFG [6:4]: Start-up mode selection: 111_B Internal start from Flash 110_B Alternate Boot Mode (ABM) 100_B Generic Bootstrap Loader 011_B ASC Bootstrap Loader
002_H	2	BMHDID	Boot Mode Header ID (Confirmation code) = $B359_H$
004_H	4	STAD	ABMHDx Start address (in case BMI.HWCFG=ABM=110_H) User Code Start address (in case BMI.HWCFG=Flash Start=111_H)
008_H	4	CRCBMHD	Check Result for the BMI Header (offset 00_H..07_H)
$00C_H$	4	CRCBMHD_N	Inverted Check Result for the BMI Header
$1F0_H$	4	Confirmation	32 bit confirmation code

ABMHD0	Address Range : $STAD0_H$ - $(STAD0 + 1F)_H$		Size = 32 Bytes
Offset Addr.	Size (Byte)	ABMHDn	ABM Header in Pflash Description
00H	4	STADABM	User Code Start Address in ABM mode
04H	4	ABMHDID	ABM Header ID (Confirmation code) = $xxxx_H$
08H	4	CHKSTART	Memory Range to be checked - Start Address
0CH	4	CHKEND	Memory Range to be checked - End Address
10H	4	CRCRANGE	Check Result for the Memory Range
14H	4	CRCRANGE_N	Inverted Check Result for the Memory Range
18H	4	CRCABMHD	Check Result for the ABM Header (offset 00H..13H)
1CH	4	CRCABMHD_N	Inverted Check Result for the ABM Header

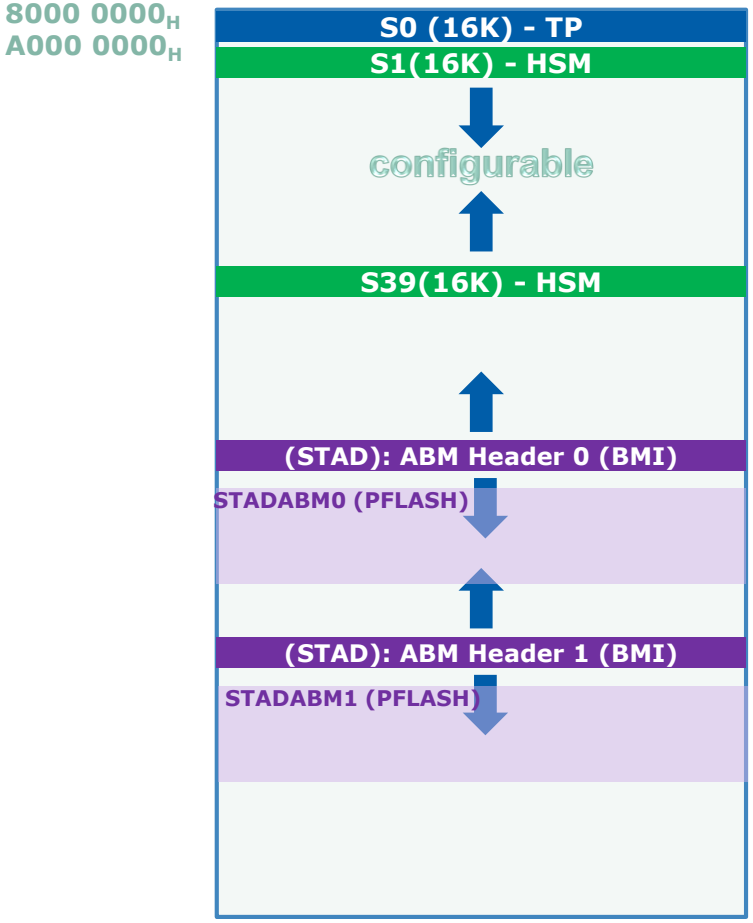
BMI and ABM Headers Configurable Addresses

› Internal Start From FLASH



PFLASH (lower 1/2/3 MB)

› Alternate Boot Mode



PFLASH (lower 1/2/3 MB)

2 BMI headers in UCB

BMIHD0C - UCB08 (Redundant Copy) Address Range : $AF40\ 1000_H$ - AF40

BMIHD0 - UCB00 Address Range : $AF40\ 0000_H$ - $AF40\ 01FF_H$ Size = 512 Bytes

Offset Addr.	Size (Byte)	BMHD0 (Bitfields)	BMI Header Description
000_H	2	BMI	Boot Mode Index (BMI) PINDIS[3] : Mode selection by configuration pins: 0 Mode selection by HWCFG pins is enabled 1 Mode selection by HWCFG pins is disabled HWCFG [6:4]: Start-up mode selection: 111_B Internal start from Flash 110_B Alternate Boot Mode (ABM) 100_B Generic Bootstrap Loader 011_B ASC Bootstrap Loader
002_H	2	BMHDID	Boot Mode Header ID (Confirmation code) = $B359_H$
004_H	4	STAD	ABMHDx Start address (in case BMI.HWCFG = ABM = 110 $_H$) User Code Start address (in case BMI.HWCFG = Flash Start = 111 $_H$)
008_H	4	CRCBMHD	Check Result for the BMI Header (offset 00_H..07_H)
$00C_H$	4	CRCBMHD_N	Inverted Check Result for the BMI Header
$1F0_H$	4	Confirmation	32 bit confirmation code

BMIHD1C (Redundant Copy) Address Range : $AF40\ 0200_H$ - $AF40\ 03FF_H$

BMIHD1 - UCB01 Address Range : $AF40\ 0200_H$ - $AF40\ 03FF_H$ Size = 512 Bytes

Offset Addr.	Size (Byte)	BMHD1 (Bitfields)	BMI Header Description
000_H	2	BMI	Boot Mode Index (BMI) PINDIS[3] : Mode selection by configuration pins: 0 Mode selection by HWCFG pins is enabled 1 Mode selection by HWCFG pins is disabled HWCFG [6:4]: Start-up mode selection: 111_B Internal start from Flash 110_B Alternate Boot Mode (ABM) 100_B Generic Bootstrap Loader 011_B ASC Bootstrap Loader
002_H	2	BMHDID	Boot Mode Header ID (Confirmation code) = $B359_H$
004_H	4	STAD	ABMHDx Start address (in case BMI.HWCFG = ABM = 110 $_H$) User Code Start address (in case BMI.HWCFG = Flash Start = 111 $_H$)
008_H	4	CRCBMHD	Check Result for the BMI Header (offset 00_H..07_H)
$00C_H$	4	CRCBMHD_N	Inverted Check Result for the BMI Header
$1F0_H$	4	Confirmation	32 bit confirmation code

Check ID,CRC, BMI,STAD

2 ABM headers in Flash

ABMHD0 Address Range : $STAD_H$ - $(STAD + 1F)_H$ Size = 32 Bytes

Offset Addr.	Size (Byte)	ABMHD0	ABM Header in Pflash Description
00H	4	STADABM	User Code Start Address in ABM mode
04H	4	ABMHDID	ABM Header ID (Confirmation code) = $xxxx_H$
08H	4	CHKSTART	Memory Range to be checked - Start Address
0CH	4	CHKEND	Memory Range to be checked - End Address
10H	4	CRCRANGE	Check Result for the Memory Range
14H	4	CRCRANGE_N	Inverted Check Result for the Memory Range
18H	4	CRCABMHD	Check Result for the ABM Header (offset 00H..13H)
1CH	4	CRCABMHD_N	Inverted Check Result for the ABM Header

User Code Start address = STADABM in case of ABM mode
= STAD in case of Internal Start from Flash

Check header & code CRC

ABMHD1 Address Range : $STAD_H$ - $(STAD + 1F)_H$ Size = 32 Bytes

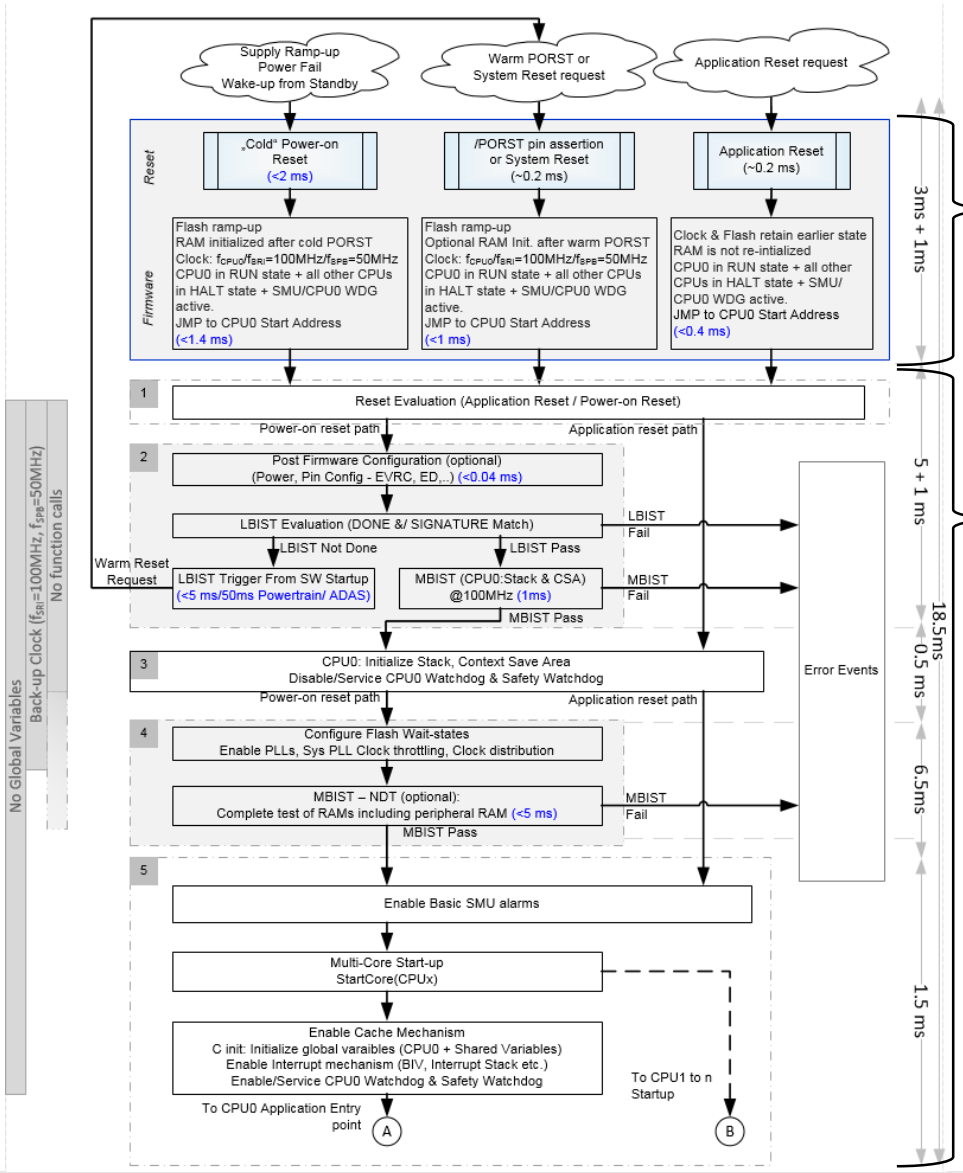
Offset Addr.	Size (Byte)	ABMHD1	ABM Header in Pflash Description
00H	4	STADABM	User Code Start Address in ABM mode
04H	4	ABMHDID	ABM Header ID (Confirmation code) = $xxxx_H$
08H	4	CHKSTART	Memory Range to be checked - Start Address
0CH	4	CHKEND	Memory Range to be checked - End Address
10H	4	CRCRANGE	Check Result for the Memory Range
14H	4	CRCRANGE_N	Inverted Check Result for the Memory Range
18H	4	CRCABMHD	Check Result for the ABM Header (offset 00H..13H)
1CH	4	CRCABMHD_N	Inverted Check Result for the ABM Header

User Code Start address = STADABM in case of ABM mode
= STAD in case of Internal Start from Flash

Agenda

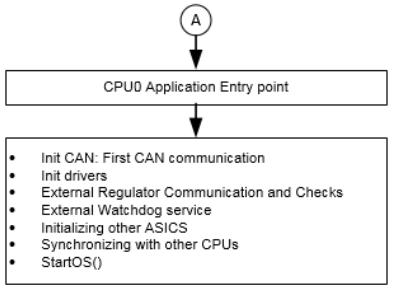
1. Power Startup
2. Boot Options and Boot Mode Headers
3. Startup Flow Complete Picture

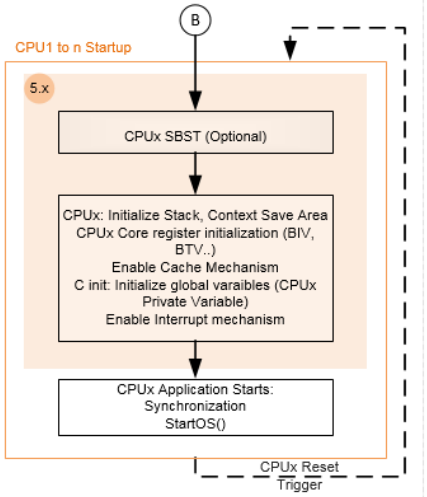
Startup Flow: Complete Picture



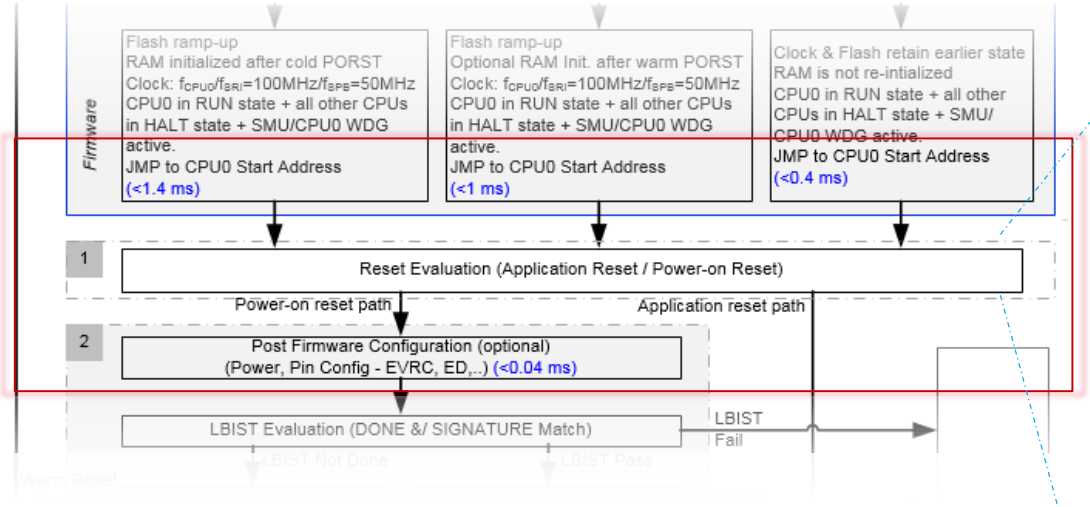
Startup Firmware:
- Non-programmable
- Configurable

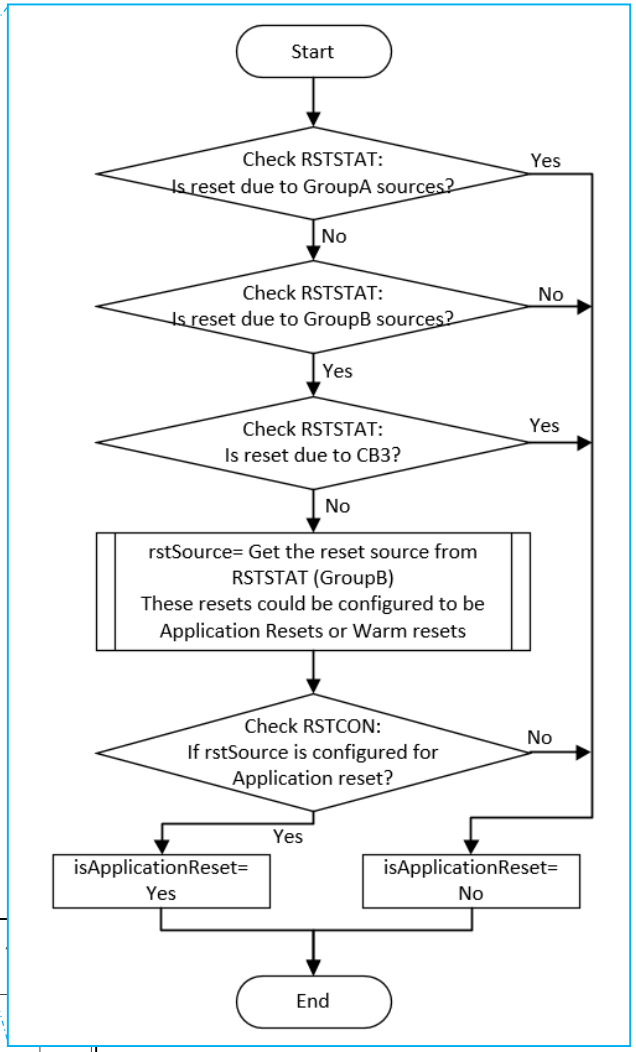
Startup Software:
- Programmable



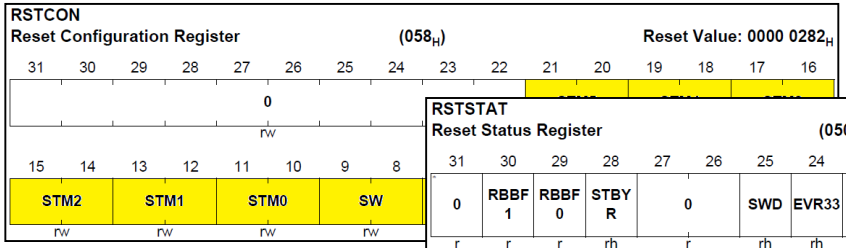


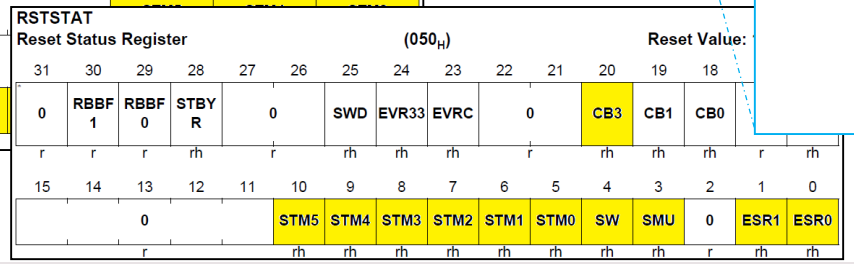
Startup Software: Phase 1
Reset Evaluation (lite)



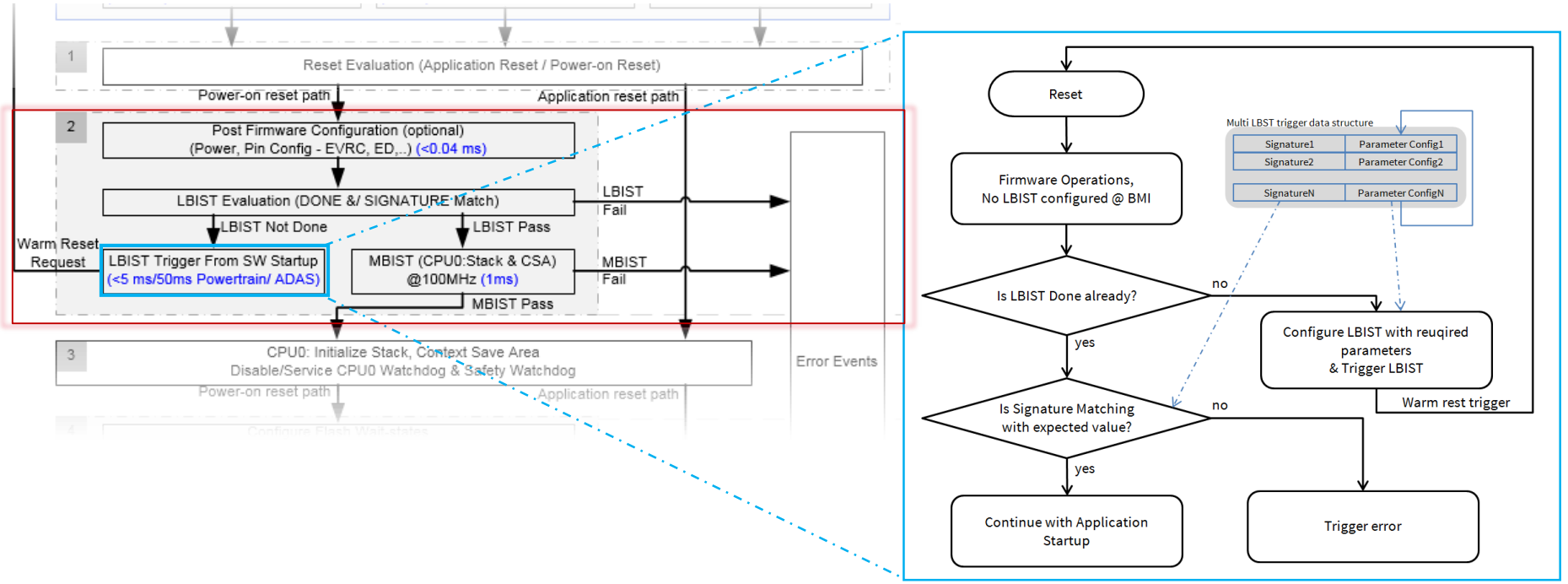


- Quick reset evaluation is done to know if last reset was Application reset OR System Reset:
  - RSTSTAT register is not cleared after this
  - Allows the detailed reset evaluation later
  - Startup flow takes different paths based on the result



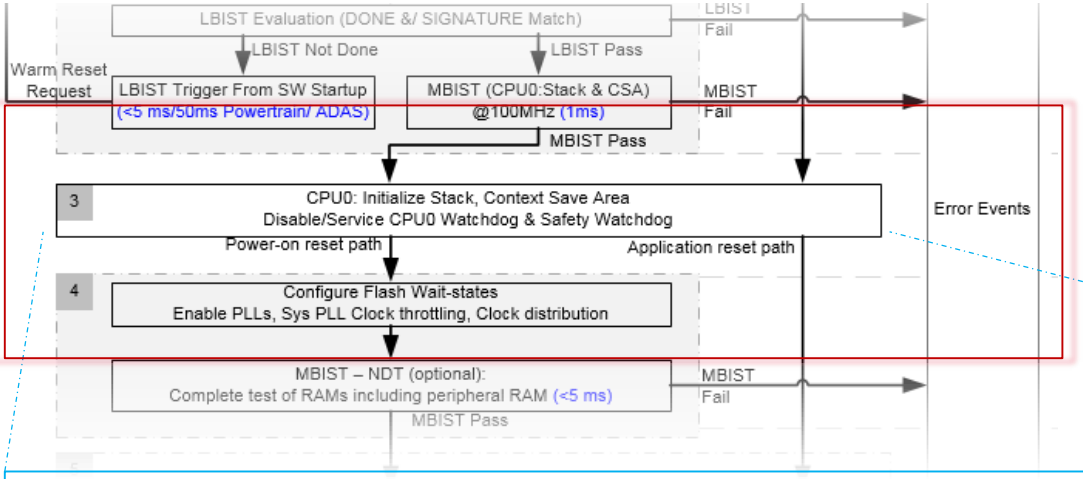


Startup Software: Phase 2
PMS, LBIST, MBIST Handling



- This part is executed only if the last reset was power-on reset
- At the beginning of this phase power supply related configurations are done
- If LBIST is started last time, which resulted in a warm power reset. This phase is being executed again.
- A call back to application is triggered in case of LBIST error
- An MBIST is triggered for the CPU0 Stack and CSA (at 100MHz SPB)

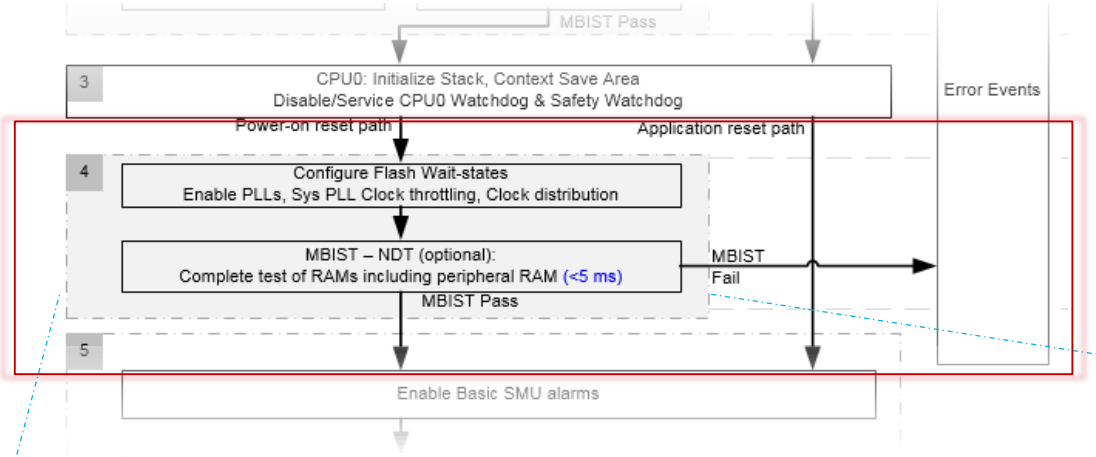
Startup Software: Phase 3
Enabling Function Calls @ CPU0

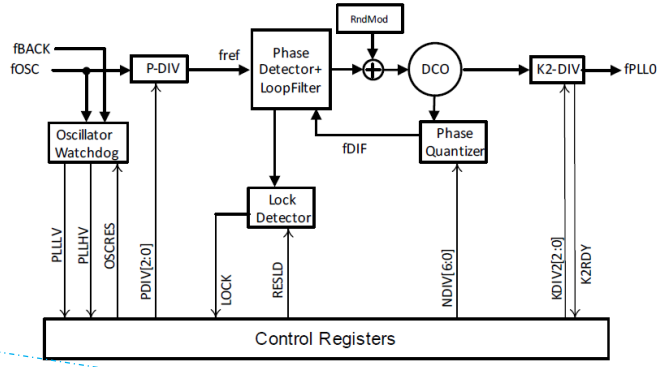


- This phase is executed for all types of resets
  - Initialization of CPU0 USTACK (user stack pointer)
  - Establish the linked lists of CPU0 CSA (context save area)
  - Handling CPU0 Watchdog and Safety Watchdog for the first time

- Note:
  - Function calls are allowed from this point of time. However Stack and CSA are not completely tested.
  - Global Variables shall not be used yet

Startup Software: Phase 4
PLL and Clocks





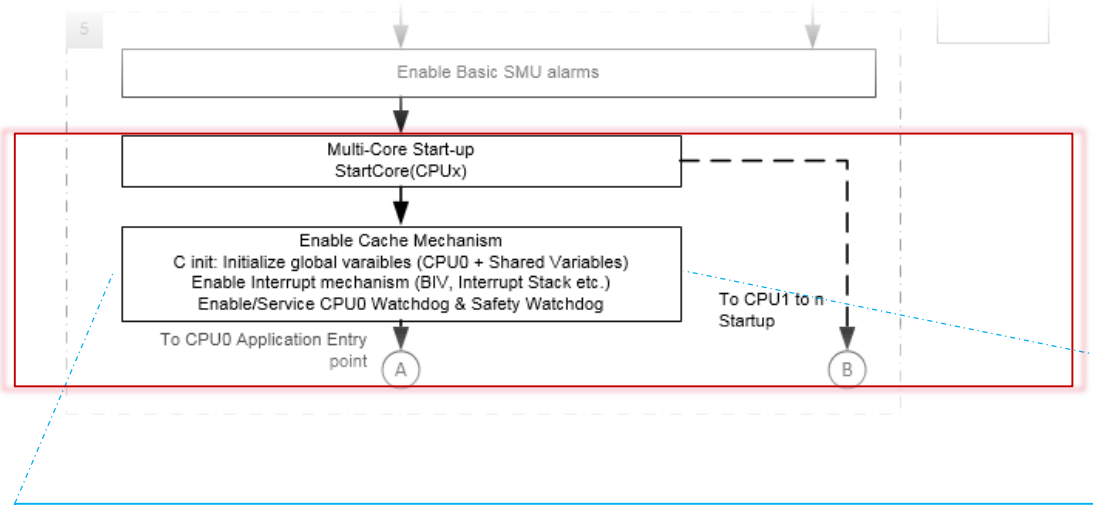
- Configure Flash Wait States (suitable for highest CPU frequency)
- Configure the System PLL and Peripheral PLL
  - Lock Both PLLs for required P and N divider values
  - Setup the clock distribution values
  - Throttle the Sys PLL (K2 divider) to required higher frequency.
    - ~100mA with 100uS settling time
- Configure the MBIST for entire RAM area with NDT (NDIT)
  - In case of error, trigger the error handling at application with call back
  - This part of the software shall not use any RAM (not even Stack and CSA)

Startup Software: Phase 5 Multicore Startup

- Starting each CPU introduces current jump of 20~100mA
- This requires the power supply settling time of >100uS
- Multicore Startup is done as chain sequence, where any CPU enables only the next enabled CPU in the chain.
- With this CPU0 could continue with its initializations.

- Cascading the startup of other CPUs saves time
- Or start remaining CPUs from CPU0 (would utilize some part of the wait time)

Startup Software: Phase 5
C Initialization



- Enable Cache Mechanism
- Initialize CPU0 “Attached” Global Variables, and CPU-Shared variables
- Enable interrupt and traps mechanism
  - Enable interrupts and vector tables
  - Initialize the vector tables (if required)

Part of your life. Part of tomorrow.

Infineon