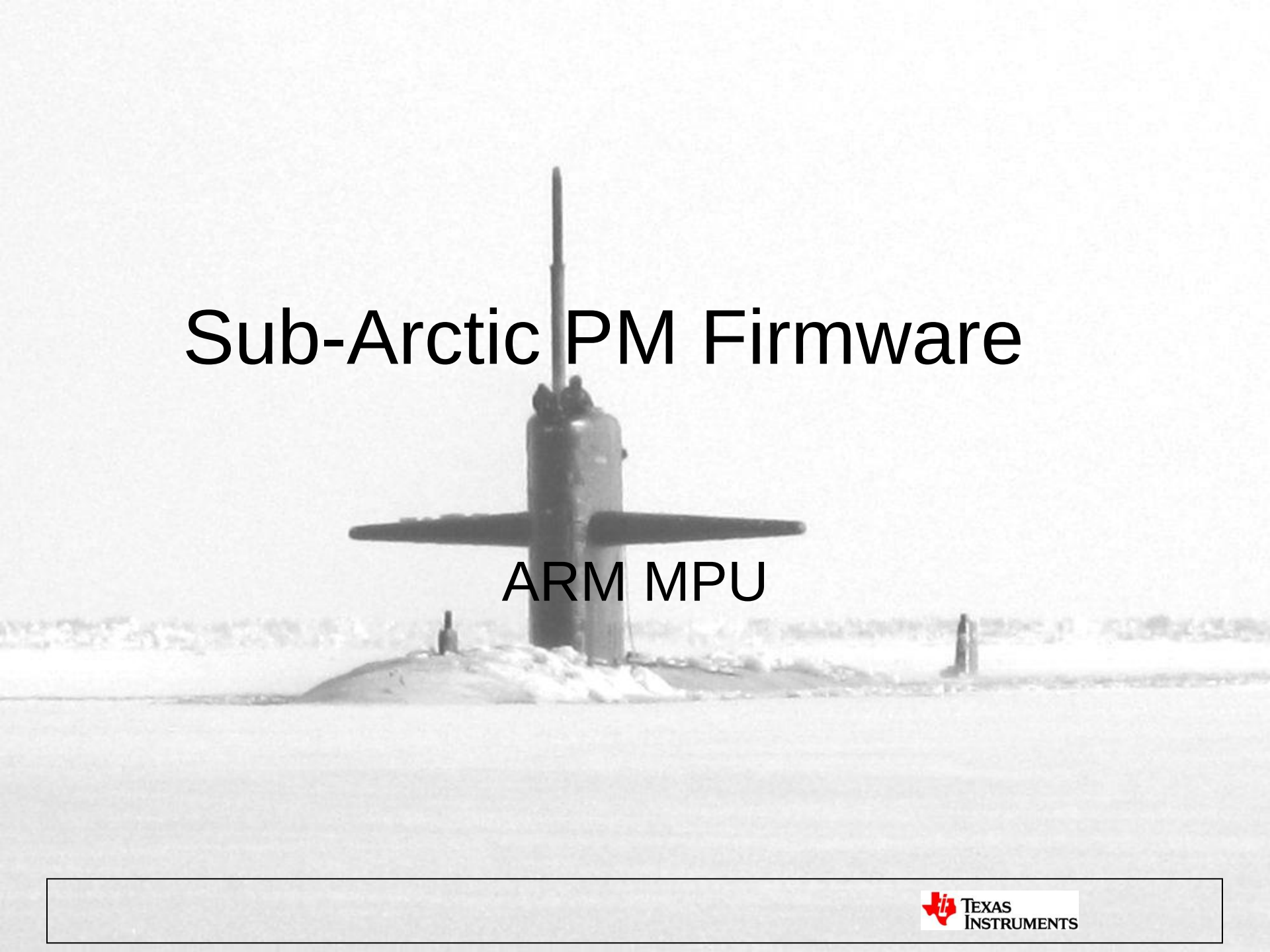


Sub-Arctic PM Firmware

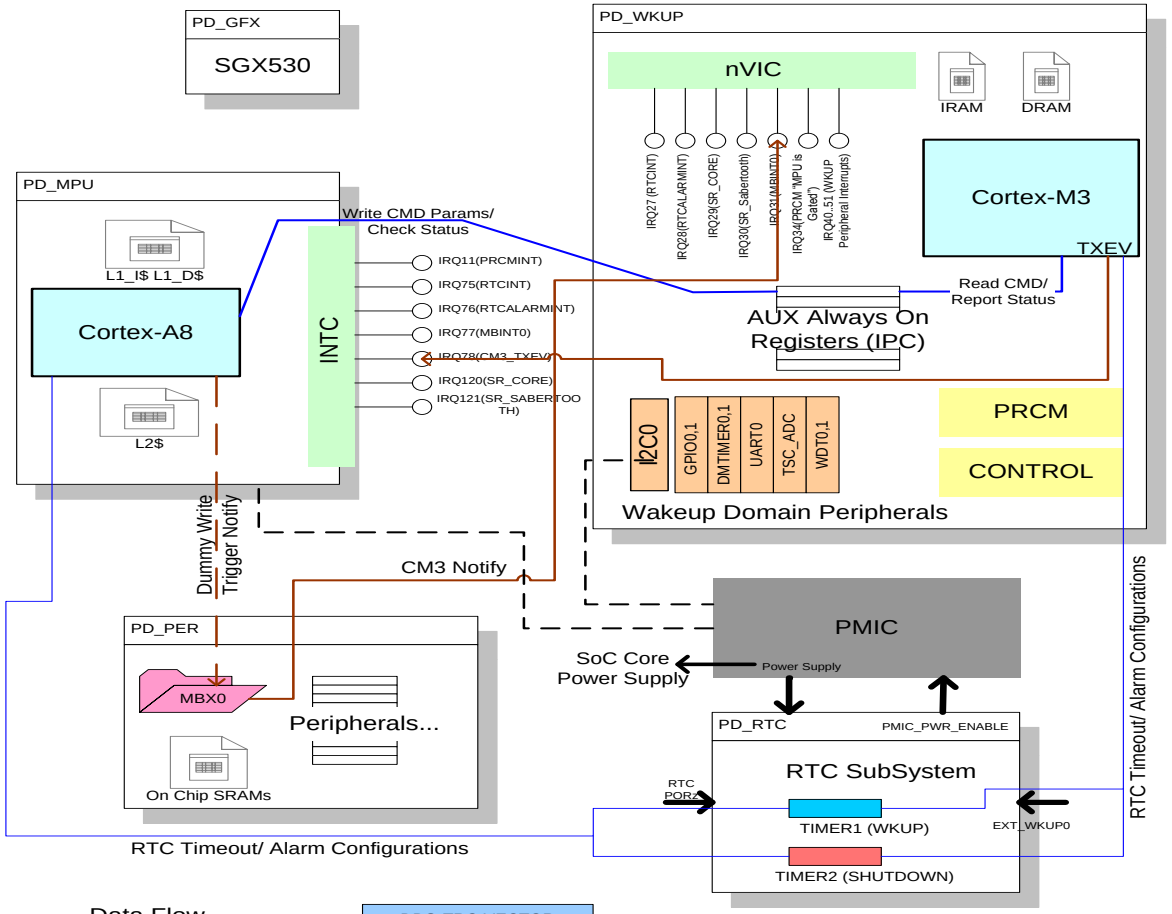
ARM MPU

Texas Instruments

Agenda

- Overview of PM features in SA
  - Interaction between different IPs
- CM3 firmware
  - Use of auxiliary register for message passing
  - Notification mechanism
  - Data structures for IPC in different low power states
- Firmware loading procedure
  - Linux as an example
- Example - Entering RTC mode

PM FW System Context…



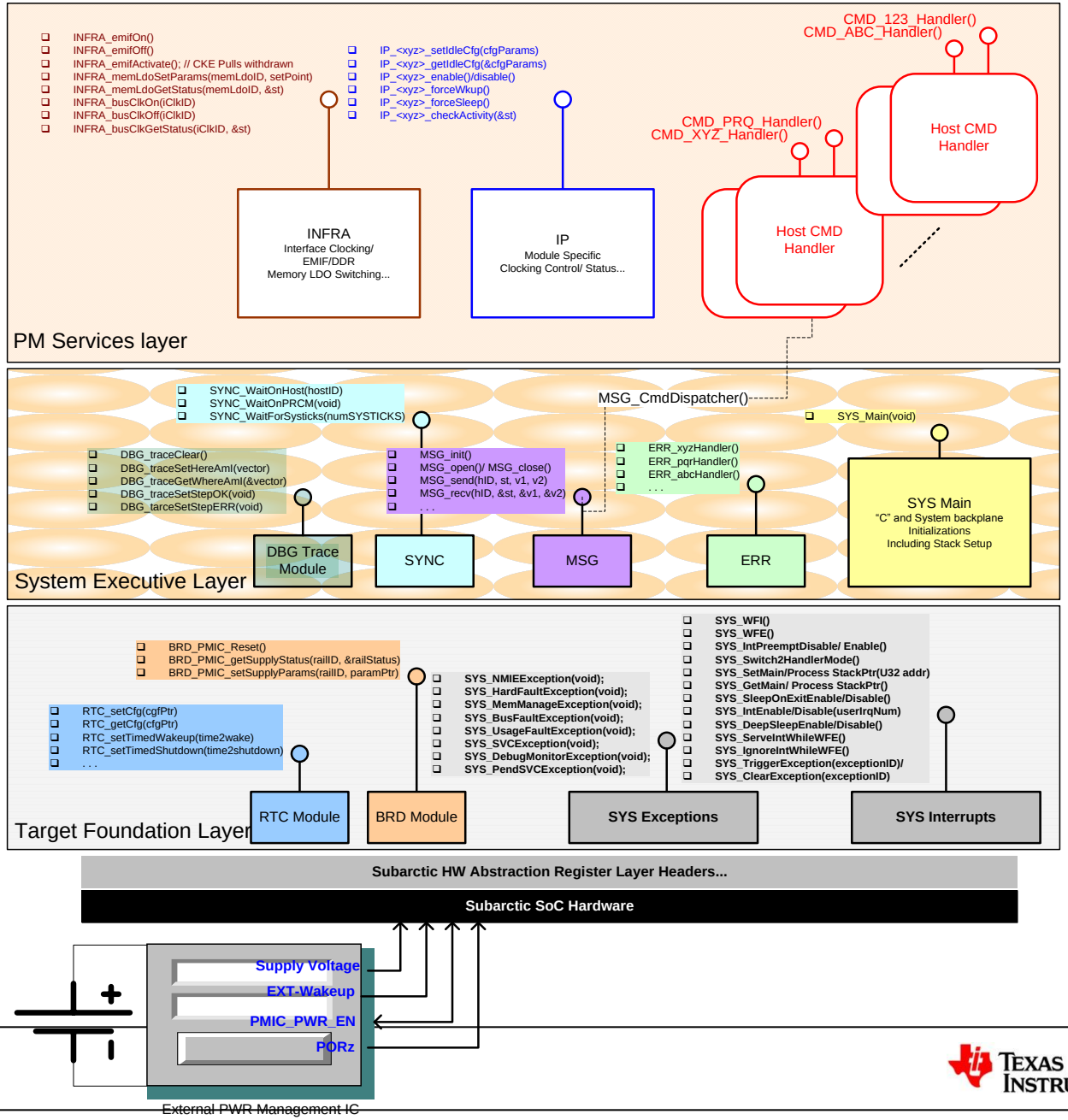
Data Flow
Control Flow

DBG-TRC-VECTOR
CMD/ STATUS
PARAMS
PARAMS
RESERVED
RESERVED
RESERVED
RESERVED FOR CUSTOMER USE

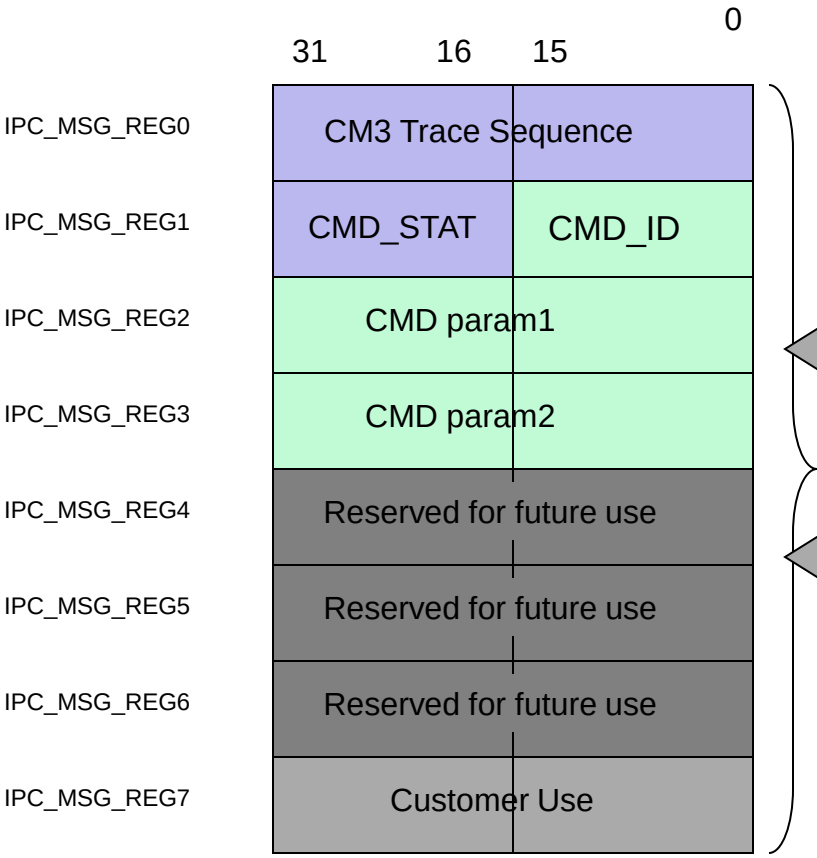
IPC Registers Use...

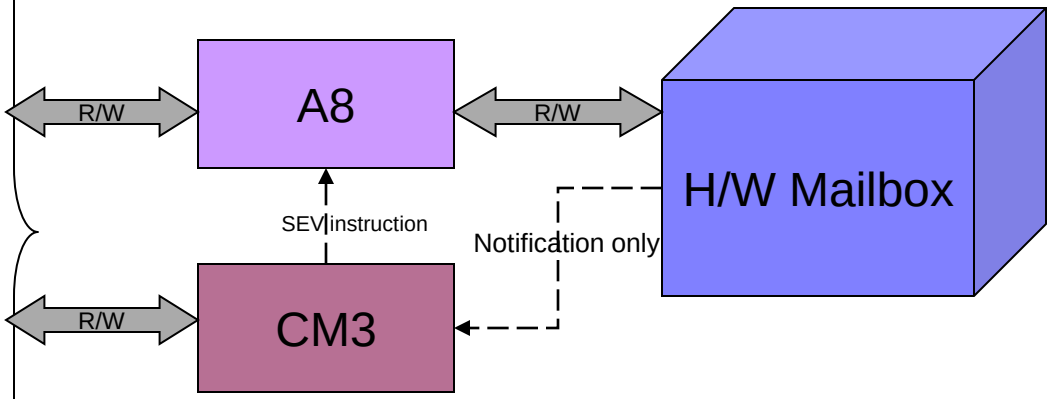
Notes:
1) SA Wakeup CM3 services only single Host viz., Cortex-A8
2) Some of RESERVED registers are TBD Use finalization during implementation.
3) SA Next follow-on device might require Wakeup CM3 to also service a CM3 Controller in it's independent power domain. Params location for exchange w/ this b/w GP and HS devices is TBD finalization.

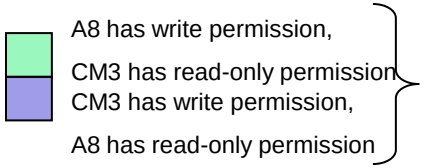
PM FW System Decomposition…



IPC mechanism



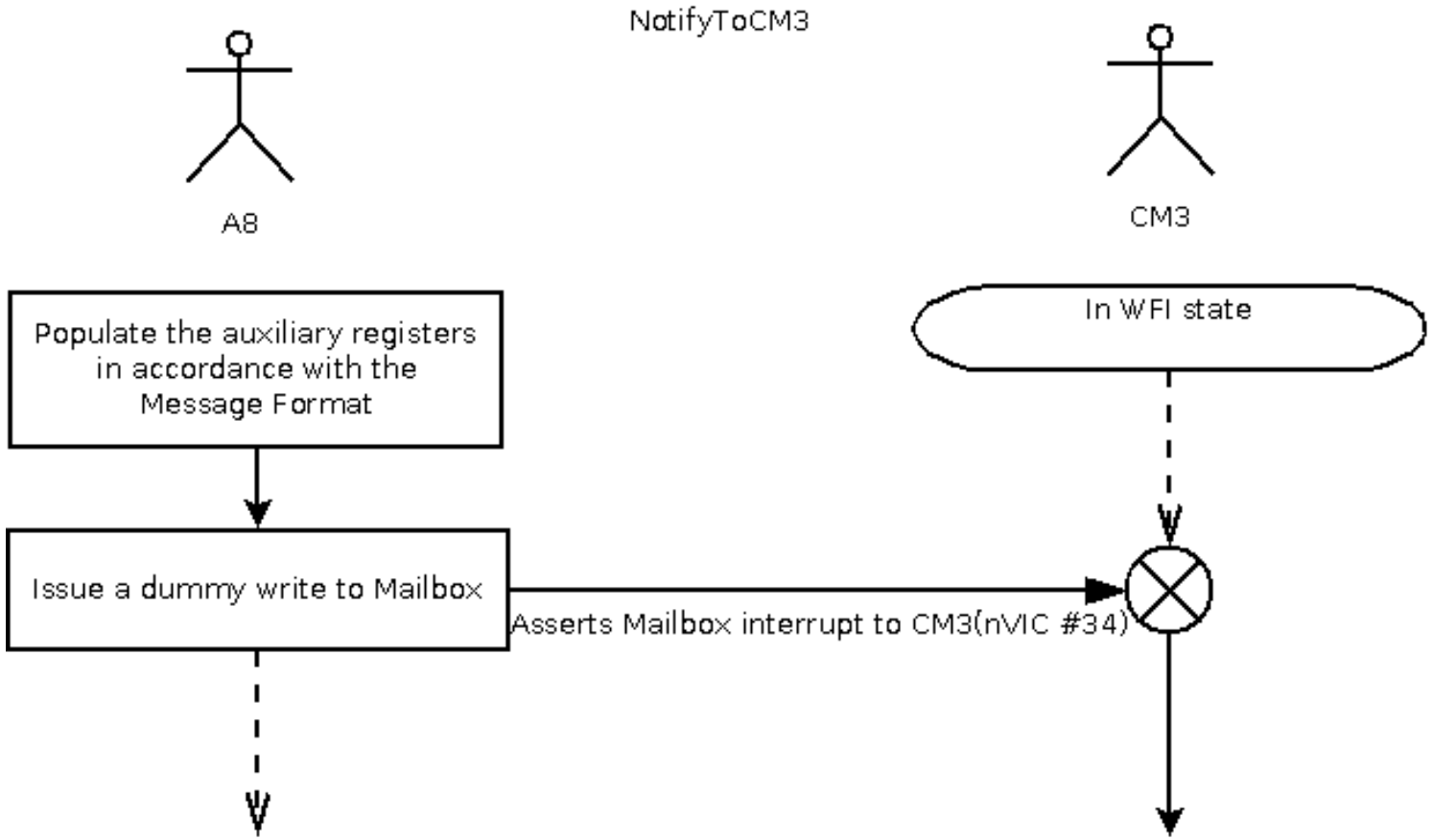




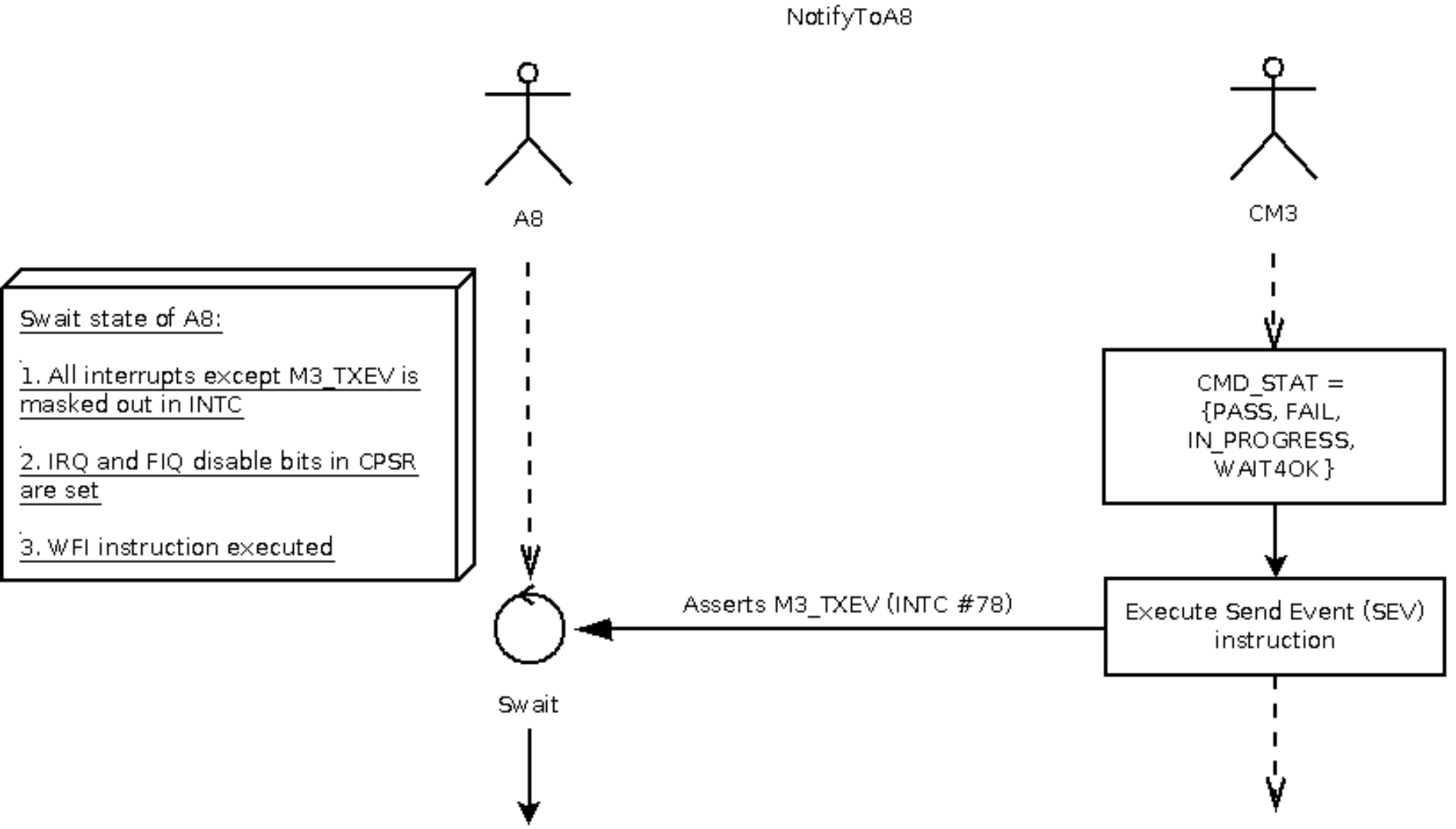
NOTE:

- IPC_MSG_REG[0..7] is in PD_WKUP domain which stays powered ON in DeepSleepx
- H/W Mailbox is in PD_PER but not accessible to CM3. However a write to MB0 by A8 notifies CM3
- Data exchange between A8 and CM3 is via IPC_MSG_REGs

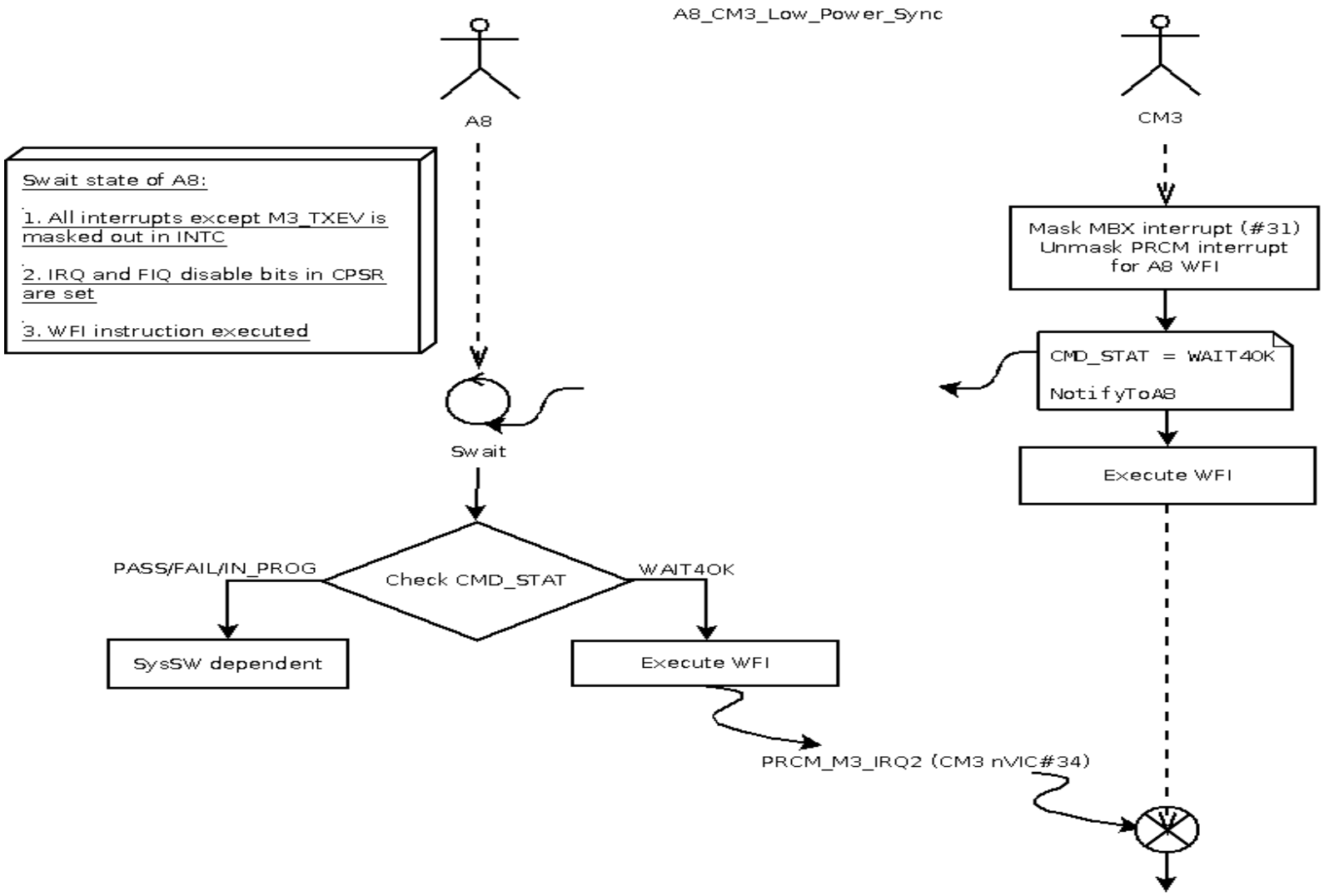
Notification from host CPU to CM3



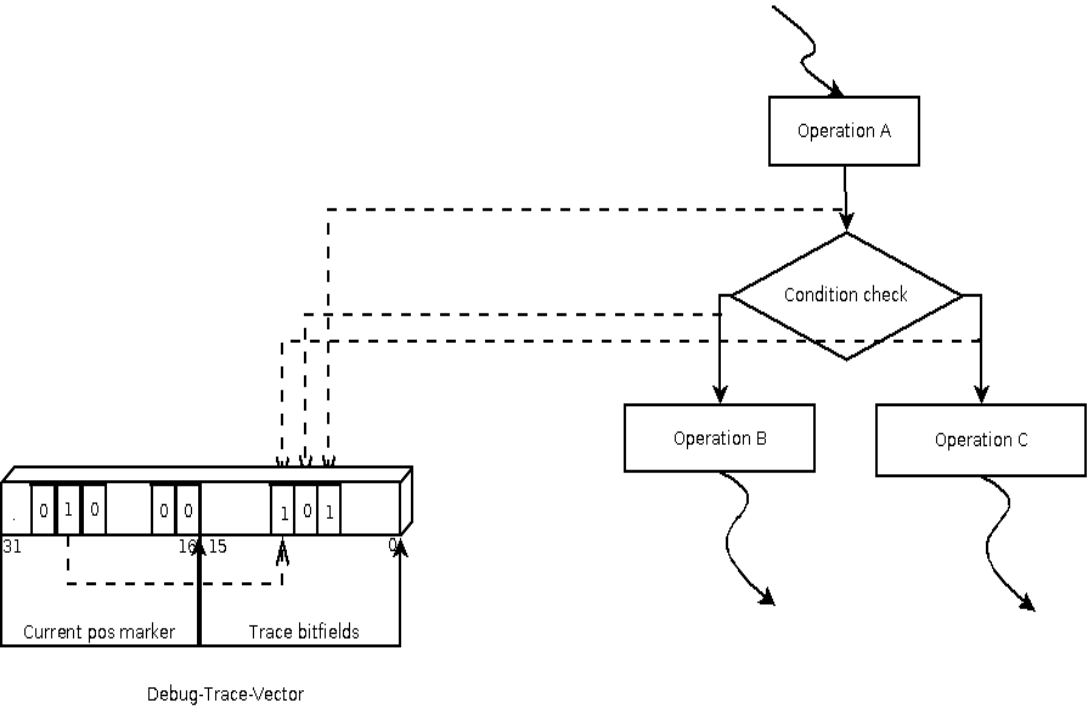
Notification from CM3 to host CPU



Sync scheme for host CPU and CM3



Debug-Trace-Vector of CM3



Debug-Trace-Vector

Note: To be finalized

Bit #	Meaning
0	Reset sequence started
1	Entering main loop
2	Waiting for MBX interrupt
3	Processing MBX interrupt
4	Waiting for trigger event from host
5	Parsing CMD_ID
7	Voltage scaling cmd received
8	Initiating transaction with PMIC
9	Received an ACK from PMIC
10	Received a NACK from PMIC
11	Clock domain transition started
12	Power domain transition started
13	Reserved
14	Reserved
15	Reserved
16-31	Mask for the current location of CM3 code in the trace bitfields. Eg: if CM3 is parsing CMD_ID, bit (5 +.16 == 21) will be set and the rest (16-20, 22-31) will be cleared.

CMD_STAT/ID field

CMD_STAT	*Value*	*Description*
PASS	0x1	In init phase this denotes that CM3 was initialized successfully. When other commands are to be executed, this indicates completion of command.
IN_PROGRESS	0x2	Early indication of command being carried out.
FAIL	0x3	In init phase 0x2 denotes CM3 could not initialize properly. When other tasks are to be done, this indicates some error in carrying out the task. *Check trace vector for details*
WAIT4OK	0x4	CM3 INTC will catch the next WFI of A8 and continue with the pre-defined sequence

CMD_ID	*Value*	*Description*
CMD_RTC	0x1	1. Initiates force_sleep on interconnect clocks. 2. Turns off MPU and PER power domains 3. Programs the RTC alarm register for deasserting pmic_pwr_enable
CMD_RTC_FAST	0x2	1. Programs the RTC alarm register for deasserting pmic_pwr_enable
CMD_DS0	0x3	1. Initates force_sleep on interconnect clocks 2. Turns off the MPU and PER power domains 3. Configures the system for disabling MOSC when CM3 executes WFI
CMD_DS1	0x5	1. Initates force_sleep on interconnect clocks 2. Turns off the MPU power domains 3. Configures the system for disabling MOSC when CM3 executes WFI
CMD_DS2	0x7	1. Configures the system for disabling MOSC when CM3 executes WFI

CM3 data structures

```
struct deep_sleep_data {
            u32 mosc_state :1;                  /* MOSC to be kept on (1) or off (0) */
            u32 deepsleep_count :16;            /* Count of how many OSC clocks needs to be seen \
                                                before exiting deep sleep mode. Default = 0x6A75 */
            u32 vdd_mpu_val :15;                /* If vdd_mpu is to be lowered, vdd_mpu in mV */

            u32 pd_mpu_state :2;                /* Powerstate of PD_MPU */
            u32 pd_mpu_ram_ret_state :1;        /* State of Sabertooth RAM memory when power domain is in retention */
            u32 pd_mpu_l1_ret_state :1;         /* State of L1 memory when power domain is in retention */
            u32 pd_mpu_l2_ret_state :1;         /* State of L2 memory when power domain is in retention */
            u32 pd_mpu_ram_on_state :2;         /* State of Sabertooth RAM memory when power domain is ON */

            u32 pd_per_state :2;                /* Powerstate of PD_PER */
            u32 pd_per_icss_mem_ret_state :1;   /* State of ICSS memory when power domain is in retention */
            u32 pd_per_mem_ret_state :1;        /* State of other memories when power domain is in retention */
            u32 pd_per_ocmc_ret_state :1;       /* State of OCMC memory when power domain is in retention */
            u32 pd_per_icss_mem_on_state :1;    /* State of ICSS memory when power domain is ON */
            u32 pd_per_mem_on_state :1;         /* State of other memories when power domain is ON */
            u32 pd_per_ocmc_on_state :1;        /* State of OCMC memory when power domain is ON */

            u32 wake_sources :13;               /* Wake sources */
                                                /* USB, I2C0, RTC_Timer, RTC_Alarm, \
                                                Timer0, Timer1, UART0, GPIO0_Wake0, \
                                                GPIO0_Wake1, MPU, WDT0, WDT1, \
                                                ADTSC*/
            u32 reserved :4;
};


 struct rtc_data {
             u32 rtc_timeout_val :4;    /* Delay for RTC alarm timeout. Default = 2secs */

 };
```

DeepSleep0

```
struct deep_sleep_data ds0_data =  {
              .mosc_state = 0,                /* MOSC off */
              .deepsleep_count = 0,           /* Default used */
              .vdd_mpu_val = x,               /* Don't care */

              .pd_mpu_state = 1,              /* Retention */
              .pd_mpu_ram_ret_state = 1,      /* Retention */
              .pd_mpu_l1_ret_state = 0,       /* OFF */
              .pd_mpu_l2_ret_state = 0,       /* OFF */
              .pd_mpu_ram_on_state = x,       /* Don't care */

              .pd_per_state = 1,              /* Retention */
              .pd_per_icss_mem_ret_state = 0, /* OFF */
              .pd_per_mem_ret_state = 0,      /* OFF */
              .pd_per_ocmc_ret_state = 1,     /* Retention */
              .pd_per_icss_mem_on_state = x,  /* Don't care */
              .pd_per_mem_on_state = x,       /* Don't care */
              .pd_per_ocmc_on_state = x,      /* Don't care */

              .wake_sources = 0x7f,           /* Any wake source */
}; /* PD_MPU is OFF (retention), PD_PER is OFF (retention) */
```

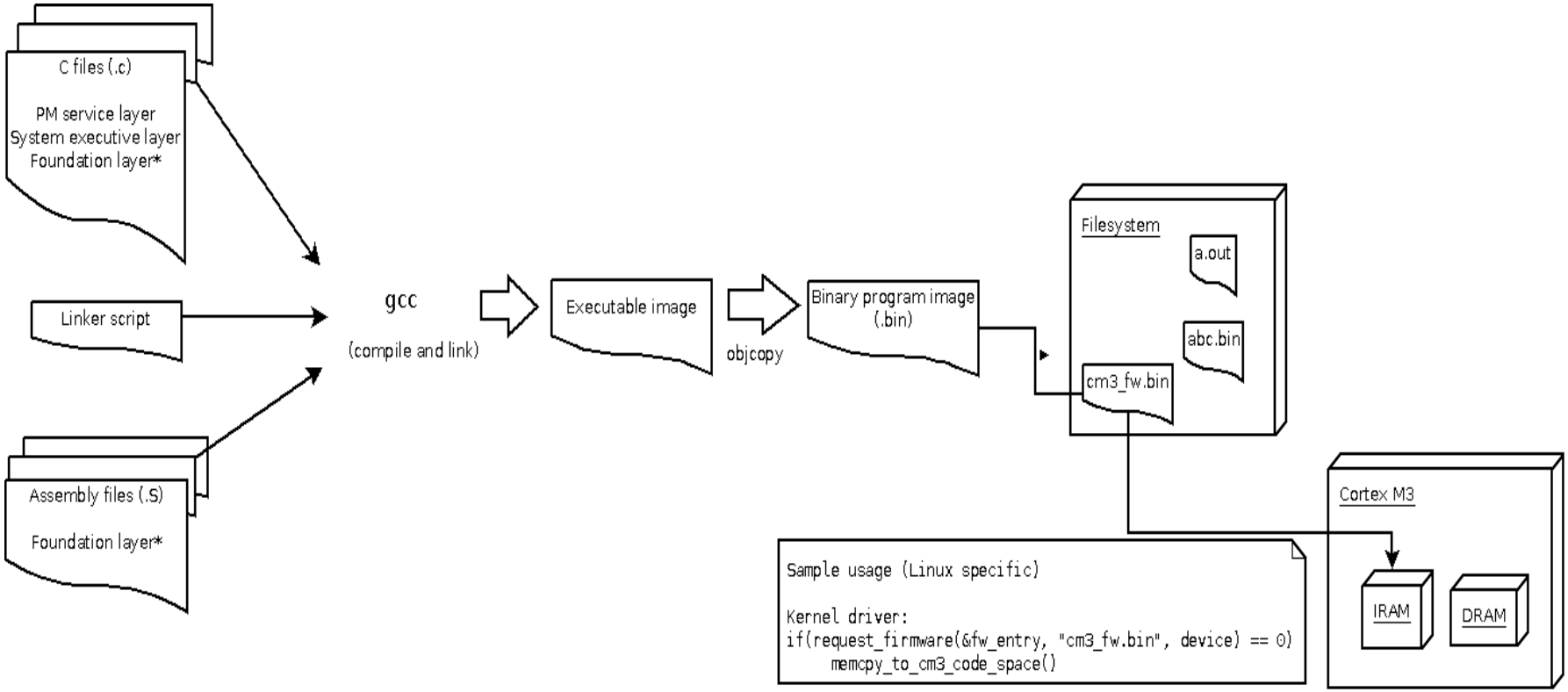
DeepSleep1

```
struct deep_sleep_data ds1_data =  {
              .mosc_state = 0,                /* MOSC off */
              .deepsleep_count = 0,           /* Default used */
              .vdd_mpu_val = x,               /* Don't care */

              .pd_mpu_state = 1,              /* Retention */
              .pd_mpu_ram_ret_state = 1,      /* Retention */
              .pd_mpu_l1_ret_state = 0,       /* OFF */
              .pd_mpu_l2_ret_state = 0,       /* OFF */
              .pd_mpu_ram_on_state = x,       /* Don't care */

              .pd_per_state = 2,              /* ON */
              .pd_per_icss_mem_ret_state = x, /* Don't care */
              .pd_per_mem_ret_state = x,      /* Don't care */
              .pd_per_ocmc_ret_state = x,     /* Don't care */
              .pd_per_icss_mem_on_state = 3,  /* ON */
              .pd_per_mem_on_state = 1,       /* ON */
              .pd_per_ocmc_on_state = 1,      /* ON */

              .wake_sources = 0x7f,           /* Any wake source */
}; /* PD_MPU is OFF (retention), PD_PER is ON */
```

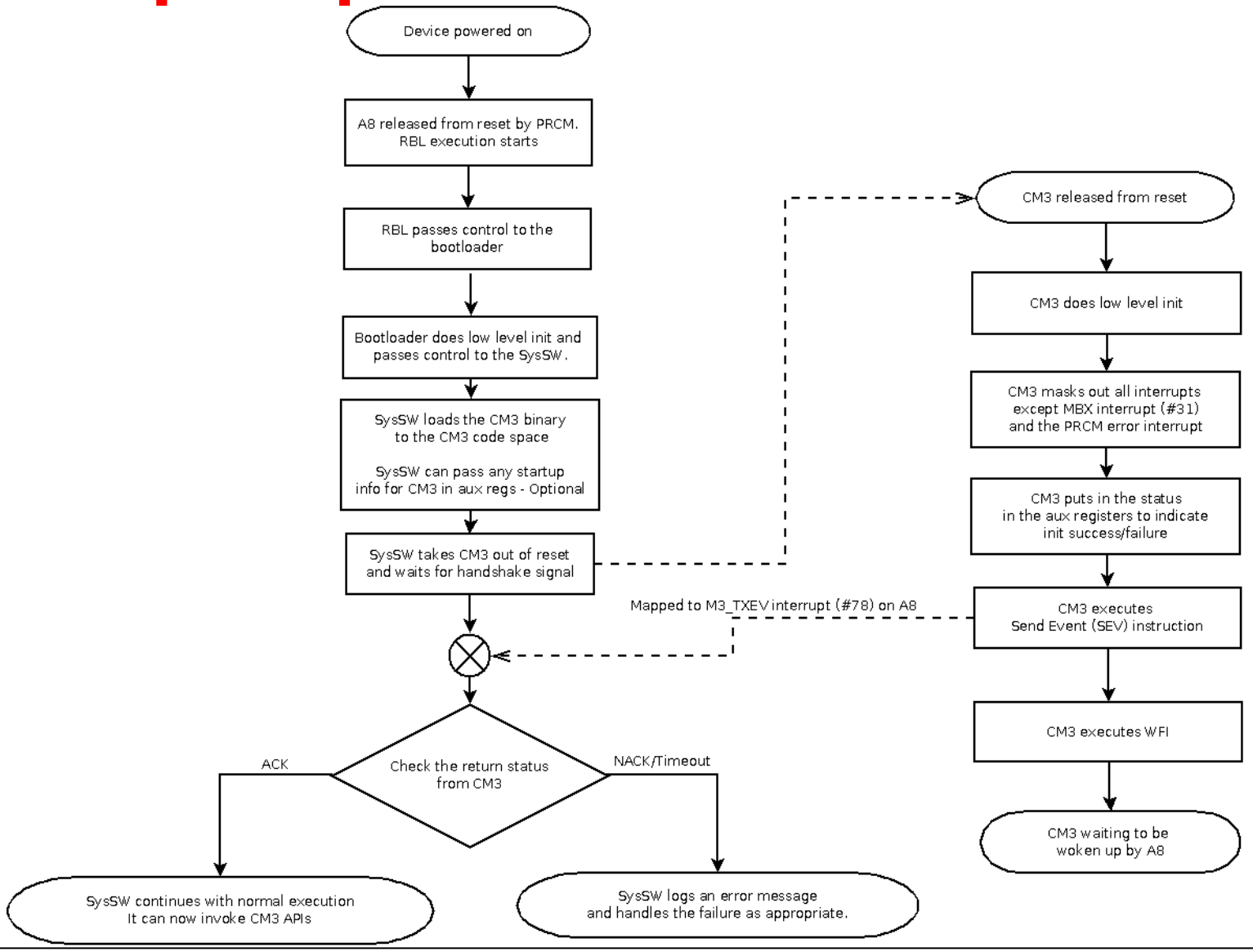
DeepSleep2

```
struct deep_sleep_data ds2_data =  {
              .mosc_state = 0,                /* MOSC off */
              .deepsleep_count = 0,           /* Default used */
              .vdd_mpu_val = 0,               /* No scaling */

              .pd_mpu_state = 2,              /* ON */
              .pd_mpu_ram_ret_state = x,      /* Don't care */
              .pd_mpu_l1_ret_state = x,       /* Don't care */
              .pd_mpu_l2_ret_state = x,       /* Don't care */
              .pd_mpu_ram_on_state = 3,       /* ON */

              .pd_per_state = 2,              /* ON */
              .pd_per_icss_mem_ret_state = x, /* Don't care */
              .pd_per_mem_ret_state = x,      /* Don't care */
              .pd_per_ocmc_ret_state = x,     /* Don't care */
              .pd_per_icss_mem_on_state = 3,  /* ON */
              .pd_per_mem_on_state = 3,       /* ON */
              .pd_per_ocmc_on_state = 3,      /* ON */

              .wake_sources = 0x7f,           /* Any wake source */
}; /* PD_MPU is ON, PD_PER is ON */
```

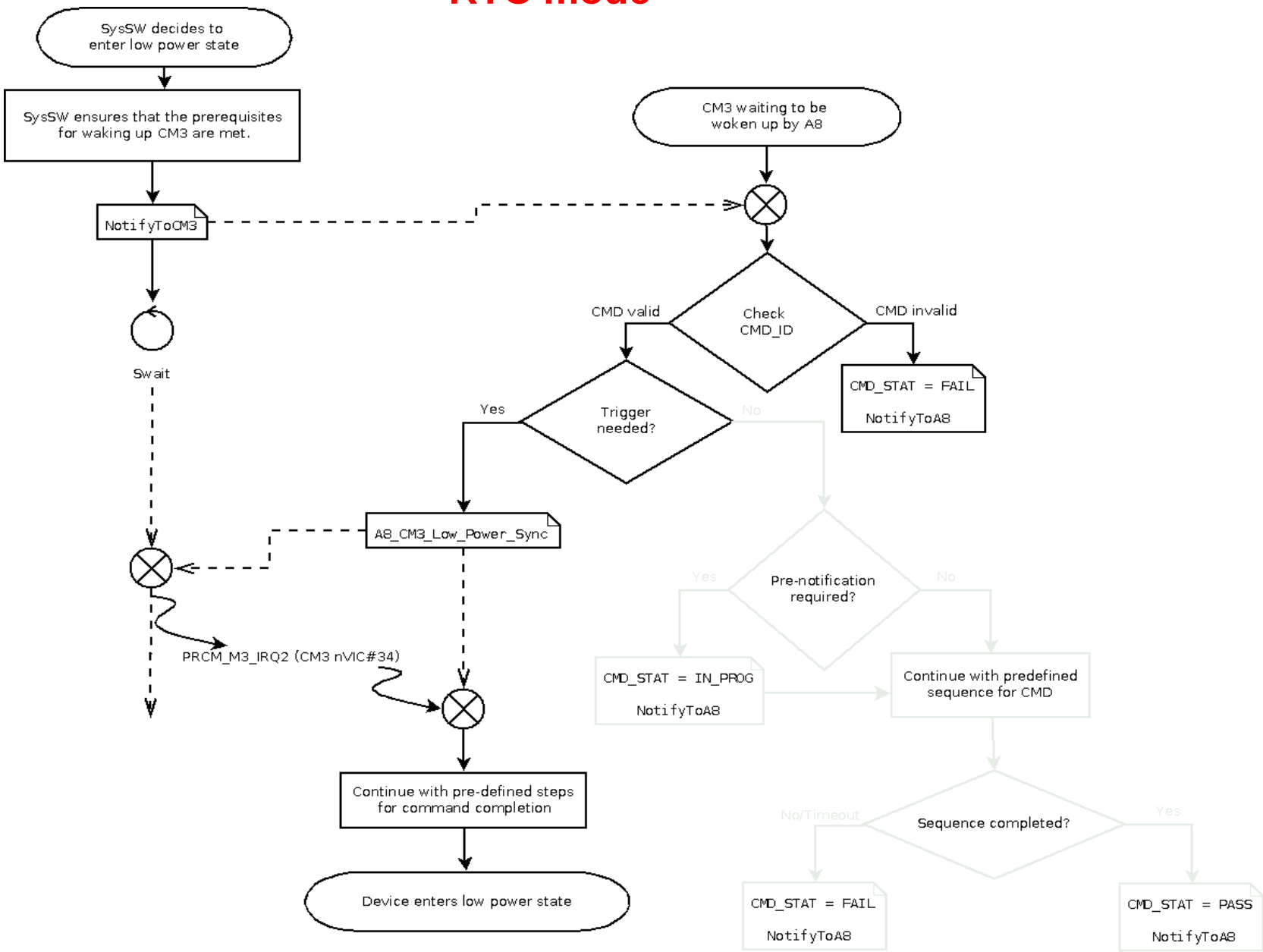
PM FW loading



Startup sequence



RTC mode


SysSW decides to enter low power state
SysSW ensures that the prerequisites for waking up CM3 are met.
NotifyToCM3
Swait
CM3 waiting to be woken up by A8
Check CMD_ID
CMD valid
CMD invalid
CMD_STAT = FAIL
NotifyToA8
Trigger needed?
Yes
No
A8_CM3_Low_Power_Sync
PRCM_M3_IRQ2 (CM3 nVIC#34)
Pre-notification required?
Yes
No
CMD_STAT = IN_PROG
NotifyToA8
Continue with predefined sequence for CMD
Continue with pre-defined steps for command completion
Sequence completed?
No/Timeout
Yes
CMD_STAT = FAIL
NotifyToA8
CMD_STAT = PASS
NotifyToA8
Device enters low power state