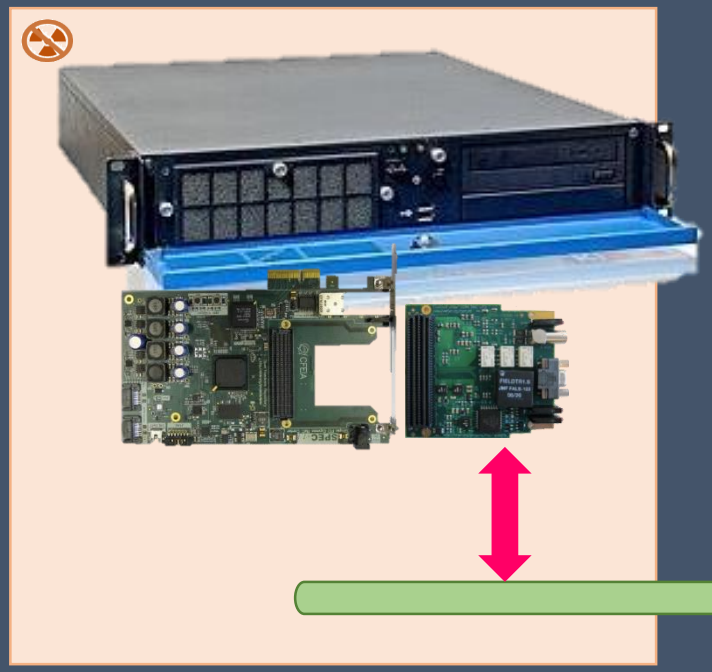
MASTERFIP
GW & SW REVIEW

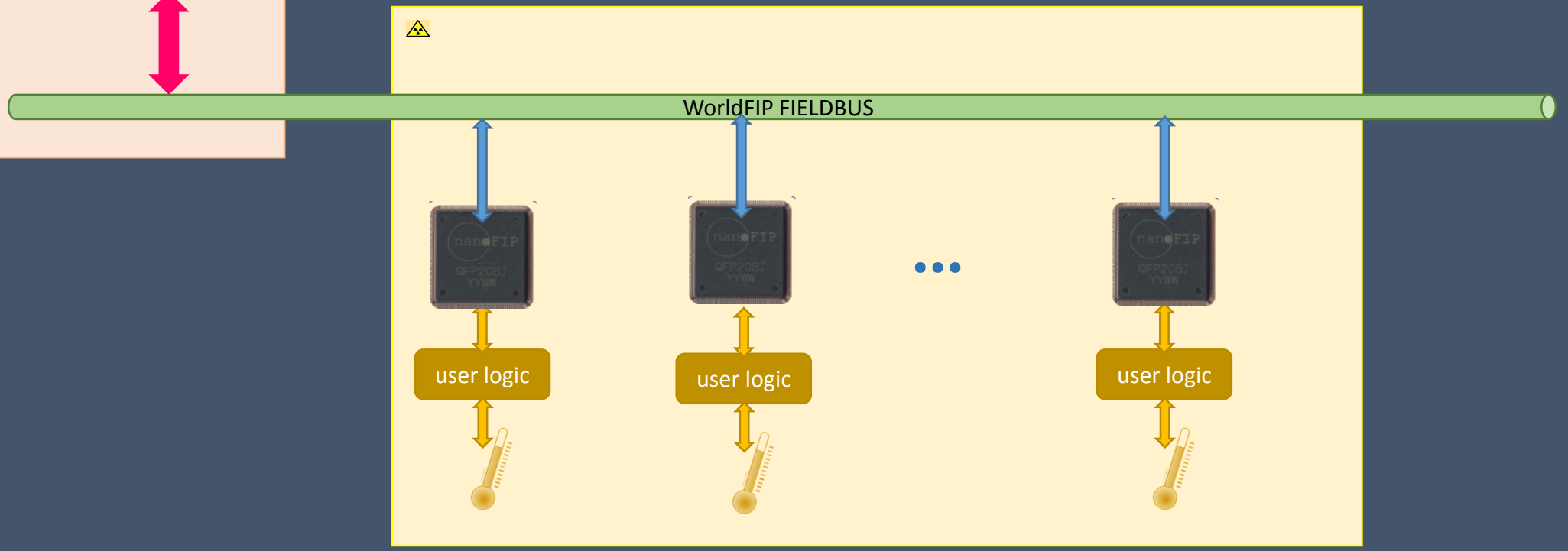
02 MAR 17

WORLDFIP BASICS

TOPOLOGY



- Twisted pair multi-drop
- 5V differential signalling
- Up to 2.5 Mb/s



WORLDFIP BASICS

DATA TRANSFER WINDOWS

WorldFIP defines three types of data transfers:

- o **deterministic** traffic through periodic variables
- o **event-type-traffic** through aperiodic messages
- o **network-management-services** through SMMPS aperiodic variables.

Aperiodic/SMMPS traffic does not disturb the determinism of the periodic part.

The macrocycle (length of each window and variables in periodic window) is configured once at startup and is not changed during bus operation.

WORLDFIP BASICS

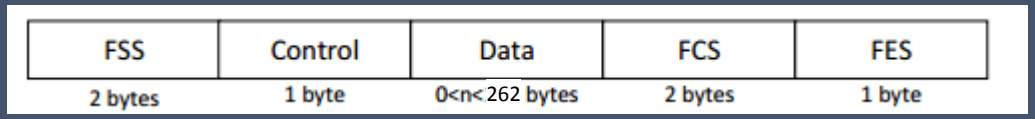
QUESTION & RESPONSE FRAMES

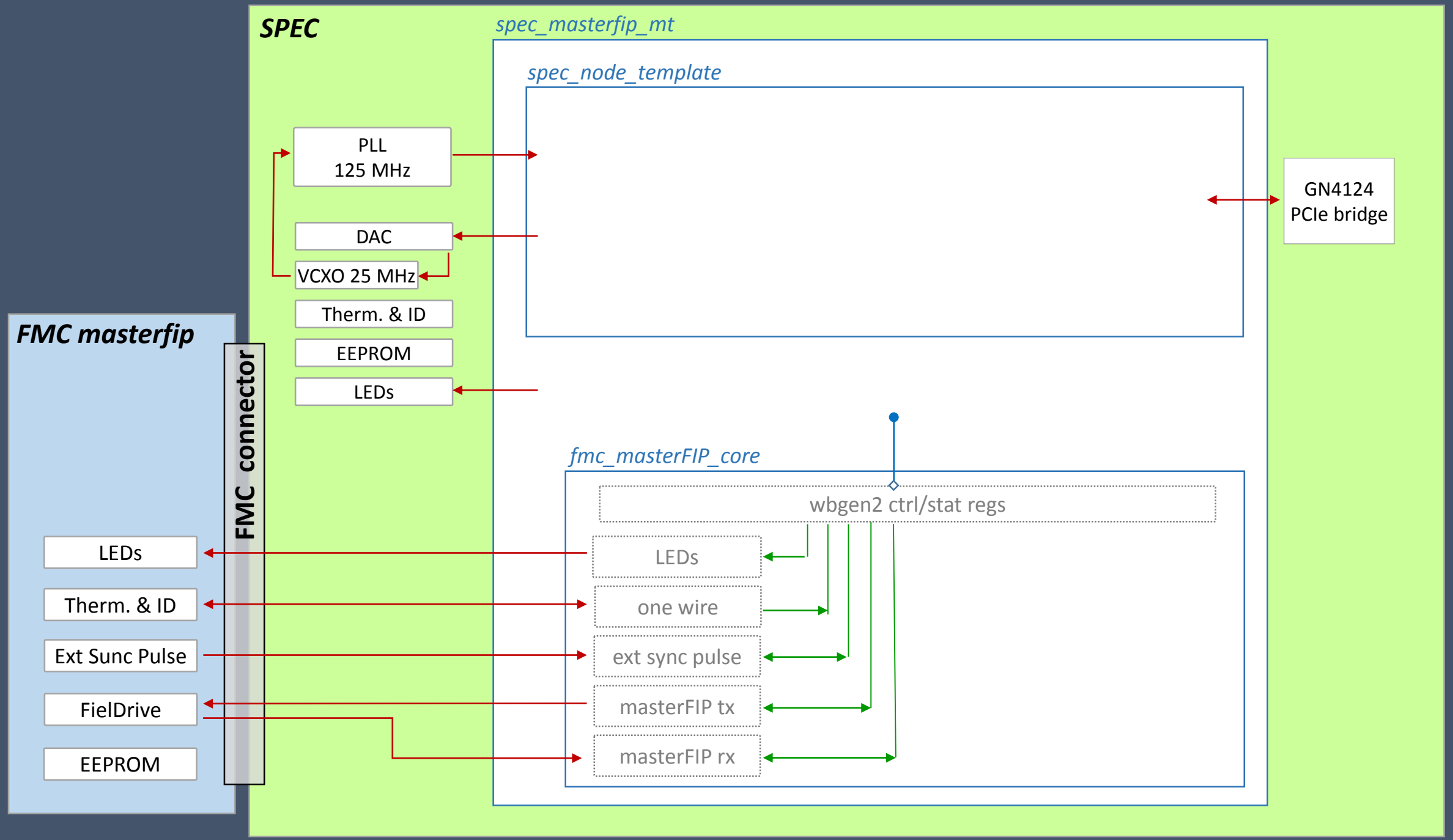
For all types of traffic, communication based on:

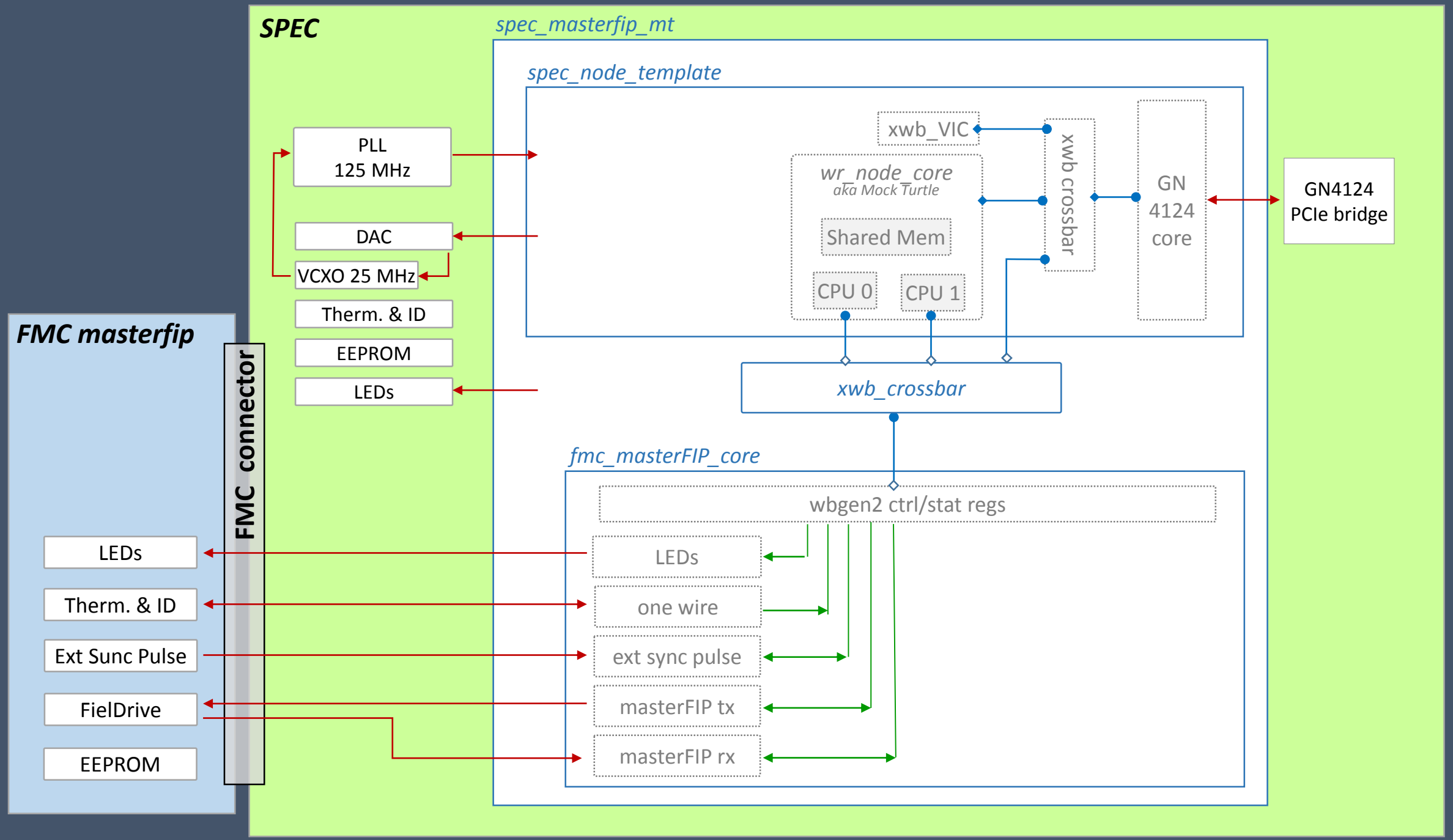
- **Question ID_DAT frame:** sent only from the master
- ID_DAT simultaneously recorded by all the nodes connected to the bus
- **Only one node/ the master** recognizes itself as being the producer
- Producer broadcasts **response RP_DAT frame** on the bus
- One/ more other nodes and/or the master recognize themselves as consumers and read the frame

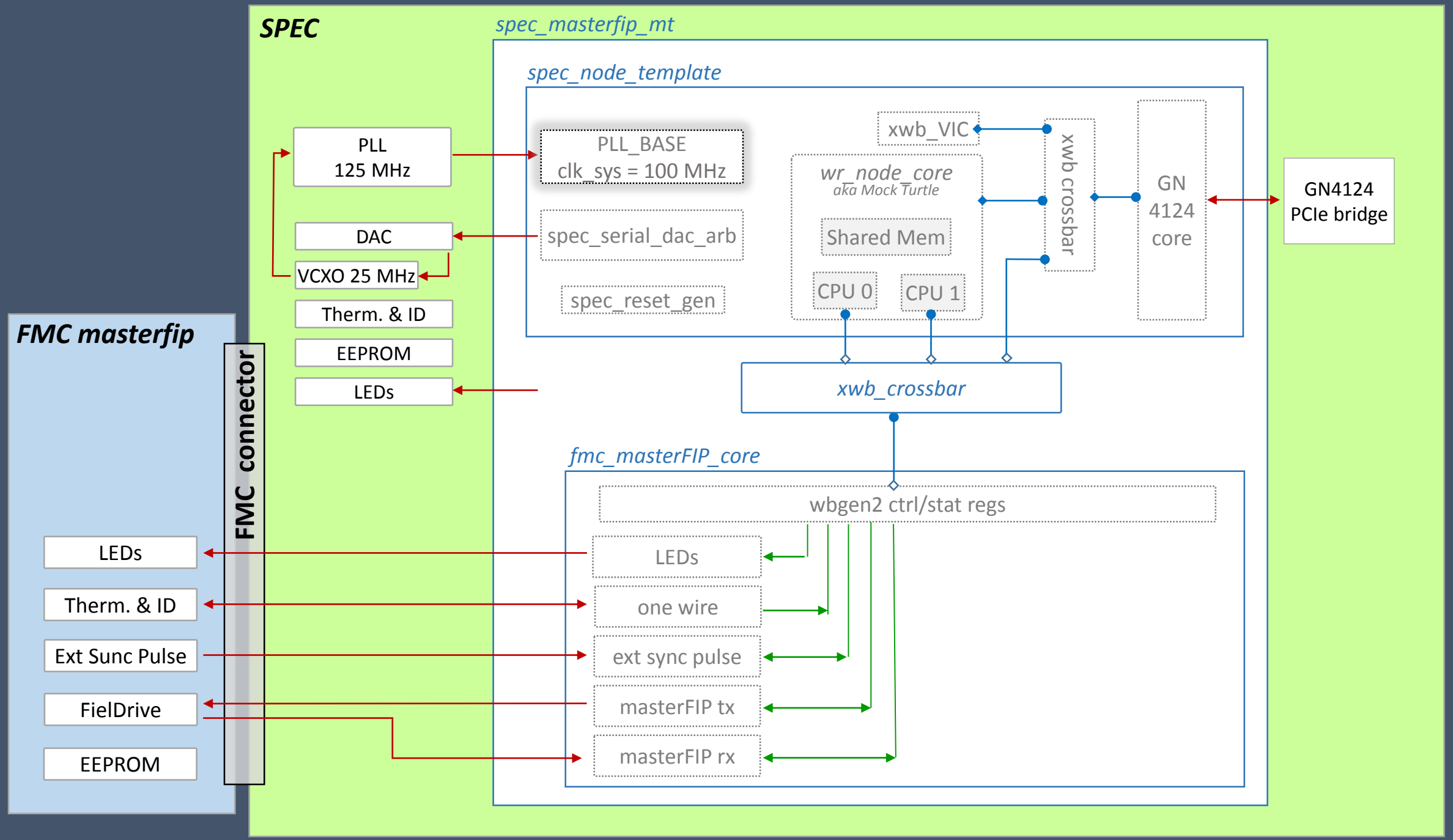
Question ID_DAT frame:

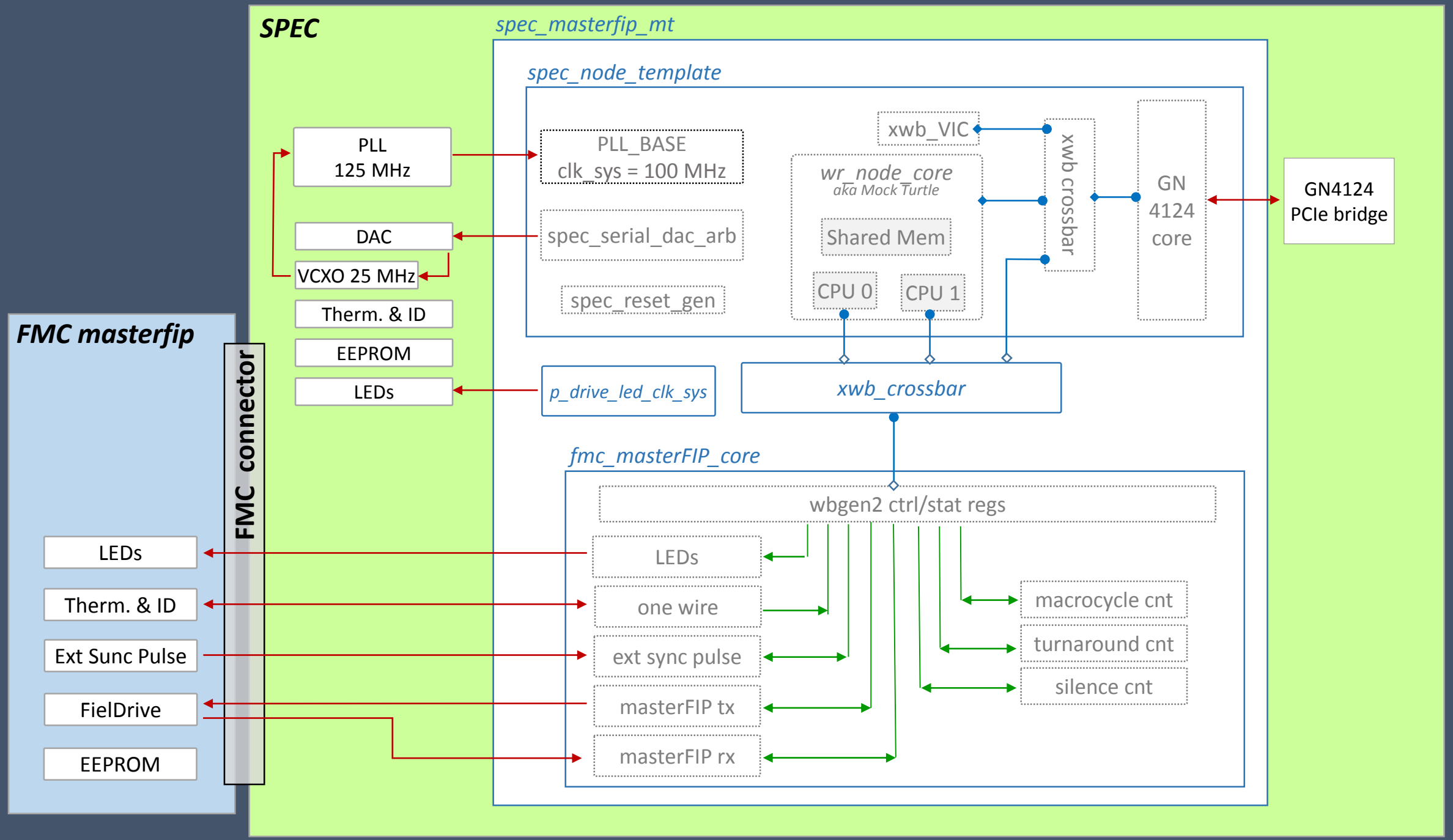
Response RP_DAT frame:

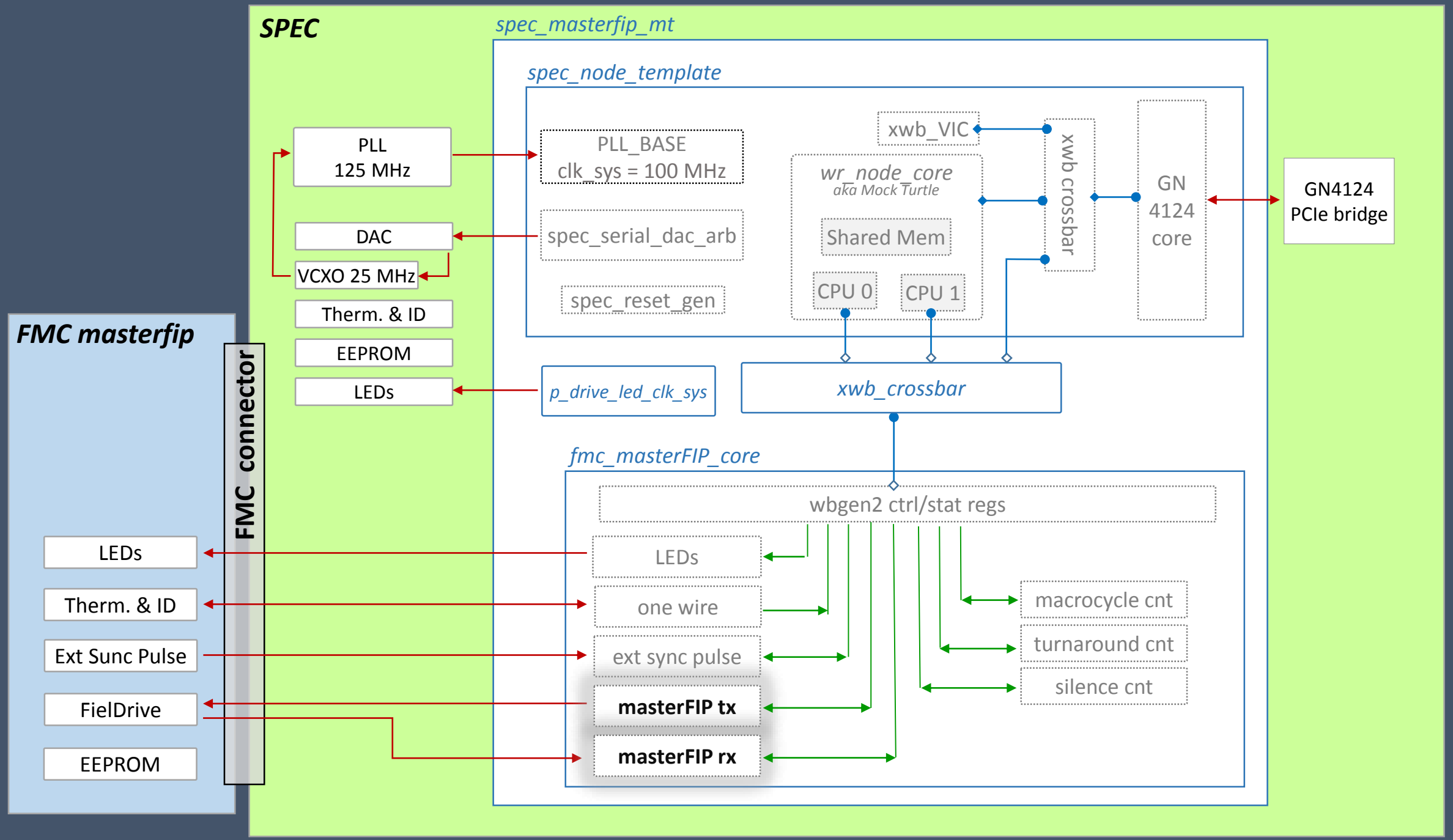


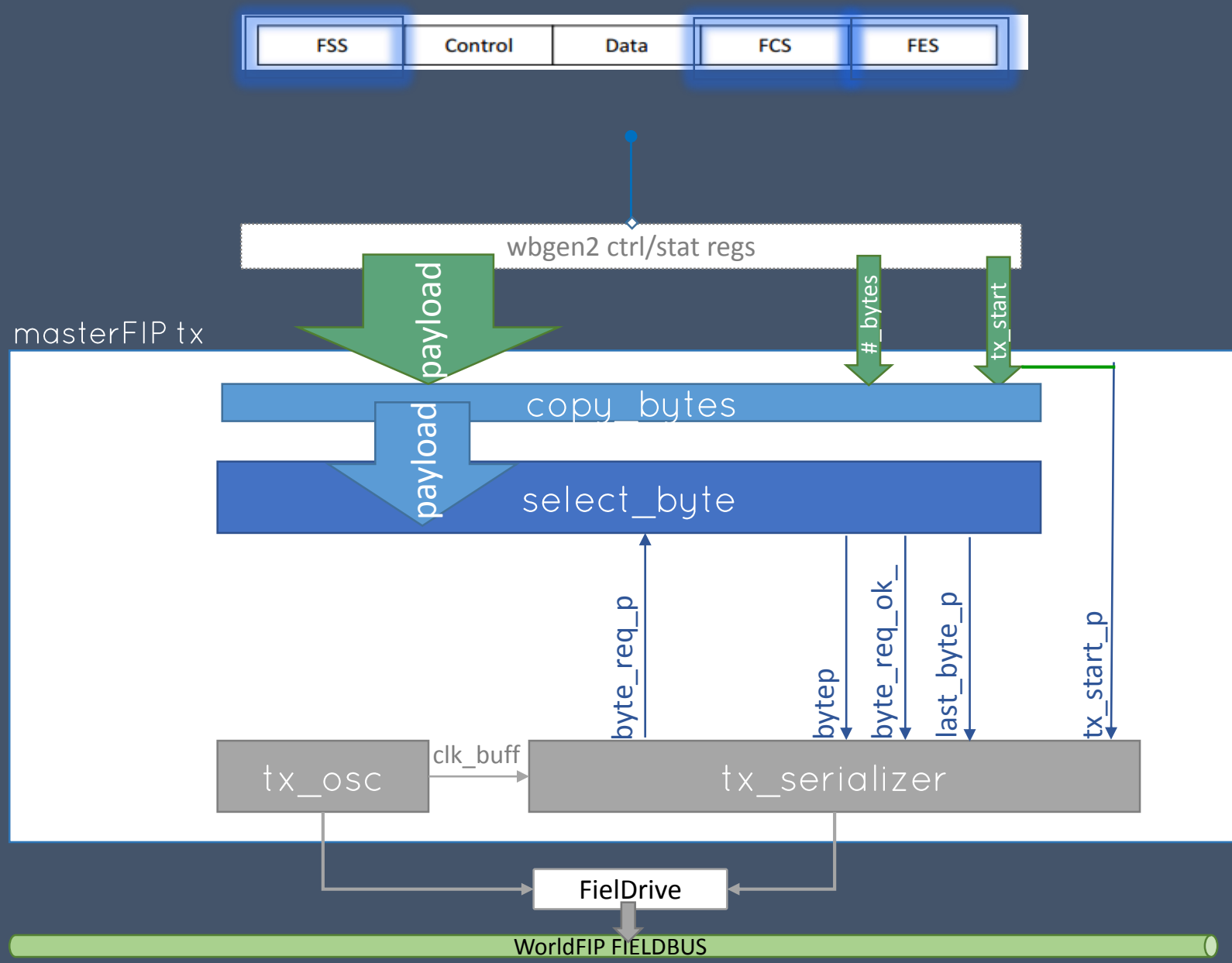

SPEC
spec_masterfip_mt
spec_node_template
PLL
125 MHz
DAC
VCXO 25 MHz
Therm. & ID
EEPROM
LEDs
GN4124
PCIe bridge
FMC masterfip
FMC connector
fmc_masterFIP_core
wbgen2 ctrl/stat regs
LEDs
one wire
ext sync pulse
masterFIP tx
masterFIP rx
LEDs
Therm. & ID
Ext Sunc Pulse
FielDrive
EEPROM

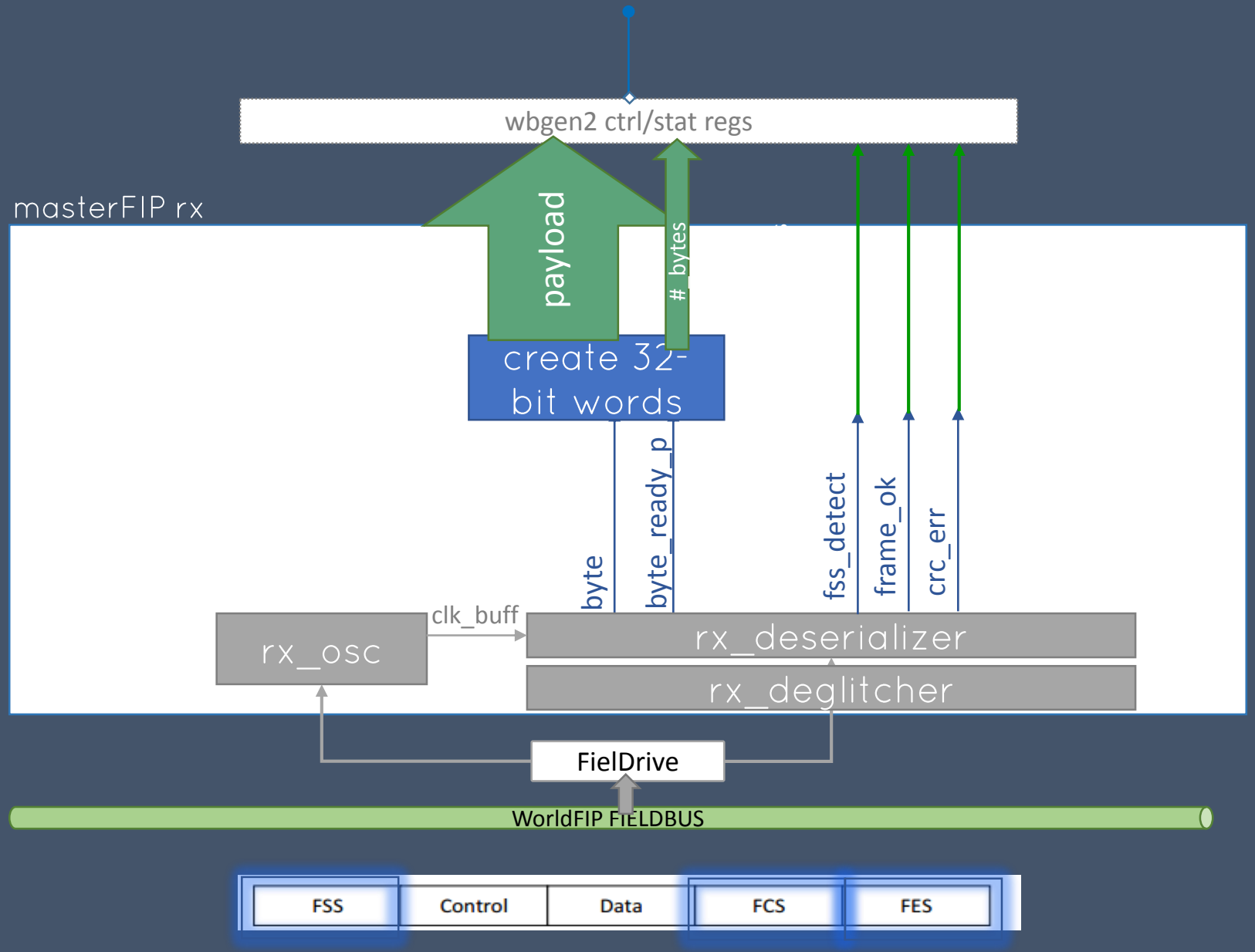

SPEC
spec_masterfip_mt
spec_node_template
PLL
125 MHz
DAC
VCXO 25 MHz
Therm. & ID
EEPROM
LEDs
xwb_VIC
wr_node_core
aka Mock Turtle
Shared Mem
CPU 0
CPU 1
xwb crossbar
GN
4124
core
GN4124
PCIe bridge
xwb_crossbar
FMC masterfip
FMC connector
LEDs
Therm. & ID
Ext Sunc Pulse
FielDrive
EEPROM
fmc_masterFIP_core
wbgen2 ctrl/stat regs
LEDs
one wire
ext sync pulse
masterFIP tx
masterFIP rx

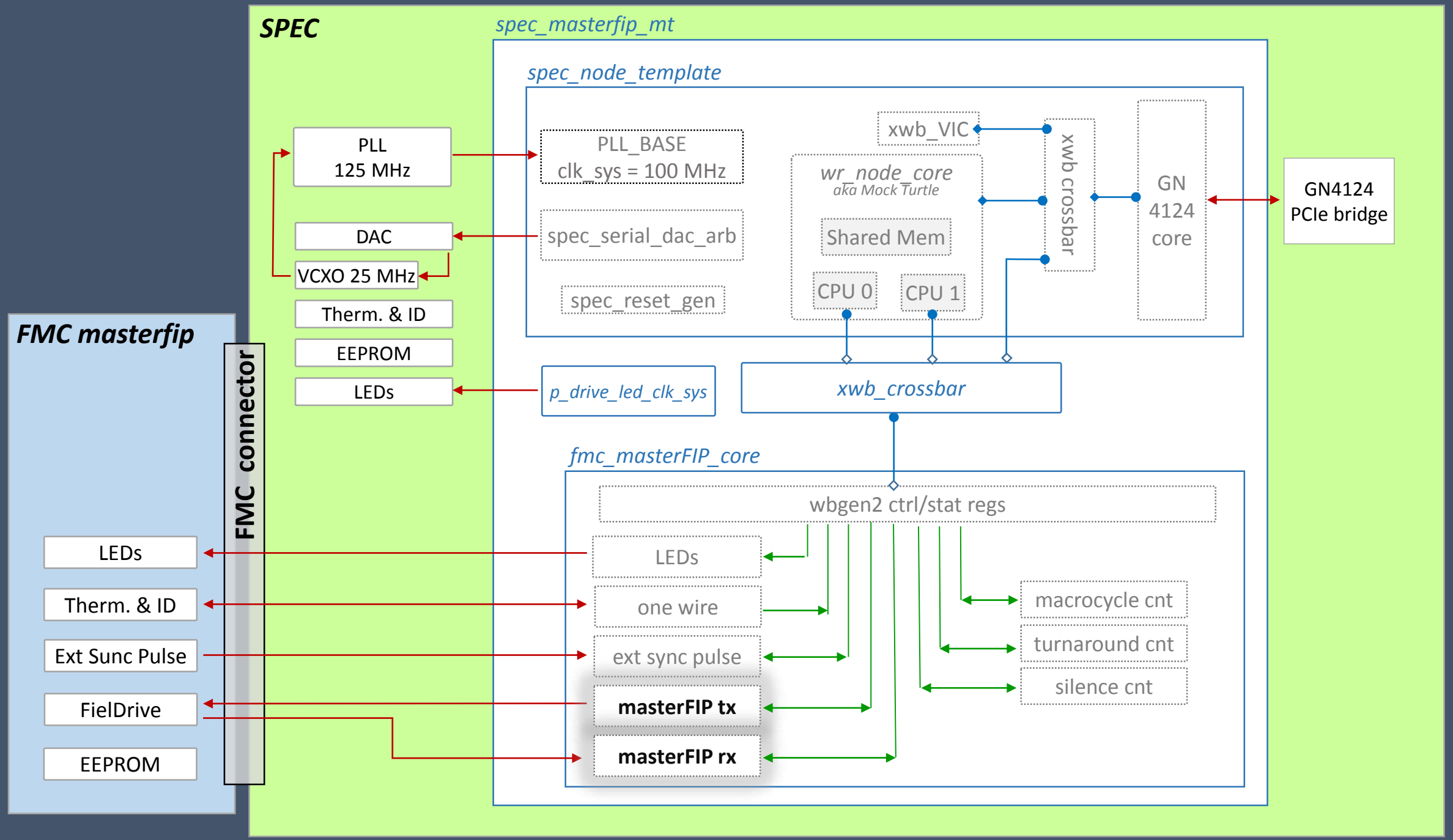

SPEC
spec_masterfip_mt
spec_node_template
PLL
125 MHz
PLL_BASE
clk_sys = 100 MHz
xwb_VIC
xwb crossbar
wr_node_core
aka Mock Turtle
Shared Mem
CPU 0
CPU 1
GN
4124
core
GN4124
PCIe bridge
DAC
spec_serial_dac_arb
VCXO 25 MHz
Therm. & ID
EEPROM
LEDs
spec_reset_gen
xwb_crossbar
FMC masterfip
FMC connector
fmc_masterFIP_core
wbgen2 ctrl/stat regs
LEDs
Therm. & ID
Ext Sunc Pulse
FielDrive
EEPROM
LEDs
one wire
ext sync pulse
masterFIP tx
masterFIP rx


SPEC
spec_masterfip_mt
spec_node_template
PLL
125 MHz
PLL_BASE
clk_sys = 100 MHz
xwb_VIC
xwb crossbar
wr_node_core
aka Mock Turtle
Shared Mem
CPU 0
CPU 1
GN
4124
core
GN4124
PCIe bridge
DAC
spec_serial_dac_arb
VCXO 25 MHz
Therm. & ID
EEPROM
LEDs
spec_reset_gen
p_drive_led_clk_sys
xwb_crossbar
FMC masterfip
FMC connector
fmc_masterFIP_core
wbgen2 ctrl/stat regs
LEDs
Therm. & ID
Ext Sunc Pulse
FielDrive
EEPROM
LEDs
one wire
ext sync pulse
masterFIP tx
masterFIP rx
macrocycle cnt
turnaround cnt
silence cnt

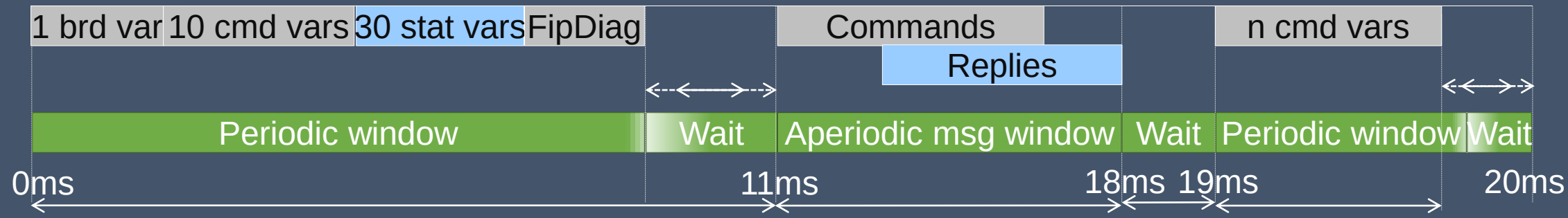

SPEC
spec_masterfip_mt
spec_node_template
PLL
125 MHz
PLL_BASE
clk_sys = 100 MHz
xwb_VIC
xwb crossbar
wr_node_core
aka Mock Turtle
Shared Mem
CPU 0
CPU 1
GN
4124
core
GN4124
PCIe bridge
DAC
spec_serial_dac_arb
VCXO 25 MHz
Therm. & ID
EEPROM
LEDs
spec_reset_gen
p_drive_led_clk_sys
xwb_crossbar
FMC masterfip
FMC connector
fmc_masterFIP_core
wbgen2 ctrl/stat regs
LEDs
Therm. & ID
Ext Sunc Pulse
FielDrive
EEPROM
LEDs
one wire
ext sync pulse
masterFIP tx
masterFIP rx
macrocycle cnt
turnaround cnt
silence cnt


FSS
Control
Data
FCS
FES
wbgen2 ctrl/stat regs
payload
#_bytes
tx_start
masterFIP tx
copy_bytes
payload
select_byte
byte_req_p
bytep
byte_req_ok_
last_byte_p
tx_start_p
tx_osc
clk_buff
tx_serializer
FielDrive
WorldFIP FIELDBUS


wbgen2 ctrl/stat regs
masterFIP rx
payload
bytes
create 32-bit words
byte
byte_ready_p
fss_detect
frame_ok
crc_err
clk_buff
rx_osc
rx_deserializer
rx_deglitcher
FielDrive
WorldFIP FIELDBUS
FSS
Control
Data
FCS
FES


SPEC
spec_masterfip_mt
spec_node_template
PLL
125 MHz
PLL_BASE
clk_sys = 100 MHz
xwb_VIC
xwb crossbar
wr_node_core
aka Mock Turtle
Shared Mem
CPU 0
CPU 1
GN
4124
core
GN4124
PCIe bridge
DAC
spec_serial_dac_arb
VCXO 25 MHz
Therm. & ID
EEPROM
LEDs
spec_reset_gen
p_drive_led_clk_sys
xwb_crossbar
FMC masterfip
FMC connector
fmc_masterFIP_core
wbgen2 ctrl/stat regs
LEDs
Therm. & ID
Ext Sunc Pulse
FielDrive
EEPROM
LEDs
one wire
ext sync pulse
masterFIP tx
masterFIP rx
macrocycle cnt
turnaround cnt
silence cnt

EXAMPLE: FGC MACROCYCLE

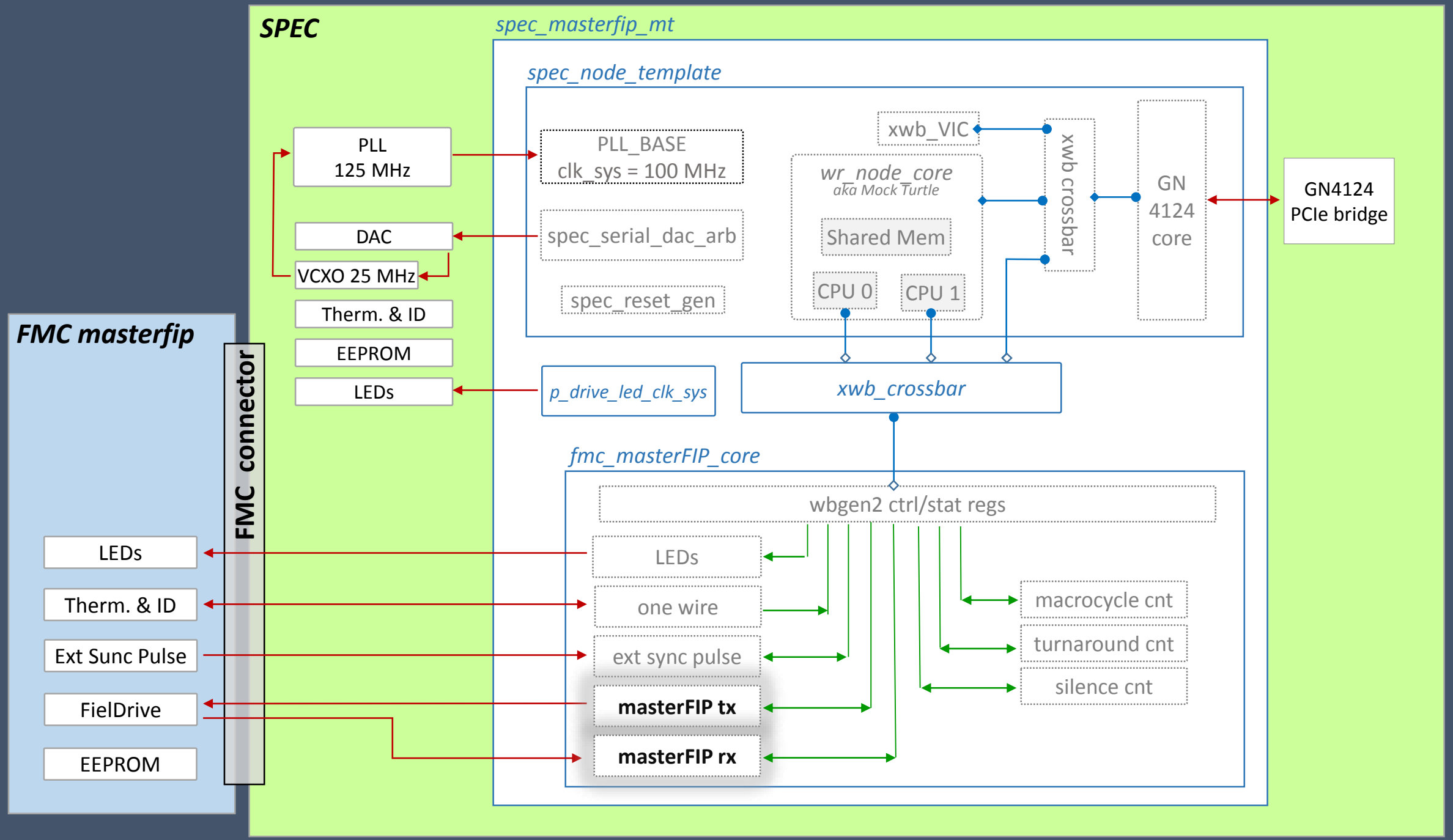


Translated into requirements:

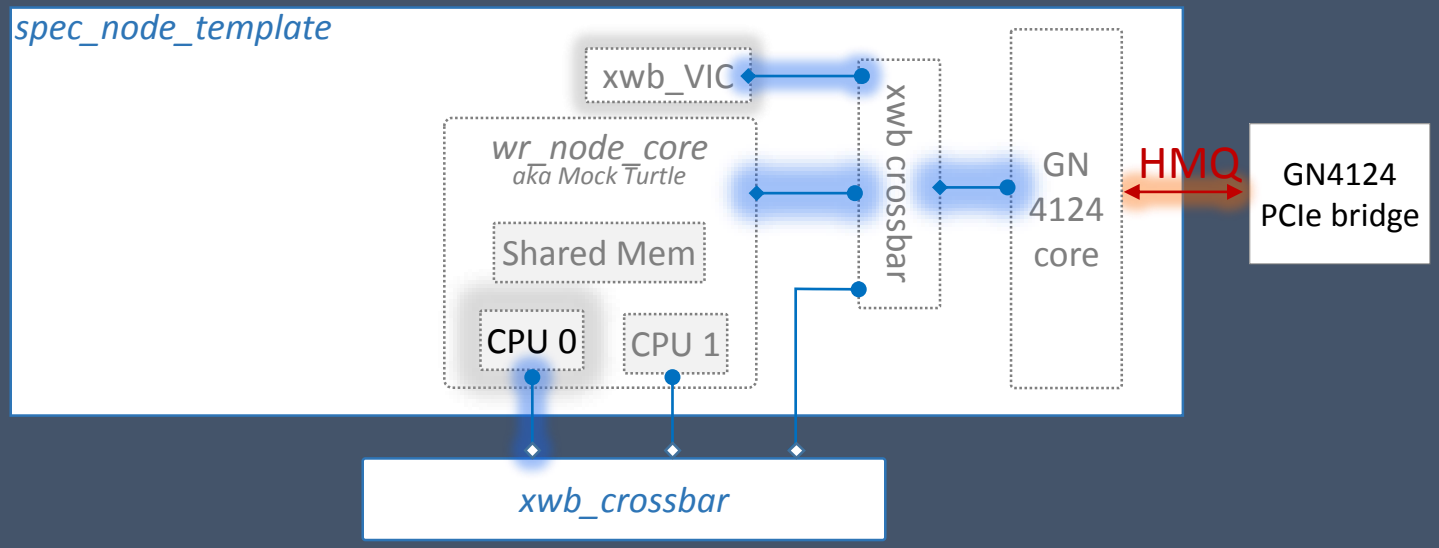
- Periodic traffic should start at T=0ms.
- Ensure that all the periodic traffic can be executed in less than 11ms.
- At T=11ms aperiodic traffic should start and end at 18ms.
- At T=18ms wait for 1ms doing nothing (or send dummy stuffing frames keeping the link active).
- Next periodic traffic should start at T=19ms.
- Ensure that it should end before T=20ms.
- Bind any variables with an irq to wake-up application for data consuming or producing.

LIB API

- **Open/close** device
- **BA config:** hw_speed_get(), hw_cfg_set(), sw_cfg_set(), response_time_get()
- **Macrocycle config:**
  - Create/delete/reset macrocycle
  - Create data FIP objects (per vars, aper msg): size, direction (prod/cons), irq flag, usr callback
- **Append windows:**
  - periodic var window: list of periodic variables to play
  - aperiodic var windows: list of aperiodic variables to play, end time
  - aperiodic message window: FIFO size (prod and cons), end time
  - wait window: silent flag, end time.
- **BA action:** load macrocycle, start, stop, reset.
- **Runtime action:** fipdata update, write var/msg payload, request for aperiodic var.

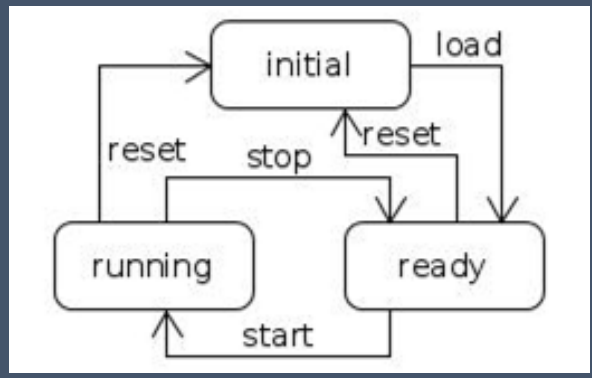

SPEC
spec_masterfip_mt
spec_node_template
PLL
125 MHz
PLL_BASE
clk_sys = 100 MHz
xwb_VIC
xwb crossbar
wr_node_core
aka Mock Turtle
Shared Mem
CPU 0
CPU 1
GN
4124
core
GN4124
PCIe bridge
DAC
spec_serial_dac_arb
VCXO 25 MHz
Therm. & ID
EEPROM
LEDs
spec_reset_gen
p_drive_led_clk_sys
xwb_crossbar
FMC masterfip
FMC connector
fmc_masterFIP_core
wbgen2 ctrl/stat regs
LEDs
Therm. & ID
Ext Sunc Pulse
FielDrive
EEPROM
LEDs
one wire
ext sync pulse
masterFIP tx
masterFIP rx
macrocycle cnt
turnaround cnt
silence cnt

MOCK TURTLE SOFTWARE

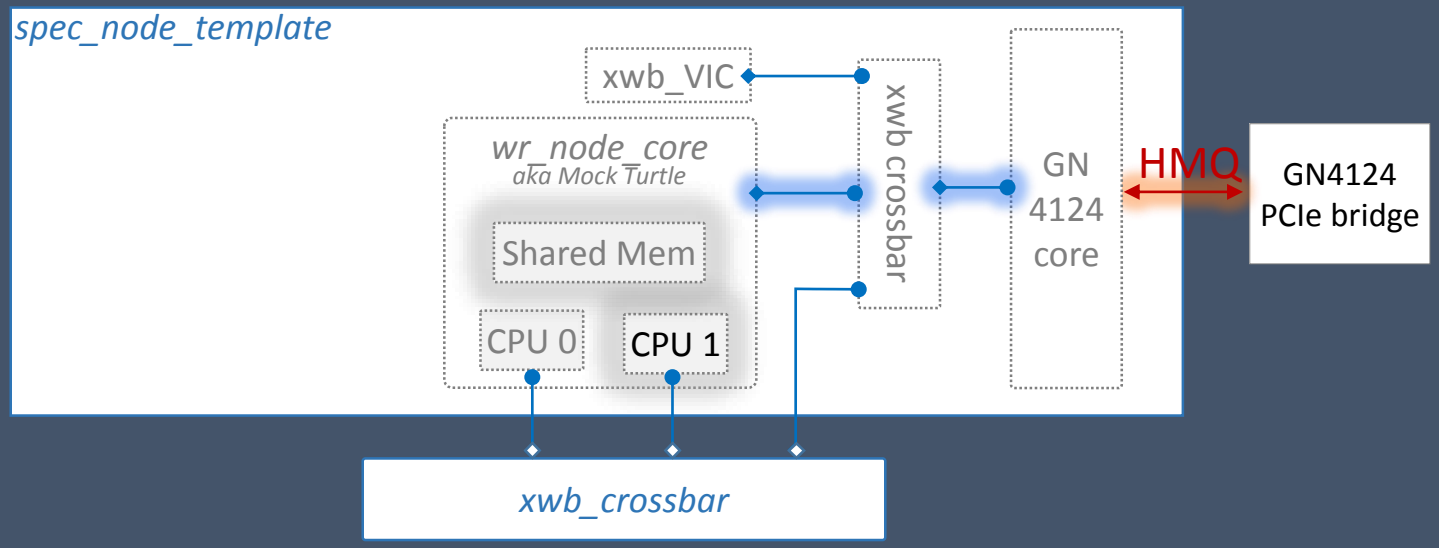


CPU0

- Handle config commands: hw_cfg(), load_macro_cycle(),start(), stop(), reset()
- Implements simple BA state machine
- Runs macrocycle
- Push **consumed** FIP data through HMQ



MOCK TURTLE SOFTWARE



CPU1

- Handle payload setting for **produced** var and messages
- Handle request for scheduling aperiodic variables
- Build diagnostic report

Shared Memory

- Payload storage for **produced** var and msg

Note: BSS and SHM are partitioned dynamically when the macro cycle is loaded.

RETRIEVE GATEWARE

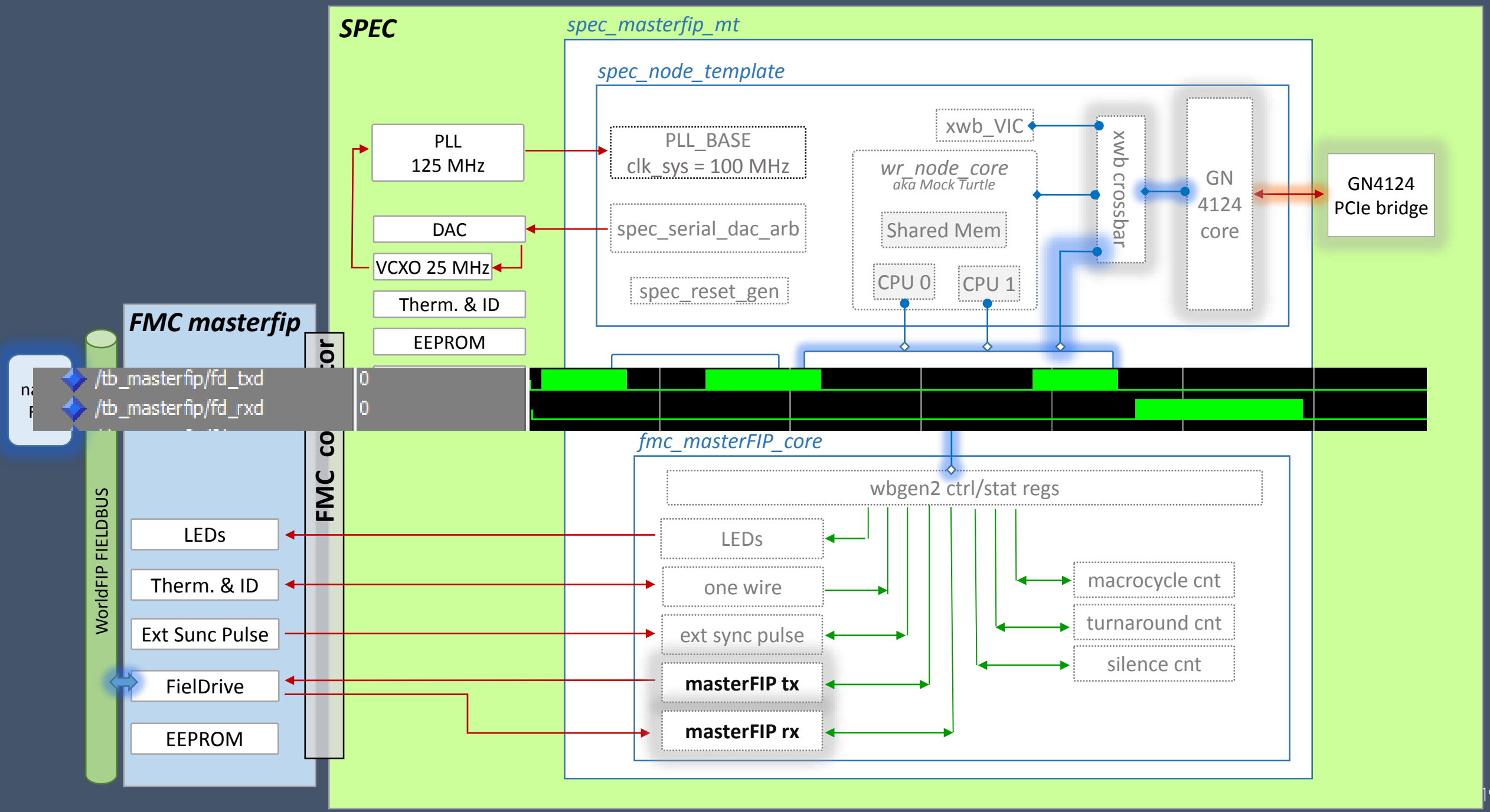
1. Clone git repo:
$ git clone git://ohwr.org/cern-fip/masterfip/masterfip-gw.git

2. Checkout 'eva_dev' branch:
$ cd masterfip-gw
$ git checkout –t origin/eva_dev.

3. Submodules initialization and update (not recursive!):
$ git submodule init
$ git submodule update

4. To synthesize, open the 'syn/spec/spec_masterfip_mt.xise' with Xilinx ISE:
$ cd syn/spec
$ ise spec_masterfip_mt.xise

5. To simulate in Modelsim
do masterfip-gw/sim/spec/tb_masterfip.do


SPEC
spec_masterfip_mt
spec_node_template
PLL
125 MHz
PLL_BASE
clk_sys = 100 MHz
xwb_VIC
xwb crossbar
wr_node_core
aka Mock Turtle
Shared Mem
CPU 0
CPU 1
GN
4124
core
GN4124
PCIe bridge
DAC
spec_serial_dac_arb
VCXO 25 MHz
spec_reset_gen
Therm. & ID
EEPROM
FMC masterfip
FMC co
WorldFIP FIELDBUS
/tb_masterfip/fd_txd
/tb_masterfip/fd_rxd
0
0
fmc_masterFIP_core
wbgen2 ctrl/stat regs
LEDs
Therm. & ID
Ext Sunc Pulse
FielDrive
EEPROM
LEDs
one wire
ext sync pulse
masterFIP tx
masterFIP rx
macrocycle cnt
turnaround cnt
silence cnt

RETRIEVE SOFTWARE

1. Clone git repo:
$ git clone https://gitlab.cern.ch/cohtdrivers/masterfip